

**Instituto de Computación – Facultad de Ingeniería
Universidad de la República**

Evaluación de una Metodología para la construcción de Data Warehouses

Data Warehouse para la plataforma EVA

Agustín Mullin

Proyecto de Grado
de Ingeniería en Computación
Mayo de 2012
Montevideo, Uruguay

Tutor: Regina Motz

Resumen

La construcción de una Data Warehouse conlleva dificultades particulares debido a sus características especiales, como ser el enfoque hacia la toma de decisiones y el manejo de grandes volúmenes de datos. En el Instituto de Computación se desarrolló la metodología Visión Global del Proceso de Construcción del Data Warehouse, que abarca todo el proceso de construcción de este tipo de sistema: el relevamiento de requerimientos guiado por metadatos, el modelado de dominio con ontologías, reglas para construir un modelo conceptual multidimensional utilizando el lenguaje CMDM, reglas para la construcción de una base de datos integrada desde donde cargar el Data Warehouse, un algoritmo semiautomático para el diseño del esquema lógico relacional del Data Warehouse y la carga del mismo desde la base de datos integrada. Además, la metodología propone una herramienta para mantener la trazabilidad, tanto entre requerimientos y conceptos, como entre requerimientos y cubos.

El principal objetivo del presente proyecto es evaluar la metodología al aplicarla en el contexto de la educación a distancia, en la plataforma Moodle. Luego de aplicada la metodología, construir el Data Warehouse y los sistemas necesarios para obtener un sistema en funcionamiento.

La aplicación de la metodología dejó a la vista las fortalezas, así como algunas limitaciones de la misma. Se destacan como aspectos muy positivos el uso de metadatos en el relevamiento del dominio y la construcción de la ontología de dominio que facilitan la comprensión del dominio; la herramienta de trazabilidad; el uso de una base de datos integrada que protege al Data Warehouse ante cambios en la realidad; y el algoritmo semiautomático para el diseño lógico relacional y la carga del Data Warehouse. Por otro lado, se critica que la aplicación de las reglas para el diseño multidimensional no permiten construir un modelo que satisfaga los requerimientos; se presenta un caso no contemplado por el algoritmo de diseño relacional; y también se muestra que el algoritmo de carga no siempre puede ser aplicado. Se presentan lineamientos para resolver estos problemas. Por último, se presenta una extensión a la herramienta de trazabilidad para que incluya también a los artefactos construidos en la capa de presentación del Data Warehouse en una herramienta de business intelligence.

Luego de aplicada la metodología se construye el Data Warehouse, se desarrolla una aplicación para realizar la carga de la base de datos integrada (proceso ETL), y se construyen reportes en la herramienta Pentaho para permitirle al usuario el acceso amigable a la información. Estos reportes son incluidos en la herramienta de trazabilidad extendida mostrando su aplicabilidad y el beneficio que produce para la gestión del sistema.

Palabras clave: Data Warehouse, metodología, modelado conceptual multidimensional, trazabilidad, ontología, educación a distancia, moodle

Tabla de Contenido

Resumen.....	3
Tabla de Contenido	5
Índice de Figuras	7
Índice de Tablas.....	9
1. Introducción.....	11
2. Conocimientos requeridos.....	15
2.1. Visión Global de Construcción de Data Warehouse (VGPCDW)	15
2.2. MD4DW	15
2.3. Tesis de Peralta y Larrañaga	16
2.4. Ontologías.....	16
2.5. CMDM.....	17
2.6. Moodle.....	18
3. Caso de estudio y requerimientos	19
3.1. Refinamiento del requerimiento	19
3.2. Descripción en detalle de la realidad	20
4. Evaluación de la metodología.....	23
4.1. Aplicación de la fase I y el relevamiento de ontologías	25
4.2. Aplicación de la fase II y el análisis del modelo	32
4.3. Fase III (Modelado conceptual multidimensional)	36
4.4. Fase IV (construcción del ODS).....	42
4.5. Fase V (diseño y carga del Data Warehouse)	43
4.6. Extensión de la Matriz de Trazabilidad.....	46
4.7. Conclusiones de la metodología	47
5. Prototipación	51
5.1. Fase de diseño	52
5.1.1. Moodle	52
5.1.2. Modelado de dominio	53
5.2. Aplicación de carga del ODS	53

5.2.1.	Datos de Bedelía.....	55
5.2.2.	Datos de Moodle	57
5.3.	Aplicación de consulta del Data Warehouse.....	59
5.3.1.	Herramientas relevadas	60
5.3.2.	Solución construida	63
5.4.	Extensión de la Matriz de Trazabilidad.....	66
5.5.	Conclusiones sobre el prototipo.....	68
6.	Conclusiones.....	69
6.1.	Limitaciones y trabajos futuros	70
7.	Glosario.....	73
8.	Bibliografía.....	75
	Anexos.....	79
	Anexo I - Aplicación de la metodología Visión Global del Proceso de Construcción de Data Warehouse al caso de estudio.....	81
	Fase I - Comprensión de la realidad.....	81
	Fase II - Modelado de Dominio	91
	Fase III - Modelado Conceptual Multidimensional	95
	Fase IV - Construcción del ODS.....	105
	Fase V – Generación del diseño lógico y la carga de datos.....	115
	E1: Esquema conceptual.....	116
	E2: Base de datos fuente	118
	E3: Lineamientos.....	119
	E4: Correspondencias.	122
	E5: Algoritmo	128
	Matriz de Trazabilidad	134
	Conclusiones del caso de estudio	135
	Anexo II - Diagrama de arquitectura de la aplicación.....	139

Índice de Figuras

Figura 1 - Fases de la metodología Visión Global del Proceso de Construcción de Data Warehouse. Tomada de (1).	23
Figura 2 - Elementos de tipo educacional de LOM. Tomado de (10)	27
Figura 3 - Ontología de Actividad.....	29
Figura 4 - Ontología denominada de Dominio. Tomada de (14).	30
Figura 5 - Ontología 1. Modela el Grado de interacción como un atributo de Curso.	34
Figura 6 - Ontología 2. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases.....	35
Figura 7 - Ontología 3. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. Además, la Métrica de Grado de interacción muestra cómo es calculado.....	36
Figura 8 - Detalle de la ontología 3.	36
Figura 9 - Dimensiones en CMDM generadas aplicando VGPCDW.	37
Figura 10 - Dimensiones Asignatura y Semestre representadas como una sola ya que comparten el nivel inferior.	38
Figura 11 - Dimensión Cursos, tiene el Grado de interacción como un nivel en la jerarquía de Cursos permitiendo realizar agregaciones desde el nivel inferior.	39
Figura 12 - Relación dimensional mostrando la asociación entre Asignaturas y Carreras.....	40
Figura 13 - Dimensiones del modelo conceptual multidimensional corregido.	41
Figura 14 - Relaciones dimensionales del modelo conceptual multidimensional corregido.	42
Figura 15 - Cubo GradoDeInteraccionCarrera. Contiene para cada Carrera, Año y Grado de Interacción, la cantidad de cursos dictados.....	44
Figura 16 - Detalle de las tablas course y quiz de la base de datos de Moodle. Tomado de (8)..	52
Figura 17 - Esquema de la base de datos integrada ODS.	54
Figura 18 - Mapeo entre archivo de datos de Bedelía y esquema del ODS.	55
Figura 19 - Mapeo entre conceptos de Moodle y el ODS, y cómo se completan los datos Recurso.tipoRecurso y Recurso.grado_de_interaccionRecurso.	58
Figura 20 - Captura de la aplicación web para la carga de los cursos Moodle al ODS.	59
Figura 21 - Reporte en el sistema Pentaho.....	65
Figura 22 - Ontología 1. Modela el Grado de interacción como un atributo de Curso.	93
Figura 23 - Ontología 2. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases.....	94
Figura 24 - Ontología 3. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. Además, la Métrica de Grado de interacción muestra cómo es calculado.....	94
Figura 25 - Detalle de la Métrica del Grado de Interacción de la ontología 3.....	95
Figura 26 - Dimensión Tiempo del modelo conceptual multidimensional.....	96
Figura 27 - Clases candidatas a niveles del modelo conceptual multidimensional.....	96
Figura 28 - Conjunto de niveles del modelo conceptual multidimensional.	97
Figura 29 - Órdenes entre los niveles del modelo conceptual multidimensional.	98

Figura 30 - Dimensiones del modelo conceptual multidimensional.	100
Figura 31 - Aplicación de la Regla 11 de la fase III de VGPCDW, que resulta en un modelo de estrella.....	101
Figura 32 - Dimensión Cursos del modelo conceptual multidimensional.	102
Figura 33 - Relaciones dimensionales del modelo conceptual multidimensional corregido.	103
Figura 34 - Dimensiones del modelo conceptual multidimensional corregido.	104
Figura 35 - Ontología de dominio resultado de aplicar la fase II de VGPCDW.....	105
Figura 36 - Esquema del ODS, resultante de aplicar la fase IV de VGPCDW.	115
Figura 37 - Relaciones dimensionales del modelo conceptual, luego de aplicar la etapa1 de la fase V.....	116
Figura 38 - Dimensiones del modelo conceptual, luego de aplicar la etapa1 de la fase V.	117
Figura 39 - Cubo CursoSemestreAsignatura.	120
Figura 40 - Cubo CarreraAsignatura.....	120
Figura 41 - Cubo GradoDeInteraccionInstituto.	121
Figura 42 - Cubo GradoDeInteraccionCarrera.	121
Figura 43 - Correspondencia entre la dimensión Asignaturas y el ODS.	122
Figura 44 - Correspondencia entre la dimensión Carreras y el ODS.....	122
Figura 45 - Correspondencia entre la dimensión Semestres y el ODS.	123
Figura 46 - Correspondencia entre el cubo CursoSemestreAsignatura y el ODS.	124
Figura 47 - Correspondencia entre el cubo CarreraAsignatura y el ODS.....	125
Figura 48 - Correspondencia entre el cubo GradoDeInteraccionInstituto por un lado, y el ODS y el cubo CursoSemestreAsignatura.....	126
Figura 49 - Correspondencia entre el cubo GradoDeInteraccionCarrera por un lado, y el ODS y los cubos CursoSemestreAsignatura y AsignaturaCarrera.	127
Figura 50 - Arquitectura de la aplicación.....	139

Índice de Tablas

Tabla 1 - Mapeo entre el Grado de interacción definido por la UE y LOM.	28
Tabla 2 - Matriz de Trazabilidad de la ontología 1.....	34
Tabla 3 - Matriz de Trazabilidad, incluyendo los cubos contruidos.	45
Tabla 4 - Modelo propuesto para Matriz de Trazabilidad, incluyendo artefactos de la capa de presentación.	46
Tabla 5 - Análisis esquemático de la metodología Visión Global del Proceso de Construcción de Data Warehouse.	47
Tabla 6 - Módulos incluidos en las suites de Business Intelligence Jasper, Pentaho y SpagoBI. Tomado de (34).....	62
Tabla 7 - Cantidad de descargas realizadas en 2011 de las suites de BI relevadas. Tomado de SourceForge (36) (37) (38) (39).....	63
Tabla 8 - Matriz que vincula los reportes creados con los requerimientos que ayudan a satisfacer.....	65
Tabla 9 - Matriz de Trazabilidad, incluyendo cubos contruidos y artefactos desarrollados en la suite Pentaho.	67
Tabla 10 - Resources en la plataforma Moodle.	84
Tabla 11 - Activities en la plataforma Moodle.....	84
Tabla 12 - Matriz de Trazabilidad de la ontología 1.....	93
Tabla 13 - Definición de cardinalidades en las relaciones interclases de la ontología.....	97
Tabla 14 - Matriz de Trazabilidad, vinculando los requerimientos con los conceptos y los cubos materializados.....	135

1. Introducción

Las organizaciones con su operativa producen gran cantidad y variedad de datos en sus sistemas de información transaccionales, que suelen ser diversos y no siempre se encuentran integrados. Sin embargo a la hora de la toma de decisiones es necesario tener una visión global de alto nivel de lo que sucede en la organización, y es necesario integrar y transformar los datos de estos sistemas transaccionales. Es en este contexto que surge el Data Warehouse.

Un Data Warehouse es una colección estructurada de datos diseñada para dar apoyo a la toma de decisiones. El Data Warehouse suele integrar datos corporativos, tanto actuales como históricos, residentes en bases de datos operacionales de la organización, en un único repositorio sobre el cual los usuarios puedan hacer análisis de información. Por lo tanto el Data Warehouse difiere de un sistema transaccional en que: 1) debe contener la información relevante para la toma de decisiones, con un nivel de agregación que no necesariamente coincide con el manejo en los sistemas transaccionales; 2) maneja grandes volúmenes de datos, ya que integra diferentes fuentes de datos y además incluye gran cantidad de datos históricos; 3) la cantidad de transacciones que debe soportar es baja pero para consultas complejas que involucran muchos datos se esperan tiempos de respuesta pequeños.

Por sus características particulares, el Data Warehouse requiere para su construcción, un abordaje diferente al utilizado para los clásicos sistemas transaccionales. Para ello, en el Instituto de Computación se desarrolló la metodología *Visión global del proceso de construcción de Data Warehouse* (VGPCDW) (1), que “presenta una guía que consta de cinco fases o pasos para construir un Data Warehouse abarcando el relevamiento de requerimientos, la representación de la realidad de la empresa, la generación del diseño conceptual, el diseño lógico y la carga de datos necesaria para el correcto funcionamiento del sistema”.

El objetivo del presente trabajo es evaluar la metodología VGPCDW. Para evaluar la metodología se aplica la misma a un requerimiento de información de la Unidad de Enseñanza (2) de Facultad de Ingeniería. La Unidad de Enseñanza (UE) es un “espacio de trabajo destinado a la mejora y el estudio de la educación en la Facultad de Ingeniería”.

En la Facultad de Ingeniería se dictan cursos on-line a través del Entorno Virtual de Aprendizaje Moodle y es de interés de la Unidad de Enseñanza conocer y entender dicha realidad. Se pretende conocer sobre los cursos: qué calidad tienen y los modelos pedagógicos que utilizan.

La aplicación de VGPCDW dejó a la vista las fortalezas de la metodología, así como algunas limitaciones de la misma. Este análisis es el principal producto del presente proyecto.

El principal problema encontrado en la metodología es que guía la construcción de un modelo conceptual multidimensional que no satisface los requerimientos debido a la confusión de CMDM (el lenguaje de modelado) con el modelo estrella. Esto lleva a que la metodología no contemple la definición de las dimensiones que son medidas en las relaciones dimensionales.

Tampoco considera relaciones dimensionales sin medida que sirven para modelar relaciones de cardinalidad N:N.

Otro problema de VGPCDW se encuentra en la etapa de diseño lógico del Data Warehouse, VGPCDW presenta un algoritmo semiautomático para el diseño del esquema relacional. En este algoritmo se encontraron casos no considerados, en particular los que requieren vincular dos relaciones dimensionales para construir un cubo.

Por último se destaca que VGPCDW realiza la carga de los datos desde una base de datos intermedia denominada ODS, y no directamente de las fuentes de datos. Si bien esto protege al Data Warehouse ante cambios en las fuentes de datos, esto se contempla como una debilidad de la metodología ya que no abarca la carga del ODS.

Este proyecto también tiene como resultado un sistema de información para la UE, que satisface los requerimientos planteados. Para poder realizar la carga del Data Warehouse se construye un prototipo que asista al usuario a en el proceso de extracción de los datos de las fuentes, la transformación y la carga.

Para poder presentar la información al usuario final se realiza un relevamiento de tecnologías open source de Business Intelligence y la selección de una, sobre la que se construye la capa de acceso a la información para el usuario final.

Se realiza una extensión a la herramienta de trazabilidad de la metodología para que abarque los artefactos creados en la capa de acceso a la información, logrando la trazabilidad completa sobre todos los elementos de la solución.

La solución presentada incluye una rica documentación que es resultado de aplicar la metodología. Esta documentación facilita el entendimiento de la solución tanto para los usuarios como para nuevos equipos de desarrollo.

En el siguiente capítulo se presentan brevemente los conocimientos requeridos para la realización del proyecto, acompañados con un comentario de por qué se utilizan y los beneficios que aportan.

En el capítulo 3 se presenta el caso de estudio sobre el que se aplica la metodología, así como los requerimientos de la Unidad de Enseñanza.

El capítulo 4 muestra la aplicación de la metodología paso por paso. Contiene los puntos claves de la aplicación de la metodología, de los cuales se arriba a las conclusiones. La aplicación detallada de la metodología se presenta como anexo. En este capítulo también se presenta un relevamiento de ontologías del dominio de educación realizado previo al desarrollo de la ontología de dominio. También se presenta la propuesta de extensión a la herramienta de trazabilidad. Por último se plantean las conclusiones generales sobre la metodología VGPCDW.

El capítulo 5 presenta el prototipo construido. Esto incluye dos herramientas, una para realizar la carga del Data Warehouse, y otra para acceder a la información contenida en el mismo. Se incluye un breve relevamiento de tecnologías open source de Business Intelligence en lo que se basó la elección de la herramienta utilizada. En este capítulo se aplica la propuesta de extensión a la herramienta de trazabilidad.

En el capítulo 6 se presentan las conclusiones generales del proyecto.

Por último se incluyen los anexos. El Anexo I muestra la aplicación detallada de la metodología. En el Anexo II se presenta la arquitectura del sistema desarrollado.

2. Conocimientos requeridos

En este capítulo se presentan brevemente los conocimientos requeridos para la realización del proyecto, acompañados con un comentario de por qué se utilizan y los beneficios que aportan.

2.1. Visión Global de Construcción de Data Warehouse (VGPCDW)

VGPCDW (1) es la metodología de construcción de Data Warehouses que se evalúa en el presente proyecto. VGPCDW “presenta una guía que consta de cinco fases o pasos para construir un Data Warehouse abarcando el relevamiento de requerimientos, la representación de la realidad de la empresa, la generación del diseño conceptual, el diseño lógico y la carga de datos”.

Además, la metodología propone la herramienta Matriz de Trazabilidad con el objetivo de mantener la trazabilidad, tanto entre requerimientos y conceptos, como entre requerimientos y cubos. Esto permite identificar los efectos de un cambio, ya sea en los requerimientos como en la realidad.

VGPCDW propone una metodología unificada de construcción de Data Warehouses, y lo hace integrando, corrigiendo y extendiendo tres proyectos: MD4DW y las tesis de Peralta y Larrañaga, que se describen brevemente a continuación.

La forma en que se construyó la metodología, como la integración de tres proyectos, genera una dificultad al momento de estudiarla. Es necesario estudiar los cuatro proyectos para obtener una visión total de la metodología y conseguir entenderla completamente. Esto se debe a que cada proyecto abarca una fase del proceso y VGPCDW los integra haciéndoles modificaciones; pero el foco de VGPCDW está puesto en los ajustes que hacen posible la integración y no en presentar la metodología de forma íntegra.

2.2. MD4DW

MetaData for Data Warehouse (MD4DW) (3) brinda una guía general para “obtener el diseño del modelo conceptual multidimensional de un Data Warehouse a partir de la realidad, teniendo en cuenta las reglas del negocio y los requerimientos del mismo”.

MD4DW plantea reglas que “permiten realizar la transformación de un modelo de dominio representado mediante una ontología, en el modelo conceptual multidimensional CMDM. La construcción de esta ontología es realizada a partir del conocimiento de la realidad guiado por el conjunto de requerimientos que el Data Warehouse deberá satisfacer.”

Esta metodología se destaca por seguir etapas o fases justificadas a través de las cuales de forma guiada se consideran, además de las necesidades iniciales, el mantenimiento y futuros cambios del Data Warehouse. Estos últimos dos puntos son considerados a través de la documentación generada a lo largo de la aplicación de la metodología, la cual permite mantener una gestión del conocimiento relativo al dominio y de las decisiones tomadas durante el diseño del Data Warehouse.

2.3. Tesis de Peralta y Larrañaga

Las tesis de maestría de Verónica Peralta (4) (5) e Ignacio Larrañaga (6) presentan un algoritmo para la generación semiautomática del esquema relacional para un Data Warehouse así como su carga y actualización. Primero, la tesis de Peralta explica cómo realizar semiautomáticamente la generación del esquema relacional para el Data Warehouse, la que se realiza a partir de una base de datos fuente integrada, un esquema conceptual modelado con CMDM y un conjunto de correspondencias entre la base de datos fuente y el esquema integrado. Luego, la tesis de Larrañaga, reutilizando el conocimiento generado durante el diseño conceptual y lógico del Data Warehouse, aborda el problema de la carga y actualización de éste.

Este algoritmo unificado presenta una solución semiautomática para generar el esquema relacional del Data Warehouse y para realizar su carga a partir de una fuente de datos integrada.

2.4. Ontologías

Una ontología es una representación formal del conocimiento como una colección de conceptos de un dominio y las relaciones entre dichos conceptos. Fue definida por Gruber como “una especificación explícita de una conceptualización compartida” (7).

VGPCDW utiliza la ontología para modelar conceptualmente el dominio de aplicación, y sobre este modelo se basa para luego construir el modelo conceptual multidimensional de la solución. El construir un modelo del dominio permite tener trazabilidad desde los conceptos de la realidad hasta las entidades del modelo conceptual multidimensional, y hasta la implementación concreta que permite satisfacer cada requerimiento.

Además, representar el dominio mediante una ontología, que es representable de forma gráfica, permite la validación con el cliente.

Otra gran fortaleza de la ontología es que es posible aplicar razonadores para detectar inconsistencias en el modelo. Esto se vuelve cada vez más importante a medida que se agregan requerimientos y el modelo crece.

Modelar el dominio del problema con ontologías en un lenguaje estándar también permite compartir y reutilizar el conocimiento adquirido. Así, uno de los aportes del proyecto es el modelo construido, que queda a disposición para futuros proyectos.

En este proyecto como lenguaje para representar la ontología se utiliza OWL 2 (8). OWL 2 es una extensión de OWL 1, con el que tiene compatibilidad hacia atrás completa. Una de las funcionalidades que agrega, que es el principal atractivo desde el punto de vista de este proyecto, es la posibilidad de definir claves para las clases. Esto simplifica los modelos construidos, al no requerir definir clases adicionales para modelar las claves.

2.5. CMDM

CMDM (Conceptual MultiDimensional Model) (9) es un modelo de datos conceptual y multidimensional que permite la especificación detallada de una base de datos multidimensional, desarrollado en el Instituto de Computación.

VGPCDW utiliza CMDM para modelar conceptualmente la solución a construir. Realizar un modelo conceptual permite abstraerse del modelo relacional en que se implemente la solución, así como de las herramientas con las que se trabaje.

CMDM brinda una visión multidimensional de los datos con dimensionalidad genérica, lo que permite buscar nuevas representaciones e interpretaciones de los datos, al convertir las medidas en dimensiones. Esta propiedad de CMDM es aprovechada en el presente proyecto, ya que la medida Grado de interacción es construida como una jerarquía y tratada como una dimensión para satisfacer uno de los requerimientos.

CMDM define formalmente sus elementos y brinda un lenguaje para definir restricciones, lo que evita ambigüedad en los modelos construidos. Pero el modelo y el lenguaje de restricciones son complejos. Como muestra de esto está el hecho de que el proyecto MD4DW utiliza CMDM de forma incorrecta, y que el proyecto de Peralta utiliza solamente el lenguaje gráfico evitando definir restricciones.

2.6. Moodle

Moodle (10) es un Learning Management System (LMS (11)) o Course Management System. Es una aplicación web para dar apoyo a la educación que permite que docentes creen cursos online, en los que estudiantes pueden inscribirse, acceder a los materiales y participar de las actividades propuestas. Es un software open source gratuito, respaldado por una gran comunidad internacional.

Moodle fue elegido institucionalmente por la Universidad de la República (UdelaR) como plataforma para educación a distancia, y es utilizado por la facultad de ingeniería como LMS. Es de esta plataforma que la Unidad de Enseñanza quiere obtener información, y para aplicar la metodología el presente proyecto trabaja con datos extraídos de Moodle.

La extracción de los datos de Moodle plantea en primer lugar dificultades dado la complejidad del sistema. Moodle almacena su información en una base de datos relacional que contiene alrededor de 200 tablas (12), sobre las cuales la documentación no cubre todos sus componentes. Para poder obtener los datos de los cursos de Moodle fue necesario estudiar la estructura de las tablas del sistema ya que no se contaba con un experto del dominio.

La carga de la base de datos intermedia (ODS) desde Moodle también plantea algunas dificultades, ya que Moodle es un sistema transaccional y su esquema relacional está pensado para soportar el dictado de cursos en la plataforma, y no el análisis de información que se quiere llevar a cabo (por ejemplo, el concepto de Grado de interacción no existe en Moodle). Es por esto que el esquema de Moodle no coincide con el esquema realizado para representar el dominio, y por lo tanto la carga del ODS no es directa.

En particular Moodle utiliza una tabla para cada tipo de recurso que puede existir en un curso. Dado los requerimientos, no es necesario diferenciar los recursos más que por el grado de interacción que ofrecen. Por lo tanto, en el modelo de dominio realizado existe una clase recurso por cada grado de interacción posible, y el tipo de recurso es modelado con un atributo. Luego, en el esquema del ODS, se crea una única tabla donde se almacenan todos los recursos indistintamente, indicándose por atributos su grado de interacción y tipo de recurso.

3. Caso de estudio y requerimientos

La Universidad de la República dispone de un Entorno Virtual de Aprendizaje llamado (EVA), que se encuentra sobre la plataforma Moodle, que los docentes utilizan para dictar los cursos de las asignaturas. La aplicación Moodle fue elegida institucionalmente, y se realizó una instalación independiente en todas las facultades de la universidad.

El cliente de este proyecto es la Unidad de Enseñanza (UE) de la Facultad de Ingeniería. La UE quiere saber cuál es el grado de interacción de los cursos en el EVA Moodle de la Facultad de Ingeniería (13).

La UE define el concepto grado de interacción, el cual se puede aplicar a un recurso de un curso en Moodle o a un curso en Moodle. El grado de interacción tiene los siguientes valores posibles: “Repositorio”, “Interacción Estudiante-Contenido”, “Interacción Estudiante-Actividad”, “Interacción Estudiante-Docente”, e “Interacción Estudiante-Estudiante”.

Para un recurso en un curso en Moodle el grado de interacción es el tipo de interacción que le permite al estudiante, y el mismo lo define la Unidad de Enseñanza para cada tipo de recurso.

Para un curso en Moodle, el grado de interacción es cuánta interacción le permite al estudiante ese curso y está determinado por el tipo de recursos que componen el curso y en qué proporción y los grados de interacción de los mismos.

Así, los recursos también tienen un grado de interacción (definido por la UE), y los cursos también tienen un grado de interacción (que depende de sus recursos).

Sabiendo el grado de interacción de los cursos, se quiere poder comparar los grados de interacción de los cursos de una misma asignatura dictados en diferentes semestres; y poder realizar comparaciones con los cursos dictados para asignaturas de institutos y carreras diferentes.

3.1. Refinamiento del requerimiento

Requerimiento1: Se quiere saber cuál es el grado de interacción de los cursos en la plataforma Moodle, y para cada curso, saber cuántos recursos de cada grado de interacción tiene.

Requerimiento 2: Se quiere saber cuál es el grado de interacción de los cursos por asignatura por semestre y año.

Requerimiento 3: Se quiere saber cuál es el grado de interacción de los cursos por asignatura por institutos y por carreras. También la cantidad de cursos por grado de interacción por año para institutos y para carreras.

3.2. Descripción en detalle de la realidad

La Universidad de la República (UdelaR) se encuentra compuesta por facultades, donde cada facultad dicta una serie de carreras. Las facultades se encuentran formadas por institutos (también llamados departamentos). Cada instituto dicta una serie de asignaturas que pertenecen a una o varias de las carreras de esa facultad.

Cada facultad tiene un nombre y una abreviación de ese nombre que se usa para identificarlo, donde ambos son únicos (por ejemplo fagro corresponde a Facultad de Agronomía). Cada carrera tiene un identificador único dentro de una facultad, un nombre, una cantidad de créditos. Cada instituto tiene un nombre que es único dentro de la facultad. Cada asignatura tiene un identificador único dentro de la facultad donde se dicta, un nombre, una cantidad de créditos, y un tipo (puede ser de grado o de posgrado).

Las facultades de la UdelaR, en particular de Facultad de Ingeniería, proveen a sus docentes de un Entorno Virtual de Aprendizaje llamado EVA, que se encuentra sobre la plataforma Moodle, que los docentes utilizan para “dictar” los cursos de las asignaturas. Se considerarán solamente las asignaturas que tengan un curso en Moodle. Un curso se identifica por la asignatura a la que corresponde y por el semestre en que se dictó. Un semestre se identifica por el año al que pertenece y el período del año (se denomina impar a la primera mitad del año, par a la segunda).

Los cursos en Moodle se encuentran formados por “Actividades” y “Recursos” (de ahora en adelante se referirá a los dos como recursos). Un recurso puede ser por ejemplo un foro de discusión, un ejercicio, un archivo para lectura, o simplemente un link. Los recursos tienen siempre un tipo (por ejemplo un foro) y un identificador único para ese tipo. Un curso siempre tiene al menos un recurso.

Se define el concepto grado de interacción, el cual se puede aplicar a un recurso de un curso en Moodle o a un curso en Moodle. El grado de interacción se mide en una escala enumerada cuyos valores posibles son: “Repositorio”, “Interacción Estudiante-Contenido”, “Interacción Estudiante-Actividad”, “Interacción Estudiante-Docente”, e “Interacción Estudiante-Estudiante”.

Para un recurso en un curso en Moodle el grado de interacción es el tipo de interacción que le permite al estudiante, y el mismo lo define la Unidad de Enseñanza para cada tipo de recurso.

Para un curso en Moodle, el grado de interacción es cuánta interacción le permite al estudiante ese curso y está determinado por el tipo de recursos que componen el curso y en qué proporción y los grados de interacción de los mismos.

Así, los recursos también tienen un grado de interacción (definido por la UE), y los cursos también tienen un grado de interacción (que depende de sus recursos).

Para definir el grado de interacción de un curso, se cuentan la cantidad de recursos que se clasifican dentro de cada tipo de grado de interacción, el grupo más poblado determinará el grado de interacción del curso.

4. Evaluación de la metodología

El objetivo de este proyecto es evaluar la metodología VGPCDW, y para ello se la aplica para satisfacer un requerimiento de la Unidad de Enseñanza. La metodología se compone de cinco fases para construir un Data Warehouse, abarcando el relevamiento de requerimientos, la representación de la realidad de la organización, la generación del diseño conceptual, el diseño lógico y la carga de datos necesaria para el correcto funcionamiento del sistema.

La metodología VGPCDW propone las siguientes cinco fases: I) Comprensión de la realidad, II) Modelado de dominio, III) Modelado conceptual multidimensional, IV) Construcción de un Operational Data Store (ODS), y V) Generación del diseño lógico y la carga de datos. La figura 1 muestra todas las fases de la metodología.

Además, la metodología propone la herramienta Matriz de Trazabilidad (en MD4DW es denominada Matriz de Requerimientos) con el objetivo de mantener la trazabilidad, tanto entre requerimientos y conceptos, como entre requerimientos y cubos. Esto permite identificar los efectos de un cambio, ya sea en los requerimientos como en la realidad.

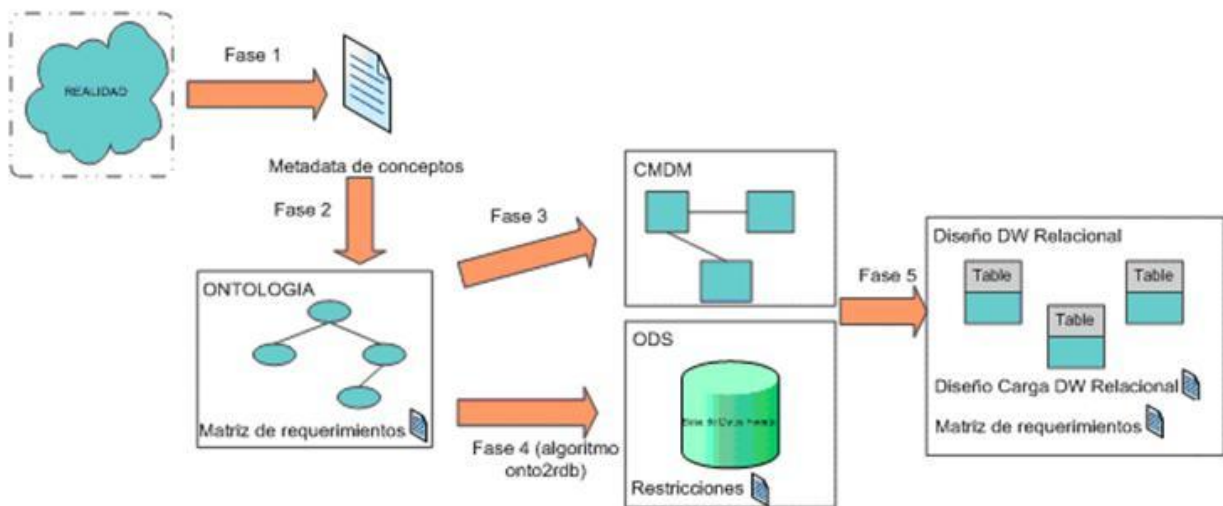


Figura 1 - Fases de la metodología Visión Global del Proceso de Construcción de Data Warehouse. Tomada de (1).

A continuación se presenta un resumen de la aplicación de la metodología, presentándose la misma en detalle en el Anexo I. En primer lugar se presenta la fase I, el relevamiento de los requerimientos del cliente y el relevamiento del dominio de aplicación. Antes de aplicar la fase II para construir la ontología de dominio, se realizó un relevamiento de ontologías existentes del dominio de educación en busca de modelos que incluyesen los conceptos necesarios para satisfacer el requerimiento. Esto plantea un escenario más cercano a la práctica en el que no se parte totalmente de cero. También permite reutilizar modelos ya construidos y aprovechar este conocimiento. Por último, utilizar ontologías ya definidas abre la posibilidad de interoperar con

otros sistemas al compartir los modelos conceptuales del dominio. Se presentan los modelos relevados.

Luego se presenta la aplicación de la fase II, que lleva a la construcción de la ontología de dominio. Se analizan tres posibles ontologías para modelar el concepto de Grado de interacción, un concepto que es central para el requerimiento y tiene la particularidad de ser la medida de una de las relaciones multidimensionales que se construyen en la fase III. Aquí ya se adelanta que uno de los problemas de la metodología radica en la falta de reglas para construir las medidas de las relaciones dimensionales en el modelo conceptual multidimensional. Se analizan los tres modelos y se elige uno para continuar aplicando la metodología.

A continuación se presenta la aplicación de la fase III para construir el modelo conceptual multidimensional. Como resultado de aplicar las reglas no se pudo construir con éxito una solución para satisfacer la totalidad de los requerimientos, ya que VGPCDW entiende erróneamente a CMDM al no entender que las medidas de las relaciones multidimensionales deben ser construidas como dimensiones. Por lo tanto VGPCDW no contempla reglas para la construcción de las medidas. Tampoco contempla reglas para la construcción de relaciones multidimensionales sin medida que sirven para modelar relaciones de cardinalidad N:N. Al terminar con esta fase se llega a un modelo que no permite satisfacer los requerimientos. Para poder evaluar las restantes fases se construye un modelo en CMDM que sí permite satisfacer los requerimientos, el mismo es presentado y se muestra cómo los satisface.

La cuarta fase de VGPCDW es la construcción de un sistema ODS desde el que realizar la carga del Data Warehouse, la que se realiza exitosamente aplicando las reglas definidas. Utilizar una base intermedia es positivo ya que protege al Data Warehouse de cambios en las fuentes de datos. Sin embargo la metodología no contempla la carga del ODS. Se presenta un breve análisis de la fase IV.

En la quinta fase de VGPCDW, mediante un algoritmo semiautomático, se genera el esquema relacional del Data Warehouse y el proceso de carga desde el ODS. Esta fase es aplicada con éxito en el proyecto, aunque se detecta que la metodología no contempla algunos casos particulares, en particular la creación de cubos que requieren vincular dos relaciones dimensionales. También se observa que la creación del esquema mediante vistas, la forma definida por VGPCDW, no siempre es posible. Por último, VGPCDW no abarca la implementación concreta del Data Warehouse, sugiriendo como solución la materialización de las vistas, lo que no es posible en los DBMS open source.

Una vez aplicada la metodología, construido y cargado el Data Warehouse, este proyecto construye una capa de acceso a los datos sobre una herramienta de Business Intelligence, lo que es presentado en el capítulo 5. Utilizando esta herramienta de Business Intelligence se crean artefactos, que pueden ser reportes, cubos OLAP, dashboards, entre otros. En esta etapa el presente proyecto propone extender la Matriz de Trazabilidad para que abarque también a estos artefactos.

A continuación se presenta: 1) la aplicación de la fase I y el relevamiento de ontologías; 2) la aplicación de la fase II y el análisis del modelo; 3) un análisis de la fase III junto con su aplicación, 4) un comentario de la fase IV (construcción del ODS); 5) un análisis de la fase V (diseño y carga del DW); 6) la propuesta de extensión de la Matriz de Trazabilidad; y por último 7) las conclusiones sobre la metodología.

4.1. Aplicación de la fase I y el relevamiento de ontologías

La fase I de VGPCDW tiene como objetivo “identificar todos los conceptos que componen el dominio del problema, junto con las relaciones existentes entre los mismos y las restricciones identificadas para cada uno. El relevamiento y comprensión de la realidad está guiado por el conjunto de requerimientos iniciales. Por esto es necesario realizar previamente un análisis de los requerimientos y en base a ello determinar el dominio del problema” (3).

El requerimiento de información del cliente refiere en términos generales al grado de interacción que ofrecen a los estudiantes, los cursos de la plataforma Moodle de Facultad de Ingeniería.

La Unidad de Enseñanza define el concepto grado de interacción, el cual se puede aplicar a un recurso de un curso en Moodle o a un curso en Moodle. El grado de interacción se mide en una escala enumerada cuyos valores posibles son: “Repositorio”, “Interacción Estudiante-Contenido”, “Interacción Estudiante-Actividad”, “Interacción Estudiante-Docente”, e “Interacción Estudiante-Estudiante”. Para un recurso en un curso en Moodle el grado de interacción es el tipo de interacción que le permite al estudiante, y el mismo lo define la Unidad de Enseñanza para cada tipo de recurso. Para un curso en Moodle, el grado de interacción es cuánta interacción le permite al estudiante ese curso y está determinado por el tipo de recursos que componen el curso y en qué proporción y los grados de interacción de los mismos. Se requiere analizar el grado de interacción de los cursos por semestre, asignatura, instituto y carrera. Una descripción detallada del requerimiento se presenta en el capítulo 3.

Como resultado de aplicar esta fase se genera el documento Metadata de conceptos que “contiene información de contexto y datos importantes identificados para cada concepto” (3), entre ellos el de la fuente de los datos. El documento Metadata de conceptos se presenta en el Anexo I que presenta la aplicación detallada de la metodología.

La aplicación de la fase I no supuso ningún problema. Se identifican como aspectos positivos de esta fase la generación de información estructurada en el documento Metadata de conceptos, donde se incentiva la temprana identificación de las fuentes de datos.

Luego de relevados los requerimientos y el dominio, y antes de aplicar la fase II para construir la ontología de dominio, se realizó un relevamiento de ontologías existentes del dominio de

educación en busca de modelos que incluyesen los conceptos necesarios para satisfacer el requerimiento.

Esto plantea un escenario más cercano a la práctica en el que no se parte totalmente de cero. También permite reutilizar modelos ya construidos y aprovechar este conocimiento, lo que puede ser muy útil en el caso de modelos complejos como es el de perfil de estudiante.

Por último, pero muy importante, utilizar ontologías ya definidas abre la posibilidad de interoperar con otros sistemas al compartir los modelos conceptuales del dominio. Modelar el dominio con una ontología permite abstraerse de la realidad concreta que se quiere modelar. En este caso, el concepto curso se refiere a los cursos de Moodle, pero puede aplicarse este modelo para referirse a cursos dictados en otro LMS, e incluso integrarse esos datos con los de Moodle. Como el grado de interacción está definido en base a las propiedades de las entidades, y no en base a las entidades en sí, se puede aplicar este modelo a otro contexto. Para poder realizarlo sólo debe definirse para los nuevos tipos de recurso, el grado de interacción que brindan.

A continuación se presentan los modelos relevados, clasificados en dos grupos: por un lado los modelos que sí cubrían conceptos del dominio necesarios para satisfacer los requerimientos. Para estos modelos se explica si se utilizaron o no los conceptos compartidos y por qué. Luego se presentan los modelos que cubren conceptos del dominio pero no son necesarios para los requerimientos actuales. A pesar de no ser necesarios, se los presenta de todas formas porque los mismos pueden ser útiles para ayudar a satisfacer un futuro requerimiento.

Ontologías aplicadas

IEEE 1484.12.1 – 2002 Standard for Learning Object Metadata (LOM)

LOM (14) es un estándar abierto de metadatos para la descripción de objetos de aprendizaje (ODA), donde un objeto de aprendizaje se define una entidad, digital o no, que puede ser utilizada para la educación. Utilizando esta definición, tanto los recursos de Moodle, los mismos cursos y la propia plataforma Moodle de facultad pueden ser considerados como ODAs. LOM no es una ontología, sino un estándar de metadatos para ODAs, y por lo tanto puede ser utilizado para describir elementos del dominio.

LOM especifica un esquema de datos conceptual que define la estructura de los metadatos para un ODA. El objetivo del estándar es facilitar la búsqueda, evaluación, adquisición y uso de los objetos de aprendizaje. En la figura 2 se presenta un detalle del esquema definido por LOM, incluyéndose únicamente los considerados de tipo educacional.

Los conceptos interesantes que tienen relevancia para el caso de estudio son los elementos Interactivity Type e Interactivity Level.

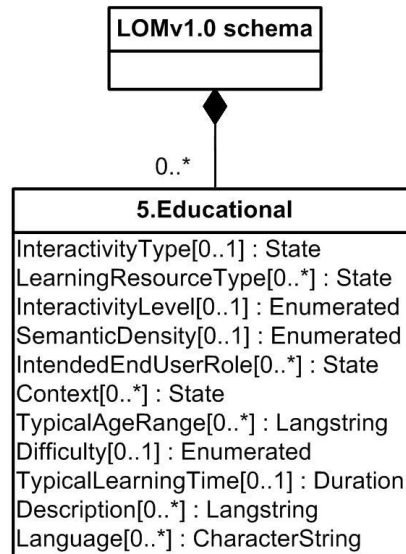


Figura 2 - Elementos de tipo educacional de LOM. Tomado de (15) .

Otro elemento interesante es el Learning Resource Type, que define el tipo de objeto de aprendizaje y algunos de sus valores posibles son: exercise, simulation, questionnaire, diagram, figure, graph y slide. Este concepto no es de interés para el presente requerimiento, pero puede resultar muy útil para uno futuro. Cabe señalar clasificar los recursos en las categorías propuestas por Learning Resource Type no siempre es automatizable, ya que por ejemplo un recurso de Moodle denominado “Link to a file or website” puede ser tanto exercise, graph o slide, u otro distinto.

Interactivity Type pretende expresar si el ODA induce la acción productiva del estudiante. Para ello ofrece tres valores posibles: active, expositive y mixed. “Un ODA de Interactivity Type tipo active requiere que el estudiante ingrese contenido semánticamente significativo o realice algún tipo de acción productiva o decisión” (14). Es de tipo expositive si “la tarea del estudiante consiste principalmente en absorber el conocimiento expuesto (generalmente a través de texto, imágenes o sonido)”.

Por otro lado el Interactivity Level pretende medir el “grado de interactividad que caracteriza al ODA. En este contexto se refiere al grado en que el estudiante puede influir en el aspecto o comportamiento del ODA” (14). Los valores posibles son very low, low, medium, high y very high.

Con estos elementos de LOM no es posible representar la interactividad definida según el requerimiento. El requerimiento pretende distinguir el tipo de interacción que el recurso de Moodle (un ODA en definitiva) le brinda al estudiante, según con quién le permite interactuar, que puede ser una actividad, el docente u otro estudiante; y esto no es posible con LOM.

De todas formas es posible realizar un crosswalk o mapeo entre los conceptos definidos por el cliente y los definidos por IEEE LOM, utilizando los elementos Interactivity Type e Interactivity Level. El mismo se muestra en la tabla 1. “Un crosswalk es un mapeo entre los elementos, semántica y sintaxis de un esquema de metadatos a otro. Un crosswalk permite que la metadata creada por un grupo sea utilizada por otro grupo que utiliza un estándar diferente” (16).

Grado de Interacción (UE)	Interactivity Type (LOM)	Interactivity Level (LOM)
Repositorio	expositive	low
Estudiante-Contenido	expositive	high
Estudiante-Actividad	mixed	high
Estudiante-Docente	active	low
Estudiante-Estudiante	active	high

Tabla 1 - Mapeo entre el Grado de interacción definido por la UE y LOM.

Este crosswalk es en cierta medida discutible, ya que no aprovecha todo el rango de opciones que brindan los elementos seleccionados de LOM, y al convertir los conceptos definidos por el cliente a LOM se pierde la riqueza de los conceptos definidos por el cliente y la posibilidad de distinguir entre un recurso que permite la interacción entre estudiantes y la que solamente la permite con el docente.

Es por esto que en el presente proyecto se decide por no utilizar el crosswalk y definir un application profile de LOM que defina el concepto de Grado de Interacción tal cual lo define el cliente.

Ontología Actividad

La ontología Actividad¹ propuesta por el proyecto Educa (17) del Instituto de Computación, apunta a la descripción de objetos de aprendizaje. Como se muestra en la figura 3, esta ontología tiene un concepto central que es el de Actividad, que es un objeto de aprendizaje.

La Actividad tiene definidos un InteractivityLevel, un InteractivityType así como un LearningResourceType. Estos tres son conceptos definidos por LOM. El Learning Resource Type refiere al tipo específico de objeto de aprendizaje, para el que algunos de los valores posibles son: exercise, simulation y questionnaire.

Por las razones expuestas en el punto anterior no es satisfactorio basar el proyecto en esta ontología. Sin embargo, una vez construida la ontología para el dominio específico, es posible integrarla con la Ontología Actividad al mapear los conceptos de RecursoMoodle y cursoMoodle con el de Actividad, como se definió anteriormente para LOM.

¹ La ontología se encuentra publicada en <http://www.fing.edu.uy/inco/grupos/csi/esp/Proyectos/OntologiasOwl/Actividad.owl>

² La ontología de Estilos de Aprendizaje se encuentra disponible en

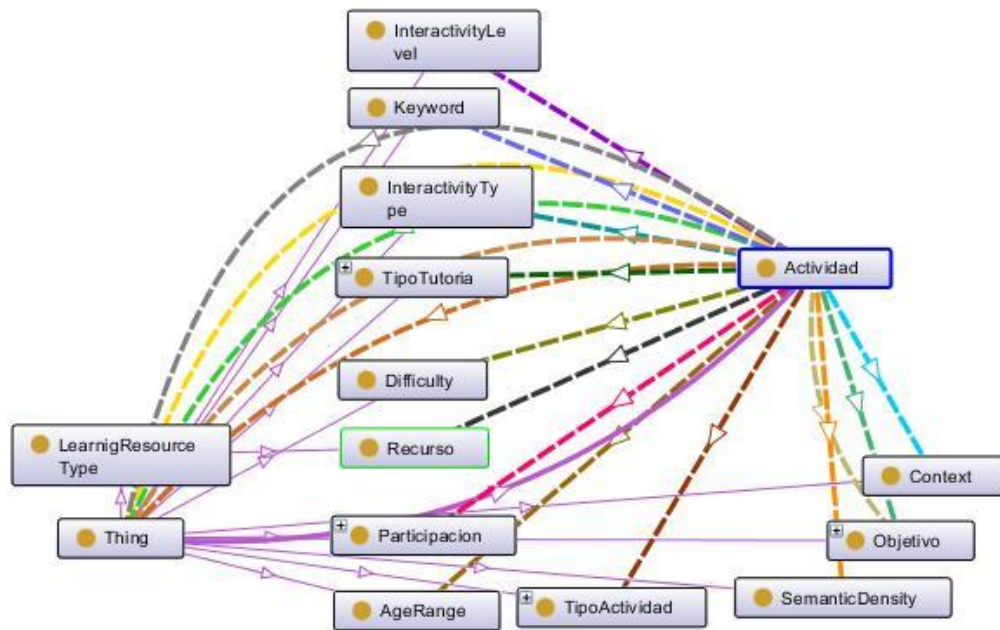


Figura 3 - Ontología de Actividad.

Ontologías no aplicadas

En esta sección se presentan los modelos de dominio relevados que no fueron aplicados para el presente proyecto. Estos modelos cubren conceptos del dominio de aplicación, como ser los perfiles de estudiantes en el contexto de e-learning, pero estos conceptos no son necesarios para satisfacer los requerimientos actuales.

A pesar de no ser necesarios, se los presenta de todas formas porque se considera que pueden ser útiles para ayudar a satisfacer un futuro requerimiento.

Ontología Estilos de Aprendizaje

La ontología de Estilos de Aprendizaje² también es parte del Proyecto Educa (17), mencionado anteriormente. Describe estilos de aprendizaje que poseen los recursos educativos. Los estilos están definidos como Analíticos u Holísticos, y cada recurso puede tener una afinidad alta o baja con los estilos. Además para los recursos se definen grados de Aplicabilidad y Generalidad. Esta ontología no posee elementos que aporten al requerimiento actual, pero pueden aplicarse para satisfacer un próximo requerimiento que refiera a este tipo de aspectos.

² La ontología de Estilos de Aprendizaje se encuentra disponible en <http://www.fing.edu.uy/inco/grupos/csi/esp/Proyectos/OntologiasOwl/EstilosDeAprendizaje.owl>

PAPI (18) es una propuesta de estándar de IEEE para tecnologías de aprendizaje que apunta a crear un modelo conceptual de los estudiantes y brindar bibliotecas para accederlo. PAPI define elementos referentes a habilidades del estudiante y a la adquisición de conocimiento, a las preferencias y estilos del estudiante, a su desempeño, entre otros.

PAPI presenta la información del estudiante en seis categorías: información personal, información de la relación del estudiante con otros (estudiantes, docentes), información de seguridad (permisos), preferencias del estudiante, rendimiento del estudiante, y el historial del estudiante de participación en la plataforma. Estos conceptos son aplicables en el contexto de educación sobre la plataforma Moodle.

Reasoning and Ontologies for Personalized E-Learning in the Semantic Web - Henze, Dolog, Nejdl (19)

Este artículo propone un framework para e-learning personalizado en la web semántica, y propone ontologías para tres tipos de recursos: de dominio, de usuario y de observación.

La primera ontología, denominada por ellos como de dominio se muestra en la figura 4. Este modelo utiliza la clase Document para referirse a recursos educativos, que describen un concepto (Concept), y se encuentran vinculados por un tópico (Subject). También permite definir relaciones de previatura entre los los Documents, así como entre los conceptos. Además define una ontología para el DocumentType, un concepto similar al Learning Resource Type de LOM. Pero no considera aspectos de la interactividad de los Documents.

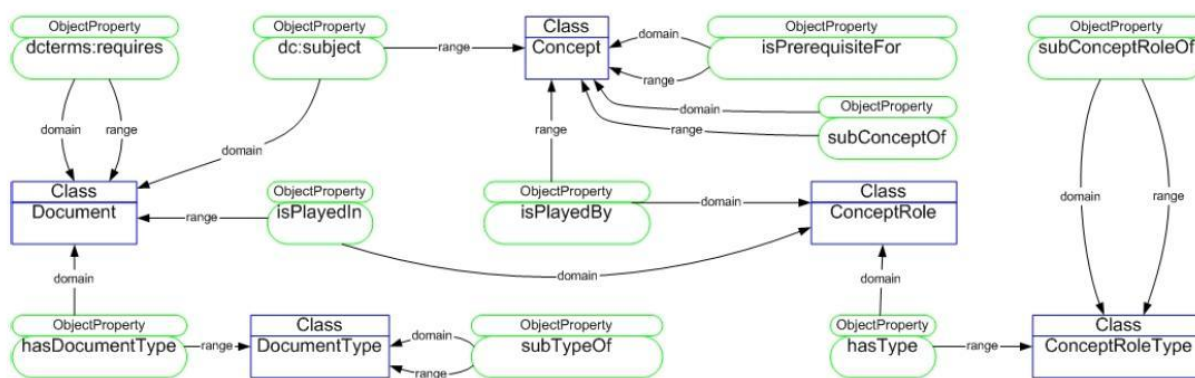


Figura 4 - Ontología denominada de Dominio. Tomada de (19).

La ontología de usuario se basa en IEEE PAPI mencionado anteriormente. La ontología de observación modela las diferentes posibles actividades de los usuarios con los recursos (ejemplos de interacciones que tiene un usuario con un recurso son access, bookmark, annotate), así como la hora de inicio y de fin de la actividad.

La ontología definida como de dominio representa conceptos del dominio de aplicación, en donde la clase Document representa los recursos de Moodle. Pero este modelo no contempla la interactividad de los recursos, que es el requerimiento del cliente. Por lo tanto esta ontología puede resultar útil para un próximo requerimiento, pero no para este. La ontología de observación tampoco resulta de utilidad para el presente proyecto, pero puede ayudar a satisfacer un requerimiento para el estudio de la actividad de los estudiantes en Moodle.

Using Ontologies for eLearning Personalization -- Paulo Gomes, Bruno Antunes, Luís Rodrigues, et al. (20)

Este artículo presenta un enfoque basado en ontologías para la personalización en e-learning. Para ello desarrolla un modelo de estudiante integrado con una ontología, lo que permite que el sistema de personalización guíe el proceso de aprendizaje del estudiante. Además, el modelo monitorea el progreso del estudiante para actualizar los conceptos que ha aprendido y decide qué conceptos debe aprender a continuación.

El modelo de estudiante maneja dos tipos de modelos: el de los datos estáticos, porque no se modifican durante la interacción del estudiante con el sistema, y el de los dinámicos, que sí se modifican según el progreso del estudiante y su interacción con el sistema. El modelo estático del estudiante incluye información personal, sobre su personalidad, cognitiva, pedagógica e información sobre sus preferencias. El modelo dinámico contiene información sobre el rendimiento del estudiante en los cursos y los conocimientos adquiridos por el estudiante.

Estas ontologías refieren a conceptos que sí aplican al dominio de la aplicación, pero no cubren los conceptos necesarios para el requerimiento actual. De todas formas, esta ontología puede resultar de interés para un futuro requerimiento que apunte a estudiar los perfiles de los estudiantes. Incorporando al presente proyecto los modelos de los estudiantes y los modelos de observación definidos en el modelo anterior, sería posible estudiar para cada perfil de estudiante, qué recursos utiliza con mayor frecuencia y para cuáles tipo de cursos obtiene mejores resultados.

Description of an Instructional Ontology and its Application in Web Services for Education - Carsten Ullrich (21)

Este artículo presenta una ontología para describir la función instruccional de recursos educativos, y explica la motivación para utilizar dicha ontología. La ontología propuesta describe un recurso educativo desde una perspectiva instruccional. La ontología no describe el contenido que es enseñado por el recurso, sino que cada clase de la ontología representa un rol instruccional (ejemplo, teorema, ejercicio). Es un concepto similar al de Learning Resource Type definido por LOM.

Esta ontología resulta muy interesante en el contexto de educación, ya que modela de alguna forma los objetivos instruccionales de los recursos, y puede aportar entendimiento a la forma

en que se arma un curso en base al tipo de recurso que tiene. Hay que señalar que algunas veces puede ser sencillo determinar el rol de un recurso, como un ejemplo o un ejercicio, pero puede no serlo tan intuitivo en otros casos.

Para el presente proyecto esta ontología no resulta de utilidad, pero sí puede resultar para satisfacer un próximo requerimiento, y puede potenciar sus resultados si se complementa con la ontología de grado de interacción desarrollada.

4.2. Aplicación de la fase II y el análisis del modelo

El objetivo principal de la presente fase es construir un modelo que represente, utilizando un modelo formal, los conceptos identificados en la Fase I detallados en el documento Metadata de Conceptos. El modelo incluye la información y características más importantes, relaciones definidas y las restricciones consideradas en el análisis del dominio, abstrayendo la realidad presentada y eliminando posibles inconsistencias y/o conflictos que no se identificaran en la Fase I.

Construir un modelo del dominio permite tener trazabilidad desde los conceptos de la realidad hasta las entidades del modelo conceptual multidimensional, y hasta la implementación concreta que permite satisfacer cada requerimiento.

Además, representar el dominio mediante una ontología, que es representable de forma gráfica, permite la validación con el cliente. Otra gran fortaleza de la ontología es que es posible aplicar razonadores para detectar inconsistencias en el modelo. Esto se vuelve cada vez más importante a medida que se agregan requerimientos y el modelo crece.

Al construir ontología de dominio dentro del marco de VGPCDW se tiene la restricción de que el modelo construido será utilizado por la fase III de la metodología, para realizar el modelo conceptual multidimensional. Esto es una restricción porque al momento de aplicar las reglas de la fase III, las clases de la ontología se convierten en niveles CMDM. De esta forma, al existir diferentes formas de modelar un mismo concepto, cada uno lleva aparejado un modelo multidimensional resultante diferente, y las decisiones que se tomen en esta fase pueden repercutir en el resto. VGPCDW no define lineamientos para afrontar este tipo de disyuntiva.

Se presentan a continuación tres modelos de dominio, que ofrecen diferentes grados de expresividad para modelar un concepto central del dominio (el grado de interacción de los cursos), y se analiza cómo los diferentes modelos pueden afectar la aplicación de la fase III.

Dado el relevamiento de dominio realizado, los conceptos sobre los que pueden surgir dudas son Semestre y Grado de interacción, que corresponden a dos dimensiones especiales, respectivamente a la dimensión tiempo y a una dimensión de medida. Ambas requieren ser

modeladas en CMDM como niveles y pero en la ontología pueden ser modelados tanto como clases o como atributos.

Respecto al Semestre, podría intentarse inicialmente representarlo como una clase de la ontología para conseguir un nivel en CMDM, pero esto es un poco forzado ya que claramente es un atributo de la clase Curso. De todas formas Semestre no plantea un problema porque VGPCDW considera reglas especiales para tratar las dimensiones de tiempo. Así que es modelado como un atributo de Curso.

Respecto al Grado de interacción de los recursos y cursos, existen tres posibilidades: 1) modelarlo como un atributo; 2) modelar el Grado de interacción como una clase y los grados de interacción posibles como subclases; 3) a la opción 2 agregarle una clase métrica que explicita cómo se realiza el cálculo de la medida Grado de interacción. Tiene sentido modelar el Grado de interacción como clase en la ontología, ya que con ello se lo explicita y se gana mucho en claridad y entendimiento por parte del usuario.

Se presentan a continuación tres ontologías que fueron construidas de forma iterativa, donde se modela de diferente forma el Grado de interacción de los recursos y cursos, donde cada una implementa una de las opciones.

La ontología 1 modela el Grado de interacción como un atributo de las clases Curso y Recurso. Este modelo tiene la desventaja de que no se explicita el concepto Grado de interacción que es central para los requerimientos. Además, al aplicar la fase III esto no genera un nivel de Grado de interacción. Se presenta en la figura 5.

La ontología 2 modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. De esta forma se explicita el concepto de Grado de interacción. Este modelo, que se presenta en la figura 6, tiene la limitación de que no se explicita el cálculo del Grado de interacción. Si sobre este modelo se aplican las reglas de la fase III, se crea el nivel Grado de interacción, que es necesario, pero se crea también un nivel para cada grado de interacción posible. Estos niveles sobran en CMDM, ya que realmente son instancias del tipo Grado de interacción, sería equivalente a tener los niveles Instituto e Instituto de Computación.

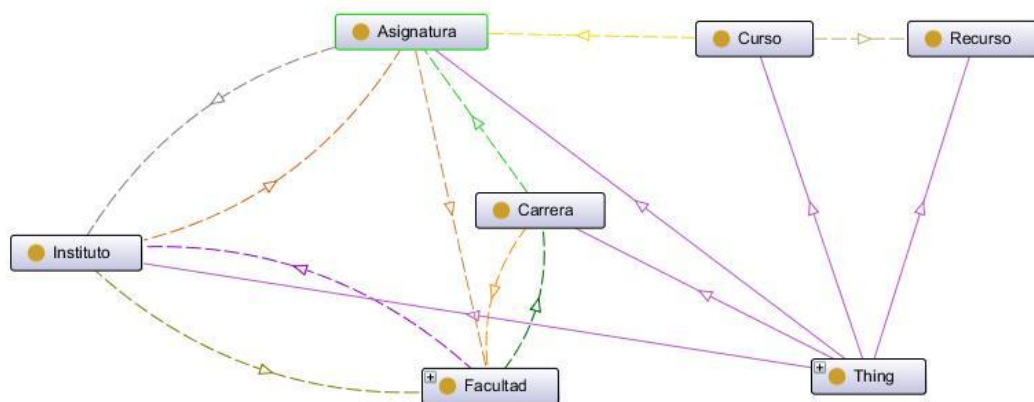


Figura 5 - Ontología 1. Modela el Grado de interacción como un atributo de Curso.

Concepto	RQ1	RQ2	RQ3
Clases			
Asignatura	√	√	√
Carrera			√
Curso	√	√	√
Facultad			√
Instituto			√
Recurso	√	√	√
Relaciones			
asignaturaCurso	√	√	√
asignaturasCarrera			√
asignaturasInstituto			√
carrerasFacultad			√
facultadAsignatura			√
facultadCarrera			√
facultadInstituto			√
institutoAsignatura			√
institutosFacultad			√
recursosCurso	√	√	√

Tabla 2 - Matriz de Trazabilidad de la ontología 1.

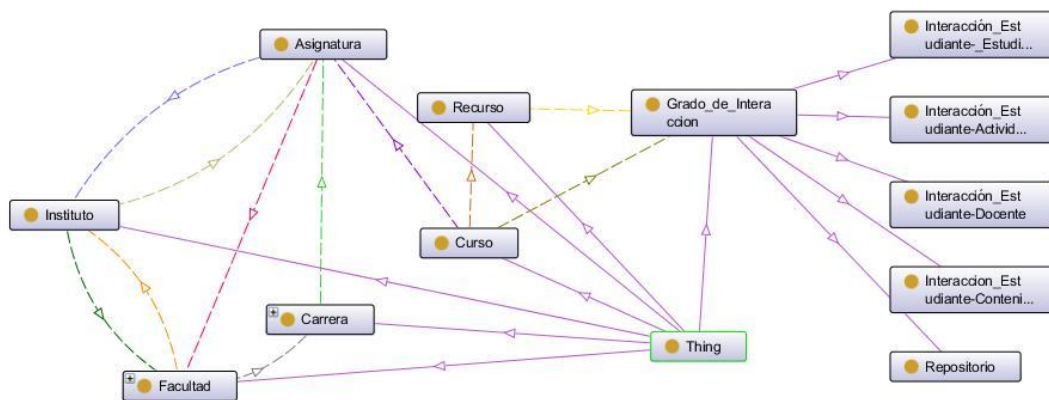


Figura 6 - Ontología 2. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases.

La ontología 3, que se muestra en la figura 7, pretende dejar explícito cómo se calcula el grado de interacción de un curso, y para ello agrega una clase “Métrica de Grado de Interacción” la cual define una fórmula basada en propiedades de los recursos para medir el grado de interacción. A esta clase se le dan tres propiedades, obtains, through y for. Estas propiedades indican qué calcula, cómo lo hace y para qué clase respectivamente. La figura 8 muestra un detalle de la ontología 3, donde se puede observar que la Métrica define el grado de interacción de un curso, basándose en el grado de interacción de sus recursos.

Como modelo del dominio, la ontología 3 es la más explícita. Sin embargo, al aplicar las reglas de la fase III también genera clases innecesarias para el modelo multidimensional.

Dado lo expuesto anteriormente, ninguna de las ontologías garantiza una posterior construcción correcta del modelo multidimensional. Esto se debe realmente a que VGPCDW no presenta reglas para construir las medidas de las relaciones dimensionales en CMDM. Además, VGPCDW no presenta lineamientos para modelar conceptos centrales del dominio. Por lo tanto, dado que la Ontología 1 no genera niveles innecesarios, este proyecto utiliza para seguir con la aplicación de VGPCDW.

En la tabla 2 se presenta la Matriz de Trazabilidad asociada a la ontología 1 de la figura 5. Esta matriz “refleja la interacción entre los conceptos y relaciones del modelo de dominio, y los requerimientos analizados” (3).

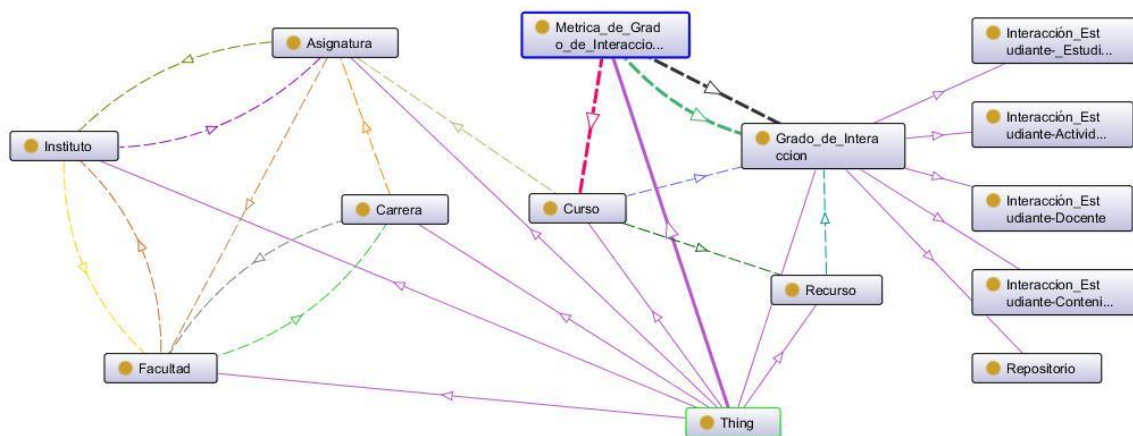


Figura 7 - Ontología 3. Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. Además, la Métrica de Grado de interacción muestra cómo es calculado.

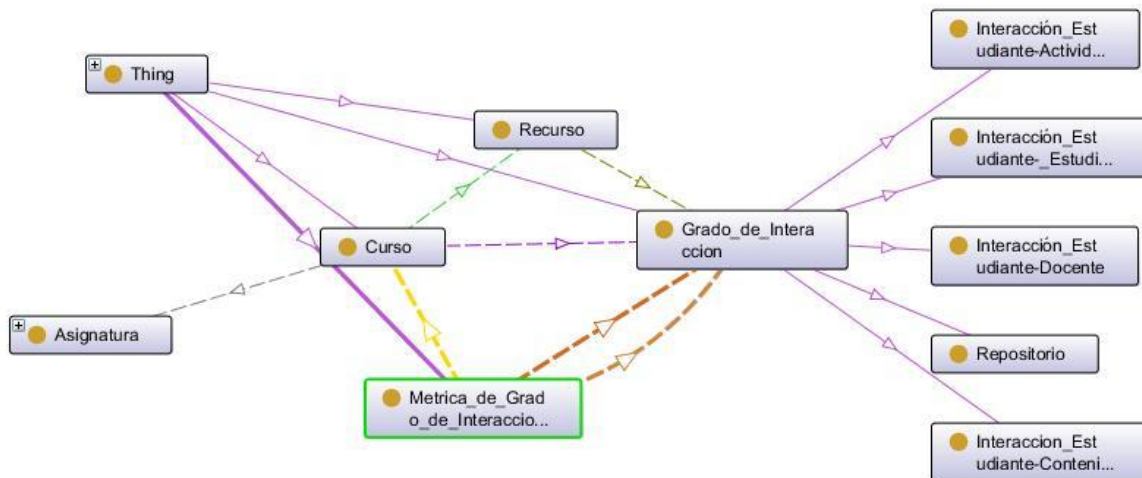


Figura 8 - Detalle de la ontología 3.

4.3. Fase III (Modelado conceptual multidimensional)

“El objetivo de la presente fase es la construcción del modelo conceptual multidimensional del Data Warehouse a partir del modelo de dominio” (1). Realizar un modelo conceptual permite abstraerse del modelo relacional en que se implemente la solución, así como de las herramientas con las que se trabaje. Para cumplir el objetivo se siguen once reglas que guían la construcción del modelo conceptual mediante la definición de: 1) los niveles, 2) las jerarquías, y 3) las dimensiones y relaciones dimensionales.

Es en las reglas de la fase III donde ocurre el problema más importante de la metodología. Las reglas de la fase III evidencian que VGPCDW confunde el enfoque multidimensional de CMDM con el enfoque de estrella. En particular no define las dimensiones que son las medidas de las relaciones dimensionales. Como consecuencia, el resultado de la aplicación de las reglas de la fase III es un modelo que es sintácticamente incorrecto.

El primer problema que se encuentra en esta fase es en la construcción de dimensiones. Las reglas llevan a construir las dimensiones que se muestran en la figura 9. Estas dimensiones tienen todas un nivel en común, el más bajo. Este nivel en realidad corresponde a la medida de la relación dimensional y no debe ubicarse dentro de las otras dimensiones.

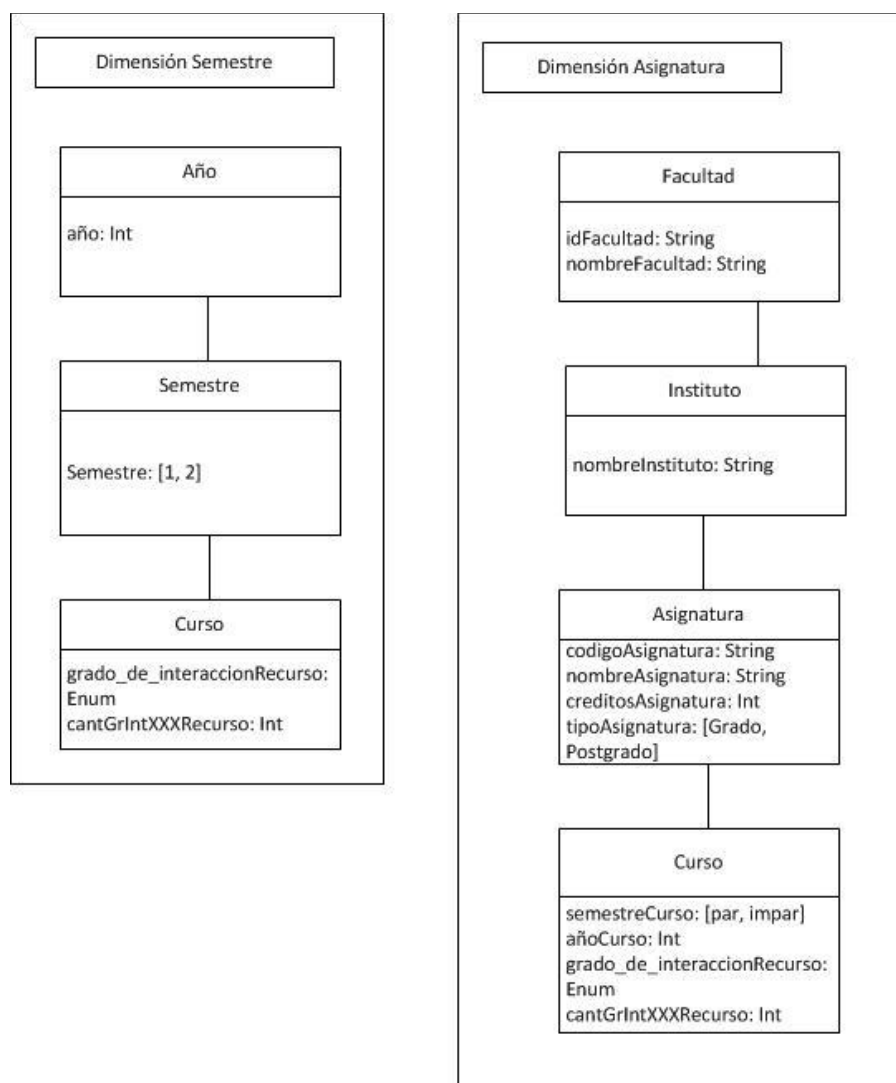


Figura 9 - Dimensiones en CMDM generadas aplicando VGPCDW.

Esta construcción carece de sentido en CMDM ya que estas dimensiones así pueden interpretarse como una sola dimensión con diferentes jerarquías como se muestra en la figura 10, y en este caso no se precisa una relación dimensional ya que no habría otras dimensiones.

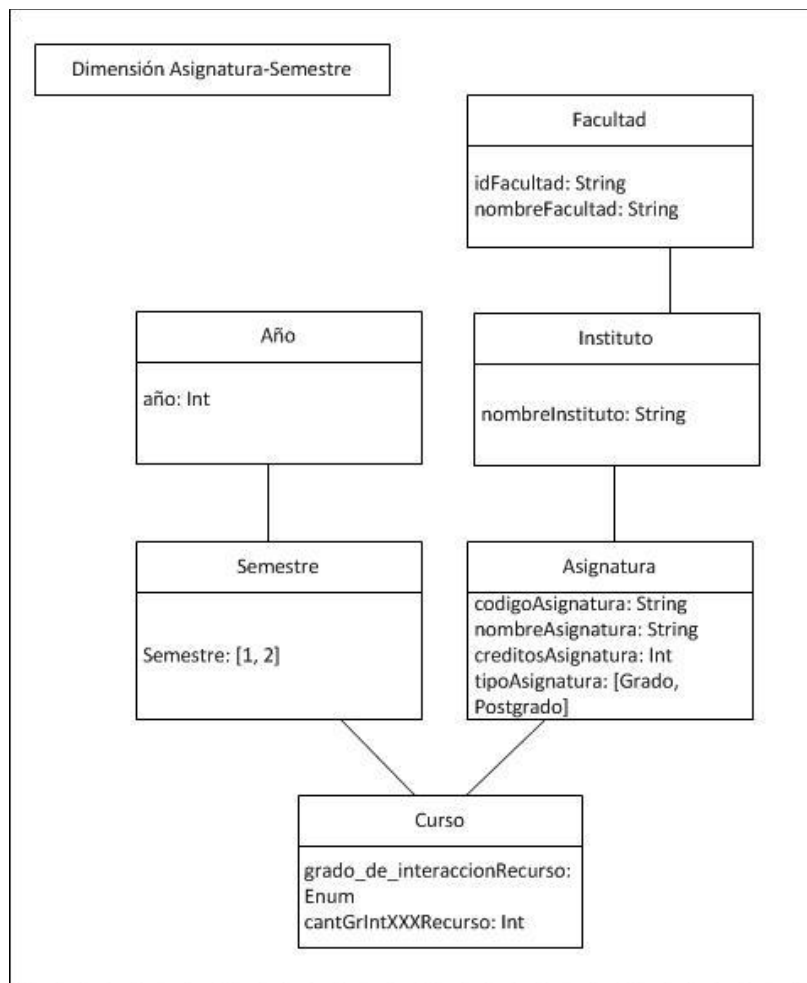


Figura 10 - Dimensiones Asignatura y Semestre representadas como una sola ya que comparten el nivel inferior.

Para el caso de estudio este modelo lleva a la construcción de una relación dimensional que no permite satisfacer los requerimientos. La relación dimensional no permite representar correctamente la realidad ya que no permite expresar el cruzamiento definido por el requerimiento 2: “por asignatura y por semestre y año”. Esto debido a que ambas dimensiones se construyen como una sola.

Otro problema de esta construcción es que no incluye la dimensión que ha de ser la medida de la relación dimensional. Tampoco plantea reglas para extraer de la ontología propiedades que deben ser modeladas como niveles en la jerarquía de la dimensión medida. De esta forma se pierde la propiedad de CMDM de dimensionalidad genérica, y la posibilidad de plantear agregaciones sobre las medidas.

En el presente caso de estudio el modelo construido no permite visualizar el Grado de interacción como una medida de la relación dimensional, y tampoco como parte de una jerarquía que parte del nivel Curso. Para satisfacer el requerimiento 3 y contar la cantidad de

cursos con cada grado de interacción para cada instituto y cada año, es necesario convertir el nivel Grado de interacción de medida a dimensión, realizando una agregación desde el nivel Cursos al nivel Grado de interacción, y considerar una nueva medida de tipo cantidad.

Para satisfacer el requerimiento se debe construir el nivel Grado de Interacción y no tratarlo como un atributo del nivel Curso. De esta forma se construye una jerarquía con los niveles Curso y Grado de Interacción, como se muestra en la figura 11. De esta forma se permite agregar cursos según sus grados de interacción

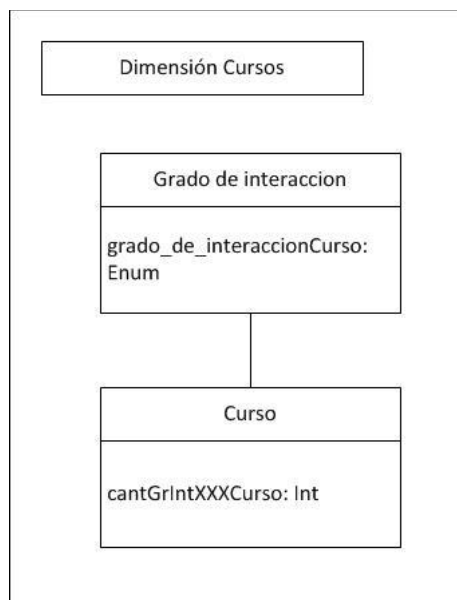


Figura 11 - Dimensión Cursos, tiene el Grado de interacción como un nivel en la jerarquía de Cursos permitiendo realizar agregaciones desde el nivel inferior.

Luego de definidas las dimensiones VGPCDW plantea la regla para definir las relaciones dimensionales y aquí se explica el por qué de los problemas mencionados anteriormente. El problema de fondo de la metodología es que confunde el enfoque multidimensional de CMDM con el enfoque de estrella, en el que existe una clase de hechos que es el centro de la relación, vinculando a las dimensiones.

La Regla 11 asocia el nivel más bajo de las dimensiones con “el eje central de la relación” (1) de CMDM. Pero en CMDM ningún nivel es asociado a una relación dimensional, ya que la relación dimensional “representa un conjunto de cubos, tomado del conjunto de todos los cubos que se pueden construir a partir de los niveles de un conjunto dado de dimensiones” (9). Por lo tanto el modelo construido es sintácticamente incorrecto, y sobre el cual no es posible de aplicar el algoritmo de Verónica Peralta.

Además de la confusión al proponer la definición de la relación dimensional, la metodología no contempla la construcción de relaciones dimensionales que tengan medida en los booleanos. Estas son las relaciones dimensionales que modelan relaciones de cardinalidad N:N.

En el caso de estudio, las reglas no contemplan cómo tratar la relación entre Asignaturas y Carreras, que tiene cardinalidad N:N. Por lo tanto VGPCDW tampoco permite satisfacer los requerimientos que involucran al concepto Carrera.

La relación entre las Carreras y las Asignaturas es una relación con cardinalidad N:N, por lo que debe modelarse según CMDM con una relación dimensional sin medida como se muestra en la figura 12.

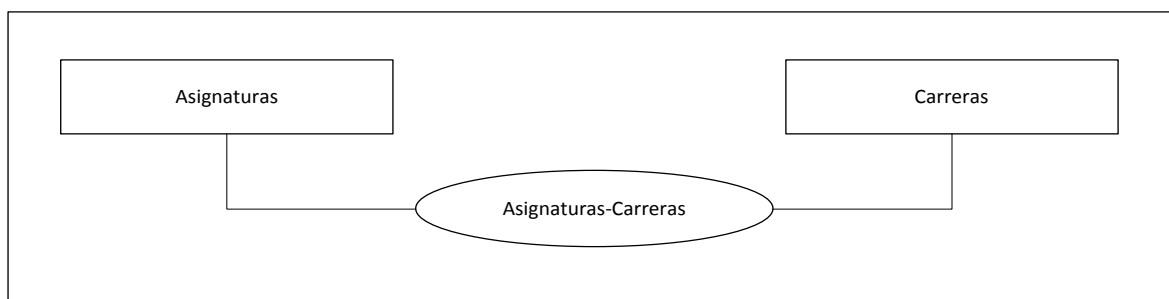


Figura 12 - Relación dimensional mostrando la asociación entre Asignaturas y Carreras.

Por lo tanto, la aplicación de la fase III de VGPCDW evidencia que la metodología no comprende completamente a CMDM ya que formula reglas que no son válidas para construir las relaciones dimensionales y pasa por alto las medidas de las relaciones, y además no contempla casos importantes como las relaciones de cardinalidad N:N.

Para resolver estos problemas VGPCDW debe:

- 1) Contemplar reglas especiales para la construcción de dimensiones de tipo medida. Estas podrían ser como las que ya contempla para dimensiones de tipo tiempo.
- 2) Redefinir la regla 11 de construcción de relaciones dimensionales. Esta debe considerar cuáles serán las dimensiones medida de las relaciones a construir. Podría considerar como candidatas a las dimensiones determinadas por los niveles en común entre las dimensiones, notando que el nivel que sea medida no debe pertenecer a las demás dimensiones.
- 3) Al construir las relaciones dimensionales, considerar también las relaciones de cardinalidad N:N para construir dimensiones con medida en los booleanos.

Como resultado de aplicar las reglas de la fase III se tiene un modelo conceptual multidimensional que no permite satisfacer los requerimientos.

Para poder seguir evaluando la metodología se corrigió el modelo para que atienda a todos los requerimientos. Para ello se definió una jerarquía para la dimensión medida, y se agregó una segunda relación dimensional que vincule las dimensiones Carreras y Asignaturas. En la figura 13 se exponen las dimensiones del modelo resultante y en la figura 14 las relaciones dimensionales.

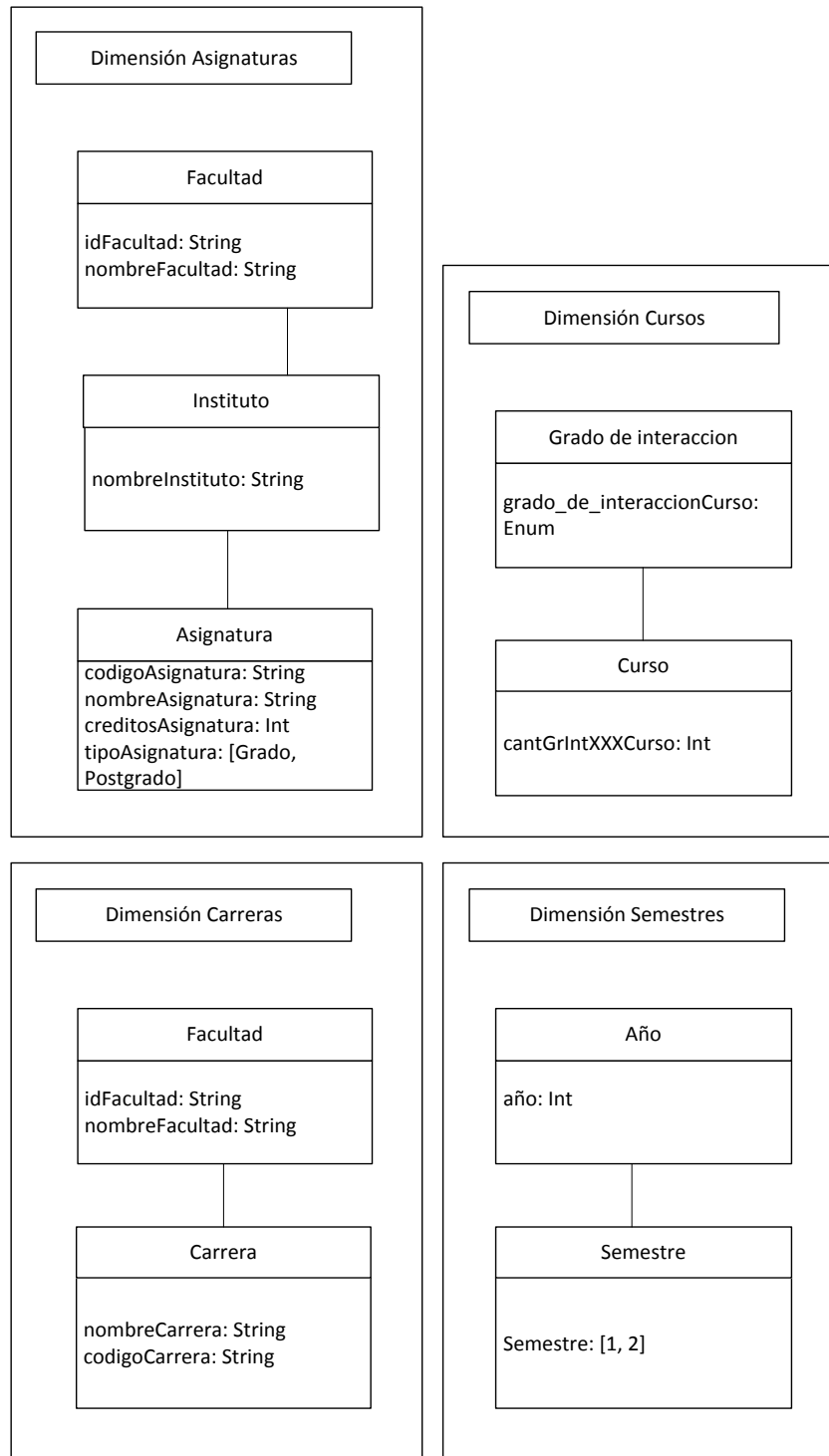


Figura 13 - Dimensiones del modelo conceptual multidimensional corregido.

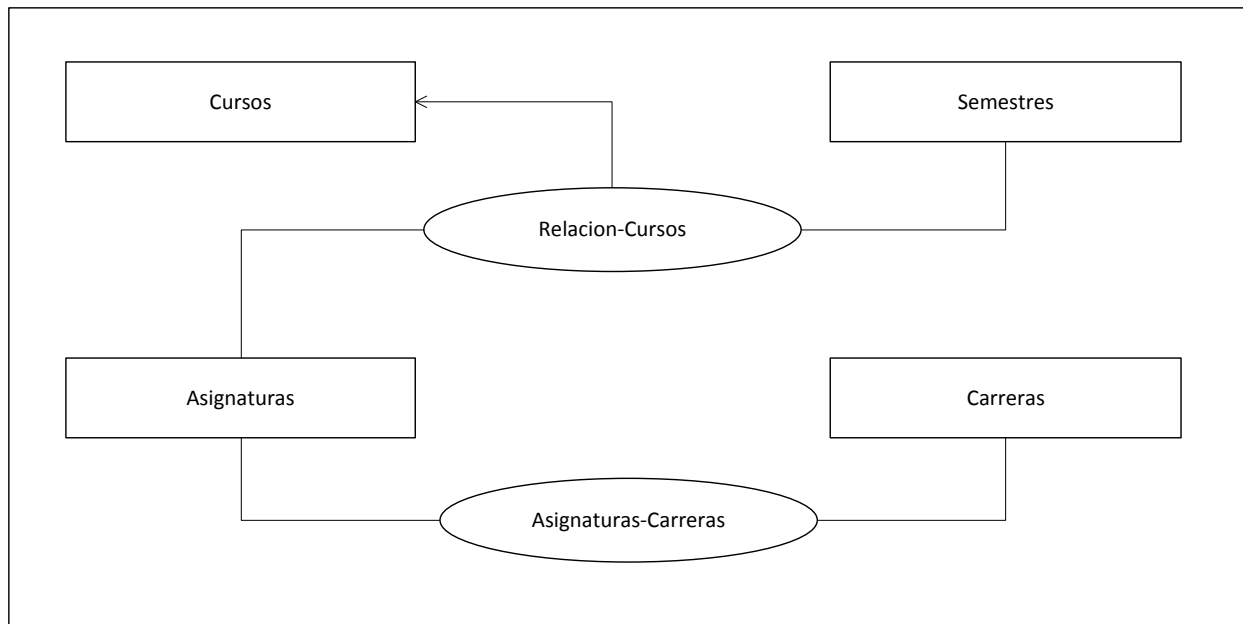


Figura 14 - Relaciones dimensionales del modelo conceptual multidimensional corregido.

El modelo construido tiene dos relaciones dimensionales: Relación-Cursos y Asignaturas-Carreras. La relación dimensional Relación-Cursos vincula las dimensiones Asignaturas y Semestres, y tiene como medida la dimensión Cursos. Con esta relación es posible dado un semestre y una asignatura obtener el curso correspondiente. Además, gracias a la dimensionalidad genérica y a que la dimensión Cursos tiene definida una jerarquía, es posible realizar una operación de agregación sobre los cursos.

La relación Asignaturas-Carreras modela la relación entre Asignaturas y Carreras. Esta relación tiene como medida los booleanos ya que su función es indicar si una Asignatura está asociada a una Carrera o no.

El requisito de visualizar los cursos dictados para una Carrera se satisface obteniendo de la relación Asignaturas-Carreras las Asignaturas asociadas a cada Carrera, y utilizando estas Asignaturas como entrada para la relación dimensional Relación-Cursos.

En base a este modelo es que se aplica el resto de la metodología.

4.4. Fase IV (construcción del ODS)

La fase IV “tiene como objetivo generar una fuente de datos relacional única que cubre toda la realidad” (1). Para ello VGPCDW define un algoritmo denominado Onto2rdb que define una serie de reglas que guían la definición del esquema del ODS.

La aplicación de estas reglas resultó intuitiva y sencilla, y fueron aplicadas con éxito.

De todas formas, el algoritmo no contempla el proceso de extracción, transformación y carga (ETL) del ODS. Este proceso es en definitiva el proceso ETL del Data Warehouse ya que la carga del Data Warehouse se realiza mediante vistas desde el ODS.

El proceso de ETL es una de las fases más críticas en la construcción de un Data Warehouse, ya que es en esta fase que se extraen y procesan los datos que conformarán la base del sistema. Además, es un proceso complejo que según algunos autores puede consumir hasta el 80% de un proyecto (22).

Si al momento de cargar los datos al ODS surgiesen dificultades y no fuese posible cargar algunos datos, esto podría, en el peor de los casos, hacer inviable el modelo multidimensional construido en la fase anterior. Esto podría suceder a pesar de los recaudos tomados en la fase I de relevamiento de requerimientos al identificar las fuentes de los datos. Esto es una debilidad de la metodología VGPCDW.

De todas formas se debe señalar como una fortaleza el uso de una fuente de datos intermedia, ya que la misma protege al Data Warehouse de cambios en las fuentes. Ante un cambio en los sistemas transaccionales solamente se debe modificar el sistema de carga del ODS (suponiendo que se sigan teniendo todos los datos necesarios), con lo que el ODS y el Data Warehouse permanecen inalterados.

4.5. Fase V (diseño y carga del Data Warehouse)

La fase V de VGPCDW “tiene como objetivo generar el diseño lógico y la carga del Data Warehouse a partir de la fuente canónica y el modelo CMDM” (1).

La última fase de la metodología aplica una serie de etapas donde: primero se definen lineamientos para los cubos que se quieren construir y los fragmentos de dimensión que se quieren definir. Luego se definen las correspondencias entre la base de datos fuente (el ODS) y el modelo conceptual. Finalmente se aplican quince pasos para definir el esquema relacional del Data Warehouse mediante vistas sobre el ODS, que son refinadas de forma sucesiva en cada paso.

Esta fase se pudo aplicar con éxito para satisfacer solamente una parte de los requerimientos. Esto se debe a que la metodología no contempla la construcción de cubos en los que sea necesario vincular varias relaciones dimensionales para construirlo.

VGPCDW carece de reglas que permitan aplicar operaciones primitivas del estilo de New-dimension-crossing (como la define Marotta (23) en su tesis, trabajo en que se basa la tesis de Peralta), operación necesaria para satisfacer los requerimientos.

Este es el caso del cubo GradoDeInteraccionCarrera representado en la figura 15. Este cubo contiene la cantidad de cursos dictados para cada Carrera, Año y Grado de Interacción. Para definir este cubo es necesario aplicar una operación del estilo de New-dimension-crossing desde la dimensión Carreras de la relación dimensional Asignaturas-Carreras, utilizando la dimensión Asignaturas como puente.

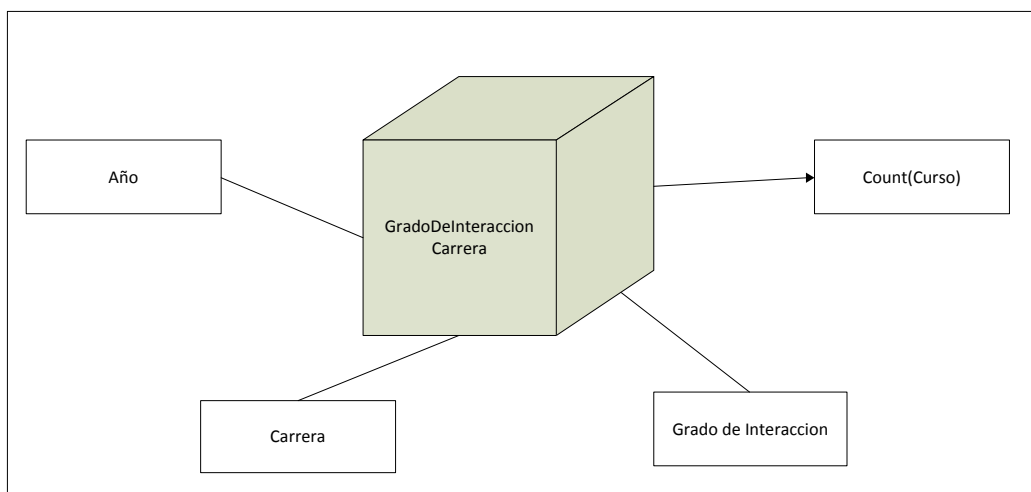


Figura 15 - Cubo GradoDeInteraccionCarrera. Contiene para cada Carrera, Año y Grado de Interacción, la cantidad de cursos dictados.

VGPCDW únicamente indica cómo construir cubos asociados a una sola relación dimensional, ya sea con el menor nivel de granularidad, o haciendo agregaciones subiendo por las dimensiones (aplicando reglas de hierarchy-drill -up o total-drill-up). Con estas reglas es posible satisfacer el resto de los requerimientos.

Una vez contruidos todos los cubos, se actualiza la Matriz de Trazabilidad. Para ello se toma la Matriz obtenida en la fase II y se agrega una columna para cada cubo a la derecha de los requerimientos. El resultado se muestra en la tabla 3.

Por último VGPCDW no aborda el problema de la implementación concreta del Data Warehouse, no contemplándose aspectos de la optimización de las consultas al Data Warehouse.

VGPCDW utiliza como solución la materialización de las vistas, pero esto no es posible en la mayoría de los sistemas DBMS, al menos no lo es en los sistemas open source (24).

Además, no siempre es posible generar las tablas mediante vistas. La fase V de VGPCDW propone 15 pasos para llegar al diseño lógico del Data Warehouse, en los que secuencialmente se van modificando las vistas que definirán el esquema resultante. El paso 6 es en el que se definen las claves de las tablas de dimensión. Al aplicar este paso del algoritmo, es necesario que las claves se encuentren dentro de los atributos existentes en el ODS. Pero en el caso de

que se requiera una clave artificial, la misma debe crearse y esto no puede hacerse mediante una vista, al menos no en MySQL (25) que es el DBMS utilizado.

Concepto	RQ1	RQ2	RQ3	Cubo Curso Semestre Asignatura	Cubo Carrera Asignatura	Cubo Grado De Interacción Instituto	Cubo Grado De Interacción Carrera
Clases							
Asignatura	✓	✓	✓	✓	✓	✓	✓
Carrera			✓		✓		✓
Curso	✓	✓	✓	✓		✓	✓
Facultad			✓		✓	✓	✓
Instituto			✓			✓	
Recurso	✓	✓	✓	✓			
Relaciones							
asignaturaCurso	✓	✓	✓				
asignaturasCarrera			✓				
asignaturasInstituto			✓				
carrerasFacultad			✓				
facultadAsignatura			✓				
facultadCarrera			✓				
facultadInstituto			✓				
institutoAsignatura			✓				
institutosFacultad			✓				
recursosCurso	✓	✓	✓				

Tabla 3 - Matriz de Trazabilidad, incluyendo los cubos construidos.

Por lo tanto, la solución de las vistas no es aplicable en todos los casos.

En el caso de estudio se observa esta situación en la definición de la dimensión Semestres. Luego de aplicar el paso 2, en el que se definen los nombres de los atributos, se obtiene la siguiente tabla:

DWSemestre(semestre, año)

y al llegar al paso 6 la forma de definir la clave es agregar una clave artificial. Esto se debe a que la clave de DWSemestre no se mapea desde la fuente de datos ODS, cosa que sí sucede en el resto de las dimensiones.

La solución que se encontró es definir la tabla con los atributos requeridos, entre ellos la clave, definida como un entero autoincremental; y cargar la tabla mediante una consulta. Esto agrega

otras dificultades a la solución, ya que se debe incluir un mecanismo para actualizar esta tabla cada vez que se agreguen nuevos datos.

Una posible solución general es no utilizar vistas, sino crear las tablas del Data Warehouse, con los índices que se consideren necesarios, y al momento de cargar nuevos datos actualizar el contenido de las tablas. Esta carga de datos se puede realizar con la consulta utilizada para definir las vistas.

De esta forma la metodología podría ser extendida brindando lineamientos para la creación de índices para la optimización de las consultas.

4.6. Extensión de la Matriz de Trazabilidad

Después de creados los cubos, este proyecto crea una capa de presentación sobre una herramienta de Business Intelligence, lo que se presenta en el capítulo 5. El objetivo de ello es facilitar el acceso a la información al usuario final. Utilizando esta herramienta de Business Intelligence se crean artefactos, que pueden ser reportes, cubos OLAP, dashboards, entre otros, que son utilizados por el usuario para acceder a la información. En esta etapa el presente proyecto propone extender la Matriz de Trazabilidad para que abarque también a estos artefactos.

La extensión propuesta consiste en agregar en la Matriz de Trazabilidad, una fila para cada uno de los artefactos construidos, que se cruzará con las columnas de requerimientos y cubos. De esta forma es posible indicar cuáles requerimientos ayuda a satisfacer cada artefacto, y de cuáles cubos se nutre para conseguirlo. Esto se muestra en la tabla 4.

	Requerimiento 1	Requerimiento n	Cubo 1	Cubo n
Concepto 1	√		√	
Concepto n		√		√
Artefacto 1	√		√	
Artefacto n		√		√

Tabla 4 - Modelo propuesto para Matriz de Trazabilidad, incluyendo artefactos de la capa de presentación.

Esto permite tener la trazabilidad completa, partiendo de los requerimientos, pasando por los cubos construidos en la base de datos, llegando a los artefactos construidos en la herramienta específica. De esta forma se gana en flexibilidad ya que se extiende la trazabilidad hasta la herramienta de Business Intelligence utilizada.

La aplicación de esta extensión se presenta en el capítulo 5, luego de construidos los artefactos de la capa de presentación.

4.7. Conclusiones de la metodología

La metodología VGPCDW tiene como objetivo guiar todo el proceso de construcción de un Data Warehouse, para ello propone cinco fases abarcando el relevamiento de requerimientos, la representación de la realidad de la organización, la generación del diseño conceptual, la construcción de una base de datos integrada (ODS), el diseño lógico y la carga de datos desde el ODS.

Se presenta un análisis esquemático en la tabla 5, donde cada fase es evaluada con las luces de un semáforo según cómo haya resultado la evaluación y aplicación de la misma. Además, se agrega un campo de observaciones donde se resumen las conclusiones.

	Evaluación	Observaciones
Fase I - Comprensión de la realidad		(+) Información estructurada (+) Temprana identificación de fuentes de datos
Fase II - Modelado de dominio		(+) Trazabilidad desde conceptos de la realidad (+) Uso de razonadores
Fase III - Modelado conceptual multidimensional		(-) No se definen medidas de relaciones dimensionales (-) No se definen relaciones dimensionales de medida booleana
Fase IV - Construcción de un Operational Data Store (ODS)		(+) El ODS protege al Data Warehouse ante cambios en las fuentes (-) No se contempla la carga del ODS
Fase V - Generación del diseño lógico y la carga de datos		(+) Generación semiautomática del esquema lógico (-) Existen casos particulares no contemplados (-) No contempla la implementación concreta del Data Warehouse

Tabla 5 - Análisis esquemático de la metodología Visión Global del Proceso de Construcción de Data Warehouse.

En este proyecto se identifican los problemas de la metodología, y aunque no se proponen soluciones generales, sí se sugieren posibles formas de solucionarlas y se construyen soluciones para el caso de estudio. Al aplicar la metodología y enfrentar estos problemas, estos son resueltos puntualmente para poder proseguir con las siguientes fases, y poder llevar a cabo la construcción del Data Warehouse.

También en el presente proyecto se propone que la herramienta de trazabilidad sea extendida para que abarque también a los artefactos utilizados en la capa de presentación de la solución, con lo que se consigue una trazabilidad completa hasta la herramienta de Business Intelligence utilizada.

Respecto a la metodología, en primer lugar hay que señalar que su aplicación no permite construir una solución que satisfaga los requerimientos. Es necesario corregir los problemas de las reglas del diseño conceptual multidimensional y plantear reglas adicionales para el diseño lógico, para que la metodología sea efectiva.

Además, es necesario extender la metodología para que contemple también la implementación concreta del Data Warehouse, tema que es de vital importancia a la hora de optimizar los tiempos de acceso a la información.

Por último, es necesario extender la metodología para que contemple el proceso de ETL del ODS, que en última instancia es el proceso de ETL del Data Warehouse (ya que el Data Warehouse se construye mediante vistas desde el ODS). El proceso de carga del ODS no es contemplado en VGPCDW actualmente, lo que se identifica como una debilidad.

Si se superan estos inconvenientes, la metodología guía la construcción una solución muy rica, que sí satisface los requerimientos. La solución incluye la trazabilidad de todo el proceso de construcción del Data Warehouse, partiendo de los requerimientos, pasando por el modelo multidimensional y llegando al diseño relacional. Esto permite que fácilmente se actualice el Data Warehouse, ya sea a partir de cambios en los requerimientos, o de nuevos requerimientos, sabiendo dónde repercuten los cambios.

La metodología construye una descripción del dominio, estructurada con metadatos, y un modelo formal del dominio de aplicación. Esto permite aplicar razonadores sobre el modelo, así como realizar la trazabilidad hasta los conceptos del dominio.

En la solución también se incluye un modelo multidimensional de los datos, que es independiente de su implementación (sea relacional o no). Además, el lenguaje utilizado - CMDM- permite una visión multidimensional de los datos con dimensionalidad genérica.

Esta rica documentación facilita el entendimiento y el uso de la solución, tanto por el usuario como por nuevos equipos de desarrollo, así como la extensión de la solución para satisfacer nuevos requerimientos.

Otra muy importante ventaja de la metodología es el diseño semiautomático del esquema relacional para el Data Warehouse.

Pero las ventajas de la metodología se obtienen a un alto costo. En primer lugar, la metodología demanda muchos conocimientos previos para poder ser aplicada. Es necesario previamente entender y manejar conceptos como ontologías, Data Warehouses y cubos, y el lenguaje CMDM; y para recién abordar el estudio de la metodología.

Además, la notación CMDM es compleja. A pesar de que el modelo se encuentra especificado perfectamente de forma lógica, los diagramas gráficos pueden prestarse a confusión, lo que

resulta patente en el proyecto MD4DW. También la notación para expresar las restricciones de integridad es compleja, y los proyectos que se utilizaron no trabajan con dichas restricciones.

Y estudiar la metodología no resulta menor ya que por la forma en que fue construida, como la integración de tres proyectos, es necesario estudiar los cuatro proyectos para obtener una visión total de la misma y conseguir entenderla completamente. Cada uno de los proyectos anteriores abarca una fase del proceso y VGPCDW los integra haciéndoles modificaciones; pero el foco de VGPCDW está puesto en los ajustes que hacen posible la integración y no en presentar la metodología de forma íntegra.

Por último, la documentación resultante, tan extensa y rica, resulta muy pesada de mantener. Ante cualquier pequeño cambio en los requerimientos, se deben actualizar muchos documentos y diagramas (que particularmente demandan mucho trabajo) para mantener toda la documentación consistente y actualizada. La documentación aporta mucho valor, pero al ser tantos documentos, es fácil que ésta se desactualice y pierda su valor.

5. Prototipación

El principal objetivo del presente proyecto es evaluar la metodología VGPCDW y para ello se la aplica a un requerimiento de la UE. Una vez aplicada la metodología se llega a un esquema relacional para el Data Warehouse, un esquema para una base de datos intermedia ODS y un procedimiento de carga del Data Warehouse desde el ODS. Además se cuenta con toda la documentación asociada al proyecto, que incluye entre otras cosas: un modelo de dominio, un modelo conceptual multidimensional, y una herramienta para realizar la trazabilidad desde los conceptos hasta el Data Warehouse.

Durante la aplicación de la metodología fue necesario estudiar y aplicar algunas tecnologías. En primer lugar fue necesario estudiar la aplicación Moodle que es una de las fuentes de datos, ya que no se contaba con un experto de ese dominio. La fase II de la metodología refiere al modelado de dominio con una ontología sobre la que se aplican razonadores, por lo que se debió aplicar una herramienta de modelado de ontologías.

Luego de aplicada la metodología, este proyecto construye los elementos restantes para poder presentarle al cliente un sistema de información en funcionamiento. Estos elementos son: el ODS y el Data Warehouse, aplicaciones para realizar la carga del ODS desde las fuentes de datos, y una aplicación para que el usuario final acceda a la información del Data Warehouse.

En primer lugar se construyen los esquemas del ODS y del Data Warehouse en el DBMS MySQL (26).

Para realizar la carga del ODS se requiere extraer datos de dos fuentes distintas, de Bedelía y del Moodle de facultad. No existe en la base de datos de Moodle un atributo que vincule las dos fuentes de datos, en particular no existe un atributo de los cursos que los vincule a una asignatura. Por lo tanto es necesario que dicha asociación sea definida por el usuario. La aplicación de carga ofrece una interfaz para facilitarle esta tarea al usuario.

El sistema requiere por último una forma de acceder a la información de forma amigable para el usuario final. Para la aplicación de acceso al Data Warehouse se presenta un relevamiento de herramientas de Business Intelligence existentes, se selecciona una de ellas y sobre ella se construyen artefactos para facilitar el acceso a la información, que pueden ser reportes, cubos OLAP, etc. Se muestra cómo esta solución satisface los requerimientos del cliente.

Estos artefactos son incluidos en la Matriz de Trazabilidad, extendiéndola, como se propone en el capítulo 4.

Finalmente se presentan las conclusiones sobre el prototipo creado.

Se presentan a continuación: 1) las herramientas estudiadas y utilizadas durante la fase de diseño, 2) la aplicación de carga del ODS, 3) la aplicación de consulta del Data Warehouse para el usuario final, 4) la extensión de la Matriz de Trazabilidad, y 5) las conclusiones del prototipo.

5.1. Fase de diseño

5.1.1. Moodle

En la fase I de la metodología se estudia el dominio de la aplicación y se describen los conceptos que en él existen. En esta fase una de las tareas es identificar las fuentes de datos vinculadas a dichos conceptos.

Por lo tanto el estudio del sistema Moodle 1.9 fue necesario desde el principio del proyecto, ya que al momento de relevar los requerimientos es necesario saber si la información que pide el cliente realmente existe y se encuentra disponible.

Para obtener los datos de los cursos de Moodle fue necesario estudiar la estructura de las tablas del sistema ya que no se contaba con un experto del dominio. Como referencia se utilizó el sitio institucional de Moodle (27) y el libro Using Moodle (28).

Este trabajo resultó bastante complejo ya que Moodle utiliza alrededor de 200 tablas (12) y la documentación no cubre todos sus componentes. Se requirió por lo tanto de un enfoque de ensayo y error para entender el funcionamiento del sistema y cómo esto se refleja en la base de datos.

Para satisfacer el requerimiento de información del cliente, es necesario saber qué tipo de recursos y en qué cantidad, tienen los cursos de la plataforma Moodle.

Para representar esta realidad Moodle utiliza una tabla de cursos, y una tabla para cada tipo de recurso. La figura 16 presenta un detalle de las tablas course y quiz.

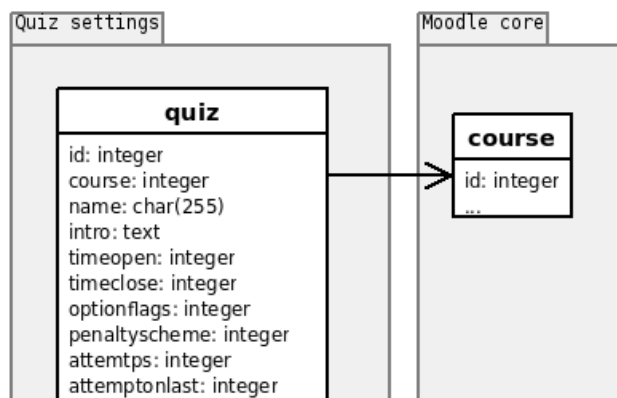


Figura 16 - Detalle de las tablas course y quiz de la base de datos de Moodle. Tomado de (12).

Para satisfacer los requerimientos del cliente también es necesario vincular los cursos de la plataforma Moodle con sus correspondientes asignaturas. Para facilitar esta tarea Moodle ofrece un atributo en los cursos denominado “Course ID number” que tiene la función de “proveer un link entre Moodle y el sistema de información de la institución” (28). Este atributo tiene la función de referenciar al identificador que la institución utilice en sus sistemas, ya sea para referenciar al curso o la asignatura.

Pero este atributo no es utilizado en Facultad de Ingeniería, por lo que esta asociación debe resolverse de otra forma. La solución acordada con el cliente es que dicha asociación se resuelva manualmente al momento de la carga.

5.1.2. Modelado de dominio

En la fase II la metodología propone realizar un modelo del dominio de aplicación utilizando una ontología. En particular, VGPCDW sugiere utilizar OWL 2 como lenguaje para representar la ontología.

OWL 2 es una extensión de OWL 1, con el que tiene compatibilidad hacia atrás completa. Una de las funcionalidades que agrega, que es el principal atractivo desde el punto de vista de este proyecto, es la posibilidad de definir claves para las clases.

Para construir la ontología se utilizó el editor Protégé (29), por contarse con experiencia previa con la herramienta. Se utilizó la versión 4.1 beta por ser el único release con soporte total para OWL 2 al momento de aplicar la fase (30).

Sobre las ontologías es posible aplicar razonadores para detectar inconsistencias. Para ello se utilizó la herramienta Hermit 1.3.1 que se encuentra incluida en el paquete instalador de Protégé.

5.2. *Aplicación de carga del ODS*

La aplicación de la metodología VGPCDW tiene entre sus resultados el esquema de una fuente de datos integrada denominada ODS (que se presenta en la figura 17), y un procedimiento de carga del Data Warehouse desde dicho ODS. Pero la carga de la base de datos ODS no es contemplada por la metodología. La solución a la carga del ODS se presenta en esta sección.

Para realizar la carga del ODS se requiere extraer datos de dos fuentes distintas, de Bedelía y del Moodle de facultad. De Bedelía se obtienen los datos de las asignaturas, las carreras y los institutos, así como qué instituto dicta cada asignatura, y a qué carreras pertenecen estas. Del

Moodle de facultad se obtienen los cursos dictados en la plataforma y los recursos que tienen, y en base a estos últimos se obtiene el grado de interacción de los cursos.

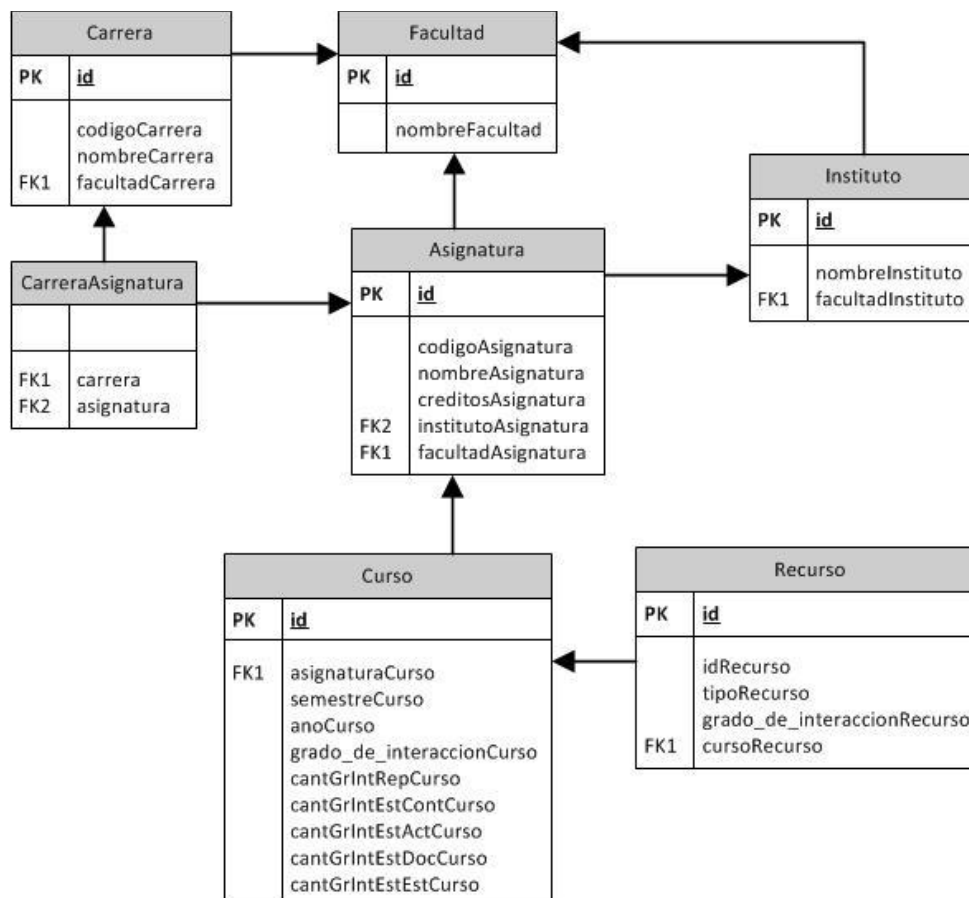


Figura 17 - Esquema de la base de datos integrada ODS.

Existe una limitante al momento de integrar las dos fuentes de datos, ya que no existe un atributo de los cursos que los vincule a una asignatura. Existe un atributo en Moodle que cumple esta función, pero no es utilizado (ver sección anterior). Tampoco es viable realizar una asociación por los nombres de las asignaturas ya que muchas veces asignaturas dictadas por institutos tienen nombres parecidos, o incluso tienen literalmente el mismo nombre. Este es el caso de la asignatura “Elementos de construcción”, dictada tanto por el Instituto de Agrimensura y el Instituto de Estructuras y Transporte.

Tampoco existe un atributo que los vincule a un semestre o a un año particular. Simplemente conociendo la asignatura no es posible determinar el semestre ya que existen asignaturas que se dictan en ambos. También es común que en la plataforma se dejen los cursos de años anteriores y si se quiere extraer esta información es necesario indicar a que año pertenecen.

Por lo tanto es necesario que para los cursos de la plataforma Moodle se asigne de forma manual, a qué asignatura pertenecen, y a qué semestre y año. La aplicación de carga ofrece una interfaz para facilitarle esta tarea al usuario.

A continuación se presenta para cada una de las fuentes de datos, el mapeo de la fuente al ODS y la aplicación desarrollada para ello. Primero se presenta la fuente de datos de Bedelía ya que se requiere que de estos datos para cargar los cursos de Moodle.

5.2.1. Datos de Bedelía

Para acceder a los datos de la primera fuente, Bedelía de facultad proporcionó un archivo de formato csv con el resultado de una consulta a su base, conteniendo información de las asignaturas, el instituto que la dicta, y a qué carreras pertenece. A continuación se muestra el esquema de la información brindada por Bedelía.

EsquemaDatosBedelia(NombreCarrera, CodigoCarrera, NombreAsignatura, CodigoAsignatura, Creditos, Instituto)

El mapeo al esquema del ODS se muestra en la figura 18. Faltaría el atributo que indique a qué facultad pertenecen las asignaturas y las carreras, pero al momento de realizar la carga se sabe que pertenecen a Facultad de Ingeniería. Este atributo existe porque se prevé la posibilidad de agregar datos de otras facultades.

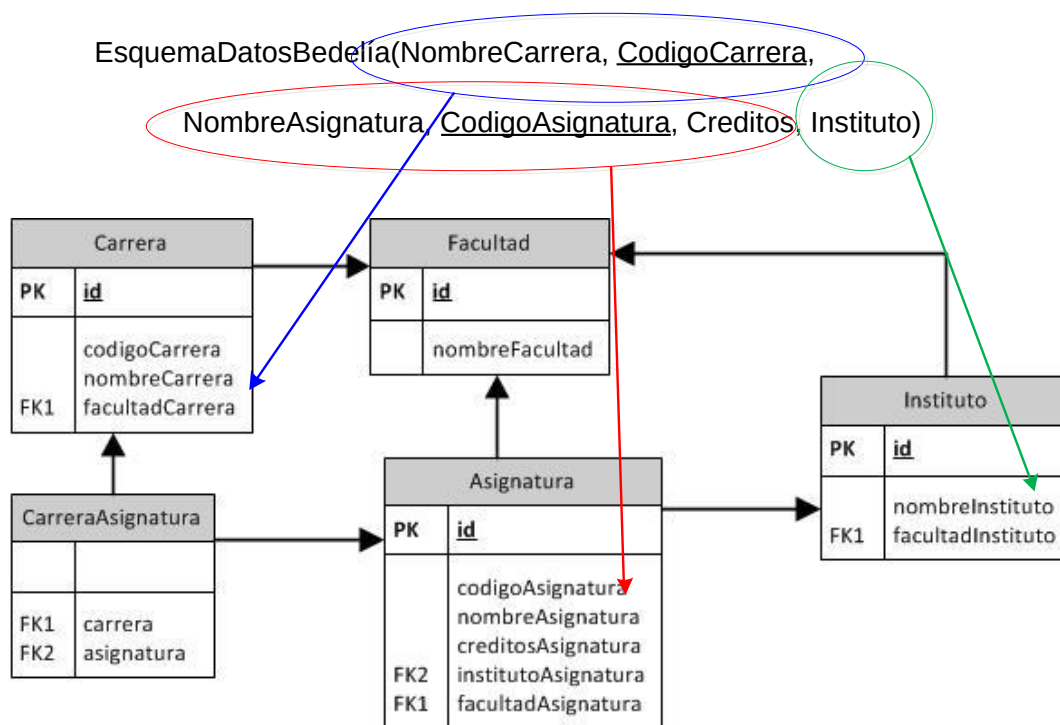


Figura 18 - Mapeo entre archivo de datos de Bedelía y esquema del ODS.

Para realizar el procesamiento de estos datos y la posterior carga al ODS se considera que una opción adecuada es la de un script sql. Ello debido a dos razones; primero la tarea es sencilla (simplemente cargar un archivo a la base de datos y sin más procesamiento distribuir estos datos en las tablas). Segundo, se requiere cargar estos datos una sola vez, que es al momento de la instalación del sistema. Luego estos datos permanecen prácticamente incambiados salvo alguna nueva asignatura que se cree o algún nuevo código que se designe. Así que no se considera que aporte el desarrollo de una aplicación con interfaz gráfica.

De todas formas, el script puede ser invocado por otra aplicación, lo que permite que eventualmente se agregue una capa de presentación para el usuario.

El script primero crea una tabla etl_bedelia con el mismo esquema de la fuente de datos y carga en ella todos los datos del archivo fuente (mediante el comando LOAD). Luego mediante consultas sobre etl_bedelia se cargan las tablas del ODS, considerando que todas las asignaturas pertenecen a Facultad de Ingeniería y que todas las claves (PK) son claves fantasía. Por lo tanto en primer lugar se debe crear la instancia de Facultad de Ingeniería en la tabla Facultad; luego se deben cargar las instancias de clases directamente vinculadas como son institutos y carreras; y recién después cargar las asignaturas. Al momento de cargar las asignaturas es necesario que en el ODS la referencia al instituto sea utilizando la clave fantasía.

Se presenta un detalle del código del script con comentarios:

```
-- Crear tabla etl_bedelia con esquema EsquemaDatosBedelia
Create table etl_asignaturas(...);

-- Cargar archivo de bedelia a etl_bedelia
LOAD DATA LOCAL INFILE '.../asignaturas.csv'
INTO TABLE etl_asignaturas

-- Mediante selects sobre etl_bedelia cargar las tablas del ODS
-- insertar instancia de Facultad de Ingenieria en tabla Facultad
insert into facultad
values ("fing", "Facultad de Ingenieria");

-- insertar institutos en tabla Instituto
insert into instituto(nombreinstituto, facultadinstituto)
select instituto, "fing" from etl_asignaturas group by instituto;

-- insertar asignaturas en tabla Asignatura
insert into asignatura (codigoasignatura, nombreasignatura, creditosasignatura,
facultadasignatura, institutoasignatura)
select etl.codasignatura, etl.nomasignatura, etl.creditos as creditosasignatura,
```



```
"fing" as facultadasignatura, ins.id -- ins.id es la clave fantasía de instituto
from etl_asignaturas etl, instituto ins
where ins.nombreinstituto = etl.instituto
group by codasignatura, nomasignatura;
```

5.2.2. Datos de Moodle

La carga del ODS desde Moodle plantea algunas dificultades, ya que Moodle es un sistema transaccional y su esquema relacional está pensado para soportar el dictado de cursos en la plataforma, y no el análisis de información que se quiere llevar a cabo. Por ejemplo, sobre los recursos almacena más información que la requerida para resolver los requerimientos y el concepto de Grado de interacción no existe en Moodle. Es por esto que el esquema de Moodle no coincide con el esquema realizado para representar el dominio, y por lo tanto es necesario realizar algunas transformaciones para realizar la carga del ODS.

En particular Moodle utiliza una tabla para cada tipo de recurso que puede existir en un curso, como se muestra en la figura 19. Dados los requerimientos, no es necesario diferenciar los recursos más que por el grado de interacción que ofrecen, según fue definido por el cliente. Por lo tanto, en el modelo de dominio realizado en la fase II de la metodología existe una clase recurso por cada grado de interacción posible, y el tipo de recurso es modelado con un atributo. Luego, en el esquema del ODS, se crea una única tabla donde se almacenan todos los recursos indistintamente, indicándose por atributos su grado de interacción y tipo de recurso.

Por lo tanto, para extraer los datos de los cursos, se debe, para cada curso, extraer los datos de sus recursos. Para extraer los recursos de cada curso es necesario iterar sobre cada una de las tablas de recursos, obteniendo los recursos asociados a ese curso. Se presenta a continuación un pseudocódigo.

```
foreach $curso {
    cargar $curso al ODS
    foreach tipo_de_recurso {
        obtener $recursos de tabla(tipo_de_recurso) asociados a $curso
        cargar $recursos al ODS asociados a $curso
    }
    calcular grado de interacción de $curso, cargarlo al ODS
}
```

Al momento de cargar los cursos en el ODS están faltando los atributos de curso: asignatura, semestre y año, que no pueden obtenerse de Moodle como se explica en la sección anterior. Estos atributos son ingresados por el usuario en una aplicación diseñada para este fin.

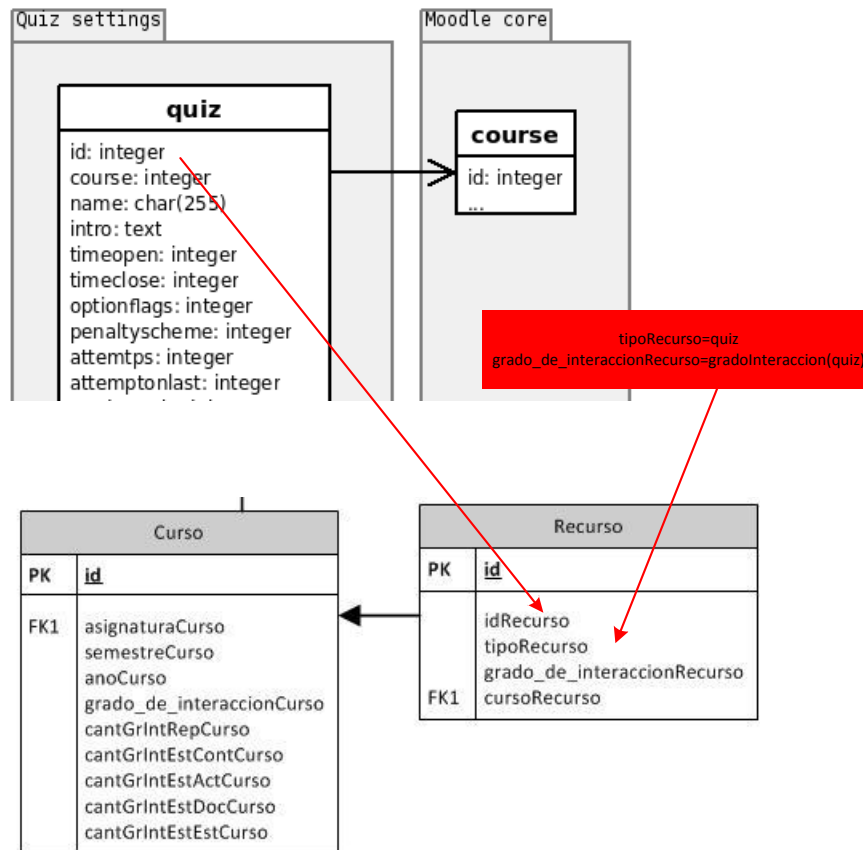


Figura 19 - Mapeo entre conceptos de Moodle y el ODS, y cómo se completan los datos Recurso.tipoRecurso y Recurso.grado_de_interaccionRecurso.

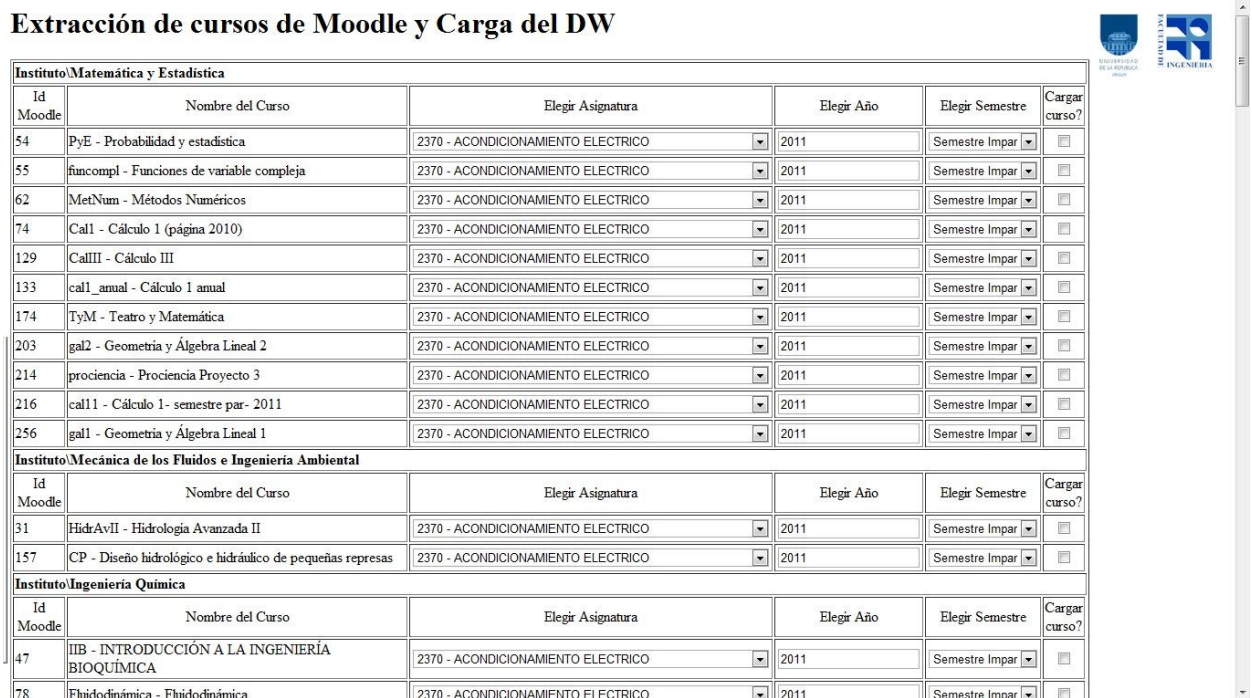
Este procedimiento de carga, a diferencia del de los datos de bedelía, requiere ser ejecutado periódicamente para ingresar datos de nuevos cursos. Por lo tanto, para esta aplicación se considera necesario contar con una interfaz gráfica para el usuario.

Por lo tanto se desarrolló una aplicación de tipo web utilizando PHP 5.3.1. Esto por un lado continúa la filosofía de la herramienta de BI seleccionada, que brinda una forma de acceso tipo web. De esta forma se presenta al usuario una forma de acceso similar en todas las etapas. Además, que la aplicación sea de acceso web permite que el usuario trabaje desde cualquier equipo sin necesidad de instalarse una aplicación. Se utilizó PHP por contar previamente con experiencia en el lenguaje, y por ser el mismo adecuado para el desarrollo de aplicaciones web (31) y ampliamente utilizado en la industria (32).

Esta aplicación tiene un diseño en dos capas, con su lógica totalmente independiente de la presentación. Esto permite que se pueda modificar su presentación fácilmente, ya sea para mejorar la apariencia de la página o convertir el programa en una aplicación de escritorio.

La interfaz de la aplicación se muestra en la figura 20. La aplicación permite que el usuario se conecte a la base de datos de Moodle y elija cuáles cursos desea ingresar al Data Warehouse entre los existentes en la plataforma, y a estos les asigne un semestre y año, y una asignatura.

Extracción de cursos de Moodle y Carga del DW



Instituto/Matemática y Estadística					
Id Moodle	Nombre del Curso	Elegir Asignatura	Elegir Año	Elegir Semestre	Cargar curso?
54	PyE - Probabilidad y estadística	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
55	funcompl - Funciones de variable compleja	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
62	MetNum - Métodos Numéricos	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
74	Cal1 - Cálculo 1 (página 2010)	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
129	CalIII - Cálculo III	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
133	cal1_anual - Cálculo 1 anual	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
174	TyM - Teatro y Matemática	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
203	gal2 - Geometría y Álgebra Lineal 2	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
214	prociencia - Prociencia Proyecto 3	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
216	cal11 - Cálculo 1- semestre par- 2011	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
256	gal1 - Geometría y Álgebra Lineal 1	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
Instituto/Mecánica de los Fluidos e Ingeniería Ambiental					
Id Moodle	Nombre del Curso	Elegir Asignatura	Elegir Año	Elegir Semestre	Cargar curso?
31	HidrAvII - Hidrología Avanzada II	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
157	CP - Diseño hidrológico e hidráulico de pequeñas represas	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
Instituto/Ingeniería Química					
Id Moodle	Nombre del Curso	Elegir Asignatura	Elegir Año	Elegir Semestre	Cargar curso?
47	IIB - INTRODUCCIÓN A LA INGENIERÍA BIOQUÍMICA	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐
78	Fhuidodinámica - Fhuidodinámica	2370 - ACONDICIONAMIENTO ELECTRICO	2011	Semestre Impar	☐

Figura 20 - Captura de la aplicación web para la carga de los cursos Moodle al ODS.

La aplicación permite que el usuario se conecte a la base de datos de Moodle, que puede ser local o remota. Esto permite que en una futura instancia la aplicación sea extendida para conectarse al EVA Moodle de diferentes facultades. Para ello es necesario previamente cargar en el ODS las asignaturas de dicha facultad. En este caso, al momento de realizar las búsquedas de asignaturas es necesario agregar filtros por facultad, lo que ya se encuentra previsto en el diseño de la aplicación mediante parámetros que recibe la lógica al ser invocada.

La aplicación también permite además levantar los datos de un archivo dump de la base de datos de Moodle, para lo cual, simplemente levanta los datos a una base de datos local de forma temporal, y se conecta a ella. Esta es la funcionalidad que se utilizó en este proyecto ya que la forma de acceso a los datos que se proveyó fue mediante un dump de la base.

5.3. Aplicación de consulta del Data Warehouse

Para que el usuario pueda consultar los datos de forma gráfica y mucho más amigable, es necesario agregar sobre la base de datos del Data Warehouse una capa de presentación. Esto se realiza con una herramienta de Business Intelligence.

Para esto primero se realizó un relevamiento de las herramientas open source existentes, estudiando de la documentación y bibliografía existente las funcionalidades de cada una. Luego se eligió una de ellas, y en esta herramienta se construyeron artefactos para el acceso a los datos para satisfacer los requerimientos de información del cliente.

Luego de contruidos estos artefactos, estos son incluidos en la Matriz de Trazabilidad, logrando una trazabilidad completa. Esta es la propuesta de extensión mencionada en el capítulo 4.

A continuación se presentan las herramientas relevadas, una comparación entre ellas y la selección de la herramienta; la solución construida con sus resultados; la extensión de la Matriz de Trazabilidad; y una evaluación de la solución construida.

5.3.1. Herramientas relevadas

Para construir la herramienta para que el usuario acceda a la información, se realizó un relevamiento de las herramientas open source que existen en el mercado. Que la herramienta sea open source es un requisito del proyecto, pero eso no debería ser una limitante ya que existe un auge de las herramientas de Business Intelligence open source y una tendencia en los últimos dos años a implantar estas herramientas (33).

El relevamiento inicial arrojó una gran cantidad de herramientas, debido a que bajo el nombre de Business Intelligence se suelen catalogar diferentes tipos de herramientas que resuelven distintas problemáticas, desde el proceso de construcción y carga de un Data Warehouse hasta el acceso al mismo. Dentro de este nombre se suelen incluir herramientas que asisten al proceso de ETL (extraction, transformation and load), sistemas DBMS, servidores de Data Warehouse (ROLAP y MOLAP), aplicaciones cliente para dichos servidores, herramientas de Data Mining, y hasta librerías de cálculo estadístico (34) (35) (36) (37).

Entre estas herramientas se incluyen:

- BIRT (herramienta de reportes y cliente para BI) (38)
- Mondrian (servidor ROLAP) (39)
- Openi (cliente para BI) (40)
- Palo (servidor MOLAP) (41)
- R (lenguaje de programación para estadística) (42)
- RapidMiner (herramienta para Data Mining y Machine Learning) (43).

También existen herramientas integradas que brindan todas (o casi todas) las funcionalidades para la construcción y utilización de un Data Warehouse, que reciben el nombre de Business Intelligence suites. En este proyecto este tipo de herramienta es preferible ya que brinda una respuesta integral a la problemática, problemática de tipo tecnológica, que no es el eje central del proyecto. En otro tipo de situación, utilizar herramientas de forma individual puede

brindarle más flexibilidad a la solución. Pero en este caso, se opta por una herramienta integrada, lista para instalar y usar.

A la hora de seleccionar la herramienta también es de interés que sea una herramienta popular de uso extendido, lo que puede facilitar la búsqueda de soluciones a problemas. A continuación se presentan brevemente las suites de Business Intelligence que se encontraron.

- **Bee Project** (44)



La suite de Business Intelligence BEE tiene licencia GNU GPL y ofrece funcionalidades para ETL, un servidor ROLAP, y funcionalidades para reportes. También se encuentra integrado con el proyecto R.

BEE es una de las primeras suites de BI open source, existiendo desde 2002. Sin embargo, es una herramienta poco popular, y como muestra de ello es la poca información que hay sobre ella y que solamente fue descargada 367 veces en el 2011. La última actualización del programa es de 2006.

- **Jasper** (45)



La suite de Business Intelligence Jasper ofrece un framework para la automatización de reportes y reportes a medida, con funcionalidades OLAP y ETL. Jasper integra las siguientes herramientas: JasperReports, iReport, JasperServer, JasperAnalysis (basado en Mondrian y JPivot) y JasperETL.

La herramienta descrita tiene licencia GNU GPL, aunque JasperSoft también tiene una edición empresarial con más funcionalidades que las mencionadas.

- **Pentaho** (46)



La suite Pentaho ofrece todas las funcionalidades para Business Intelligence: Reporting, Análisis, Dashboards, ETL y Data Mining. La suite incluye las siguientes herramientas: Pentaho Data Integration (también conocido como Kettle) para ETL, Pentaho Analysis (basado en Mondrian y JPivot), una herramienta de reportes basada en JFreeReports, y la herramienta Weka para DataMining. La suite no incluye un DBMS en el paquete.

La herramienta descrita tiene licencia GNU GPL y Mozilla Public Licence, aunque al igual que JasperSoft, Pentaho tiene una edición empresarial con más funcionalidades.

Esta herramienta es utilizada dentro del grupo CSI del Instituto de Computación.

- SpagoBI (47)



SpagoBI es una suite totalmente open source (Jasper y Pentaho tienen versiones comerciales con funcionalidades adicionales a las comentadas) perteneciente a SpagoWorld Initiative.

Entre las herramientas de la suite se incluyen SpagoBI Server (que concentra la gran mayoría de las funcionalidades: ETL, Reporting, Analysis, Dashboards y DataMining), SpagoBI Meta (para la gestión de metadata), y SpagoBI Studio (para la edición de los documentos analíticos -como ser los reportes). Además, SpagoBI Server brinda algunas funcionalidades para apoyar la toma de decisiones al nivel operativo.

Para analizar estas opciones se recurrió a la bibliografía existente, en particular a los relevamientos y comparaciones presentados en los trabajos de Golfarelli (48) y Thomsen (24) (49). En estos proyectos se realiza una comparación entre Jasper, Pentaho y SpagoBI (no hacen mención a Bee Project).

Por lo descrito en estos trabajos las tres herramientas difieren poco en sus funcionalidades, cubriendo todas el espectro de funcionalidades básicas de Business Intelligence: ETL, OLAP y Reporting. Incluso, sus módulos más importantes son compartidos por las tres herramientas, al ser otros productos open source, como se muestra en la tabla 6.

La principal diferencia en funcionalidad ocurre en temas de seguridad, donde solamente SpagoBI brinda la posibilidad de manejar perfiles de usuario.

Según Golfarelli (48) las tres herramientas son sencillas de instalar y configurar. También los manuales son de buena calidad, permitiendo resolver la mayoría de los problemas.

Módulos	Jasper	Pentaho	SpagoBI
Servidor de Aplicaciones	JBoss	JBoss	JBoss
Dashboard	JFreeChart	JFreeChart	Openlaszlo
DataMining	-	Weka	Weka
DBMS	MySQL, Oracle, SQLServer, PostgreSQL, etc.	MySQL, Oracle, SQLServer, PostgreSQL, etc.	MySQL, Oracle, SQLServer, PostgreSQL, etc.
OLAP	Mondrian y JPivot	Mondrian y JPivot	Mondrian y JPivot
Reporting	JasperReport	Pentaho Report Designer, JasperReport, BIRT	JasperReport, BIRT
Servidor Web	Tomcat	Tomcat	Tomcat

Tabla 6 - Módulos incluidos en las suites de Business Intelligence Jasper, Pentaho y SpagoBI. Tomado de (48).

Por último, según Golfarelli las comunidades de usuarios de Jasper y Pentaho son significativamente mayores que la de SpagoBI, y dentro de estos dos primeros el foro más activo, donde antes obtuvieron respuesta a las preguntas planteadas es el de Pentaho.

Para intentar medir el uso de cada herramienta se obtuvieron en SourceForge la cantidad de descargas de cada herramienta en el año 2011, lo que se muestra en la tabla 7. Esta comparación arroja pocos resultados ya que Jasper y SpagoBI son descargadas desde sus sitios oficiales. Queda evidenciada la baja popularidad de Bee Project.

Bee Project	Jasper	Pentaho	SpagoBI
367	No disponible ³	368mil	560 ⁴

Tabla 7 - Cantidad de descargas realizadas en 2011 de las suites de BI relevadas. Tomado de SourceForge (50) (51) (52) (53).

Como conclusión sobre las herramientas, tanto Jasper, Pentaho y SpagoBI son herramientas completas que brindan funcionalidades similares. En base a este análisis, Jasper y Pentaho son las herramientas más populares, y serían las candidatas más claras, sin presentar ninguna una ventaja clara sobre la otra.

Por lo tanto, dado que Pentaho es la herramienta que es utilizada en el grupo CSI, y que esto permitiría obtener respuesta a posibles problemas, se opta por la suite Pentaho para implementar la solución.

5.3.2. Solución construida

Una vez elegida Pentaho como la herramienta con la que brindar el acceso al Data Warehouse se procedió a estudiar sus funcionalidades y elegir la forma de acceder a la información. Para ello se recurrió al libro Pentaho Solutions de Roland Bouman y Jos van Dongen (54).

Para presentar la información al usuario, Pentaho brinda, al igual que las otras suites de Business Intelligence, las siguientes alternativas: reportes, cubos OLAP y dashboards.

Los reportes son informes que presentan una información determinada, de una forma determinada. Son la forma más común de presentar información al usuario final. Incluso, en un ambiente típico de Business Intelligence, entre el 75 y 80% del contenido utilizado por los usuarios es accedido a través de reportes. Entre un 15 y 20% es accedido con herramientas OLAP, y menos de 5% trabajado con sistemas de Data Mining (54).

³ No es posible saber cuántas veces Jasper fue descargada en 2011, ya que la descarga no se realiza desde SourceForge, sino desde su sitio oficial.

⁴ No es posible saber cuántas veces SpagoBI fue descargada en 2011, ya que la descarga también se realiza desde su sitio oficial, en el que no se encuentran disponibles estos datos.

Las herramientas OLAP permiten que el usuario analice de forma interactiva datos multidimensionales, realizando las operaciones de consolidación (o roll-up), drill-down, slicing y dicing. La consolidación significa la agregación de los datos que pueden ser acumulados y computados en una o más dimensiones. Drill-down es la operación opuesta, que permite que el usuario navegue a mayor detalle de los datos. Slicing y dicing son las operaciones que permiten que el usuario seleccione (slicing) un conjunto específico de datos y los presente (dicing) desde diferentes perspectivas (55).

Un dashboard es una aplicación utilizada para presentar contenido de alto nivel al usuario. Los dashboard contienen solamente unos pocos indicadores sobre el desempeño de algún aspecto del negocio. Suelen mostrar su información de forma exclusivamente gráfica, con el objetivo de brindar una visión condensada del negocio (54).

Los reportes y los dashboards son herramientas atractivas debido a que son sencillas de utilizar ya que brindan la información de forma predefinida y requieren muy poca interacción del usuario. Pero esta forma de acceder es rígida y no permite que el usuario navegue por los datos. Los dashboards no tienen sentido para el contexto planteado, pero los reportes pueden satisfacer perfectamente los requerimientos.

Las herramientas OLAP resultan más interesantes que los reportes al brindarle mayor interactividad al usuario, brindando la posibilidad de agregar datos por las dimensiones de los cubos. Pero este tipo de herramienta no puede aplicarse (o al menos no directamente) al contexto del requerimiento. Esto se debe a que las medidas que se definieron para las tablas de hechos no son aditivas en ninguna de las dimensiones. Esto significa que sumar o promediar el Grado de Interacción de dos cursos carece de sentido. Por esta razón no tiene sentido disponer la información en un cubo cuyo objetivo es precisamente agregar las medidas.

Por lo tanto, la solución disponible para resolver los requerimientos de información del cliente son los reportes. La solución incluye cuatro reportes que presentan la siguiente información:

- Reporte 1 - Cursos por Asignatura: todos los cursos, agrupados por asignatura,
- Reporte 2 - Cursos por Año y Semestre: todos los cursos, agrupados por semestre,
- Reporte 3 - Cursos por Instituto: todos los cursos, agrupados por instituto que los dicta,
- Reporte 4 - Cursos por Carrera: todos los cursos, agrupados por carrera a la que pertenecen.

En la figura 21 se presenta la interfaz que Pentaho brinda al usuario final, donde se muestra uno de los reportes.

La solución construida satisface los requisitos del cliente, el que quedó muy satisfecho con la información generada por el Data Warehouse.

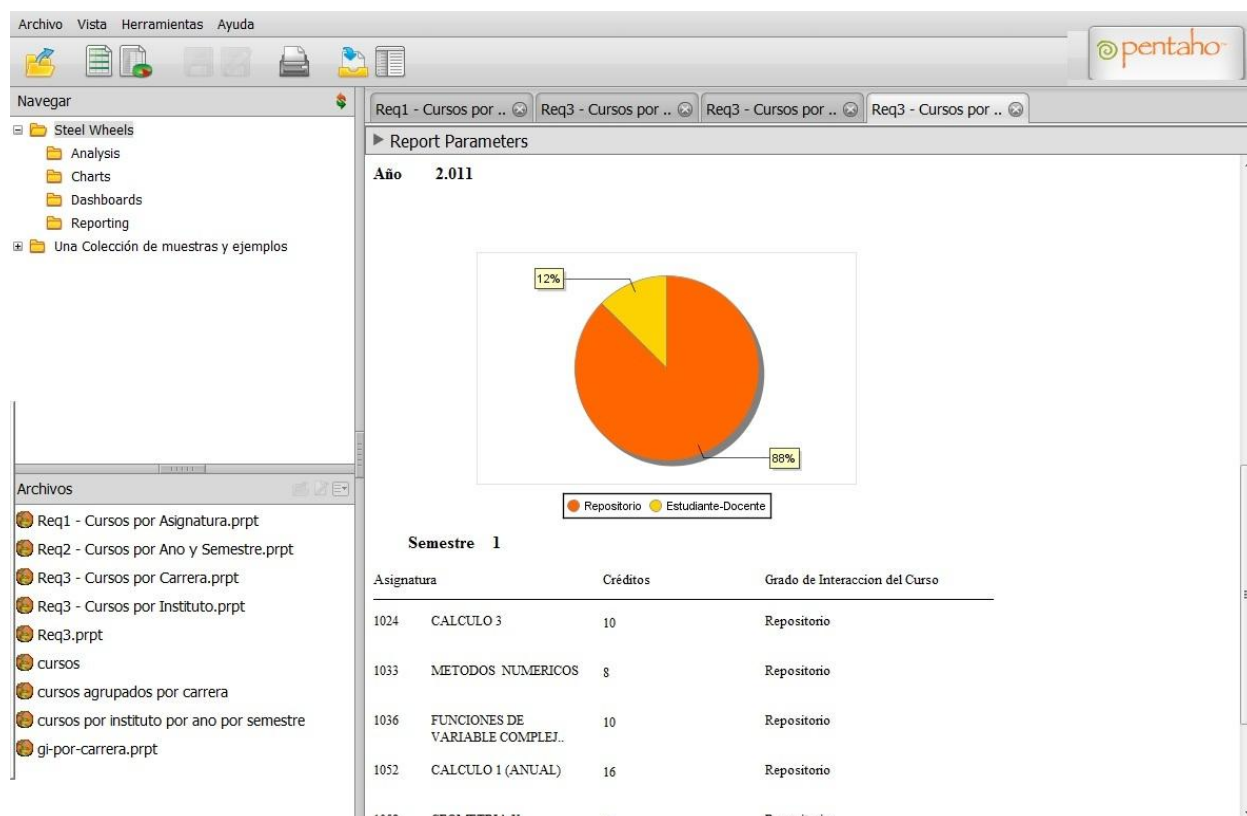


Figura 21 - Reporte en el sistema Pentaho.

Los requerimientos iniciales se presentan nuevamente a continuación. Luego, en la tabla 8 se muestra los requerimientos que los reportes creados ayudan a satisfacer.

Requerimiento1: Se quiere saber cuál es el grado de interacción de los cursos en la plataforma Moodle, y para cada curso, saber cuántos recursos de cada grado de interacción tiene.

Requerimiento 2: Se quiere saber cuál es el grado de interacción de los cursos por asignatura por semestre y año.

Requerimiento 3: Se quiere saber cuál es el grado de interacción de los cursos por asignatura por institutos y por carreras. También la cantidad de cursos por grado de interacción por año para institutos y para carreras.

	Reporte 1	Reporte 2	Reporte 3	Reporte 4
<i>Requerimiento1</i>	X			
<i>Requerimiento2</i>		X		
<i>Requerimiento3</i>			X	X

Tabla 8 - Matriz que vincula los reportes creados con los requerimientos que ayudan a satisfacer.

La aplicación resulta sencilla de utilizar ya que todo el acceso es mediante un portal web, y presenta la información de forma accesible y comprensible. En particular, las gráficas de torta ayudan a visualizar la información.

En lo que respecta a la aplicación para la creación de los reportes, la misma no resulta tan sencilla de utilizar como la aplicación para el usuario final. La herramienta es sencilla de utilizar para realizar reportes básicos, pero para diseñar reportes complejos la dificultad de uso de la herramienta aumenta ya que la documentación no cubre en detalle a toda la aplicación. Se encontró particular dificultad al momento de crear gráficas, ya que existen diferencias en las forma de crear cada una y la documentación solamente cubre algunas.

5.4. Extensión de la Matriz de Trazabilidad

Luego de contruidos los reportes en la suite Pentaho se identifica (se muestra en la tabla 6) qué requerimientos satisfacen. También se sabe de cuáles cubos obtienen los datos. Se propone agregar esta información a la Matriz de Trazabilidad. Esta propuesta de extensión es presentada en el capítulo 4.

La propuesta consiste en agregar a la Matriz de Trazabilidad los artefactos de la plataforma BI, ya sean reportes, dashboards o cubos OLAP, que se vinculan a cada requerimiento inicial. De esta forma se logra una trazabilidad completa desde los requerimientos y los conceptos del dominio de aplicación, pasando por los cubos del modelo relacional, llegando a los artefactos que satisfacen cada requerimiento inicial. El resultado se presenta en la tabla 9.

La Matriz de Trazabilidad expresa entre otras dependencias, la siguiente: el Requerimiento 1 se satisface con el Reporte 1 (Cursos por Asignatura), que obtiene su información del Cubo CursoSemestreAsignatura. Si surgiese un cambio en el concepto de Asignatura, este podría afectar al Cubo CursoSemestreAsignatura y por lo tanto al Reporte 1, consiguiendo quizás que no se satisfaga más el Requerimiento 1.

Es importante destacar la utilidad que adquiere una herramienta de este estilo cuando la cantidad de requerimientos, cubos y artefactos crece, volviéndose cada vez más difícil entender las dependencias entre todos los elementos del sistema.

Concepto	RQ1	RQ2	RQ3	Cubo Curso Semestre Asignatura	Cubo Carrera Asignatura	Cubo Grado De Interacción Instituto	Cubo Grado De Interacción Carrera
Clases							
Asignatura	✓	✓	✓	✓	✓	✓	✓
Carrera			✓		✓		✓
Curso	✓	✓	✓	✓		✓	✓
Facultad			✓		✓	✓	✓
Instituto			✓			✓	
Recurso	✓	✓	✓	✓			
Relaciones							
asignaturaCurso	✓	✓	✓				
asignaturasCarrera			✓				
asignaturasInstituto			✓				
carrerasFacultad			✓				
facultadAsignatura			✓				
facultadCarrera			✓				
facultadInstituto			✓				
institutoAsignatura			✓				
institutosFacultad			✓				
recursosCurso	✓	✓	✓				
Artefactos							
Reportes							
R1 - Cursos por Asignatura	✓			✓			
R2 - Cursos por Año y Semestre		✓		✓			
R3 - Cursos por Instituto			✓	✓		✓	
R4 - Cursos por Carrera			✓	✓	✓		✓

Tabla 9 - Matriz de Trazabilidad, incluyendo cubos construidos y artefactos desarrollados en la suite Pentaho.

5.5. Conclusiones sobre el prototipo

Se consiguió construir una solución que integra los datos de las distintas fuentes, y ofrece al usuario una forma amigable de acceder a la información en forma de reportes, satisfaciendo de este modo los requerimientos. Se presenta la arquitectura de toda la solución en el Anexo II.

Para realizar la carga del Data Warehouse se construyó una aplicación que permite que el usuario seleccione los cursos que desea cargar, los vincule a una asignatura, y les asigne un semestre. El diseño de la aplicación tiene la lógica independiente de la presentación, lo que permite que la interfaz sea mejorada o modificada, permitiendo otro tipo de acceso.

Para el acceso a la información se relevaron las suites de Business Intelligence más utilizadas y se seleccionó la herramienta Pentaho. Por lo relevado esta herramienta no parece diferenciarse mayormente de JasperSoft, la otra suite que surgía como alternativa. La decisión de utilizar Pentaho se basó principalmente en que ya era utilizada en CSI. Sobre esta herramienta se construyeron reportes, que presentan la información requerida por el usuario, de la forma definida por el usuario, en un formato amigable.

Estos reportes son fáciles de utilizar para el usuario final ya que presentan toda la información requerida ya procesada y ordenada. Sin embargo, los reportes son una forma de acceder a la información que brinda escasa interacción, ya que no permite que el usuario busque otras representaciones o agregaciones. De todas formas, los reportes son la herramienta más adecuada dado el caso de estudio.

Por otro lado, por más que los reportes resultan sencillos de utilizar, no es sencilla su construcción. Para ello es necesario familiarizarse con la herramienta Pentaho Report Designer y tener conocimientos de SQL y del modelo relacional del Data Warehouse. Además, la documentación de Pentaho no siempre cubre en detalle todos los aspectos de la herramienta. Todo esto resulta en que la creación de nuevos reportes sea difícil para un usuario no técnico.

Si se hubiera elegido otra de las suites relevadas, es de suponer que el resultado hubiera sido igualmente satisfactorio.

Como resultado, luego de aplicar VGPCDW (con algunas correcciones) y construir las soluciones mencionadas en este capítulo, se tiene un sistema de información en funcionamiento que satisface los requerimientos planteados. El sistema abarca la extracción de datos desde las fuentes de datos y carga del ODS, la carga desde el ODS al Data Warehouse, y el acceso al mismo mediante reportes.

Por último, se tiene una herramienta que permite realizar la trazabilidad desde los requerimientos hasta los artefactos construidos en Pentaho, lo que cobra vital importancia a medida que la cantidad de requerimientos, cubos y artefactos crece, volviéndose cada vez más difícil entender las dependencias entre todos los elementos del sistema.

6. Conclusiones

El resultado principal de este proyecto es la evaluación de la metodología Visión Global del Proceso de Construcción de Data Warehouse (VGPCDW) desarrollada en el InCo. En base al estudio y aplicación de la metodología, se concluye que utilizando la misma no es posible construir una solución que satisfaga los requerimientos.

Esto se debe a que la metodología padece de una mala interpretación del lenguaje de modelado conceptual que utiliza, por lo que plantea reglas inconsistentes y conduce a una solución que no es correcta.

Otro problema en la metodología ocurre al momento de construir el diseño relacional de Data Warehouse, ya que existen casos que no son contemplados por la misma.

Por último, a la hora de cargar el Data Warehouse la metodología presenta varias carencias. Primero, plantea construir la solución como vistas desde la fuente intermedia ODS, pero esto no siempre es posible. Tampoco aborda el problema de la implementación concreta del Data Warehouse.

Además, la metodología no contempla el proceso de extracción, transformación y carga (ETL) del ODS, que en sí es el ETL del Data Warehouse, que es una fase crucial en la construcción de este tipo de soluciones.

En este proyecto se sugieren líneas de acción para resolver los problemas planteados.

Dejando de lado estos problemas, la metodología construye una solución muy rica en documentación, con trazabilidad desde los requerimientos hasta la solución, con modelos de dominio, un modelo conceptual multidimensional, un algoritmo para el diseño semiautomático del modelo relacional y la carga. Estos son elementos muy positivos, y hacen que esta metodología una vez corregida, pueda guiar la construcción de una solución tan rica como satisfactoria.

Pero las ventajas de la metodología se obtienen a un alto costo. En primer lugar, la metodología demanda muchos conocimientos previos para poder ser aplicada, como ontologías y el lenguaje CMDM. Además el estudio de la metodología resulta difícil ya que es necesario estudiar los cuatro proyectos para obtener una visión total de la misma y conseguir entenderla completamente. Por último, la documentación resultante, tan extensa y rica, resulta muy pesada de mantener.

Otro resultado del proyecto es la extensión de la Matriz de Trazabilidad propuesta por VGPCDW, para que abarque los artefactos construidos en la aplicación de consulta del Data Warehouse. De esta forma se logra una trazabilidad completa desde los requerimientos hasta los reportes construidos en Pentaho.

Este proyecto también tiene como resultado la construcción de un Data Warehouse para la Unidad de Enseñanza, que satisface el requerimiento de información planteado. La solución fue construida con la metodología VGPCDW como guía, y por ello el producto final incluye una rica documentación, con una descripción del dominio, modelos de dominio, un modelo conceptual multidimensional correcto, un diseño lógico relacional, y la trazabilidad desde los requerimientos hasta la solución final.

Además, la solución incluye una aplicación para facilitar la carga del Data Warehouse, y una aplicación de consulta del mismo, basado en la suite Pentaho. Ambas herramientas resultaron intuitivas y fáciles de utilizar para el usuario.

Como otros aportes de este proyecto se tienen los modelos de dominio construidos, que pueden ser reutilizados, permitiendo compartir el conocimiento adquirido.

El proyecto también presenta un relevamiento de ontologías del dominio de educación, con una breve descripción de cada una.

6.1. Limitaciones y trabajos futuros

Como trabajo a futuro surge naturalmente el proponer soluciones de tipo general para los problemas detectados, con el objetivo de tener una metodología correcta. El punto central de esta corrección debe ser corregir las reglas para la elaboración del modelo multidimensional. En particular la definición de las medidas de las relaciones dimensionales, y la definición de relaciones dimensionales sin medida.

Otro aspecto a tener en cuenta, es definir las reglas faltantes para el diseño lógico en lo que refiere a la definición de cubos que se construyen a partir de dos o más relaciones dimensionales.

Por último resolver la implementación concreta del Data Warehouse.

También sería deseable extender la metodología para que abarque el proceso de ETL del ODS, y de esta forma tener completo el proceso de carga del Data Warehouse.

Luego de resueltos estos aspectos, es necesario presentar toda la metodología de forma unificada e integrada, de forma que pueda ser más accesible su comprensión y utilización.

Otros trabajos a futuro son los planteados en VGPCDW y todavía pendientes, sobre aspectos de calidad de datos, y la utilización de restricciones en el modelo CMDM.

Una limitación que presenta la solución construida es que no existe una herramienta que asocie los datos con el modelo multidimensional CMDM. Esta le brindaría semántica a los datos para que el usuario pueda buscar nuevas representaciones de los datos y obtener así nuevo conocimiento. Si se contase con una herramienta de este estilo, sería posible explotar todas las posibilidades que ofrece CMDM, en particular aprovechar las posibilidades que ofrece la dimensionalidad genérica, al convertir las medidas en dimensiones y una dimensión cualquiera en medida.

Se destaca la dimensionalidad genérica como uno de los puntos fuertes de CMDM, pero se realiza un uso limitado de la misma. Se limita a las posibilidades encontradas al momento de diseñar los cubos, y no se permite la búsqueda de nuevas interpretaciones de los datos por parte del usuario en el momento de la consulta del Data Warehouse.

7. Glosario

BI - Business Intelligence. Refiere principalmente a técnicas informáticas utilizadas para identificar, extraer y analizar información de negocio. Business Intelligence apunta a mejorar la toma de decisiones.

CMDM - Conceptual MultiDimensional Model. Modelo de datos conceptual y multidimensional para la especificación de una base de datos multidimensional. Fue desarrollado por Carpani en el Instituto de Computación.

Cubo - Estructura básica en el modelo multidimensional. Una instancia de un cubo, queda determinada por un conjunto de datos para cada eje y un conjunto de datos para la matriz. A los ejes se les llama Dimensiones y al dato que se presenta en la matriz, se le llama Medida. A los elementos del producto cartesiano de los ejes (dimensiones) se le llama Coordenadas.

DSS - Decision Support System. Sistema de información orientado al apoyo de la toma de decisiones.

DW - Data Warehouse.

EVA - Entorno Virtual de Aprendizaje.

Matriz de requerimientos (o trazabilidad) - Matriz que vincula los requerimientos con los conceptos del dominio. En VGPCDW se le agrega la vinculación con los cubos, y se le cambia en nombre a Matriz de trazabilidad.

Matriz de Trazabilidad - Ver Matriz de requerimientos.

MD4DW - Metadata for Data Warehouse, Proyecto de Grado de 2006, realizado en el Instituto de Computación.

Metadata de Conceptos - Documento que contiene información de contexto y datos importantes identificados para cada concepto. Incluye una descripción del dominio relevado, sobre el cual se deben resolver los requerimientos. Es el resultado de aplicar la fase I de VGPCDW.

Moodle - Plataforma tecnológica del EVA.

ODS - Operational Data Store. Base de datos diseñada para integrar datos de múltiples fuentes con el objetivo de realizar posteriores operaciones sobre ellos.

OLAP - OnLine Analytical Processing. Es un enfoque para responder consultas de tipo multidimensional.

Onto2rdb - Ontology to Relational database. Algoritmo que a partir de una ontología de dominio crea un esquema relacional para un ods, que permita la carga del almacén de datos.
Glosario LOAD.

UdelaR - Universidad de la República.

UE - Unidad de Enseñanza de la Facultad de Ingeniería.

VGPCDW - Visión Global del Proceso de Construcción de Data Warehouse, Proyecto de Grado de 2010, realizado en el Instituto de Computación.

8. Bibliografía

1. **Deppen, Nathalie; Ricca, Paola y Trías, Diego.** *Visión global del proceso de construcción de Data Warehouse.* Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2010. Proyecto de Grado.
2. **Unidad de Enseñanza de Facultad de Ingeniería.** FIng.edu.uy. *Unidad de Enseñanza.* [En línea] [Citado el: 22 de Abril de 2012.] http://www.fing.edu.uy/uni_ens/presentacion.htm.
3. **Colombatto, Cecilia; Gayoso, Diego y Giménez, Sebastián.** *MD4DW - Metadata For Data Warehouse Guide.* Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2006. Proyecto de Grado.
4. **Peralta, Verónica.** *Diseño Lógico de Data Warehouses a partir de Esquemas Conceptuales Multidimensionales.* Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2001. Tesis de Maestría.
5. —. *Un caso de estudio sobre diseño logico de Data Warehouses.* InCo-Pedeciba, Universidad de la República. Montevideo : s.n., 2001. Reporte técnico.
6. **Larrañaga, Ignacio.** *Carga de un Data Warehouse a partir de la Traza de Diseño.* Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2006. Tesis de Maestría.
7. *A translation approach to portable ontology specifications.* **Gruber, Thomas.** 2, s.l. : Elsevier, 1993, Knowledge Acquisition, Vol. 5, págs. 199-220.
8. **W3C.** W3.org. *OWL 2 Web Ontology Language - Document Overview.* [En línea] [Citado el: 22 de Abril de 2012.] <http://www.w3.org/TR/owl-overview/>.
9. **Carpani, Fernando.** *CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales.* Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2001. Tesis de maestría.
10. **Moodle.** Moodle.org. *Moodle.* [En línea] [Citado el: 22 de Abril de 2012.] <http://moodle.org/>.
11. **Wikipedia.** Wikipedia.org. *Definición de Learning Management System.* [En línea] [Citado el: 22 de Abril de 2012.] http://en.wikipedia.org/wiki/Learning_management_system.
12. **Moodle.** Moodle.org. *Database schema introduction.* [En línea] [Citado el: 2012 de Abril de 18.] http://docs.moodle.org/dev/Database_schema_introduction.
13. **Facultad de Ingeniería, Universidad de la República.** Facultad de Ingeniería. *Sitio del Entorno Virtual de Aprendizaje (EVA) de Facultad de Ingeniería.* [En línea] [Citado el: 22 de Abril de 2012.] <http://eva.fing.edu.uy/>.
14. **IEEE.** *IEEE 1484.12.1-2002 - Draft Standard for Learning Object Metadata.* Learning Technology Standards Committee of the IEEE, IEEE. 2002. Draft Standard.
15. **Wikipedia.** Wikipedia.org. *LOM base schema.* [En línea] [Citado el: 2012 de Abril de 18.] http://en.wikipedia.org/wiki/File:LOM_base_schema.png.
16. **National Information Standards Organization.** *Understanding Metadata.* Bethesda, MD : NISO Press, 2004. 1-880124-62-9.
17. **Proyecto EduCa.** Red de Educación con Calidad Cultural (EduCa). *Desarrollos.* [En línea] [Citado el: 2012 de abril de 18.] <http://www.fing.edu.uy/inco/grupos/csi/esp/Proyectos/EducaDesarrollos.htm>.

18. **Farance, Frank.** IEEE.org. *IEEE Learning Technology Standards Committee (LTSC)*. [En línea] [Citado el: 21 de Abril de 2012.] ltsc.ieee.org/meeting/199912/doc/papi--199912--farance.ppt.
19. *Reasoning and Ontologies for Personalized E-Learning in the Semantic Web.* **Henze, N.; Dolog, P. y Nejd, W.** 4, s.l. : International Forum of Educational Technology & Society, 2004, Educational Technology & Society, Vol. 7, págs. 82-97.
20. *Using Ontologies for eLearning Personalization.* **Gomes, Paulo, y otros.** Coimbra, Portugal : s.n., Setiembre de 2006, E-learning Conference.
21. *Description of an Instructional Ontology and its Application in Web Services for Education.* **Ullrich, Carsten.** Eindhoven, Holanda : s.n., 2004, International Workshop on Applications of Semantic Web technologies for E-Learning.
22. **Demarest, Marc.** *The politics of data warehousing*. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.uncg.edu/ism/ism611/politics.pdf>.
23. **Marotta, Adriana.** *Data Warehouse Design and Maintenance through Schema Transformations*. Instituto de Computación, Facultad de Ingeniería, Universidad de la República. 2000. Tesis de maestría.
24. *A Survey of Open Source Tools for Business Intelligence.* **Thomsen, Christian y Bach Pedersen, Torben.** 3, s.l. : Information Resources Management Association, 2009, International Journal of Data Warehousing and Mining, Vol. 5.
25. **MySQL.** MySQL.com. *Restrictions on Views*. [En línea] [Citado el: 22 de Abril de 2012.] <http://dev.mysql.com/doc/refman/5.0/en/view-restrictions.html>.
26. —. MySQL. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.mysql.com/>.
27. **Moodle.** docs.Moodle.org. *Moodle Documentation*. [En línea] [Citado el: 22 de Abril de 2012.] http://docs.moodle.org/19/en/Main_page.
28. **Cole, Jason y Foster, Helen.** *Using Moodle*. 2da. Sebastopol, CA : O'Reilly, 2008. Free Online Edition. 978-0-596-52918-5.
29. **Protégé.** protégé.stanford.edu. [En línea] [Citado el: 22 de Abril de 2012.] <http://protege.stanford.edu/>.
30. **protégé.** Protege 4.1 Release. [En línea] [Citado el: 22 de Abril de 2012.] http://protegewiki.stanford.edu/wiki/P4_1_Release_Announcement.
31. **PHP.** PHP. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.php.net/>.
32. **Wikipedia.** PHP. [En línea] [Citado el: 22 de Abril de 2012.] <http://en.wikipedia.org/wiki/PHP>.
33. **enterprise APPS today.** EnterpriseAppsToday.com. *Open Source Business Intelligence Heats Up*. [En línea] [Citado el: 30 de Noviembre de 2011.] <http://www.enterpriseappstoday.com/business-intelligence/open-source-business-intelligence.html>.
34. **Wikipedia.** Wikipedia.org. *Business intelligence tools*. [En línea] [Citado el: 22 de Abril de 2012.] http://en.wikipedia.org/wiki/Business_intelligence_tools.
35. **All, Ann.** ITBusinessEdge.com. *Who Is Using Open Source Business Intelligence, and Why*. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.itbusinessedge.com/cm/community/features/interviews/blog/who-is-using-open-source-business-intelligence-and-why/?cs=35889>.

36. **Dean, Sam.** Ostatic.com. *Open Source Business Intelligence Software On the Rise*. [En línea] [Citado el: 22 de Abril de 2012.] <http://ostatic.com/blog/open-source-business-intelligence-software-on-the-rise>.
37. Squidoo.com. *Open Source Business Intelligence*. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.squidoo.com/osbi>.
38. **BIRT.** Sitio oficial de BIRT. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.eclipse.org/birt/phoenix/>.
39. **Mondrian.** Sitio oficial de Mondrian. [En línea] [Citado el: 22 de Abril de 2012.] <http://mondrian.pentaho.com/>.
40. **Openi.** Sitio oficial de Openi. [En línea] [Citado el: 22 de Abril de 2012.] <http://openi.sourceforge.net/>.
41. **Palo.** Sitio oficial de Palo. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.palo.net/>.
42. **R.** Sitio oficial de R. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.r-project.org/>.
43. **Rapid Miner.** Sitio oficial de Rapid Miner. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/rapidminer/>.
44. BEE Project. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/bee/>.
45. JasperForge. [En línea] [Citado el: 22 de Abril de 2012.] <http://jasperforge.org/>.
46. Pentaho. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.pentaho.com/>.
47. SpagoWorld. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.spagoworld.org/>.
48. **Golfarelli, Matteo.** Open Source BI Platforms: A Functional and Architectural Comparison. *Data Warehousing and Knowledge Discovery*. s.l. : Springer Berlin / Heidelberg, 2009, págs. 287-297.
49. **Thomsen, Christian y Pedersen, Torben.** A Survey of Open Source Tools for Business Intelligence. *Data Warehousing and Knowledge Discovery*. s.l. : Springer Berlin / Heidelberg, 2005, págs. 74-84.
50. SourceForge.org. *BEE Project Stats*. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/bee/files/stats/timeline?dates=2011-01-01+to+2011-12-31>.
51. SourceForge.org. *JasperSoft Stats*. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/jaspersoft/files/stats/timeline?dates=2011-01-01+to+2011-12-31>.
52. SourceForge.org. *Pentaho Stats*. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/pentaho/files/stats/timeline?dates=2011-01-01+to+2011-12-31>.
53. SourceForge.org. *SpagoBI Stats*. [En línea] [Citado el: 22 de Abril de 2012.] <http://sourceforge.net/projects/spagobi/files/stats/timeline?dates=2011-01-01+to+2011-12-31>.
54. **Bouman, Roland y van Dongen, Jos.** *Pentaho Solutions*. Indianápolis : Wiley, 2009. 978-0-470-48432-6.
55. **Wikipedia.** Wikipedia.org. *Online analytical processing*. [En línea] [Citado el: 22 de Abril de 2012.] http://en.wikipedia.org/wiki/Online_analytical_processing.

Anexos

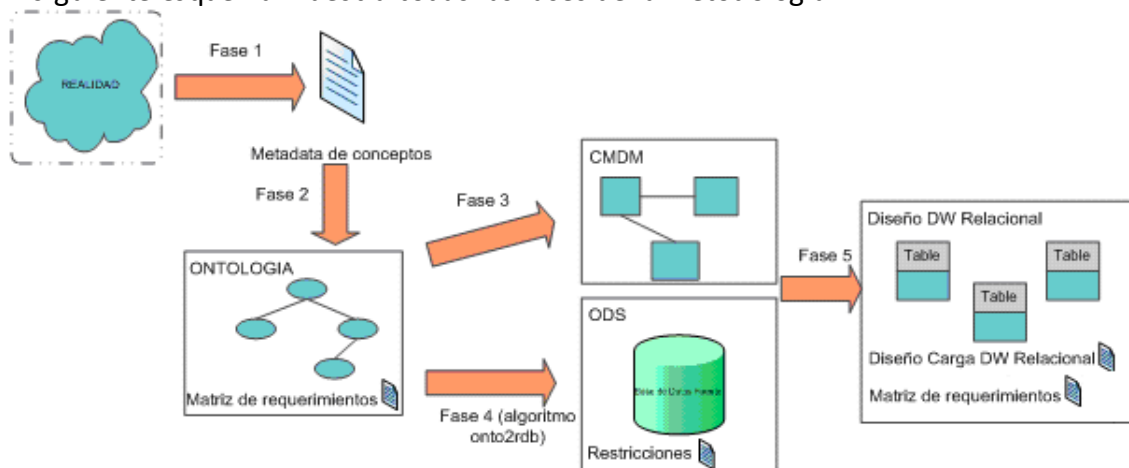
Anexo I - Aplicación de la metodología Visión Global del Proceso de Construcción de Data Warehouse al caso de estudio

Se trabajará para satisfacer algunas de las necesidades de información que tiene la Unidad de Enseñanza respecto a los cursos dictados en la plataforma Moodle. La contraparte de la Unidad de Enseñanza es Valery Buhl que trabaja en el área de educación.

En este caso de estudio se trabajará con un requerimiento y se relevará principalmente la realidad que concierne al mismo.

Se aplicará la metodología *Visión global del proceso de construcción de Data Warehouse*, la que se compone de cinco fases para construir un Data Warehouse abarcando el relevamiento de requerimientos, la representación de la realidad de la empresa, la generación del diseño conceptual, el diseño lógico y la carga de datos necesaria para el correcto funcionamiento del sistema.

El siguiente esquema muestra todas las fases de la metodología:



Las cinco fases de la metodología son: I) Comprensión de la realidad, II) Modelado de dominio, III) Modelado conceptual multidimensional, IV) Construcción de un ODS, y V) Generación del diseño lógico y la carga de datos.

Fase I - Comprensión de la realidad

La fase I de VGPCDW tiene como objetivo “identificar todos los conceptos que componen el dominio del problema, junto con las relaciones existentes entre los mismos y las restricciones identificadas para cada uno. El relevamiento y comprensión de la realidad está guiado por el conjunto de requerimientos iniciales. Por esto es necesario realizar previamente un análisis de los requerimientos y en base a ello determinar el dominio del problema” (3). Se presenta a continuación el resultado de esta fase, el documento Metadata de Conceptos.

Requerimiento

Se quiere saber cuál es el grado de interacción de los cursos en el Entorno Virtual de Aprendizaje (EVA) Moodle de la Facultad de Ingeniería; y con esto poder comparar el grado de interacción de los cursos de una misma asignatura dictados en diferentes semestres; y poder hacer comparaciones con cursos de asignaturas de institutos y carreras diferentes.

Descripción de la realidad

“Esta sección contiene el relevamiento, consideraciones, axiomas e hipótesis realizadas, permitiendo obtener una comprensión correcta de la realidad. En esta sección se recoge todo aquello que brinde información sobre el dominio del problema, incluyendo las opiniones de los expertos conocedores del dominio.

En base a la información proporcionada en esta parte se realizará la conceptualización de la realidad.” (3)

La Universidad de la República (UdelaR) se encuentra compuesta por facultades, donde cada facultad dicta una serie de carreras. Las facultades se encuentran formadas por institutos (también llamados departamentos). Cada instituto dicta una serie de asignaturas que pertenecen a una o varias de las carreras de esa facultad.

Cada facultad tiene un nombre y una abreviación de ese nombre que se usa para identificarlo, donde ambos son únicos (por ejemplo fagro corresponde a Facultad de Agronomía). Cada carrera tiene un identificador único dentro de una facultad, un nombre, una cantidad de créditos. Cada instituto tiene un nombre que es único dentro de la facultad. Cada asignatura tiene un identificador único dentro de la facultad donde se dicta, un nombre, una cantidad de créditos, y un tipo (puede ser de grado o de posgrado).

La UdelaR provee a los docentes de un Entorno Virtual de Aprendizaje llamado EVA, que se encuentra sobre la plataforma Moodle, que los docentes utilizan para “dictar” los cursos de las asignaturas. Se considerarán solamente las asignaturas que tengan un curso en Moodle. Un curso se identifica por la asignatura a la que corresponde y por el semestre en que se dictó. Un semestre se identifica por el año al que pertenece y el período del año (se denomina impar a la primera mitad del año, par a la segunda).

Los cursos en Moodle se encuentran formados por “Actividades” y “Recursos” (de ahora en adelante se referirá a los dos como recursos). Un recurso puede ser por ejemplo un foro de discusión, un ejercicio, un archivo para lectura, o simplemente un link. Los recursos tienen siempre un tipo (por ejemplo un foro) y un identificador único para ese tipo. Un curso siempre tiene al menos un recurso.

La Unidad de Enseñanza define el concepto grado de interacción, el cual se puede aplicar a un recurso de un curso en Moodle o a un curso en Moodle. El grado de interacción se mide en una escala enumerada cuyos valores posibles son: “Repositorio”, “Interacción Estudiante-Contenido”, “Interacción Estudiante-Actividad”, “Interacción Estudiante-Docente”, e “Interacción Estudiante-Estudiante”. Para un recurso en un curso en Moodle el grado de interacción es el tipo de interacción que le permite al estudiante, y el mismo lo define la Unidad de Enseñanza para cada tipo de recurso. Para un curso en Moodle, el grado de interacción es cuánta interacción le permite al estudiante ese curso y está determinado por el tipo de recursos que componen el curso y en qué proporción y los grados de interacción de los mismos.

Así, los recursos también tienen un grado de interacción (definido por la UE), y los cursos también tienen un grado de interacción (que depende de sus recursos).

Para definir el grado de interacción de un curso, se contarán la cantidad de recursos que caigan dentro de cada tipo de grado de interacción, el grupo más poblado determinará el grado de interacción del curso.

Refinamiento del requerimiento.

Requerimiento 1

Se quiere saber cuál es el grado de interacción de los cursos en la plataforma Moodle, y para cada curso, saber cuántos recursos de cada grado de interacción tiene.

Requerimiento 2

Se quiere saber cuál es el grado de interacción de los cursos por asignatura por semestre y año.

Requerimiento 3

Se quiere saber cuál es el grado de interacción de los cursos por asignatura por institutos y por carreras. También la cantidad de cursos por grado de interacción por año para institutos y para carreras.

Las tablas 10 y 11 muestran respectivamente los Resources (recursos) y Activities (actividades) disponibles en la plataforma Moodle, junto con sus grados de interacción. En este proyecto tanto a los dos se los denomina recursos.

Recursos (Resources)		
Nombre en Español	Nombre en Inglés	Grado de interacción
Componer una página de	Compose a text page	Repositorio

texto		
Componer una página web	Compose a web page	Repositorio
Enlazar un archivo o una web	Link to a file or web site	Repositorio
Insertar una etiqueta	Insert a label	No se considera
Mostrar un directorio	Display a directory	Repositorio
Desplegar paquete de contenidos IMS	Add an IMS Content Package	Repositorio

Tabla 10 - Resources en la plataforma Moodle.

Actividades		
Nombre en Español	Nombre en Inglés	Grado de interacción
Tareas (Assignments)		
Subida avanzada de archivos	Advanced uploading of files	Int. Est-Activ
Texto en línea	Online text	Int. Est-Activ
Subir un solo archivo	Uploading a single file	Int. Est-Activ
Actividad no en línea	Offline activity	Int. Est-Activ
Base de Datos	Database	Repositorio
Consulta	Choice	Int. Est-Cont
Cuestionario	Quiz	Int. Est-Cont
Chat	Chat	Int. Est-Doc
Diario	Journal	Int. Est-Activ
Encuesta	Survey	No se considera
Foro	Forum	Int. Est-Doc
Foro de Novedades	News Forum	No se considera
Glosario	Glossary	Int. Est-Activ
Hot Potatoes Quiz	Hot Potatoes Quiz	Int. Est-Cont
LAMS	LAMS	No se considera
Lección	Lesson	Int. Est-Cont
Módulo de encuesta	Feedback	No se considera.
SCORM	SCORM/AICC	Int. Est-Cont
Taller	Workshop	Int. Est-Est
Wiki	Wiki	Int. Est-Est

Tabla 11 - Activities en la plataforma Moodle.

Conceptos identificados

facultad, instituto, asignatura, carrera, curso, recurso.

Definición de los conceptos de la realidad

“Este capítulo se encuentra conformado por la descripción del conjunto de conceptos participantes del marco conceptual sobre el cual se trabajará.

Para poder satisfacer las necesidades planteadas se propone el uso de una plantilla que cubra las necesidades planteadas y que brinde la información necesaria para alimentar el modelo conceptual. La plantilla contiene tanto la descripción de cada concepto como las características que permiten gestionar el conocimiento, de cada concepto en particular o de las relaciones y restricciones entre ellos.” (3)

Convención

Los nombres de las clases se escriben con mayúscula, las propiedades con minúscula. Las propiedades se escriben con el nombre aquí descrito, seguidas del nombre de la clase. eg: la propiedad carreras de Facultad, se escribe carrerasFacultad.

Nombre:	Facultad	Fecha:	17/12/2010
Descripción:	Cada una de las grandes divisiones de una universidad, correspondiente a una rama del saber, y en la que se dan las enseñanzas de una carrera determinada o de varias carreras afines.		
Responsable:	Agustín Mullin		
Fuente:	[Origen del requisito]		
Usos:	Representa el concepto facultad dentro del sistema.		
Supuestos:			
Identificador:	Id		
Atributos:			
Nombre:	Id		
Descripción:	Identificador de la facultad, es una sigla.		
Cardinalidad:	1		
Obligatoriedad:	Este dato es obligatorio		
Dominio:	fagro, farq, enba, eubca, fcien, ccee, eda, liccom, fcs, fder, fenf, eumus, fhum, fing, fmed, eunut, egradu, tecmed, odon, psico, fq, fvet.		
Nombre:	Nombre		
Descripción:	Nombre de la facultad		
Cardinalidad:	1		
Obligatoriedad:	Este dato es obligatorio		

Dominio:	<i>Agronomía, Arquitectura, Bellas Artes, Bibliotecología, Ciencias, C. Económicas, Administración, C. de la Comunicación, C. Sociales, Derecho, Enfermería, Música, Humanidades, Ingeniería, Medicina, Nutrición, Graduados de Medicina, Tecnología Médica, Odontología, Psicología, Química, Veterinaria.</i>
Nombre:	<i>Carreras</i>
Descripción:	<i>Las carreras que se dictan en esa facultad.</i>
Cardinalidad:	<i>N</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<u><i>Carrera</i></u>
Nombre:	<i>Institutos</i>
Descripción:	<i>Institutos que forman esa facultad</i>
Cardinalidad:	<i>N</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<u><i>Instituto</i></u>

Nombre:	Carrera	Fecha:	17/12/2010
Descripción:	Representa el concepto carrera dentro del sistema. Es un tipo de recurso educacional.		
Responsable:	Agustín Mullin		
Fuente:	Información proporcionada por Bedelía		
Usos:	Representa las carreras que son dictadas en cada facultad.		
Supuestos:			
Identificador:	[Facultad, Código]		
Atributos:			
Nombre:	Nombre		
Descripción:	Nombre de la carrera		
Cardinalidad:	1		
Dominio:	[Define el conjunto de valores posibles el concepto que se define]		
Obligatoriedad:	Este dato es obligatorio		
Nombre:	Código		
Descripción:	Código de la carrera		
Cardinalidad:	1		
Dominio:	String		
Obligatoriedad:	Este dato es obligatorio		
Nombre:	Facultad		
Descripción:	Facultad que dicta la carrera		
Cardinalidad:	1		
Obligatoriedad:	Este dato es obligatorio		

Concepto Rel. :	<u>Facultad</u>
Nombre:	<i>Asignaturas</i>
Descripción:	<i>Asignaturas que componen la carrera</i>
Cardinalidad:	<i>N</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<u>Asignatura</u>

Nombre:	Instituto	Fecha:	17/12/2010
Descripción:	División dentro de una facultad, agrupa grupos son afines entre sí.		
Responsable:	Agustín Mullin		
Fuente:	Información proporcionada por Bedelía		
Usos:	Representa los institutos que componen cada facultad. También agrupa grupos afines. Los grupos dictan asignaturas afines, así que las asignaturas también pueden agruparse por instituto.		
Supuestos:	Cada instituto pertenece a una sola facultad.		
Identificador:	Nombre, Facultad		
Atributos:			
Nombre:	Nombre		
Descripción:	Nombre del instituto		
Cardinalidad:	1		
Dominio:	String		
Obligatoriedad:	Este dato es obligatorio		
Nombre:	Facultad		
Descripción:	Facultada la que pertenece el instituto.		
Cardinalidad:	1		
Obligatoriedad:	Este dato es obligatorio		
Concepto Rel. :	<u>Facultad</u>		
Nombre:	Asignaturas		
Descripción:	Asignaturas que son dictadas por ese intituto.		
Cardinalidad:	N		
Obligatoriedad:	Este dato es obligatorio		
Concepto Rel. :	<u>Asignatura</u>		

Nombre:	Asignatura	Fecha:	17/12/2010
Descripción:	<i>Cada una de las materias que se dictan o forman un plan académico de estudios.</i>		
Responsable:	<i>Agustín Mullin</i>		

Fuente:	<i>Información proporcionada por Bedelía</i>
Usos:	<i>Representa el concepto de asignatura en el sistema.</i>
Supuestos:	
Identificador:	<i>[Código, Facultad]</i>
Atributos:	
Nombre:	<i>Código</i>
Descripción:	<i>Código de la asignatura</i>
Cardinalidad:	<i>1</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Dominio:	<i>String</i>
Nombre:	<i>Facultad</i>
Descripción:	<i>Facultad donde se dicta esa asignatura</i>
Cardinalidad:	<i>1</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<i><u>Facultad</u></i>
Nombre:	<i>Nombre</i>
Descripción:	<i>Nombre de la asignatura</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>String</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Instituto</i>
Descripción:	<i>Instituto que dicta la asignatura. El instituto determina directamente la facultad a la que pertenece.</i>
Cardinalidad:	<i>1</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<i><u>Instituto</u></i>
Nombre:	<i>Créditos</i>
Descripción:	<i>Cantidad de créditos asignados a la asignatura.</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>El dato es un entero</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Tipo</i>
Descripción:	<i>La asignatura puede ser de grado o de postgrado.</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>Grado, Postgrado</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>

Nombre:	Recurso	Fecha:	08/09/2011
----------------	----------------	---------------	------------

Descripción:	<i>Recurso digital que se encuentra disponible en un curso para la participación de los estudiantes. Es lo que compone a los cursos. Cada recurso pertenece a un solo curso.</i>
Responsable:	<i>Agustín Mullin</i>
Fuente:	<i>Base de datos de Moodle</i>
Usos:	<i>Cada recurso tiene un grado de interacción, mediante el grado de interacción de todos los recursos de un curso se calcula el grado de interacción del curso.</i>
Supuestos:	
Identificador:	<i>[Id, Tipo, Curso]</i>
Atributos:	
Nombre:	<i>Id</i>
Descripción:	<i>Identifica unívocamente al recurso.</i>
Cardinalidad:	<i>1</i>
Dominio:	
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Tipo</i>
Descripción:	<i>Dice de qué tipo es el recurso.</i>
Cardinalidad:	<i>1</i>
Dominio:	
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Grado de interacción</i>
Descripción:	<i>Es el grado de interacción que el recurso le brinda al estudiante.</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>Repositorio, Interacción Estudiante-Contenido, Interacción Estudiante-Actividad, Interacción Estudiante-Docente, Interacción Estudiante-Estudiante</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Curso</i>
Descripción:	<i>Es el curso al que pertenece el recurso</i>
Cardinalidad:	<i>1</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<u><i>Curso</i></u>

Nombre:	Curso	Fecha:	28/02/2011
Descripción:	<i>Un curso es el dictado de una asignatura en un EVA.</i>		
Responsable:	<i>Agustín Mullin</i>		

Fuente:	<i>Base de datos de Moodle</i>
Usos:	<i>Representa el concepto de curso dentro del sistema</i>
Supuestos:	
Identificador:	<i>[Asignatura, Semestre, Año]</i>
Atributos:	
Nombre:	<i>Asignatura</i>
Descripción:	<i>Es la asignatura que corresponde al curso</i>
Cardinalidad:	<i>1</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<i><u>Asignatura</u></i>
Nombre:	<i>Semestre</i>
Descripción:	<i>El semestre en el que se dictó el curso</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>[1,2]</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Año</i>
Descripción:	<i>Año en que se dictó el curso</i>
Cardinalidad:	<i>1</i>
Dominio:	<i>Naturales</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Grado de interacción</i>
Descripción:	<i>Representa el grado de interacción del curso. El mismo se calcula en base a las actividades que se proponen en el curso.</i>
Cardinalidad:	<i>1</i>
Dato calculado:	<i>Sí, se calcula en base a los grados de interacción de los recursos del curso.</i>
Dominio:	<i>Repositorio, Interacción Estudiante-Contenido, Interacción Estudiante-Actividad, Interacción Estudiante-Docente, Interacción Estudiante-Estudiante</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Cantidad Gr.Int. Repositorio, Cantidad Gr.Int.Int.Est.Cont., etc.</i>
Descripción:	<i>Representa a 5 atributos. Se tiene uno por cada grado de interacción definido. Su valor es la cantidad de recursos del curso, que tienen ese grado de interacción.</i>
Cardinalidad:	<i>1</i>
Dato calculado:	<i>Sí, se cuentan los recursos del curso con el grado de interacción correspondiente.</i>
Dominio:	<i>Naturales</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Nombre:	<i>Recursos</i>
Descripción:	<i>Son los recursos del curso.</i>

Cardinalidad:	<i>N</i>
Obligatoriedad:	<i>Este dato es obligatorio</i>
Concepto Rel. :	<u>Recurso</u>

Fase II - Modelado de Dominio

“El objetivo principal de la presente fase es construir un modelo que represente, utilizando un modelo formal, los conceptos identificados en la Fase I detallados en el documento Metadata de Conceptos.

El modelo incluye la información y características más importantes, relaciones definidas y las restricciones consideradas en el análisis del dominio, abstrayendo la realidad presentada y eliminando posibles inconsistencias y/o conflictos que no se identificaran en la Fase I.

Como resultado de la fase se obtiene el modelo semántico de dominio que representa la realidad relevada, modelando la información de contexto y datos importantes para cada concepto, las relaciones entre estos y las restricciones identificadas.

Adicionalmente, al finalizar la fase se cuenta con una herramienta denominada Matriz de Requerimientos que refleja la interacción entre los conceptos y relaciones del modelo de dominio, y los requerimientos analizados.” (3) A la Matriz de Requerimientos se la denominará Matriz de Trazabilidad, que es el nombre que adopta VGPCDW. (1)

Al construir ontología de dominio dentro del marco de VGPCDW se tiene la restricción de que el modelo construido será utilizado por la fase III de la metodología, para realizar el modelo conceptual multidimensional. Esto es una restricción porque al momento de aplicar las reglas de la fase III, las clases de la ontología se convierten en niveles CMDM. De esta forma, al existir diferentes formas de modelar un mismo concepto, cada uno lleva aparejado un modelo multidimensional diferente, y las decisiones que se tomen en esta fase pueden repercutir en el resto. VGPCDW no define lineamientos para afrontar este tipo de disyuntiva.

Se presentan a continuación tres modelos de dominio, que ofrecen diferentes grados de expresividad para modelar un concepto central del dominio (el grado de interacción de los cursos), y se analiza cómo los diferentes modelos pueden afectar la aplicación de la fase III.

Dado el relevamiento de dominio realizado, los conceptos sobre los que pueden surgir dudas son Semestre y Grado de interacción, que corresponden a dos dimensiones especiales, respectivamente a la dimensión tiempo y a una dimensión de medida. Ambas requieren ser modeladas en CMDM como niveles.

Respecto al Semestre, podría intentarse inicialmente representarlo como una clase de la ontología para conseguir un nivel en CMDM, pero esto es un poco forzado ya que claramente

es un atributo de la clase Curso. De todas formas Semestre no plantea un problema porque VGPCDW considera reglas especiales para tratar las dimensiones de tiempo. Así que es modelado como un atributo de Curso.

Respecto al Grado de interacción de los recursos y cursos, existen tres posibilidades: 1) modelarlo como un atributo; 2) modelar el Grado de interacción como una clase y los grados de interacción posibles como subclases; 3) a la opción 2 agregarle una clase métrica que explicita cómo se realiza el cálculo de la medida Grado de interacción. Tiene sentido modelar el Grado de interacción como clase en la ontología, ya que con ello se lo explicita y se gana mucho en claridad y entendimiento por parte del usuario.

Se presentan a continuación tres ontologías que fueron construidas de forma iterativa, donde se modela de diferente forma el Grado de interacción de los recursos y cursos, donde cada una implementa una de las opciones.

La ontología 1 que se muestra en la figura 22, modela el Grado de interacción como un atributo de las clases Curso y Recurso. Este modelo tiene la desventaja de que no se explicita el concepto Grado de interacción que es central para los requerimientos. Además, al aplicar la fase III esto no genera un nivel de Grado de interacción.

La ontología 2 que se muestra en la figura 23, modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. De esta forma se explicita el concepto de Grado de interacción. Este modelo tiene el problema de que no se explicita el cálculo del Grado de interacción. Si sobre este modelo se aplican las reglas de la fase III, se crea el nivel Grado de interacción, que es necesario, pero se crea también un nivel para cada grado de interacción posible. Estos niveles sobran en CMDM, ya que realmente son instancias del tipo Grado de interacción, sería equivalente a tener los niveles Instituto e Instituto de Computación.

La ontología 3 que se muestra en la figura 24, pretende dejar explícito cómo se calcula el grado de interacción de un curso, y para ello agrega una clase “Métrica de Grado de Interacción” la cual define una fórmula basada en propiedades de los recursos para medir el grado de interacción. A esta clase se le dan tres propiedades, obtains, through y for. Estas propiedades indican qué calcula, cómo lo hace y para qué clase respectivamente, lo que se muestra en la figura 25. Como modelo del dominio, esta ontología es la más explícita. Sin embargo, al aplicar las reglas de la fase III también genera clases innecesarias para el modelo multidimensional.

Dado lo expuesto anteriormente, ninguna de las ontologías garantiza una posterior construcción correcta del modelo multidimensional. Esto se debe realmente a que VGPCDW no presenta reglas para construir las medidas de las relaciones dimensionales en CMDM. Además, VGPCDW no presenta lineamientos para modelar conceptos centrales del dominio. Por lo tanto, dado que la Ontología 1 no genera niveles innecesarios, este proyecto utiliza para seguir con la aplicación de VGPCDW. Se presenta junto con la ontología, en la tabla 12, la Matriz de Trazabilidad asociada a este modelo.

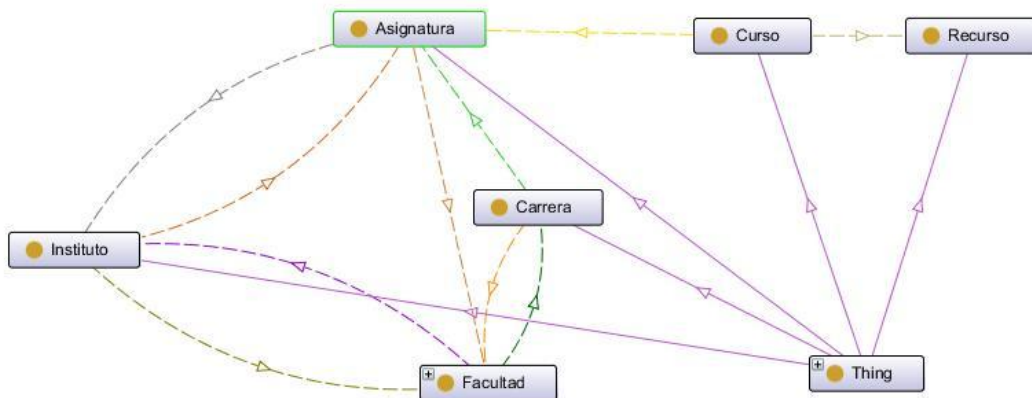


Figura 22 - Ontología 1. Modela el Grado de interacción como un atributo de Curso.

Concepto	RQ1	RQ2	RQ3
Clases			
Asignatura	√	√	√
Carrera			√
Curso	√	√	√
Facultad			√
Instituto			√
Recurso	√	√	√
Relaciones			
asignaturaCurso	√	√	√
asignaturasCarrera			√
asignaturasInstituto			√
carrerasFacultad			√
facultadAsignatura			√
facultadCarrera			√
facultadInstituto			√
institutoAsignatura			√
institutosFacultad			√
recursosCurso	√	√	√

Tabla 12 - Matriz de Trazabilidad de la ontología 1.

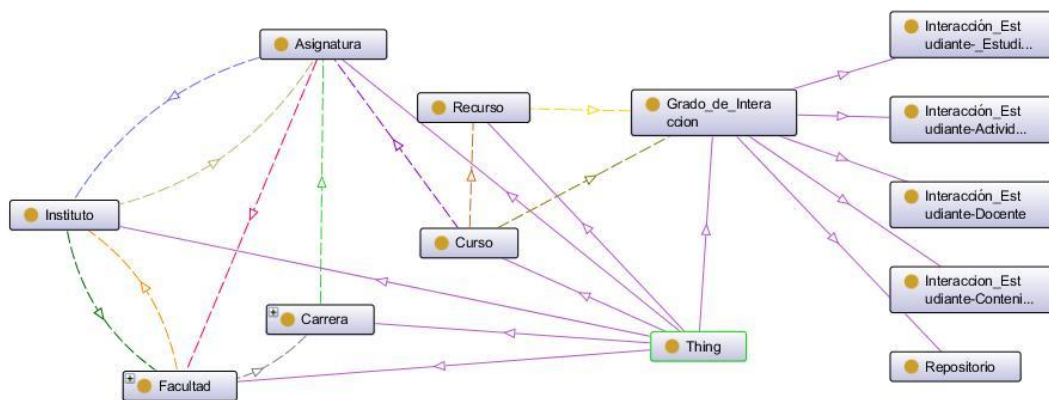


Figura 23 - Ontología 2 . Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases.

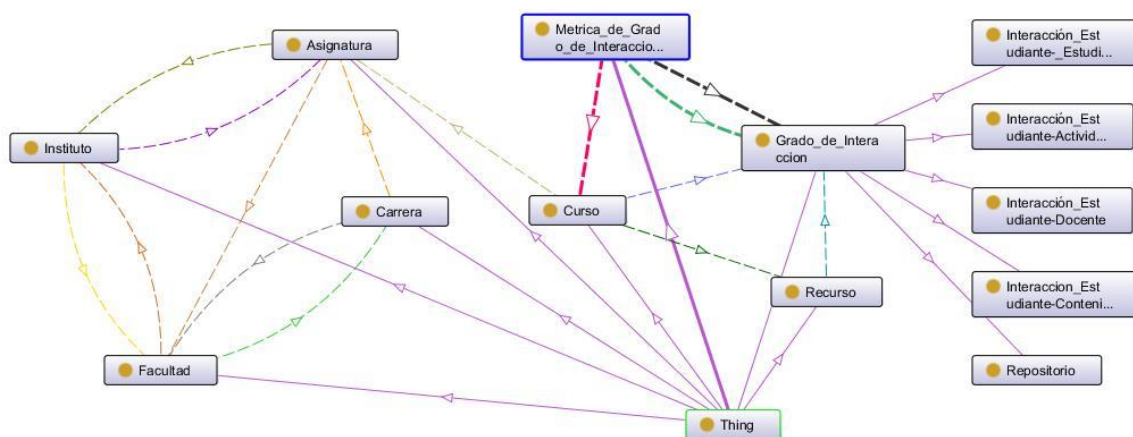


Figura 24 - Ontología 3 . Modela el Grado de interacción como una clase y los grados de interacción posibles como subclases. Además, la Métrica de Grado de interacción muestra cómo es calculado.

En detalle podemos observar que la Medida define el grado de interacción de un curso, basándose en el grado de interacción de sus recursos.

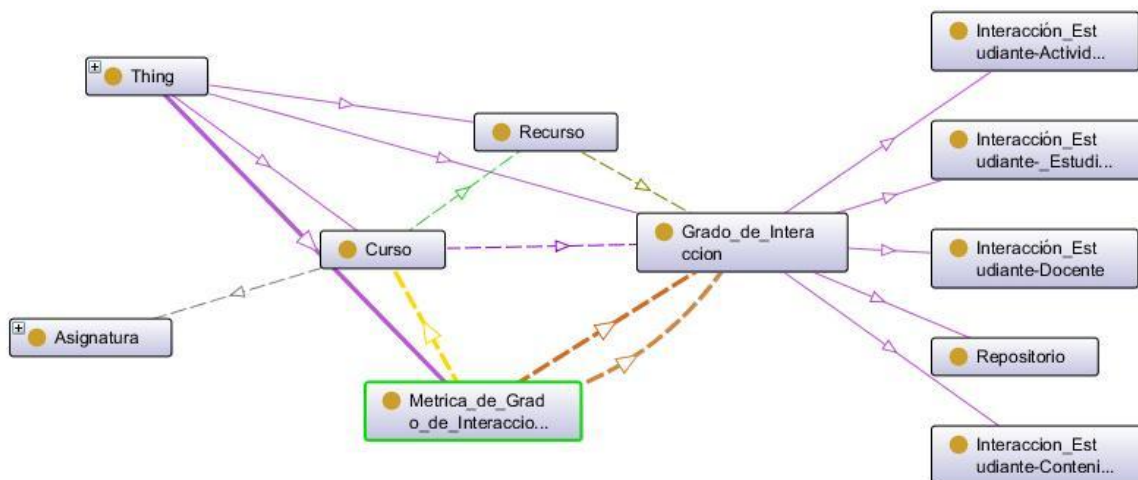


Figura 25 - Detalle de la Métrica del Grado de Interacción de la ontología 3.

Fase III - Modelado Conceptual Multidimensional

“El objetivo de la presente fase es construir el modelo multidimensional a partir del modelo de dominio, guiado por los requerimientos” (3). Para esto se tendrá en cuenta el conjunto de requerimientos iniciales, el documento Metadata de conceptos, el modelo de dominio (la ontología 1) y la Matriz de Trazabilidad (Matriz de Requerimientos).

“Se utiliza un conjunto de reglas que permiten la obtención del modelo conceptual multidimensional a partir del modelo de dominio, guiando el proceso de transformación de los elementos del modelo en estructuras del modelo conceptual. Con estas reglas debe ser posible traducir los conceptos y relaciones representados en el modelo de dominio a las estructuras requeridas para el modelo conceptual.” (3)

Consideraciones Iniciales

Se considera el concepto Tiempo especial, que puede considerarse como un objeto de la realidad. En CMDM una dimensión representa un objeto que participa en el problema por lo que es factible considerar al concepto Tiempo directamente como una dimensión del modelo. Por este motivo las reglas propuestas en MD4DW no se aplican a este concepto.

Se identifica la siguiente dimensión asociada al concepto Tiempo, denominada Semestre, que se muestra en la figura 26. Esta dimensión se construye en concordancia con las reglas para la construcción de dimensiones que se verán a continuación. Que el nivel Curso aparezca dentro de la dimensión Semestre es consistente con las reglas definidas en MD4DW. Este punto es abordado nuevamente más adelante.

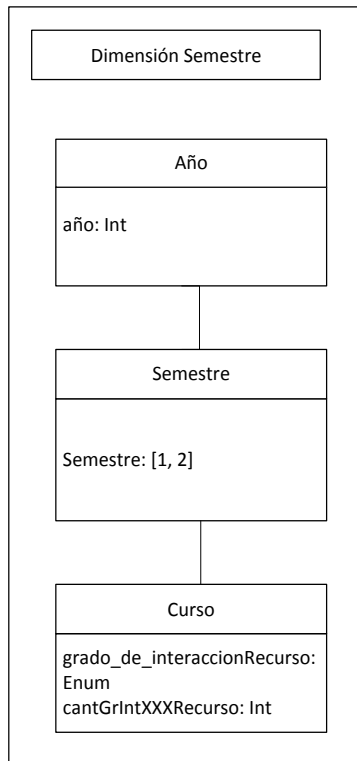


Figura 26 - Dimensión Tiempo del modelo conceptual multidimensional.

A continuación se procede a aplicar las reglas de MD4DW.

Definición de Niveles

Regla R1 - Definición de clases candidatas a niveles del modelo multidimensional

Definición de clases candidatas a niveles del modelo multidimensional.

Clases candidatas:

Área de facultades, Asignatura, Carrera, Curso, Facultad, Instituto, Recurso. Se muestran en la figura 27.

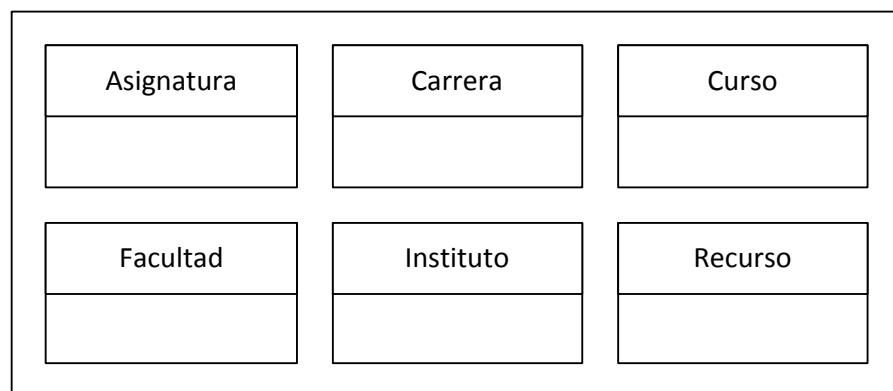


Figura 27 - Clases candidatas a niveles del modelo conceptual multidimensional.

Regla R2 - Definición de niveles del modelo multidimensional

Definición del conjunto de niveles del modelo CMDM.

Se muestran en la figura 28. Para mejorar la simplicidad del diagrama, en lugar de poner los cinco atributos de la clase Curso, correspondientes a la cantidad de recursos asociados al curso para cada grado de interacción, solamente se incluye uno a modo de referencia con el nombre XXX.

Se incluyen los atributos de Curso vinculados a la dimensión de tiempo Semestres, ya que ninguna regla especifica que deban retirarse. De todas formas, se retirarán posteriormente.

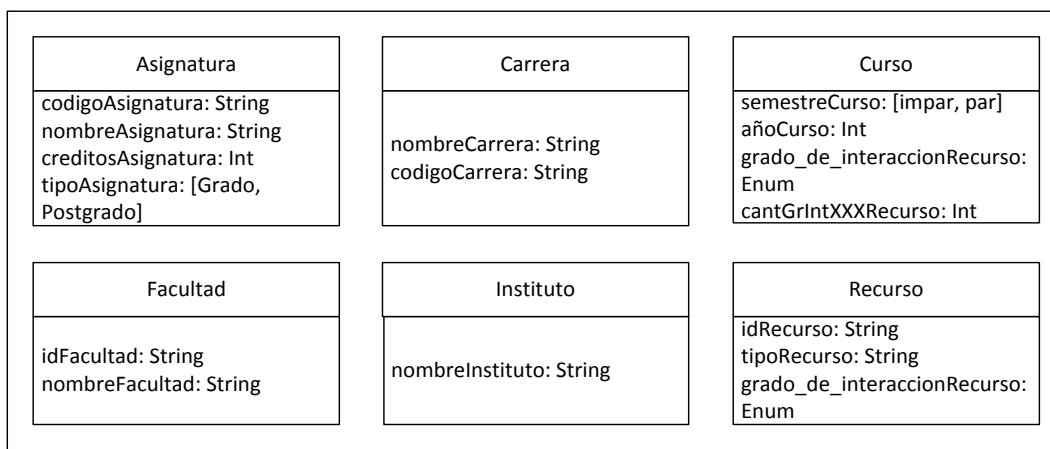


Figura 28 - Conjunto de niveles del modelo conceptual multidimensional.

Definición de Jerarquías

Regla R3 - Definición de cardinalidades

Identificar y definir cardinalidades en las relaciones interclases de la ontología.

Clase	Clase	Cardinalidad	Relación
Curso	Asignatura	N:1	asignaturaCurso
Facultad	Asignatura	N:1	facultadAsignatura
Carrera	Facultad	N:1	facultadCarrera
Instituto	Facultad	N:1	facultadInstituto
Asignatura	Instituto	N:1	institutoAsignatura
Recurso	Curso	N:1	recursosCurso
Asignatura	Carrera	N:N	asignaturaCarrera

Tabla 13 - Definición de cardinalidades en las relaciones interclases de la ontología.

Regla R4 - Identificación de orden entre niveles a partir de cardinalidad N:1
 Identificar órdenes entre los niveles del modelo multidimensional.

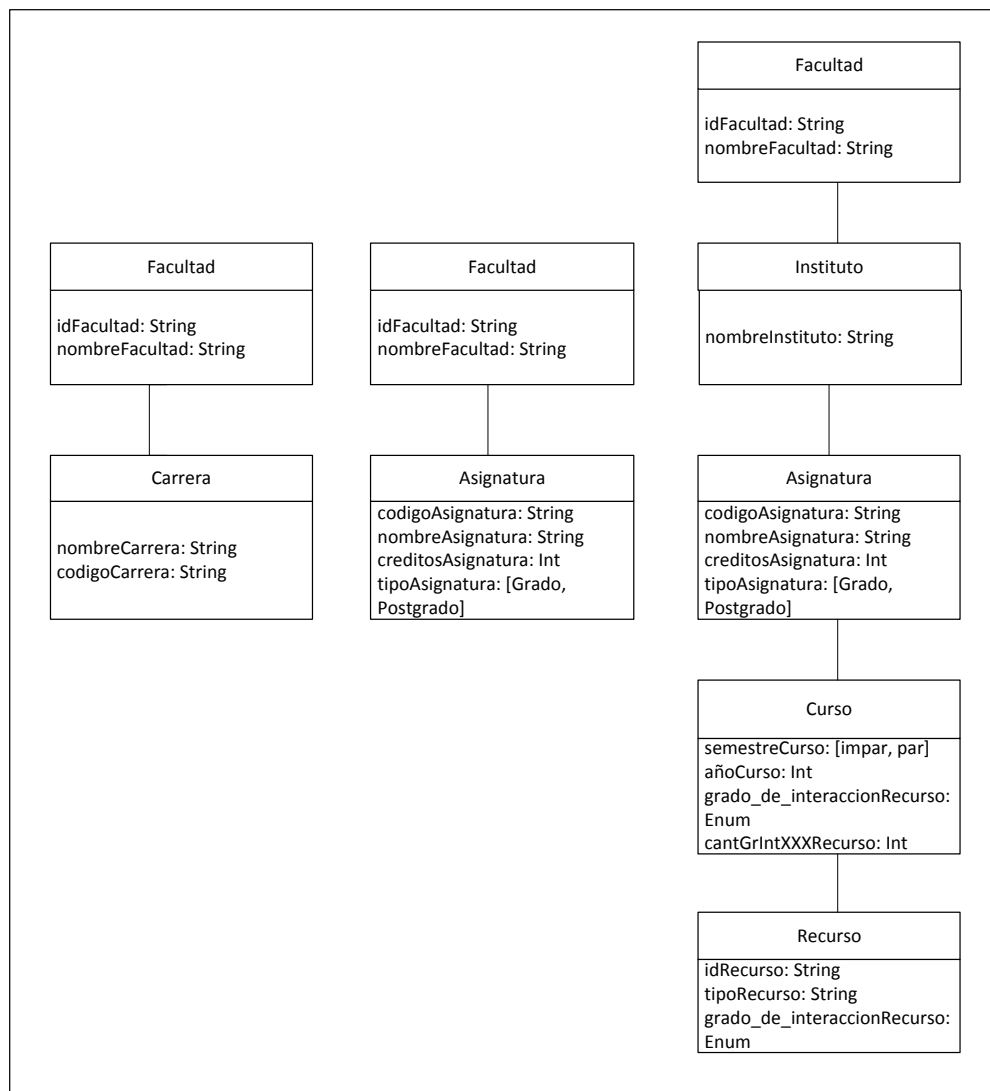


Figura 29 - Órdenes entre los niveles del modelo conceptual multidimensional.

Regla R5 - Identificación de orden entre niveles a partir de cardinalidad 1:N
 Identificar órdenes entre los niveles del modelo multidimensional.
 Regla aplicada.

Regla R6 - Identificación de orden entre niveles a partir de cardinalidad N:N
 Identificar órdenes entre los niveles del modelo multidimensional.

En este paso se identifican las siguientes relaciones de cardinalidad N:N:
 Asignatura - Carrera
 Carrera - Curso

Carrera - Instituto
Carrera - Recurso

Según la metodología estas clases solamente se pueden relacionar de la siguiente forma:

- a) las clases se mapean como niveles de jerarquías diferentes en una misma dimensión
- b) las clases se mapean como niveles de diferentes dimensiones.

No es posible que se cumpla el caso a), donde Carrera pertenezca a la misma dimensión que Asignatura, pero que pertenezcan a jerarquías distintas, así que debe ser el caso b).

De esta forma Carrera debe pertenecer a una dimensión diferente.

Regla R7 - Identificación de orden entre niveles a partir de cardinalidad 1:1 con Totalidad

Identificar órdenes entre los niveles del modelo multidimensional.

No aplica.

Regla R8 - Identificación de orden entre niveles a partir de cardinalidad 1:1

Identificar órdenes entre los niveles del modelo multidimensional.

No aplica.

Regla R9 - Categorización de subclases (a partir de subclases distintas)

Identificar órdenes entre niveles del modelo multidimensional.

No aplica.

Definición de dimensiones y relaciones dimensionales

Regla R10 - Identificación de dimensiones en CMDM

Identificar dimensiones del modelo multidimensional.

En base a lo definido en la Regla 6, Carrera debe pertenecer a una dimensión diferente de Asignatura. Así se lo modela en la figura 30.

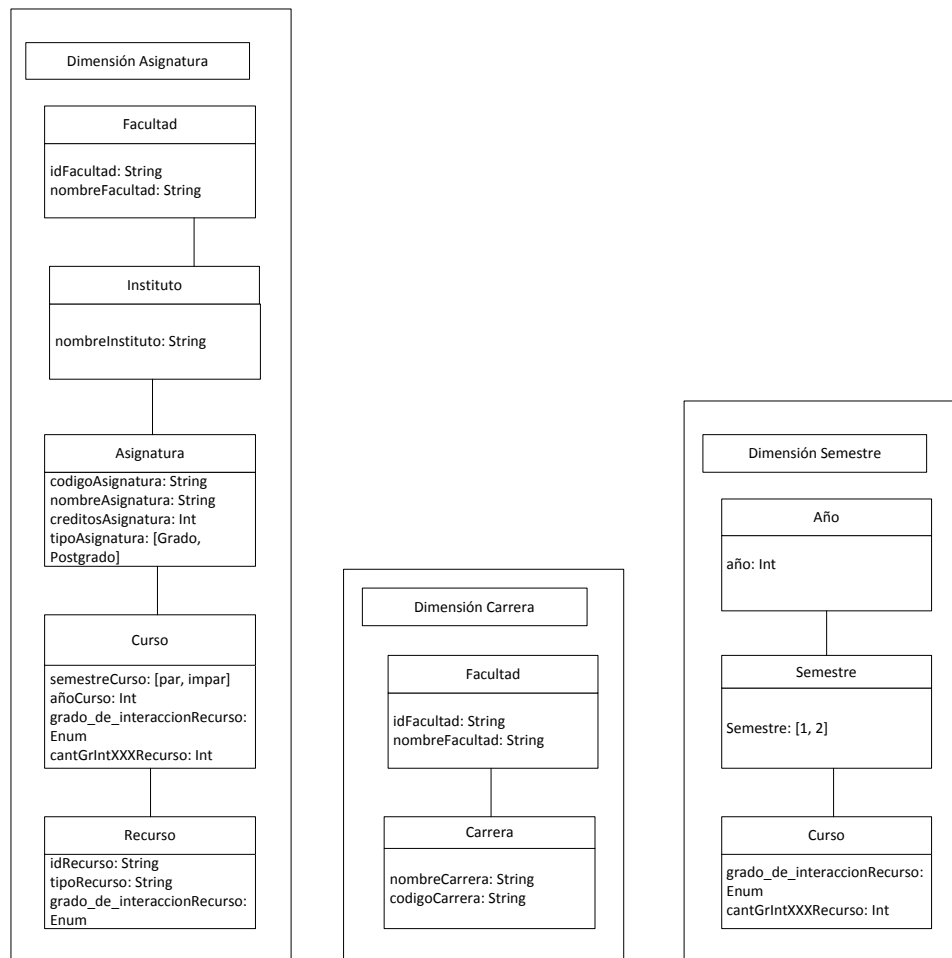


Figura 30 - Dimensiones del modelo conceptual multidimensional.

Regla R11 – Identificación de relaciones dimensionales en CMDM

Identificar relaciones dimensionales del modelo multidimensional CMDM.

La Regla 11 establece lo siguiente:

En el caso de contar con una dimensión donde a partir de su nivel inferior I se desprendan varias jerarquías, y a su vez la cardinalidad entre I con los niveles superiores sea $N:1$ en todos los casos, es posible definir una relación dimensional donde I sea el eje central de la relación. Las dimensiones que definen dicha relación dimensional quedan formadas por el conjunto de niveles superiores a partir del eje central.

Esta regla sufre de dos problemas, en primer lugar no propone cómo tratar las relaciones de cardinalidad $N:N$. Este es el caso de la relación entre Asignaturas y Carreras, que no es contemplada por esta regla.

Segundo, pero más importante, la regla falla al entender el modelo multidimensional CMDM, planteando la construcción de una estructura que no existe en el modelo, la de “eje central de la relación”.

Esta regla confunde el enfoque multidimensional de CMDM con el enfoque de estrella, donde la tabla de hechos es el centro de la relación, vinculando a las dimensiones. La regla asocia el nivel más bajo de las jerarquías con “el eje central de la relación” de CMDM, pero en CMDM ningún nivel es asociado a una relación dimensional, ya que la relación dimensional representa el conjunto de cubos que se pueden construir con los niveles de las dimensiones participantes.

Siguiendo la propuesta de esta regla se llega al modelo de la figura, que parece un diseño de estrella con Curso como el hecho de la misma.

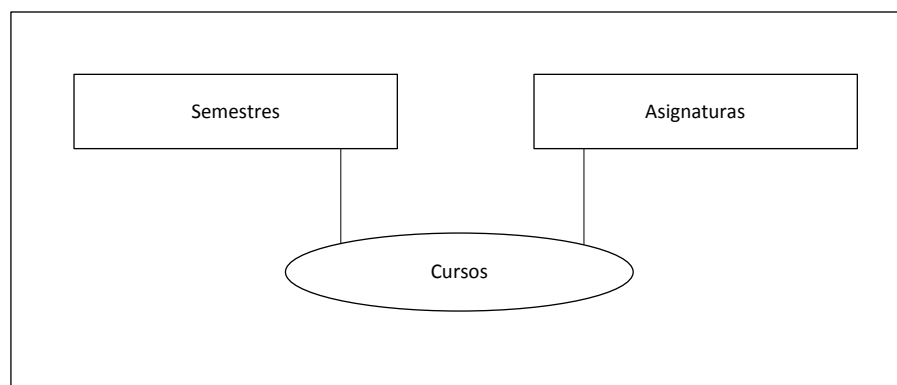


Figura 31 - Aplicación de la Regla 11 de la fase III de VGPCDW, que resulta en un modelo de estrella.

CMDM define dimensiones y medidas para las relaciones dimensionales, y al permitir dimensionalidad genérica, las medidas pueden ser vistas como dimensiones, y viceversa.

Pero esta regla no define como establecer la medida de la relación dimensional. En el resultado de la aplicación de esta regla, se obtiene una relación dimensional sin medida. Siendo claro de la especificación que la medida de la relación es el curso y su grado de interacción.

Esta regla no contempla que las medidas, tal cual las dimensiones, puedan también tener jerarquías, como está definido en CMDM.

Además de los problemas expuestos, aplicando la regla como está definida, no se consigue ubicar en la relación dimensional a la dimensión de Carreras, y por ende al nivel Carrera. Por lo que, con el resultado obtenido no se consigue satisfacer los requerimientos iniciales.

La Regla 11 solamente sugiere crear relaciones dimensionales en el caso de haber dimensiones que partan del mismo nivel, siempre y cuando estas dimensiones empiecen con relaciones N:1. La regla no considera otro tipo de situaciones para sugerir relaciones dimensionales.

En este caso se tiene la relación entre Asignaturas y Carreras, que tiene una cardinalidad N:N y no es considerada por la metodología. La forma de modelarlo sería creando otra relación dimensional que contemple esta relación.

Se observa además que el resultado obtenido por la aplicación pura de las reglas, obtiene un resultado que no es sintácticamente correcto en CMDM, y sobre el cual no es posible de aplicar el algoritmo de Verónica Peralta.

En base a lo analizado, se modifica el modelo para que consiga satisfacer los requerimientos. Se realizan las siguientes modificaciones al modelo. Para que la relación dimensional definida cuente con una medida, se define la dimensión Cursos, que se muestra en la figura 32. Además, corresponde quitar el nivel Recurso de la dimensión, porque la medida de la relación dimensional es el Curso, y no el Recurso. Esto responde a que el Recurso no es necesario para el Data Warehouse, solamente se lo precisa para hacer el cálculo del Grado de interacción de los Cursos.

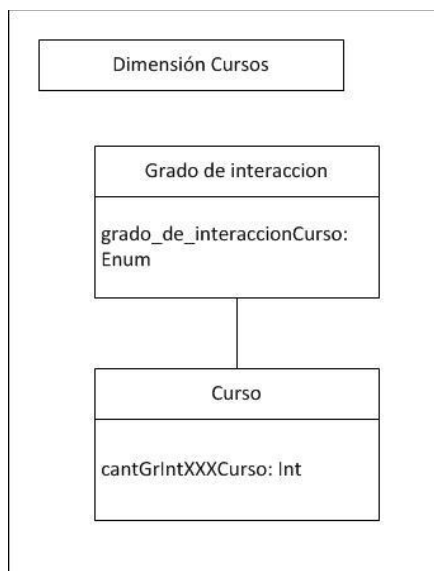


Figura 32 - Dimensión Cursos del modelo conceptual multidimensional.

La dimensión tiene el nivel Curso, por lo que este debe ser retirado de todas las otras dimensiones. Además, se agrega el nivel Grado de Interacción, que permite agregar los cursos por su grado de interacción. Esta relación es N:1 como debe ser en el modelo CMDM. Este caso muestra un ejemplo de una dimensión utilizada como medida, la cual cuenta con una jerarquía.

Por lo tanto, todas las dimensiones definidas hasta ahora son redefinidas, retirando el nivel Curso de cada una.

Además, se agrega la relación Asignaturas-Carreras, que vincula las dimensiones Asignaturas y Carreras. La medida de esta dimensión no se especifica, ya que es el tipo bool. Esto significa que

la relación muestra los cruzamientos entre los niveles Asignatura y Carrera, cuya relación tiene cardinalidad N:N.

El modelo obtenido, que permite satisfacer los requerimientos, se presenta en las figuras 33 y 34, en las que se presentan respectivamente las relaciones dimensionales y las dimensiones.

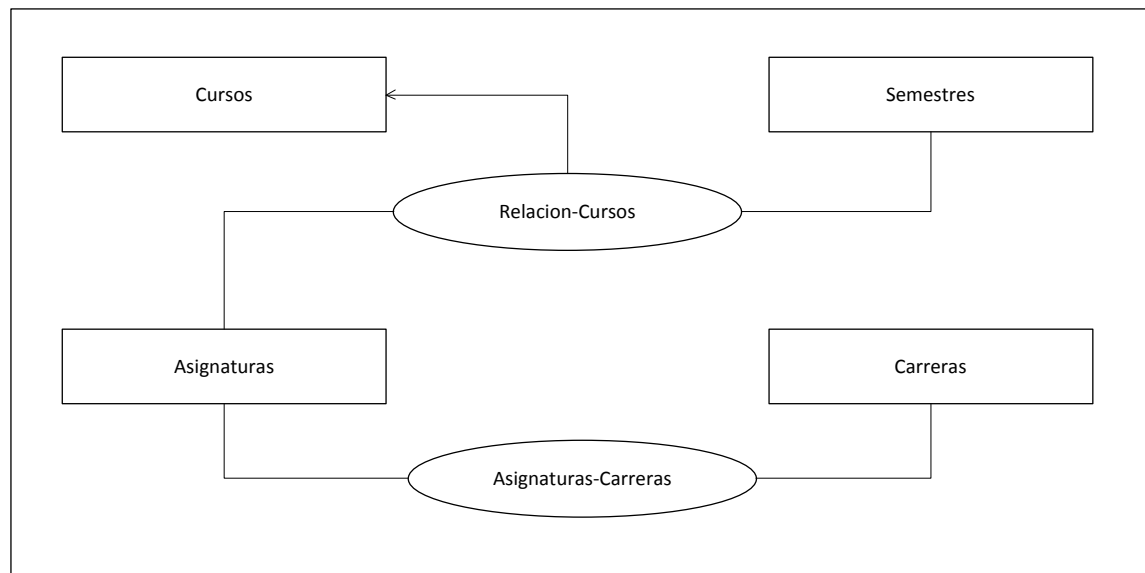


Figura 33 - Relaciones dimensionales del modelo conceptual multidimensional corregido.

El modelo construido tiene dos relaciones dimensionales: Relación-Cursos y Asignaturas-Carreras. La relación dimensional Relación-Cursos vincula las dimensiones Asignaturas y Semestres, y tiene como medida la dimensión Cursos. Con esta relación es posible dado un semestre y una asignatura obtener el curso correspondiente. Además, gracias a la dimensionalidad genérica y a que la dimensión Cursos tiene definida una jerarquía, es posible realizar una operación de agregación sobre los cursos.

La relación Asignaturas-Carreras modela la relación entre Asignaturas y Carreras. Esta relación tiene como medida los booleanos ya que su función es indicar si una Asignatura está asociada a una Carrera o no.

El requisito de visualizar los cursos dictados para una Carrera se satisface obteniendo de la relación Asignaturas-Carreras las Asignaturas asociadas a cada Carrera, y utilizando estas Asignaturas como entrada para la relación dimensional Relación-Cursos.

Luego de las modificaciones mencionadas, se obtiene un modelo CMDM correcto, sobre el que es aplicado el resto del algoritmo.

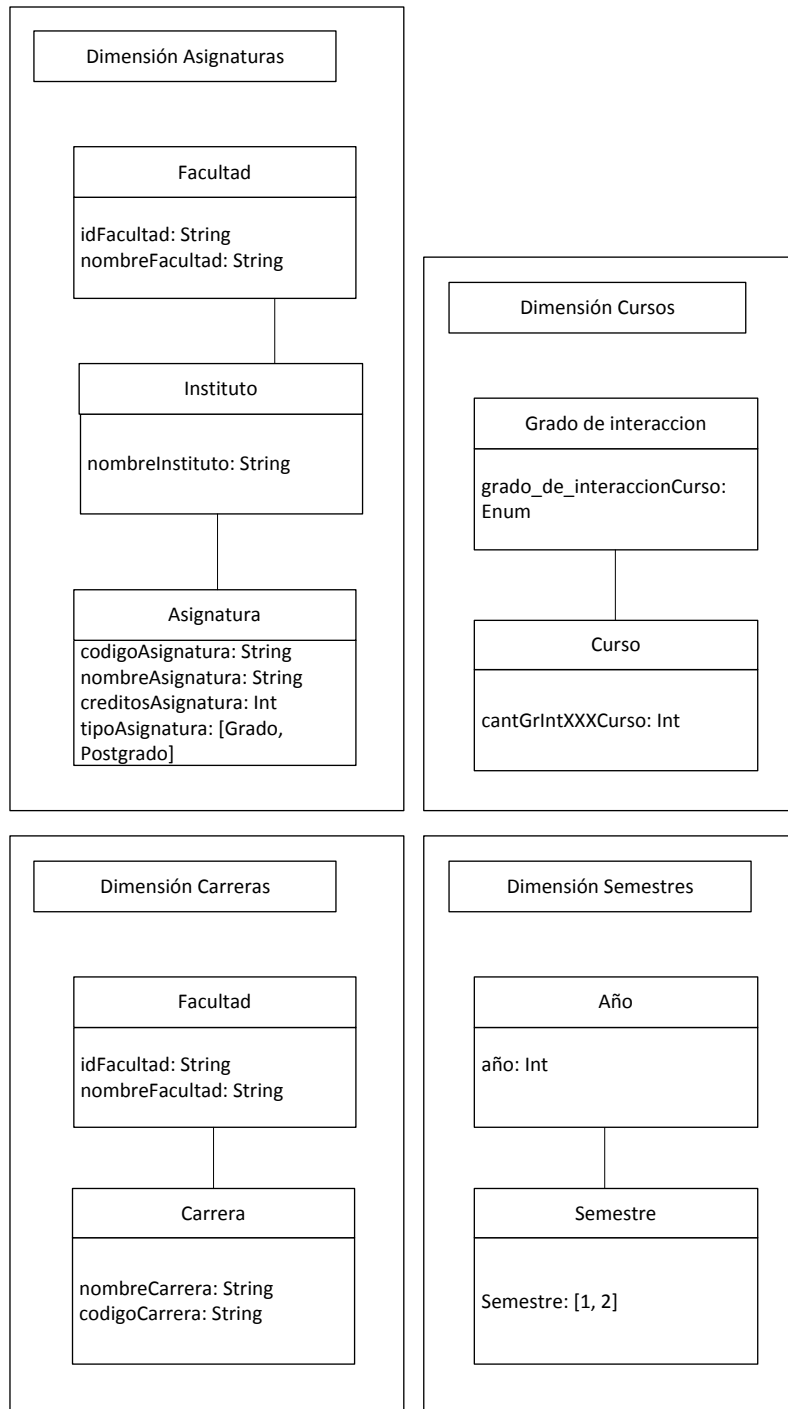


Figura 34 - Dimensiones del modelo conceptual multidimensional corregido.

Fase IV - Construcción del ODS

“Esta fase tiene como objetivo generar una fuente de datos relacional única que cubra toda la realidad” (1). Como entrada se utilizará la ontología OWL2 creada en la Fase II, y como salida se obtendrá una base de datos relacional que pueda almacenar toda la información del Data Warehouse. Para ello se aplica el algoritmo Onto2rdb.

Se trabaja en base a la Ontología 1 de la Fase II, que se muestra en la figura 35.

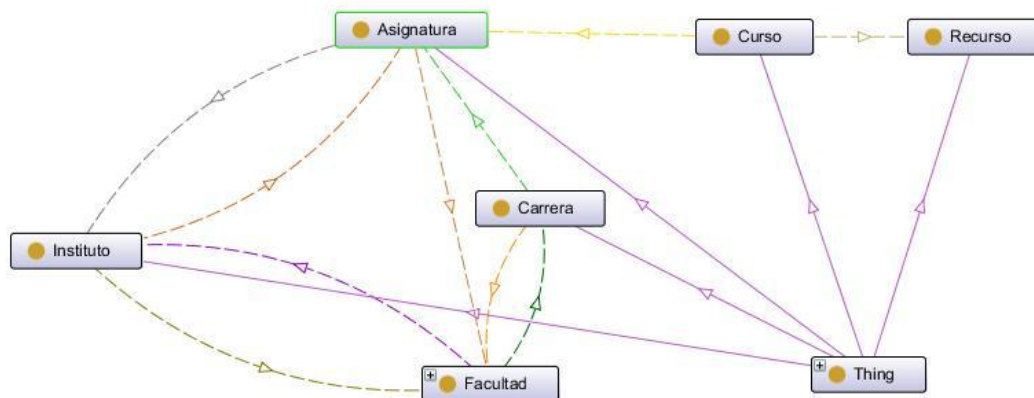


Figura 35 - Ontología de dominio resultado de aplicar la fase II de VGPCDW.

Precondiciones del algoritmo

Se asume que las clases de la ontología son disjuntas dos a dos con la excepción de subclase y superclase en las jerarquías. La precondition se cumple.

Algoritmo

Se presenta a continuación la aplicación detallada del algoritmo Onto2rdb.

1. Nuevo esquema

Se crea un esquema de base de datos vacío para agregarle las tablas en los siguientes pasos.

```
CREATE DATABASE IF NOT EXISTS `ods` DEFAULT CHARACTER SET latin1;
grant all on quartz.* to 'ods_user'@'localhost' identified by 'password';
```

```
USE `ods`;
```

2. Una tabla por cada clase

Se recorren las clases alfabéticamente por su nombre y por cada una, se agrega al nuevo esquema de base de datos (creado en el paso anterior), una tabla con el mismo nombre de la clase. Para la nueva tabla se agrega un campo por cada datatype property de la clase con el mismo nombre y tipo (rango de la property) de esta. Si existe una restricción en la ontología de exactly 1 para esa datatype property, el campo se agrega no admitiendo nulos, de otra forma, el campo admitirá nulos.

```
DROP TABLE IF EXISTS Asignatura;
```

```
DROP TABLE IF EXISTS Carrera;
```

```
DROP TABLE IF EXISTS Curso;
```

```
DROP TABLE IF EXISTS Facultad;
```

```
DROP TABLE IF EXISTS Instituto;
```

```
DROP TABLE IF EXISTS Recurso;
```

```
CREATE TABLE Asignatura
```

```
(  
    codigoAsignatura VARCHAR(80) NOT NULL,  
    nombreAsignatura VARCHAR(80) NOT NULL,  
    creditosAsignatura UNSIGNED INT NOT NULL,  
    tipoAsignatura UNSIGNED INT NOT NULL  
);
```

```
CREATE TABLE Carrera
```

```
(  
    codigoCarrera VARCHAR(80) NOT NULL,  
    nombreCarrera VARCHAR(80) NOT NULL  
);
```

```
CREATE TABLE Curso
```

```
(  
    semestreCurso INT UNSIGNED NOT NULL,  
    anoCurso INT UNSIGNED NOT NULL,  
    grado_de_interaccionCurso INT UNSIGNED NOT NULL,  
    cantGrIntRepCurso INT UNSIGNED NOT NULL,  
    cantGrIntEstContCurso INT UNSIGNED NOT NULL,  
    cantGrIntEstActCurso INT UNSIGNED NOT NULL,  
    cantGrIntEstDocCurso INT UNSIGNED NOT NULL,  
    cantGrIntEstEstCurso INT UNSIGNED NOT NULL  
);
```

```
CREATE TABLE Facultad
```

```
(
```

```

    idFacultad VARCHAR(80) NOT NULL,
    nombreAsignatura VARCHAR(80) NOT NULL
);

CREATE TABLE Instituto
(
    nombreInstituto VARCHAR(80) NOT NULL
);

CREATE TABLE Recurso
(
    idRecurso VARCHAR(80) NOT NULL,
    tipoRecurso VARCHAR(80) NOT NULL,
    grado_de_interaccionRecurso UNSIGNED INT NOT NULL
);

```

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura)
 Carrera(codigoCarrera, nombreCarrera)
 Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso,
 cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso)
 Facultad(idFacultad, nombreAsignatura)
 Instituto(nombreInstituto)
 Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

3. Claves, relaciones 1 a 1 y relaciones n a 1

Mientras haya alguna tabla que no tiene clave primaria definida o queden relaciones N:1 o 1:1 que no se han reflejado en la base de datos, se ejecutan iterativamente las siguientes 3 acciones:

- 3.1. Claves
- 3.2. Relaciones N a 1
- 3.3. Relaciones 1 a 1

3.1. Claves

// Facultad tiene como identificador a un solo atributo, se lo usa como clave
 ALTER TABLE Facultad
 ADD PRIMARY KEY (idFacultad);

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura)
Carrera(codigoCarrera, nombreCarrera)
Curso(semesterCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso,
cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso)
Facultad(idFacultad, nombreAsignatura)
Instituto(nombreInstituto)
Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

3.2. Relaciones N a 1

```
ALTER TABLE Asignatura  
ADD facultadAsignatura VARCHAR(80) NOT NULL;
```

```
ALTER TABLE Asignatura  
ADD FOREIGN KEY (facultadAsignatura)  
REFERENCES Facultad (idFacultad);
```

```
ALTER TABLE Carrera  
ADD facultadCarrera VARCHAR(80) NOT NULL;
```

```
ALTER TABLE Carrera  
ADD FOREIGN KEY (facultadCarrera)  
REFERENCES Facultad (idFacultad);
```

```
ALTER TABLE Instituto  
ADD facultadInstituto VARCHAR(80) NOT NULL;
```

```
ALTER TABLE Instituto  
ADD FOREIGN KEY (facultadInstituto)  
REFERENCES Facultad (idFacultad);
```

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura,
facultadAsignatura)
Carrera(codigoCarrera, nombreCarrera, **facultadCarrera**)
Curso(semesterCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso,
cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso)
Facultad(idFacultad, nombreAsignatura)
Instituto(nombreInstituto, **facultadInstituto**)
Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)

Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)

3.3 Relaciones 1 a 1

No hay.

3.1 Claves

```
// Asignatura usa dos atributos como identificador,  
// se crea un atributo nuevo como identificador, de tipo autoincremental  
ALTER TABLE Asignatura  
ADD id VARCHAR(80) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE Asignatura  
ADD PRIMARY KEY id  
ADD UNIQUE (codigoAsignatura, facultadAsignatura);
```

```
ALTER TABLE Carrera  
ADD id VARCHAR(80) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE Carrera  
ADD PRIMARY KEY id  
ADD UNIQUE (codigoCarrera, facultadCarrera);
```

```
ALTER TABLE Instituto  
ADD id VARCHAR(80) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE Instituto  
ADD PRIMARY KEY id  
ADD UNIQUE (nombreInstituto, facultadInstituto);
```

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id)
Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)
Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso)
Facultad(idFacultad, nombreAsignatura)
Instituto(nombreInstituto, facultadInstituto, id)

Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)

Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)

3.2 Relaciones N a 1

ALTER TABLE Asignatura

ADD institutoAsignatura VARCHAR(80) NOT NULL;

ALTER TABLE Asignatura

ADD FOREIGN KEY (institutoAsignatura)

REFERENCES Instituto (id);

ALTER TABLE Curso

ADD asignaturaCurso VARCHAR(80) NOT NULL;

ALTER TABLE Curso

ADD FOREIGN KEY (asignaturaCurso)

REFERENCES Asignatura (id);

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id, institutoAsignatura)

Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)

Curso(semesterCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso)

Facultad(idFacultad, nombreAsignatura)

Instituto(nombreInstituto, facultadInstituto, id)

Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)

Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)

Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)

Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)

3.3 Relaciones 1 a 1

No hay.

3.1 Claves

```
ALTER TABLE Curso
ADD id VARCHAR(80) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE Curso
ADD PRIMARY KEY id
ADD UNIQUE (semestreCurso, añoCurso, asignaturaCurso);
```

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id, institutoAsignatura)
Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)
Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso, id)
Facultad(idFacultad, nombreAsignatura)
Instituto(nombreInstituto, facultadInstituto, id)
Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)
Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)
Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)
Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)
Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)

3.2 Relaciones N a 1

```
ALTER TABLE Recurso
ADD cursoRecurso VARCHAR(80) NOT NULL;
```

```
ALTER TABLE Recurso
ADD FOREIGN KEY (cursoRecurso)
REFERENCES Curso (id);
```

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id, institutoAsignatura)

Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)
 Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso, id)
 Facultad(idFacultad, nombreAsignatura)
 Instituto(nombreInstituto, facultadInstituto, id)
 Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso, **cursoRecurso**)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)
 Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)
 Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)
 Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)
 Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)
Recurso(cursoRecurso) FOREIGN KEY Curso(id);

3.3 Relaciones 1 a 1

No hay.

3.1 Claves

ALTER TABLE Recurso
 ADD id VARCHAR(80) NOT NULL AUTO_INCREMENT;

ALTER TABLE Recurso
 ADD PRIMARY KEY id
 ADD UNIQUE (idRecurso, tipoRecurso, cursoRecurso);

Resultado parcial

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id, institutoAsignatura)
 Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)
 Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso, id)
 Facultad(idFacultad, nombreAsignatura)
 Instituto(nombreInstituto, facultadInstituto, id)
 Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso, cursoRecurso, id)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)
 Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)
Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)
Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)
Recurso(cursoRecurso) FOREIGN KEY Curso(id);

3.2 Relaciones N a 1

No hay.

3.3 Relaciones 1 a 1

No hay.

3.1 Claves

No hay.

4 Relaciones N a N

DROP TABLE IF EXISTS CarreraAsignatura;

CREATE TABLE CarreraAsignatura
(
 carrera VARCHAR(80) NOT NULL,
 asignatura VARCHAR(80) NOT NULL
);

ALTER TABLE CarreraAsignatura
ADD PRIMARY KEY (carrera, asignatura);

ALTER TABLE CarreraAsignatura
ADD FOREIGN KEY (carrera)
REFERENCES Carrera (id);

ALTER TABLE CarreraAsignatura
ADD FOREIGN KEY (asignatura)
REFERENCES Asignatura (id);

Resultado final

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura,
facultadAsignatura, id, institutoAsignatura)
Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)

Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso, id)

Facultad(idFacultad, nombreAsignatura)

Instituto(nombreInstituto, facultadInstituto, id)

Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso, cursoRecurso, id)

CarreraAsignatura (carrera, asignatura)

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)

Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)

Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)

Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)

Recurso(cursoRecurso) FOREIGN KEY Curso(id)

CarreraAsignatura (carrera) FOREIGN KEY Carrera(id)

CarreraAsignatura (asignatura) FOREIGN KEY Asignatura(id)

6 Restricciones no consideradas

El grado de interacción de un curso se calcula en base a los grados de interacción de los recursos asociados a él.

Los atributos de Curso, tipo cantGrIntXXXCurso, se calcula como la cantidad de recursos del grado de interacción XXX del curso.

El esquema de la base de datos obtenida por el algoritmo se presenta en la figura 36.

La fase IV, para la construcción de una base de datos ODS, se aplicó con total éxito, siendo las indicaciones claras y precisas.

De todas formas, el algoritmo no contempla el proceso de extracción, transformación y carga (ETL) del ODS. Este proceso es en definitiva el proceso ETL del Data Warehouse ya que la carga del Data Warehouse se realiza mediante vistas desde el ODS.

Si al momento de cargar los datos al ODS surgiesen dificultades y no fuese posible cargar algunos datos, esto podría, en el peor de los casos, hacer inviable el modelo multidimensional construido en la fase anterior. Esto podría suceder a pesar de los recaudos tomados en la fase I de relevamiento de requerimientos al identificar las fuentes de los datos. Esto es una debilidad de la metodología VGPCDW.

Sin embargo se debe señalar como una fortaleza el uso de una fuente de datos intermedia, ya que la misma protege al Data Warehouse de cambios en las fuentes. Ante un cambio en los

sistemas transaccionales solamente se debe modificar el sistema de carga del ODS (suponiendo que se sigan teniendo todos los datos necesarios), con lo que el ODS y el Data Warehouse permanecen inalterados.

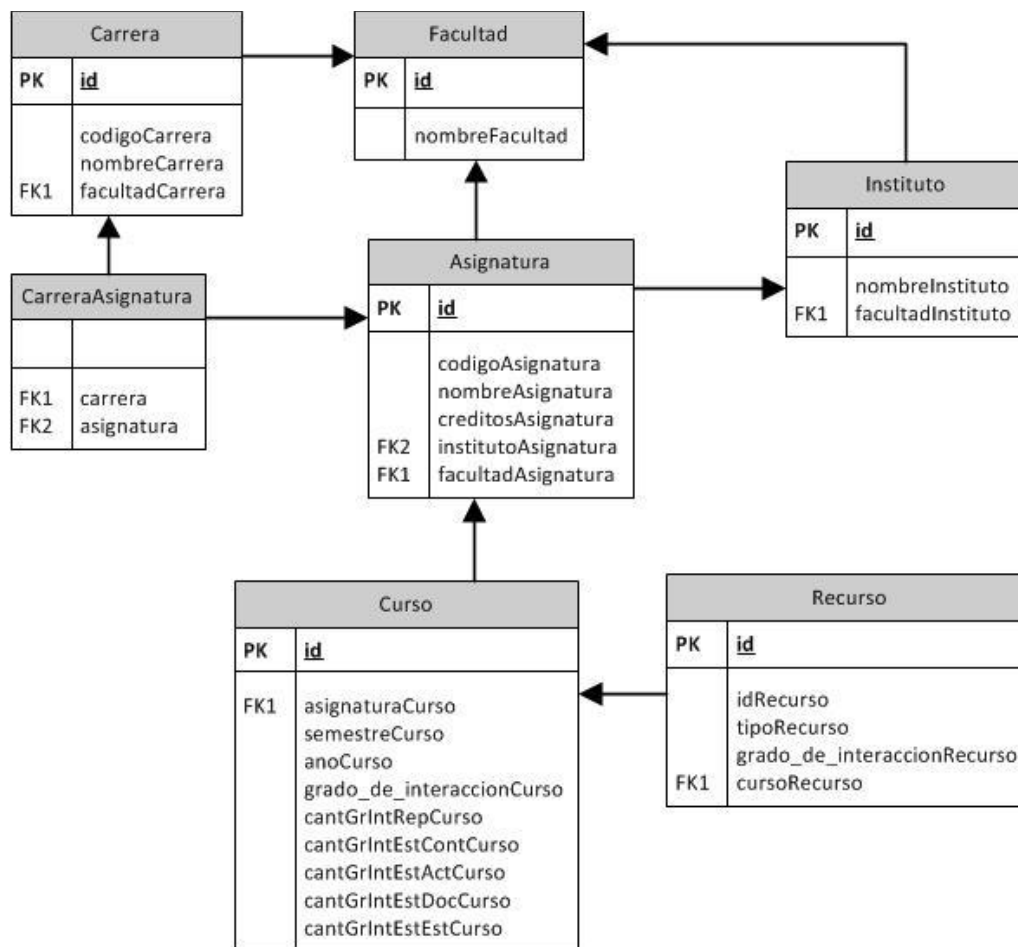


Figura 36 - Esquema del ODS, resultante de aplicar la fase IV de VGPCDW.

Fase V – Generación del diseño lógico y la carga de datos

“Esta fase tiene como objetivo generar el diseño lógico y la carga del Data Warehouse a partir de la fuente canónica y el modelo CMDM” (1).

Como entrada recibe:

- la matriz de trazabilidad (o matriz de requerimientos) (obtenida en la fase II)
- el modelo multidimensional en CMDM (obtenido en la fase III)
- la base de datos relacional que contiene toda la información del Data Warehouse (obtenida en la fase IV).

El algoritmo cuenta con cinco etapas, en las primeras cuatro se define todo lo necesario para aplicar el algoritmo en la quinta etapa.

E1: Esquema conceptual

Se toma el modelo conceptual en CMDM, se define una clave por cada nivel y se cambian aquellos nombres de atributos que aparecen repetidos.

El resultado de este paso son las dimensiones (con los cambios antes descritos) y las relaciones dimensionales del modelo en CMDM.

El modelo obtenido se presenta a continuación. Al aplicar esta etapa, las relaciones dimensionales no se modifican, pero sí cambian las dimensiones. El modelo resultante se muestra en las figuras 37 y 38, donde se presentan respectivamente las relaciones dimensionales y las dimensiones.

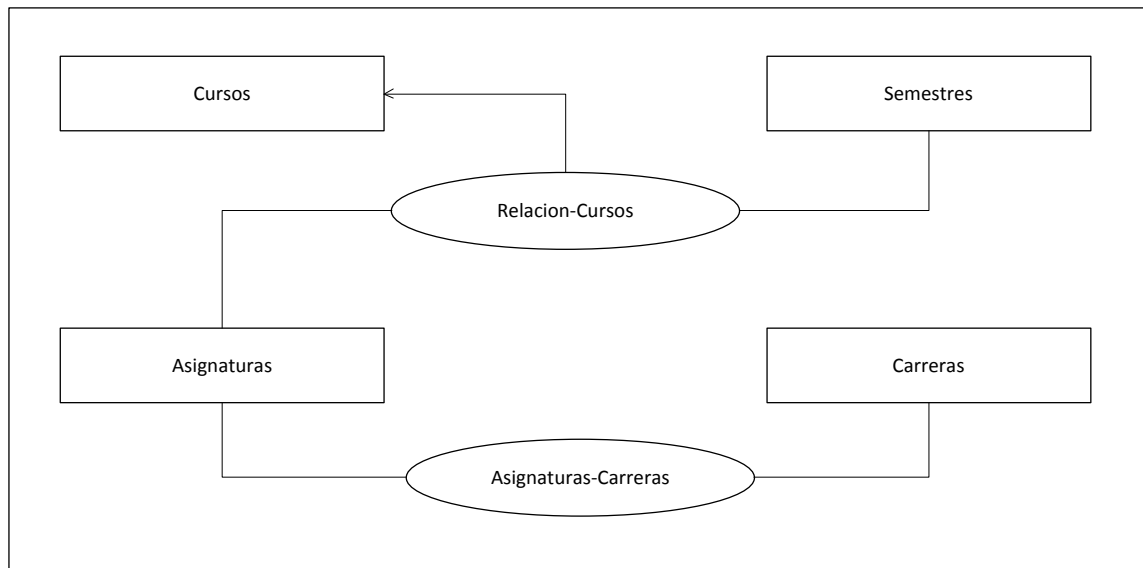


Figura 37 - Relaciones dimensionales del modelo conceptual, luego de aplicar la etapa1 de la fase V.

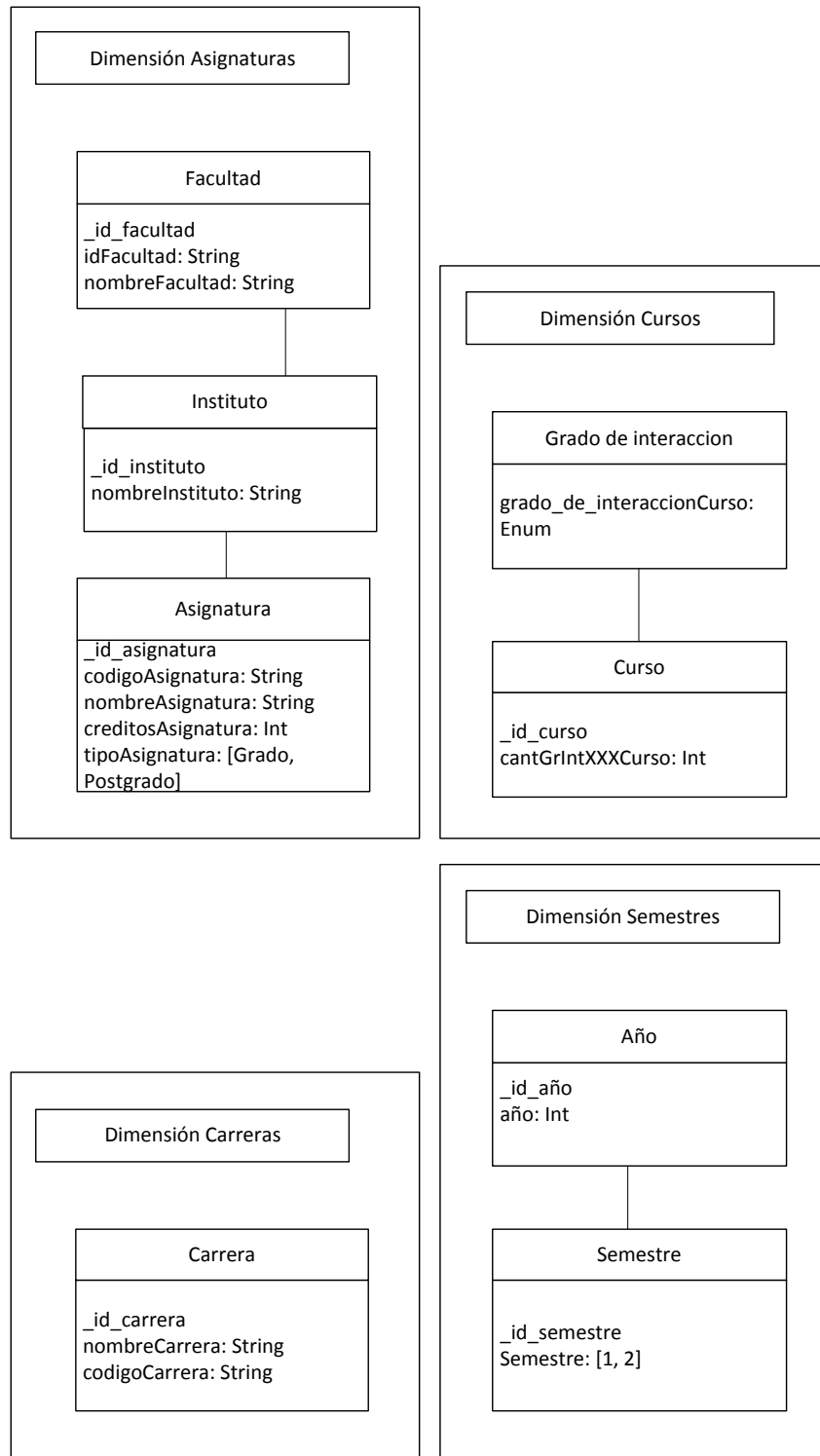


Figura 38 - Dimensiones del modelo conceptual, luego de aplicar la etapa1 de la fase V.

Se definió una clave por cada nivel, denominada id_CLASE. Además se quitó el nivel Facultad de la dimensión Carreras, ya que es la misma información que se obtiene a través de la dimensión Asignaturas.

Por último, se decidió retirar el nivel Recurso del modelo, ya que en sí no es necesario para satisfacer ningún requerimiento. Es necesario contar con él en el relevamiento, y también en el ods, ya que se pretende que la función para el cálculo del grado de interacción de un curso puede redefinirse en el futuro. Pero al momento de mostrarle la información al usuario del Data Warehouse, alcanza con los valores agregados que se obtienen de los recursos, estando éstos ya entre los atributos relevados para curso.

Se observa que no existe ninguna regla que contemple retirar niveles del modelo multidimensional. Todas las clases existentes en la ontología son mapeadas mediante el algoritmo para el modelo CMDM, independientemente de si el concepto es necesario para la solución a presentar en el Data Warehouse. De todos modos, esta es una situación muy especial, donde se requiere el concepto para eventuales redefiniciones de la media de grado de interacción, pero el nivel de granularidad especificado para el DW es superior al del concepto.

E2: Base de datos fuente

Se describe la base de datos canónica en la cual se aplicará el algoritmo.

Un diagrama del esquema se presenta en la figura 36.

Asignatura(codigoAsignatura, nombreAsignatura, creditosAsignatura, tipoAsignatura, facultadAsignatura, id, institutoAsignatura)

Representa el concepto de asignatura, tiene nombre y cantidad de créditos, está asociada a un instituto y a una facultad.

Carrera(codigoCarrera, nombreCarrera, facultadCarrera, id)

Representa el concepto de carrera, tiene un código y un nombre, y está asociada a una facultad.

Curso(semestreCurso, añoCurso, grado_de_interaccionCurso, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso, asignaturaCurso, id)

Representa el concepto de un curso dictado en Moodle, está asociado a una asignatura, a un semestre y a un año. Tiene un grado de interacción según los recursos que tenga.

Facultad(idFacultad, nombreAsignatura)

Representa el concepto de facultad, una facultad está dividida en institutos.

Instituto(nombreInstituto, facultadInstituto, id)

Representa el concepto de instituto, cada instituto dicta un grupo de asignaturas.

Recurso (idRecurso, tipoRecurso, grado_de_interaccionRecurso, cursoRecurso, id)

Representa el concepto de recurso dentro de un curso de Moodle, tiene un grado de interacción según el tipo de recurso que sea.

CarreraAsignatura (carrera, asignatura)

Establece a cuáles carreras pertenece cada asignatura.

Restricciones

Asignatura (facultadAsignatura) FOREIGN KEY Facultad (idFacultad)

Carrera(facultadCarrera) FOREIGN KEY Facultad (idFacultad)

Instituto(facultadInstituto) FOREIGN KEY Facultad (idFacultad)

Asignatura(institutoAsignatura) FOREIGN KEY Instituto(id)

Curso(asignaturaCurso) FOREIGN KEY Asignatura (id)

Recurso(cursoRecurso) FOREIGN KEY Curso(id)

CarreraAsignatura (carrera) FOREIGN KEY Carrera(id)

CarreraAsignatura (asignatura) FOREIGN KEY Asignatura(id)

Restricciones no consideradas

El grado de interacción de un curso se calcula en base a los grados de interacción de los recursos asociados a él.

Los atributos de Curso, tipo cantGrIntXXXCurso, se calcula como la cantidad de recursos del grado de interacción XXX del curso.

E3: Lineamientos

Para cada relación dimensional, definir los cubos que se van a construir y cuál será el detalle de información de cada uno. Además se deben definir las franjas de información para cada cubo (año actual, anteriores, clientes uruguayos, clientes extranjeros, etc.). Y se deben establecer las estrategias de diseño, lo que implica definir fragmentos en cada dimensión. Fragmentar la dimensión es agrupar niveles de la misma.

Materialización de Relaciones

En base a los requerimientos se decide materializar cuatro cubos, uno de la relación Relación-Cursos, uno de la relación Asignatura-Carrera, y dos de la composición de las dos relaciones del modelo, Relación-Cursos y Asignaturas-Carreras.

- CursoSemestreAsignatura: que presenta el grado de interacción de los cursos, con detalle de Semestre y Asignatura. Se presenta en la figura 39.
- CarreraAsignatura: representa cuáles asignaturas se dictan para cada carrera. Se presenta en la figura 40.

- GradoDeInteraccionInstituto: que presenta la cantidad de cursos para cada grado de interacción, con detalle de Año e Instituto. Se presenta en la figura 41.
- GradoDeInteraccionCarrera: que presenta la cantidad de cursos para cada grado de interacción, con detalle de Año y Carrera. Se presenta en la figura 42.

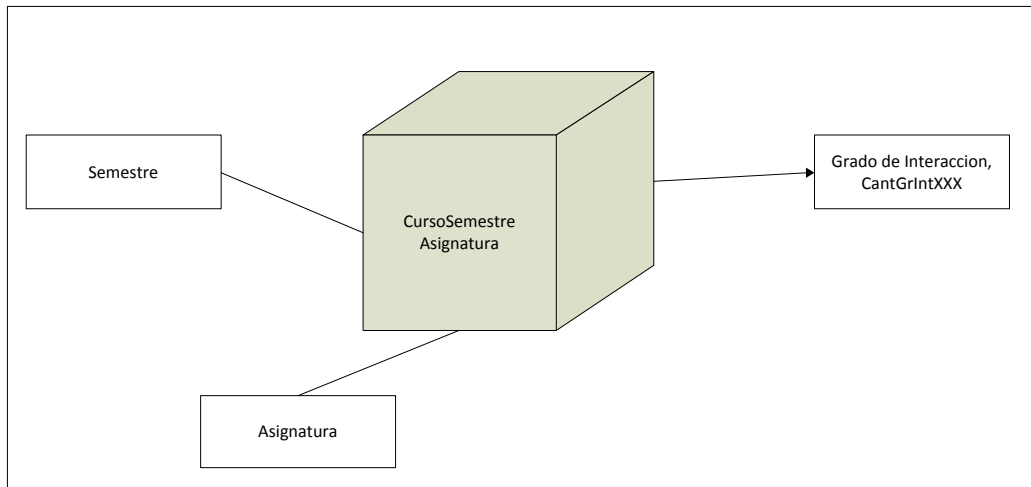


Figura 39 - Cubo CursoSemestreAsignatura.

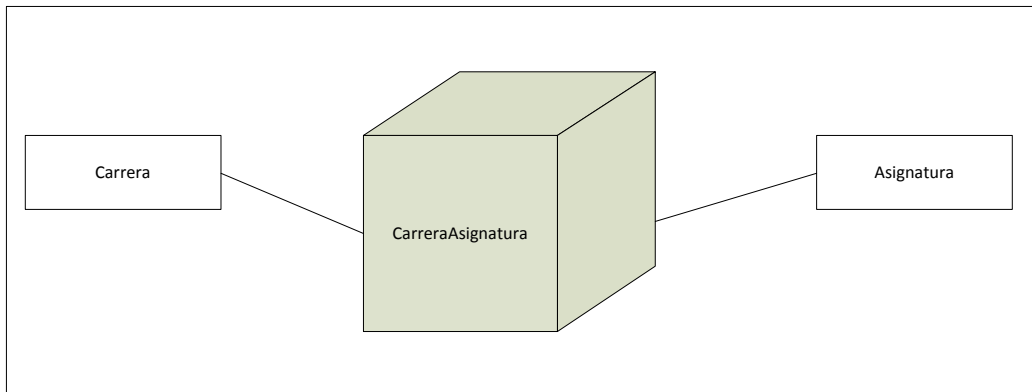


Figura 40 - Cubo CarreraAsignatura.

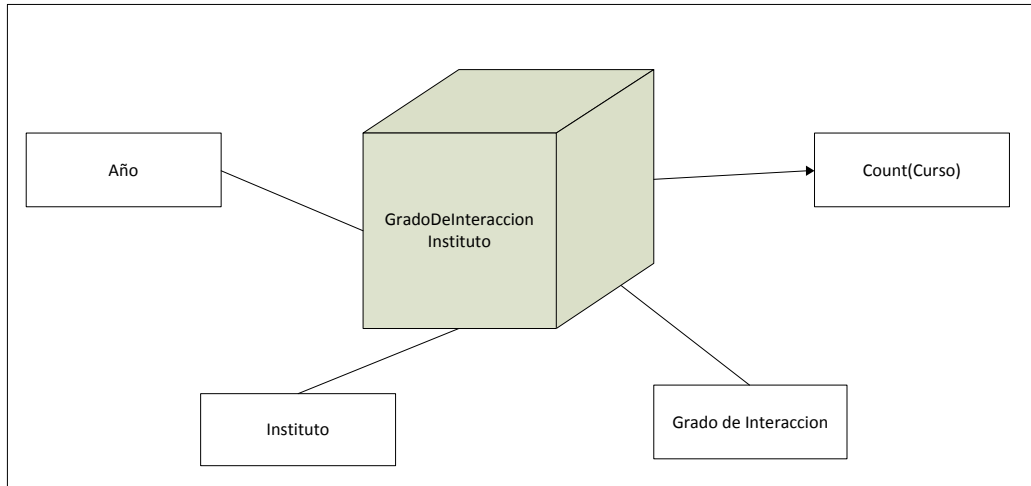


Figura 41 - Cubo GradoDeInteraccionInstituto.

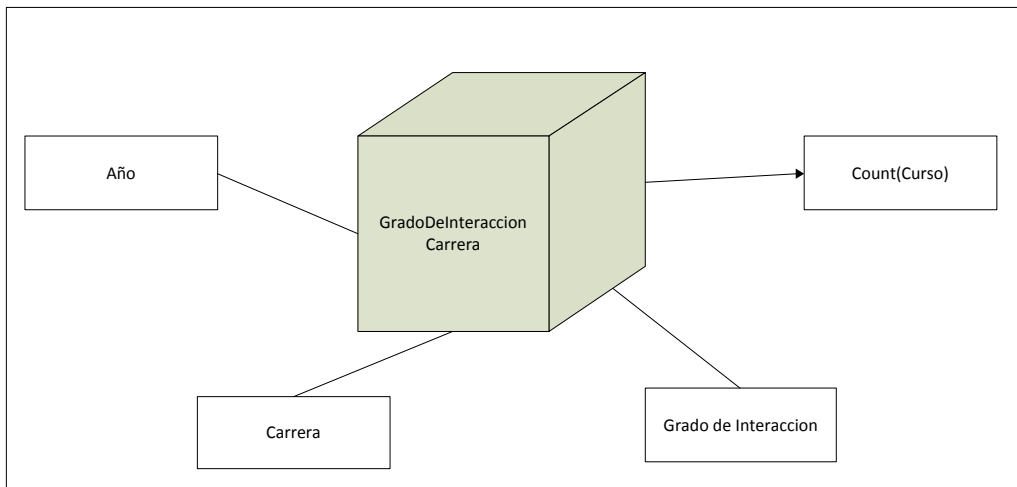


Figura 42 - Cubo GradoDeInteraccionCarrera.

Fragmentación de Cubos

Se decide no fragmentar ningún cubo.

Fragmentación de Dimensiones

Se decide no fragmentar ninguna dimensión, por lo tanto se tendrán las siguientes dimensiones:

- Asignaturas, desnormalizada
- Carreras, desnormalizada,
- Semestres, desnormalizada
- Cursos, no se utiliza ya que es la medida.

E4: Correspondencias.

Se encuentran las correspondencias entre la BDC y el modelo en CMDM. Si existen atributos calculados, se definen en este paso.

Mapeos de dimensiones

Se presentan las correspondencias entre el ODS y las dimensiones Asignaturas (figura 43), Carreras (figura 44) y Semestres (figura 45).

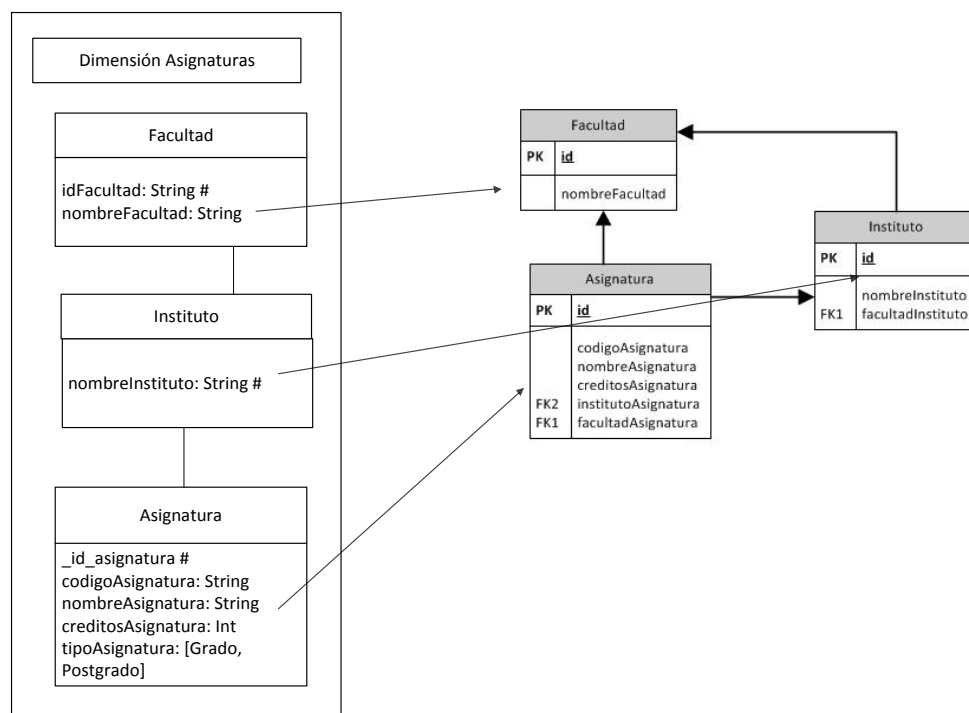


Figura 43 - Correspondencia entre la dimensión Asignaturas y el ODS.

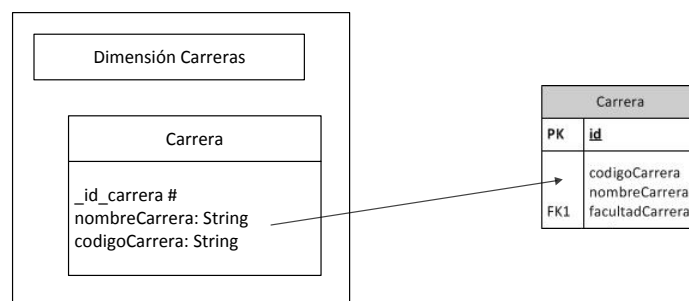


Figura 44 - Correspondencia entre la dimensión Carreras y el ODS.

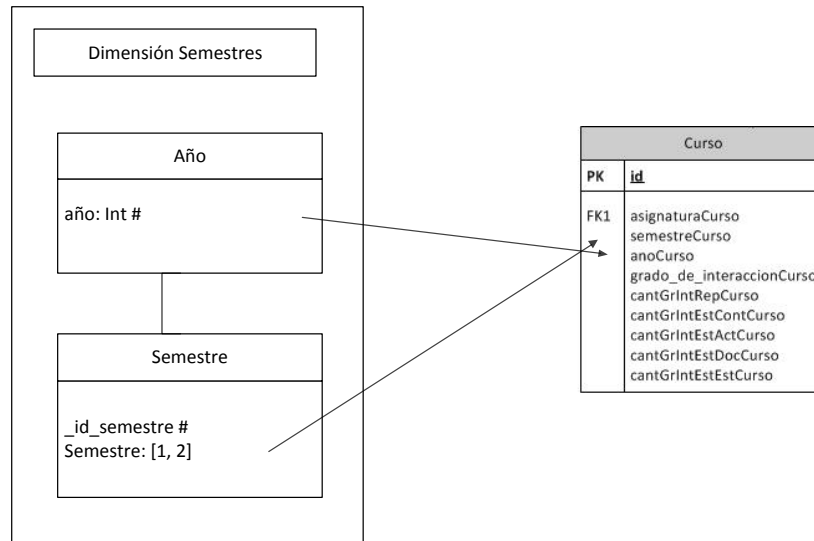


Figura 45 - Correspondencia entre la dimensión Semestres y el ODS.

Mapeos de cubos

Se presentan las correspondencias de los cubos con mapeo base. La correspondencia es entre el ODS y los cubos CarreraSemestreAsignatura (figura 46) y AsignaturaCarrera (figura 45).

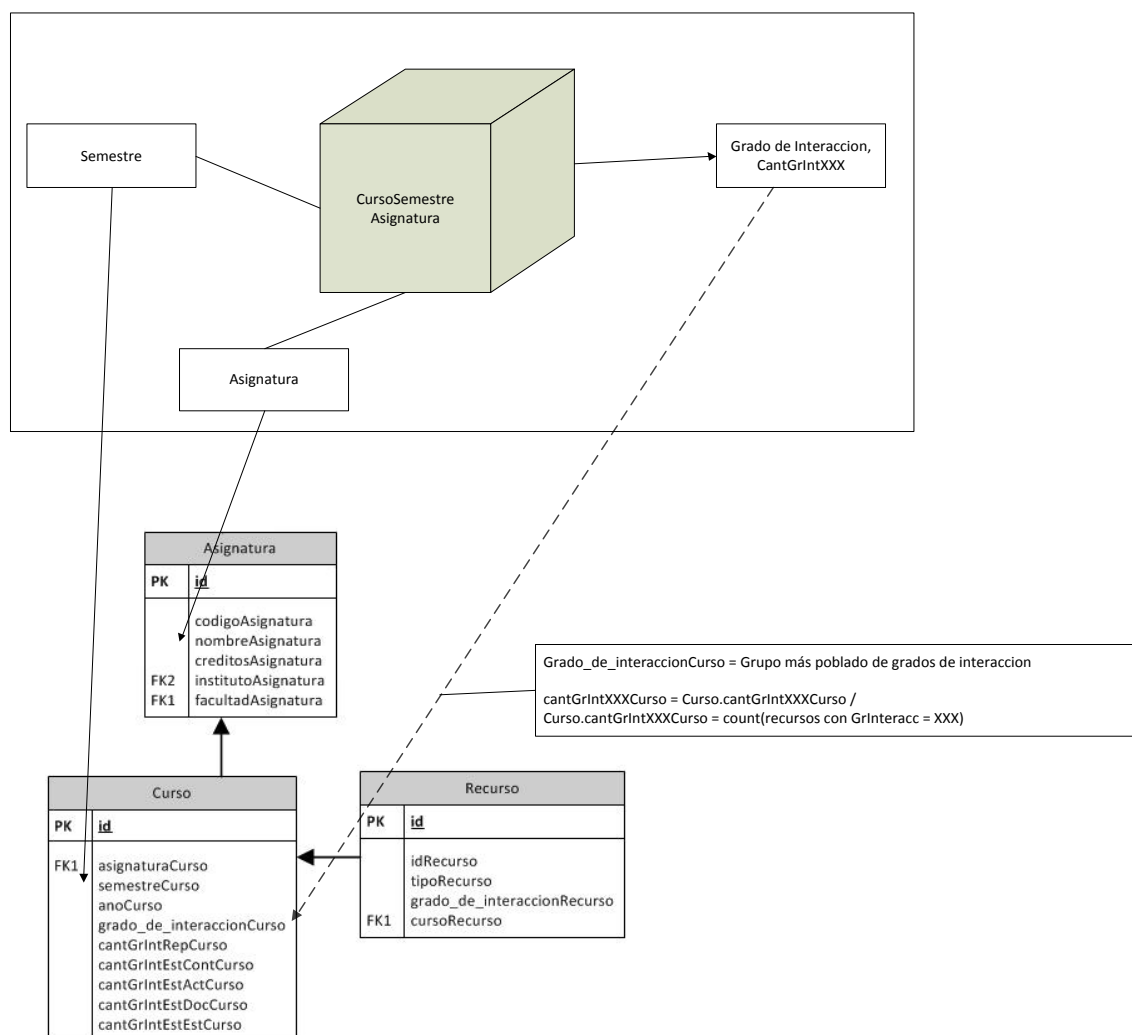


Figura 46 - Correspondencia entre el cubo CursoSemestreAsignatura y el ODS.

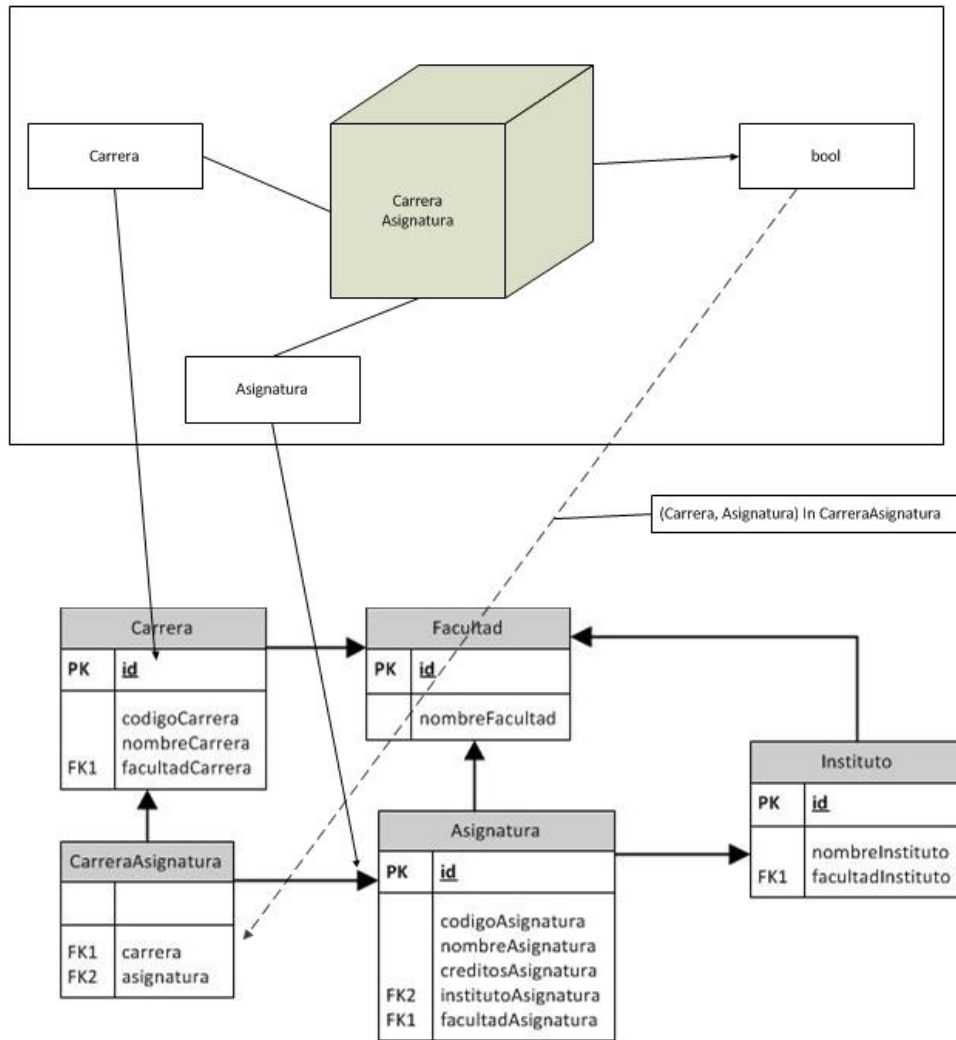


Figura 47 - Correspondencia entre el cubo CarreraAsignatura y el ODS.

Cubos con mapeo recursivo

Se presentan los cubos con mapeo recursivo, que son cubos que se basan en otros cubos. Primero en la figura 48 se presenta el cubo GradoDeInteraccionInstituto que se mapea al ODS y al cubo CursoSemestreAsignatura. En la figura 49 se presenta el cubo GradoDeInteraccionCarrera, que se mapea al ODS y a los cubos CursoSemestreAsignatura y AsignaturaCarrera.

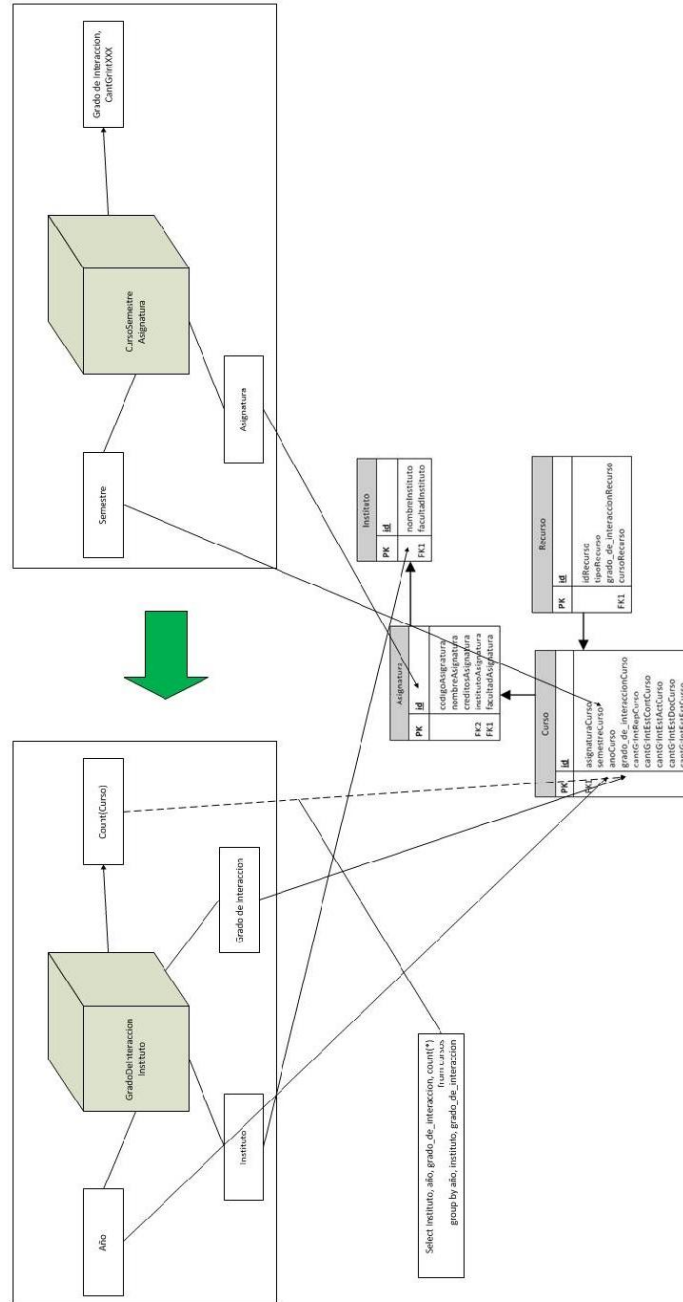


Figura 48 - Correspondencia entre el cubo GradoDeInteraccionInstituto por un lado, y el ODS y el cubo CursoSemestreAsignatura.

E5: Algoritmo

El algoritmo se divide en 4 partes con un total de 15 pasos:

I. Pasos 1- 6 para armar las tablas de dimensiones

Paso 1: Construcción de Esqueletos: Tomar cada fragmento y armar una única vista que lo mapee, definiendo así el esqueleto de cada dimensión con los atributos de mapeo directo.

Dimensión Asignaturas

```
CREATE VIEW DWAsignatura
AS
SELECT asig.id, asig.codigoAsignatura, asig.nombreAsignatura, asig.creditosAsignatura,
ins.id, ins.nombreInstituto, fac.idFacultad, fac.nombreFacultad
FROM
Asignatura asig, Instituto ins, Facultad fac
WHERE
asig.institutoAsignatura=ins.id
and ins.facultadInstituto=fac.idFacultad;
```

Dimensión Carreras

```
CREATE VIEW DWCarrera
AS
SELECT id, codigoCarrera, nombreCarrera
FROM Carrera;
```

Dimensión Semestres

```
CREATE VIEW DWSemestre
AS
SELECT semestreCurso, anoCurso
FROM Curso
GROUP BY semestreCurso, anoCurso; -- la clave es los dos atributos!
```

Paso 2: Nombres Atributos: Generar vistas si existen ítems del fragmento con correspondencia directa que tengan distinto nombre en la relación de la BDF de la que provienen (nombre(ítem) <> nombre(atributo)). Se hace el cambio de nombre directamente en la vista resultado del paso 1.

Dimensión Asignaturas

Renombrar identificadores

```
CREATE VIEW DWAsignatura
AS
SELECT asig.id as _id_asignatura, asig.codigoAsignatura, asig.nombreAsignatura,
asig.creditosAsignatura, ins.nombreInstituto, fac.idFacultad, fac.nombreFacultad
FROM
```



```
Asignatura asig, Instituto ins, Facultad fac
WHERE
asig.institutoAsignatura=ins.id
and ins.facultadInstituto=fac.idFacultad;
```

Dimensión Carreras

Renombrar identificadores

```
CREATE VIEW DWCarrera
AS
SELECT id as _id_carrera, codigoCarrera, nombreCarrera
FROM Carrera;
```

Dimensión Semestres

```
CREATE VIEW DWSemestre
AS
SELECT semestreCurso AS semestre, anoCurso AS ano
FROM Curso
GROUP BY semestreCurso, anoCurso;
```

Paso 3: Correspondencias: Generar atributos para ítems con mapeo calculado o externo. Para esto agrega cada nuevo atributo en la vista resultado del paso 2. Para seguir con una única vista, propone utilizar sub consultas en el FROM para resolver los atributos resultados de operaciones de agrupamiento.

No aplica.

Paso 4: Aplicar Filtros: Modificar las vistas para cubrir la necesidad de que las instancias deseadas para el DW cumplan ciertas restricciones ($f(Ci1, \dots, Cik)$), expresadas como condiciones sobre los datos de la BDF. Los filtros se agregan directamente en la vista resultado del paso 3. Este paso aplica únicamente en el tratamiento de cubos, no en el tratamiento de dimensiones

Este paso no es necesario en la metodología, ya que las restricciones ya se encuentran reflejadas en la base de datos ODS.

Paso 5: Eliminar Atributos: Aplicar esta regla si existen atributos en el resultado parcial que no están dentro de las correspondencias definidas, es decir, que no deben estar en el DW.

Este paso aplica únicamente en el tratamiento de cubos, no en el tratamiento de dimensiones.

Paso 6: Claves: Finalmente para cada fragmento cuya clave difiera de la clave en el DW resultado, definir la clave necesaria.

Dimensión Semestres

Todavía no tiene su clave, porque la misma no se mapea desde el ods. Por lo tanto, para DWSemestre se creará una tabla con un valor autoincremental que actúe como clave.

```
CREATE TABLE DWSemestre (  
    _id_semestre INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    semestre INT UNSIGNED NOT NULL,  
    ano INT UNSIGNED NOT NULL  
);  
INSERT INTO DWSemestre(semestre, ano)  
SELECT semestreCurso AS semestre, anoCurso AS ano  
FROM Curso  
GROUP BY semestreCurso, anoCurso  
ORDER BY anocurso, semestrecurso;
```

El esquema resultado es el siguiente:

```
DWAsignatura(id_asignatura, codigoAsignatura, nombreAsignatura, creditosAsignatura,  
nombreInstituto, idFacultad, nombreFacultad)  
DWCarrera(id_carrera, codigoCarrera, nombreCarrera)  
DWSemestre(id_semestre, semestre, ano)
```

II. Pasos 7-12 para armar los cubos con mapeo simple

Paso 7: Construcción de Esqueletos: Tomar cada fragmento y armar una única vista que lo mapee, definiendo así el esqueleto de cada dimensión con los atributos de mapeo directo.

Cubo CursoSemestreAsignatura

```
CREATE VIEW DWCursoSemestreAsignatura  
AS  
SELECT cur.asignaturacurso, dwsem._id_semestre, cur.grado_de_interaccionCurso,  
cur.cantGrIntRepCurso, cur.cantGrIntEstContCurso, cur.cantGrIntEstActCurso,  
cur.cantGrIntEstDocCurso, cur.cantGrIntEstEstCurso  
  
FROM curso cur, dwsemestre dwsem  
  
WHERE  
dwsem.semestre=cur.semestrecurso  
and dwsem.ano=cur.anocurso;
```

Cubo CarreraAsignatura

```
CREATE VIEW DWCarreraAsignatura  
AS  
SELECT carrera, asignatura
```

FROM carreraasignatura;

Cubo GradoDeInteraccionInstituto

Este cubo tiene mapeo recursivo, se construye con las reglas 13 y 14.

Cubo GradoDeInteraccionCarrera

Este cubo tiene mapeo recursivo, se construye con las reglas 13 y 14.

Paso 8: Nombres Atributos: Generar vistas si existen ítems del fragmento con correspondencia directa que tengan distinto nombre en la relación de la BDF de la que provienen (nombre(ítem) <> nombre(atributo)). Se hace el cambio de nombre directamente en la vista resultado del paso 1.

Cubo CursoSemestreAsignatura

```
CREATE VIEW DWCursoSemestreAsignatura  
AS
```

```
SELECT cur.asignaturacurso as Asignatura,  
dwsem._id_semestre as Semestre,  
cur.grado_de_interaccionCurso as Grado_de_interaccion,  
cur.cantGrIntRepCurso, cur.cantGrIntEstContCurso, cur.cantGrIntEstActCurso,  
cur.cantGrIntEstDocCurso, cur.cantGrIntEstEstCurso
```

```
FROM curso cur, dwsemestre dwsem
```

```
WHERE
```

```
dwsem.semestre=cur.semestrecurso  
and dwsem.ano=cur.anocurso;
```

Paso 9: Correspondencias: Generar atributos para ítems con mapeo calculado o externo. Para esto agrega cada nuevo atributo en la vista resultado del paso 2. Para seguir con una única vista, propone utilizar sub consultas en el FROM para resolver los atributos resultados de operaciones de agrupamiento.

No aplica.

Paso 10: Aplicar Filtros: Modificar las vistas para cubrir la necesidad de que las instancias deseadas para el DW cumplan ciertas restricciones (f(Ci1, ..., Cik)), expresadas como condiciones sobre los datos de la BDF. Los filtros se agregan directamente en la vista resultado del paso 3.

Este paso aplica únicamente en el tratamiento de cubos, no en el tratamiento de dimensiones

No aplica.

Paso 11: Eliminar Atributos: Aplicar esta regla si existen atributos en el resultado parcial que no están dentro de las correspondencias definidas, es decir, que no deben estar en el DW.
Este paso aplica únicamente en el tratamiento de cubos, no en el tratamiento de dimensiones.

No aplica.

Paso 12: Claves: Finalmente para cada fragmento cuya clave difiera de la clave en el DW resultado, definir la clave necesaria.

El esquema resultante es:

DWCursoSemestreAsignatura(Asignatura, Semestre, Grado_de_interaccion,
cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso,
cantGrIntEstEstCurso)
DWCarreraAsignatura(carrera, asignatura)

III. Pasos 13-14 para armar los cubos con mapeo recursivo

Se analizan los cubos con mapeo recursivo, o sea aquellos cubos que se basan en otros cubos.

En estos casos, se buscará reducir el detalle de una dimensión, eliminando la misma (Drill-up total) o bien subiendo por niveles de la jerarquía dentro de una dimensión (Drill-up parcial).

Paso 13: Fragmentos Ficticios: En una primera (paso 13) se construyen fragmentos de dimensión denominados Fragmentos Ficticios, dado que sólo se utilizan para construir los cubos recursivos.

Para esto, se ejecutan nuevamente los pasos 1 al 6 del algoritmo por lo cual las vistas que se construyen son equivalentes a las ya vistas para estos pasos.

```
Cubo DWGradoDeInteraccionInstituto
CREATE VIEW DWAsignaturaInstituto
AS
SELECT asig.id as _id_asignatura, ins.nombreInstituto
FROM
Asignatura asig, Instituto ins, Facultad fac
WHERE
asig.institutoAsignatura=ins.id
and ins.facultadInstituto=fac.idFacultad;
```

Paso 14: Se utilizan estos fragmentos para efectuar las operaciones de Drill-Up sobre el cubo base y así obtener el cubo recursivo. Estos fragmentos ficticios se construyen sólo para las dimensiones en que sea necesario hacer un Drill-Up Parcial, donde esto significa no eliminar la dimensión en el cubo sino ir a un nivel superior.

Dichos fragmentos también están especialmente diseñados para contener sólo los niveles origen y el de destino del Drill-Up.

En este paso se aplican las operaciones de Drill-Up al cubo base para obtener el cubo recursivo, con la particularidad de que las operaciones de Drill-Up se hacen en una dimensión a la vez.

Cubo DWGradoDeInteraccionInstituto

```
CREATE VIEW DWGradoDeInteraccionInstituto
```

```
AS
```

```
SELECT sem.ano, asig.nombreinstituto, cur.grado_de_interaccion, count(*) as cantidad
```

```
FROM dwcursosemestreasignatura cur, dwsemestre sem, dwasignatura asig
```

```
WHERE cur.semestre=sem._id_semestre
```

```
and cur.asignatura=asig._id_asignatura
```

```
GROUP BY sem.ano, asig.nombreinstituto, cur.grado_de_interaccion;
```

El esquema resultante es:

DWGradoDeInteraccionInstituto(año, nombreInstituto, grado de interaccion, cantidad)

Luego de aplicados los pasos 13 y 14 para armar cubos recursivos, no fue posible armar el cuarto cubo que se había propuesto: el cubo GradoDeInteraccionCarrera. Esto se debe a que para construirlo es no se realiza solamente un drill-up por una jerarquía, sino que también es necesario aplicar una regla del estilo de New dimension crossing, como la define Marotta (23) en el trabajo en el que se basa esta metodología.

Marotta nombra la regla como "Primitive 14. New dimension crossing" y la define de la siguiente manera: "Muchas veces se requiere materializar un cruce de dimensiones en una nueva relación. Esto se puede realizar a través de un join de relaciones. Por ejemplo, si existe una relación de medida en la que participa la relación producto junto con otra, y existe otra relación donde el proveedor se determina por el producto. La relación proveedor se puede agregar a la relación de medida retirando la dimensión producto, resultando en un cruce entre proveedor y las otras dimensiones existentes en la relación de medida" (23).

En este caso es necesario realizar una operación análoga, ya que se pretende vincular los cursos a las carreras a las que pertenecen. En este caso, asignatura sale del cubo y es remplazado por carrera, tal como lo hacen producto y proveedor respectivamente en el ejemplo.

La metodología no considera casos donde sea necesario aplicar una regla de este tipo.

De todas formas, se materializa el cubo a pesar de no estar contemplado en la metodología. Aplicando una idea análoga, se materializa el cubo.

Cubo DWGradoDeInteraccionCarrera

```
CREATE VIEW DWGradoDeInteraccionCarrera
AS
SELECT sem.ano, carasig.carrera, cur.grado_de_interaccion, count(*) as cantidad
FROM dwcursosemestreasignatura cur, dwsemestre sem, dwasignatura asig,
dwcarreraasignatura carasig
WHERE cur.semestre=sem._id_semestre
and cur.asignatura=asig._id_asignatura
and carasig.asignatura=asig._id_asignatura
GROUP BY sem.ano, carasig.carrera, cur.grado_de_interaccion;
```

El esquema resultante es:

DWGradoDeInteraccionCarrera(año, carrera, grado de interaccion, cantidad)

IV. Paso 15 para armar las franjas de cubos

No aplica.

Esquema final resultante

Dimensiones

DWAsignatura(id asignatura, codigoAsignatura, nombreAsignatura, creditosAsignatura, nombreInstituto, idFacultad, nombreFacultad)

DWCarrera(id carrera, codigoCarrera, nombreCarrera)

DWSemestre(id semestre, semestre, ano)

Cubos

DWCursoSemestreAsignatura(Asignatura, Semestre, Grado_de_interaccion, cantGrIntRepCurso, cantGrIntEstContCurso, cantGrIntEstActCurso, cantGrIntEstDocCurso, cantGrIntEstEstCurso)

DWCarreraAsignatura(carrera, asignatura)

DWGradoDeInteraccionInstituto(año, nombreInstituto, grado de interaccion, cantidad)

DWGradoDeInteraccionCarrera(año, carrera, grado de interaccion, cantidad)

Matriz de Trazabilidad

Una vez finalizada la ejecución de este algoritmo, se está en condiciones de crear la Matriz de Trazabilidad completa. Para esto sólo basta tomar la Matriz de requerimientos de la fase II y agregar una columna llamada Cubos a la derecha de los requerimientos. El resultado se presenta en la tabla 14.

Concepto	RQ1	RQ2	RQ3	Cubo Curso Semestre Asignatura	Cubo Carrera Asignatura	Cubo Grado De Interacción Instituto	Cubo Grado De Interacción Carrera
Clases							
Asignatura	√	√	√	√	√	√	√
Carrera			√		√		√
Curso	√	√	√	√		√	√
Facultad			√		√	√	√
Instituto			√			√	
Recurso	√	√	√	√			
Relaciones							
asignaturaCurso	√	√	√				
asignaturasCarrera			√				
asignaturasInstituto			√				
carrerasFacultad			√				
facultadAsignatura			√				
facultadCarrera			√				
facultadInstituto			√				
institutoAsignatura			√				
institutosFacultad			√				
recursosCurso	√	√	√				

Tabla 14 - Matriz de Trazabilidad, vinculando los requerimientos con los conceptos y los cubos materializados.

Conclusiones del caso de estudio

El trabajo realizado consistió en aplicar la metodología del proyecto Visión Global de Construcción de Data Warehouse para satisfacer un requerimiento de la Unidad de Enseñanza, con el objetivo de evaluar la metodología.

La metodología del proyecto Visión Global de Construcción de Data Warehouse, surge como la fusión de otros cuatro proyectos: primero, el diseño de un lenguaje de modelado multidimensional CMDM hecho por Carpani. Segundo, la metodología MD4DW que propone cómo llegar al diseño multidimensional en CMDM de una realidad partiendo de un análisis de los requerimientos. Tercero, el algoritmo de Peralta para asistir al diseño lógico del DW de forma semiautomática a partir del modelo multidimensional hecho en CMDM. Y por último, en base a ésta última metodología, el algoritmo de Larrañaga para la carga del DW desde una base de datos fuente.

VGPCDW encadena los proyectos mencionados, revisándolos, corrigiéndolos y extendiéndolos, para lograr una metodología completa. Como resultado VGPCDW propone cinco fases: I) Comprensión de la realidad, II) Modelado de Dominio, III) Modelado Conceptual Multidimensional, IV) Construcción del ODS y V) Generación del diseño lógico y la carga de datos.

Aplicando la metodología tal cual se expone en el proyecto VGPCDW, y revisando también los proyectos anteriores para ahondar en puntos en los que la documentación de VGPCDW era escasa, no se pudo construir con éxito una solución para satisfacer la totalidad de los requerimientos.

En primera instancia se aplicaron las tres primeras fases de VGPCDW (correspondientes a la versión revisada del algoritmo de MD4DW), las primeras dos sin encontrar dificultades, sí encontrándolas en la tercera.

Las reglas de la Fase III evidencian que MD4DW confunde el enfoque multidimensional de CMDM con el enfoque de estrella. En este sentido intenta mapear los niveles definidos a un concepto no existente en CMDM como “eje central de una relación multidimensional”. También pierde el concepto de las medidas de la relación dimensional. El resultado de la aplicación de las reglas de la fase III es un modelo que es sintácticamente incorrecto.

Si se pasa por alto las confusiones que lleva en su interior MD4DW, y se identifica al “eje central de la relación” como la medida de la misma, se obtiene una metodología que no permite que las medidas, tal cual las dimensiones, puedan también tener jerarquías, como está definido en CMDM por la dimensionalidad genérica.

El modelo conceptual obtenido no expresa la posibilidad de realizar un roll-up a través de la jerarquía de la dimensión medida, siendo esto parte de los requerimientos del caso de estudio.

Además de los problemas expuestos, aplicando las reglas como están definidas, no se consigue modelar relaciones que tengan cardinalidad N:N. Estas relaciones, por cómo está definido CMDM, representan cruzamientos que deben modelarse en relaciones dimensionales. MD4DW no incluye reglas para abordar este tipo de problemas y construir este tipo de relaciones dimensionales.

Por lo tanto, aplicando las tres primeras fases del VGPCDW se llega (si pasamos por alto los problemas antes expuestos) a un modelo multidimensional que no consigue satisfacer los requerimientos.

Para seguir evaluando la metodología se completó el modelo para que atienda a todos los requerimientos. Se definió una jerarquía para la dimensión medida, y se agregó una segunda relación dimensional que vincule las dimensiones Carreras y Asignaturas.

A continuación se aplicó la fase IV con éxito para la construcción de una base de datos ODS. De todas formas, la metodología no contempla la carga del ODS, lo que se identifica como una debilidad.

Por último se aplicó la fase V para generar el diseño lógico y la carga de los datos, en base al ODS y al modelo multidimensional corregido. Esta fase se pudo aplicar con éxito solamente para una parte del modelo.

Esto se debe a que la metodología no cubre la construcción de cubos en los que sea necesario vincular varias relaciones dimensionales para construirlo. Únicamente indica cómo construir cubos asociados a una sola relación dimensional, ya sea con el menor nivel de granularidad, o haciendo agregaciones subiendo por las dimensiones (aplicando reglas de *hierarchy-drill -up* o *total-drill-up*).

Esta fase carece de reglas que permitan aplicar operaciones primitivas del estilo de *New-dimension-crossing* (como la define Marotta en su tesis, trabajo en que se basa el proyecto de Peralta), cosa necesaria para satisfacer los requerimientos.

Como conclusión, la metodología Visión Global de Construcción de Data Warehouse tiene en su interior algunos problemas conceptuales debido a la incorrecta interpretación de CMDM en MD4DW. Pasando por alto estos inconvenientes, la aplicación de la metodología dio como resultado una solución que permitió satisfacer sólo parcialmente los requerimientos.

Anexo II - Diagrama de arquitectura de la aplicación

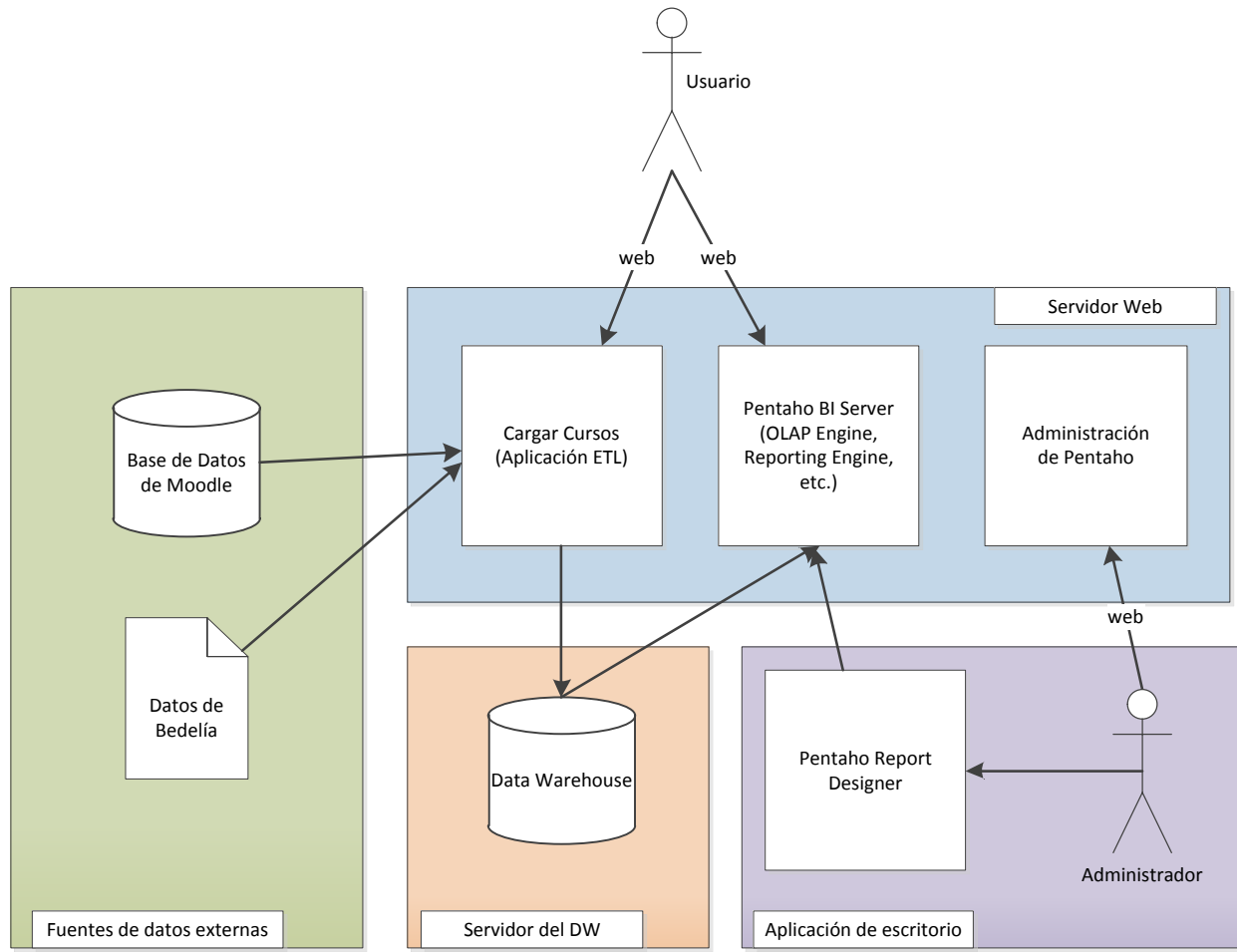


Figura 50 - Arquitectura de la aplicación.

