

Universidad de las Ciencias Informáticas

Facultad 3



Título: Herramienta para la edición de ontologías
del modelo multidimensional

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Leonardo Ricardo García

Tutor(es): M.Sc. Julio César Díaz Vera

Co-tutor(es): Ing. Ricardo Pablo Ávila Alfaro

Ing. Andy Fernández Garabote

Julio, 2014

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Leonardo Ricardo García

Firma del Autor

M.Sc. Julio César Díaz Vera

Firma del Tutor

DATOS DE CONTACTO

Julio César Díaz Vera, graduado de Ingeniería en Telecomunicaciones en la Universidad Martha Abreu ubicada en Villa Clara, Cuba. Máster en Gestión de Proyectos, grado científico alcanzado en la Universidad de la Ciencias Informáticas, La Habana, Cuba. Posee varias publicaciones nacionales e internacionales referentes al tema abordado en esta investigación, alcanzando una experiencia de más de 10 años.

Correo electrónico: jcdiaz@uci.cu

AGRADECIMIENTOS

Puede ser que desde fuera parezca fácil solo agradecer a aquellos que de una forma u otra han puesto su granito de arena para que en este día pueda llegar al punto máximo de estudios. No existen palabras capaces de describir o retribuir dichas acciones. Espero no quede nadie a quien agradecer, pero si eso llegara a suceder ruego que se sienta agradecido, esa es mi intención.

Quisiera comenzar con la mujer que me trajo a este mundo, a quien cariñosamente le digo “bin” sin saber por qué, que ha sabido ser madre, padre, amiga, consejera, etc. Ella fue y ha sido capaz de dejar su vida a un lado para primeramente dármele todo a mí y luego a mi hermanito Gerald. No encuentro manera de decirle que este día es más suyo que mío y que ahora seré yo quien vele por ella. A Gerald que no me vea como un reto que encuentre sus propios pasos y que lo quiero mucho. A mis abuelitos, “papi Alfredo” y “mamá Santa”, aunque para mí sólo papi y mami, que más que abuelos fueron padres que me enseñaron mis primeras lecciones de la vida junto a mi mamá. Ellos aguantaron mis etapas de niño, adolescente y joven, con amor y muchos consejos. Por fin ya no tendrán que llorar más porque el niño se les va para la UCI. Le agradezco a toda mi familia que de una manera u otra siempre me ha apoyado para poder cumplir este sueño.

Gracias Lisy por existir, brindarme tu amor incondicional y dejarme ser partícipe de tu familia, a la que orgullosamente puedo decir que pertenezco. Tu apoyo ha sido clave y le has dado sentido a mi vida. Si de sacrificarse por lo que uno desea se trata, tú sin dudas, eres la mejor, de ti aprendí a querer más mi carrera. Mariela, para usted tampoco tengo palabras de gratitud, prácticamente me ha hecho sentir su hijo durante todo el noviazgo. En las buenas y en las malas han estado ahí para apoyarme, cuidar de mí y de mi familia, tanto en lo personal como en lo profesional. A toda la familia, en especial a ustedes, gracias, me han hecho sentir el hombre más afortunado del mundo.

A mi papá Gaspar por darme apoyo y sentirse orgulloso de su hijo. A Yi, que a veces dice que no la quiero pero ella sabe que siempre tendrá un lugarcito en mi corazón, al igual que Mime. Por supuesto, a mis hermanos paternos, Yoisel, Zeni y Geisita, que de una forma u otra han estado ahí para mí. Le pido a mis familiares paternos que no crean que me olvido de ellos, siempre los tengo en mente a pesar de que la lejanía suponga un obstáculo.

A Gerar que como un padre me enseñó la mayoría de las cosas que forjaron mi carácter y se convirtió en un ejemplo a seguir. A los amigos de mi provincia que a pesar de no poder estar aquí hoy, siempre preguntaban y se preocupaban.

“Camarada”, como cariñosamente te llamamos y nos llamas, gracias por ser nuestro profesor, amigo, tutor, guía, en tantos sentidos que nunca pensamos. Mi primera impresión de ti es que eras un “pesa’o”, tenía razón, pero con el tiempo me di cuenta que de esa pesadez surgiría una gran amistad, a pesar de que me gustaba desafiarte, y te la desquitabas por ello, claro. Sinceramente te agradezco que hayas invertido tu tiempo y paciencia en esta tesis. Siempre recordaré estas palabras tuyas: “la universidad sólo te enseña a aprender, lo demás depende de ti”. Andy y Ricardo, no me olvido de ustedes, que en la ausencia de Julio asumieron y nos ayudaron a afrontar las dificultades.

Gracias Leandro, Rayner, Guille, Yoalvy, Nilson y Daniel, compañeros de apartamento que me soportaron y soporté. De ustedes aprendí muchas cosas y odié otras, pero al final del día siempre anduvimos juntos y prácticamente se convirtieron en hermanos para mí, aún después de irse de la universidad como es el caso de Leo y el Daniko. Espero no se les olvide la promesa del 2018. Rayner y Guille, prácticamente fuimos compañeros de tesis, nos reímos y preocupamos mucho, gracias por esos momentos. Yoalvy por mucho que lo intenté nunca pude igualar tus horarios de sueño y despreocupación. Nilson, a pesar de ser medio peleón siempre ayudaste en lo que podías sin peros, tumbaré mangos en tu nombre.

Juan y Devorah, les encantó siempre molestarme, gracias por hacerme reír cuando no tenía ganas. Les agradezco a todos los integrantes del grupo 3502 que de alguna forma me enseñaron que ser y que no, incluso a aquellos que ya no están en la universidad. Al grupito que me ayudaba a pasar lista en la “recre”, les pido que no cambien.

A los amigos de la dieta (alumnos, profesores y trabajadores) que se cuiden y muchas gracias por esos ratos de risas y “chucho”, por los cuales valía la pena caminar tanto.

A los miembros del tribunal les ofrezco mi gratitud, pues gracias a sus críticas constructivas este trabajo tuvo lugar con la calidad esperada. Gracias a todos los profesores de los cuales tuve el honor de ser discípulo, por darle forma al conocimiento que pude adquirir hasta el momento. De la misma manera quiero agradecer a todos aquellos amigos que hice en la universidad, por

enseñarme el valor de ese recurso para la vida. Para todos aquellos que de una forma u otra han influido en mi desarrollo como profesional, MUCHAS GRACIAS.

DEDICATORIA

A mi mamá y abuelos que siempre quisieron verme así...

RESUMEN

El minado de reglas de asociación es una de las técnicas de minería de datos más estudiadas y difundidas. En los últimos años se ha extendido su incorporación a herramientas de inteligencia de negocio que utilizan almacenes de datos como fuentes de datos para aprovechar las ventajas de la misma. Sin embargo la simbiosis entre los almacenes de datos y las reglas de asociación exagera uno de sus problemas históricos, la generación de modelos que contienen demasiadas reglas. Con vista a reducir los modelos, una de las variantes que ha arrojado buenos resultados, es la utilización de conocimiento previo para reducir la cantidad de reglas en los modelos finales. Las ontologías constituyen el mecanismo estándar para representar el conocimiento existente sobre un dominio particular. Pero las mismas tienen una alta complejidad técnica que dificulta que puedan ser utilizadas por especialistas del dominio para representar su conocimiento. En este trabajo se propone una herramienta, que es capaz de servir de interfaz entre las aplicaciones que utilizan la información de dominio expresada como ontologías del modelo multidimensional y los especialistas del negocio que necesitan representar su conocimiento en estas ontologías. Se construye una herramienta computacional que interpreta el código OWL de la ontología del modelo multidimensional y expresa también, de manera gráfica, los elementos que son relevantes para el especialista del dominio al que da la oportunidad de enriquecer la modelación del dominio incorporándole su conocimiento sobre las relaciones que existen entre los componentes.

PALABRAS CLAVE

Conocimiento previo, minería de datos, ontología, reglas de asociación, representación visual.

TABLA DE CONTENIDOS

Introducción.....	10
1 Capítulo 1: Fundamentación teórica	18
1.1 Almacenes de datos y OLAP.....	18
1.2 Minería de datos (MD).....	20
1.2.1 Reglas de asociación	21
1.3 Ontología	23
1.3.1 Lenguajes para la representación de ontologías	25
1.4 Esquemas conceptuales	29
1.4.1 Distintas aproximaciones para la representación de esquemas conceptuales	30
1.4.2 Tipos de esquemas conceptuales	32
1.4.3 Representación de esquemas conceptuales usando ontologías	34
1.5 Tecnologías y herramientas empleadas para el desarrollo.....	35
1.5.1 Marcos de trabajo	35
1.5.2 Lenguajes de programación	37
1.5.3 Otras herramientas	38
1.6 Conclusiones parciales	39
2 Capítulo 2: Herramienta para la edición de ontologías del modelo multidimensional	40
2.1 Ontología del modelo multidimensional	40
2.2 Metodología de desarrollo	41
2.3 Arquitectura.....	44
2.4 Usuarios y funcionalidades.....	45
2.4.1 Subsistema de carga de la ontología	46
2.4.2 Subsistema de gestión de la ontología.....	48
2.4.3 Subsistema de guardado	53
2.5 Conclusiones parciales	55
3 Capítulo 3: Validación de la solución	57
3.1 Métodos de validación.....	57
3.2 Recursos computacionales utilizados para la demostración.....	58
3.3 Conjuntos de datos utilizados y resultados obtenidos	58
3.4 Conclusiones parciales	64
4 Conclusiones	65
5 Recomendaciones.....	66

6 Referencias67

INTRODUCCIÓN

Los almacenes de datos (AD) se han establecido como una tecnología clave en las organizaciones, generando gran interés en su uso como fuente de datos en el proceso de minería de datos. En (Inmon, 2002) se define al almacén de datos como *"una colección de datos orientados a temas, integrados, no volátiles y variables en el tiempo, organizados para dar apoyo al proceso de toma de decisiones"*. Cada una de estas características puede interpretarse de la siguiente manera:

- **Orientado a temas:** los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- **Variante en el tiempo:** los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.
- **No volátil:** la información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura, y se mantiene para futuras consultas.
- **Integrado:** la base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.

Los AD son diseñados para explotar las ventajas de la tecnología OLAP (*On-Line Analytical Processing*) y lograr un mayor rendimiento en las consultas analíticas. En contraposición con lo que ocurre en los sistemas transaccionales u OLTP (*On-Line Transaction Processing*) que están optimizados para operaciones transaccionales que gestionan el día a día de la empresa, centrándose en su actividad fundamental y optimizando las operaciones que modifican el estado de los datos.

Por otro lado la minería de datos (MD) es una familia de métodos computacionales que tiene como objetivo recolectar y analizar datos relacionados a un sistema de interés con el propósito de ganar mejor entendimiento del mismo (Triantaphyllou, 2010). La comunidad científica acepta de manera natural la clasificación de minería de datos en función de las tareas que realiza como (Han, 1997):

- **Caracterización:** generaliza un conjunto de datos relevantes para la tarea en un cubo de datos generalizado que puede ser utilizado para la extracción de los distintos tipos de reglas o ser vistos en múltiples niveles de abstracción desde diferentes ángulos. En particular, se deriva un conjunto de reglas características que resumen los aspectos generales de un conjunto de datos especificados por el usuario (clase destino). Por ejemplo, los síntomas de una enfermedad específica se pueden resumir en una regla característica.
- **Comparación:** explota un conjunto de reglas discriminantes que resumen las características que distinguen a la clase que se examina (la clase destino) de otras clases (llamadas clases contraste). Por ejemplo, para distinguir una enfermedad de las demás, una regla discriminante resume los síntomas que discriminan esta enfermedad de los demás.
- **Asociación:** descubre un conjunto de reglas de asociación (en la forma " $A_1 \wedge \dots \wedge A_i \rightarrow B_1 \wedge \dots \wedge B_j$ ") en varios niveles de abstracción de conjuntos pertinentes de los datos en una base de datos. Por ejemplo, se puede descubrir una serie de síntomas que a menudo ocurren junto con ciertos tipos de enfermedades y estudiar más a fondo las razones detrás de ellos.
- **Clasificación:** analiza un conjunto de datos de entrenamiento, (es decir, un conjunto de objetos cuya etiqueta de clase se conoce) y construye un modelo para cada clase en base a las características de los datos. Un conjunto de reglas de clasificación se genera por un proceso de clasificación, que se puede utilizar para clasificar los datos futuros y desarrollar una mejor comprensión de cada clase en la base de datos. Por ejemplo, se pueden clasificar las enfermedades y proporcionar los síntomas que describe cada clase o subclase.
- **Predicción:** predice los posibles valores, algunos datos que faltan o la distribución de valores de ciertos atributos en un conjunto de objetos. Consiste en encontrar el conjunto de atributos relevantes con los atributos de interés y la predicción de la distribución de valor en base al conjunto de datos similares a objetos seleccionados. Por ejemplo, el salario potencial de un empleado se puede predecir sobre la base de la distribución salarial de empleados similares en la empresa.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- **Agrupamiento:** agrupa a un conjunto de datos seleccionados en la base de datos o almacén de datos en un conjunto de grupos para asegurar que la similitud entre clases es baja y la similitud intraclase es alta.

Varios han sido los trabajos realizados en aras de unir ambos paradigmas, desde que Han en (Han, 1997) introdujera el término OLAM (*On-line Analytical Mining*). Las razones para el uso de OLAM que plantea son:

- Las herramientas de extracción de datos necesitan trabajar sobre datos integrados, consistentes y sobre los que previamente se ha aplicado una limpieza (Fayyad, y otros, 1996). Un AD se construye realizando este tipo de pre procesamiento sobre los datos. Por esto, constituye una fuente para obtener los datos en el análisis.
- Los usuarios habitualmente requieren moverse sobre la información de la base de datos, seleccionar porciones de los datos y analizarlos a diferentes niveles de abstracción. Los sistemas OLAP permiten realizar este movimiento de forma flexible. La integración de técnicas de minería de datos dará mecanismos para ver las relaciones entre los datos manteniendo la flexibilidad de cara al usuario.
- Para los usuarios a veces es difícil predecir previamente que tipo de conocimiento quieren obtener. Integrando OLAP con diferentes métodos de minería de datos, se da la libertad al usuario de seleccionar diferentes técnicas de minería de datos.

Han estableció además cinco posibles áreas de desarrollo dentro de la temática: caracterización y comparación basada en OLAP, asociación basada en OLAP, clasificación basada en OLAP, predicción basada en OLAP y agrupación basada en OLAP.

La comunidad científica ha prestado especial interés a la combinación de las técnicas OLAP con las reglas de asociación, pues dichas técnicas ofrecen una fuente de datos previamente procesados e integrados que reducen considerablemente el esfuerzo asociado al pre procesamiento de datos en la aplicación de técnicas de minería de datos. En la Figura 1 se muestran las diferentes fases de la extracción de conocimiento en bases de datos según (Fayyad, y otros, 1996), de forma tal que el esfuerzo asociado al pre procesamiento de los datos pueda ser contrastado.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

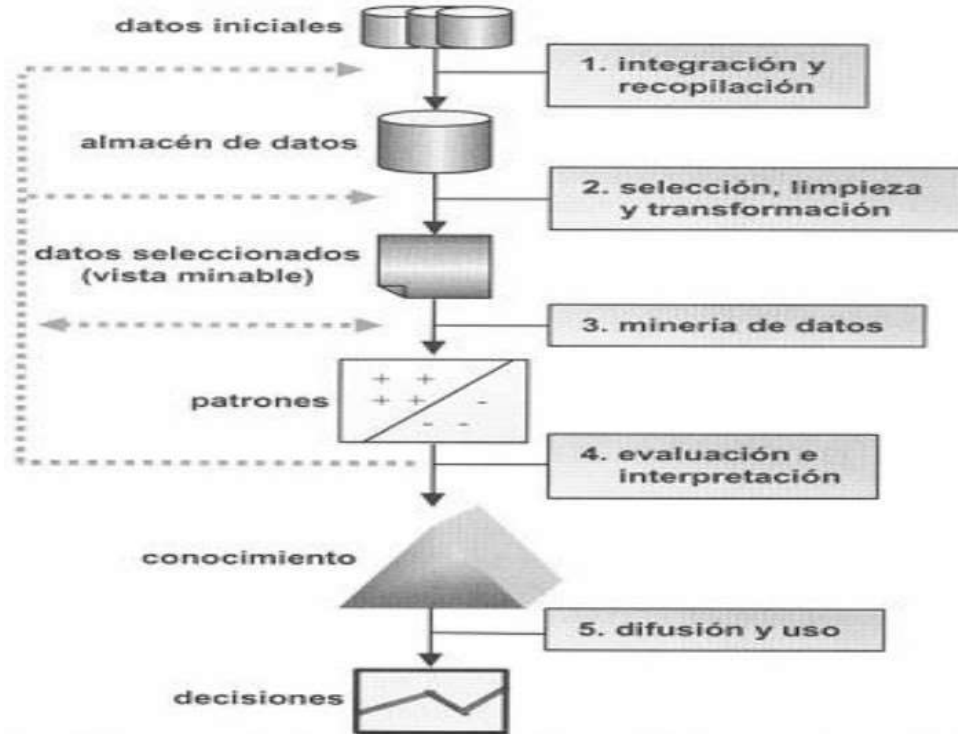


Figura 1. Fases del proceso de extracción de conocimiento en bases de datos

Una regla de asociación (Agrawal, y otros, 1994) en minería de datos se utiliza para establecer relaciones entre elementos que tienen ocurrencia común dentro de un grupo de transacciones. Es decir que la aparición de unos elementos dentro de una transacción implica la aparición de otros grupos de elementos en la misma transacción con cierta medida de certeza.

Los criterios que conforman la base para la extracción de dichas reglas son el soporte y la confianza. El soporte no es más que la proporción de transacciones en la base de datos que contiene dicha regla. Y la confianza puede ser interpretada como la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda.

La extracción de reglas de asociación trae consigo una dificultad implícita que consiste en identificar las que satisfagan un umbral de evaluación para el soporte y la confianza. Para este

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

proceso primero se encuentran los itemset¹ frecuentes, es decir, los que tienen un soporte mayor que un umbral de evaluación. Se generan las reglas de asociación fuertes que son aquellas que satisfacen de igual manera un umbral de confianza.

Uno de los grandes problemas que se presentan en la extracción es la cantidad de reglas que se generan y la dificultad que representa su análisis. Para paliar dicha dificultad se han hecho varias propuestas (Balcazar, 2010), (Lee, y otros, 2006), (Bonchi, y otros, 2004), (Li, y otros, 2009) y (Berrado, y otros, 2007), resaltando en todas la introducción de conocimiento previo del dominio dentro del proceso de generación de reglas por los resultados obtenidos.

De manera natural las ontologías han ganado un lugar dentro de los trabajos que proponen hacer uso del conocimiento previo en la extracción de reglas de asociación gracias a su capacidad de representar de manera formal un dominio determinado.

En (Guarino, N., 1998) se presenta a la ontología como un artefacto ingenieril, constituido por un vocabulario específico que se usa para describir una realidad, más un conjunto de supuestos explícitos en lo que concierne al entendimiento del significado de las palabras del vocabulario. Este conjunto de supuestos tiene usualmente forma de lógica de primer orden, donde las palabras del vocabulario aparecen como nombres de predicado unarios o binarios, llamados conceptos y relaciones respectivamente. En el más simple de los casos, una ontología describe una jerarquía de conceptos relacionados jerárquicamente. Por otro lado en casos más complejos, se añaden axiomas para expresar otras relaciones entre conceptos.

El lenguaje de representación ontológica más difundido es el OWL (OWL Working Group, 2009) que está diseñado para aplicaciones que necesitan procesar el contenido de la información. Tiene

¹ Un ítemset o conjunto de ítems está compuesto por atributos, variables o campos de cualquier base de datos. Su tamaño se determina de acuerdo a la cantidad de ítems que contiene, definiéndose como k-ítemset, donde k es igual al número de ítems (Medina, y otros, 2007).

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

varias representaciones que ayudan a una mejor interpretación y fuerza expresiva por ejemplo RDF (*Resource Description Framework*) y XML (*eXtensible Markup Language*).

Para los seres humanos es difícil el procesamiento, análisis e interpretación de la información representada en OWL. Actualmente se desarrollan alternativas que puedan facilitar dichas tareas. Siendo la representación gráfica o visualización en forma de diagramas, la técnica más utilizada.

Existen varias herramientas para la visualización de ontologías entre ellas destacan Ontolingua, Chimaera y Protégé. Pero las mismas no están concebidas o diseñadas para personas que no sean expertas en la rama de la informática dedicada a las formas de representar la información, específicamente a las ontologías. Es decir que cuando un decisor tiene que manipularlas y analizarlas primero tiene que conocer y poseer un nivel elevado de lógica descriptiva para trabajar con la herramienta escogida.

En la Figura 2 se puede observar la complejidad que implica para los seres humanos el manejo y entendimiento del código OWL comparado con el equivalente visual de la misma ontología.

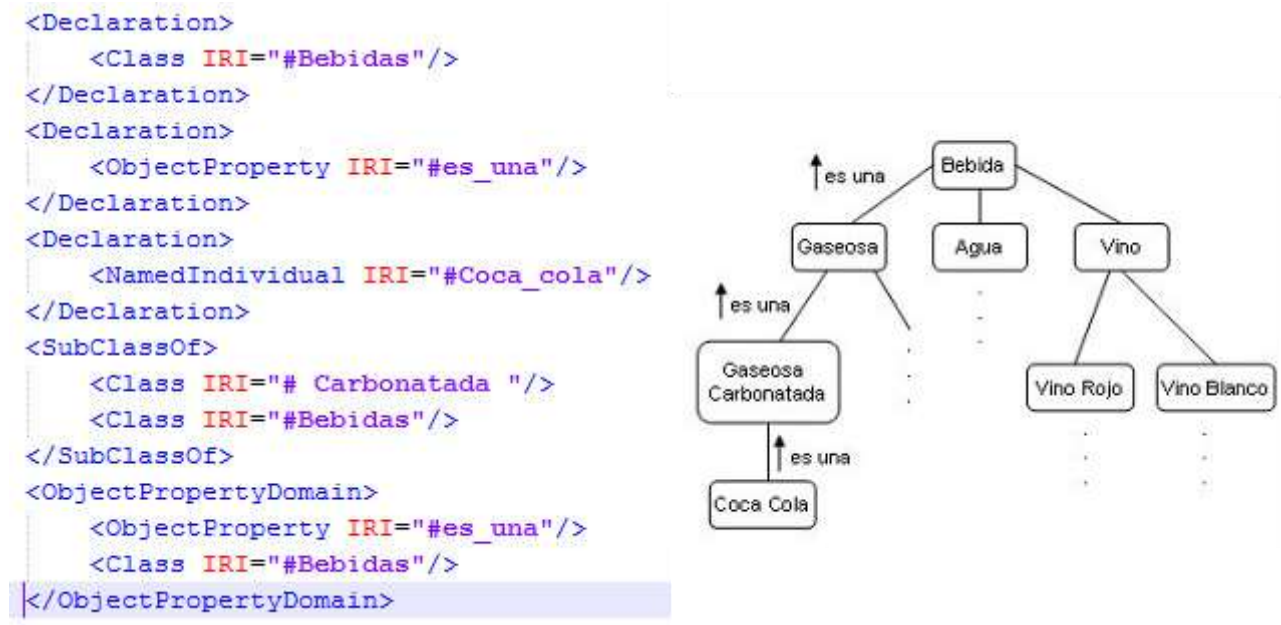


Figura 2. Ontología sobre bebidas y parte del código OWL relacionado

Para esta investigación se identifica como **problema**: la alta complejidad técnica de los modelos de representación del conocimiento, en bases de datos multidimensionales, dificulta a los

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

especialistas la incorporación, a los mismos, de su conocimiento del dominio. Enfocándose como **objeto de estudio** la ontología del modelo multidimensional de datos y **campo de acción** la representación gráfica de ontologías.

Se persigue como **objetivo general** desarrollar una aplicación que permita representar el conocimiento del dominio relacionado con las asociaciones de elementos en un cubo de datos. Y sirven de complemento a este objetivo los siguientes **objetivos específicos**:

- Establecer el marco conceptual de referencia.
- Establecer un modelo teórico para la representación de asociaciones en el modelo multidimensional.
- Definir equivalencias entre elementos visuales y los objetos de la ontología.
- Definir un mecanismo para obtener la representación visual de la ontología de un cubo de datos y las relaciones entre sus elementos.
- Constatar la efectividad de la solución en un caso de estudio.

Los cuales derivan en las siguientes **tareas a cumplir**:

1. Recopilación de la bibliografía referente al tema
2. Selección de la bibliografía
3. Análisis de la bibliografía
4. Definición de los elementos esenciales que componen la ontología y asociación a una representación gráfica
5. Definición de un mecanismo de representación gráfico del conocimiento previo sobre asociaciones de elementos en cubos de datos
6. Implementación de una aplicación que permita editar la representación ontológica de un cubo de datos y las asociaciones entre sus elementos
7. Diseño de experimentos que validen la solución
8. Presentación de los resultados

Se propone como **idea a defender** que si se construye una aplicación que permita representar gráficamente los elementos del modelo multidimensional y sus relaciones, se facilitará la extensión del modelo de dominio por parte de los especialistas. Se espera tener como **posible**

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

resultado una aplicación que permita editar de manera gráfica una ontología del modelo multidimensional.

En el **Capítulo 1** se desarrolla el marco conceptual a partir del análisis crítico de los principales conceptos de ontología, esquemas conceptuales, almacenes de datos y OLAP, minería de datos, reglas de asociación; propuestos por diferentes especialistas reconocidos en el tema. Además se definen las herramientas y tecnologías a usar en la implementación de la herramienta para la edición de ontologías del modelo multidimensional (*HEOMM*).

El **Capítulo 2** describe la arquitectura utilizada para el desarrollo de la HEOMM, así como los usuarios que harán uso de ella y las funcionalidades que brinda. Se mencionan las principales estructuras de la ontología del modelo multidimensional, definiéndose los elementos que la componen, siguiendo el estándar propuesto por el lenguaje OWL.

El **Capítulo 3** presenta los diferentes métodos utilizados para validar los resultados de una investigación: los experimentos, modelos matemáticos, el razonamiento lógico y la demostración. Este último es el seleccionado para validar la solución propuesta en esta investigación por adaptarse a sus características. Se describen los recursos computacionales necesarios para la ejecución de la HEOMM y para desarrollar la demostración. Se incluye la descripción de todos los elementos que conforman el dominio de la demostración. Por último se presentan los resultados obtenidos y se realiza una discusión de los mismos, demostrando el cumplimiento del objetivo general.

1 CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace alusión a los diversos conceptos que sirven de soporte teórico para la presente investigación. Se realizan búsquedas en varias fuentes bibliográficas para contrastar criterios de varios autores y llegar a un consenso que recoja los elementos afines al tema en desarrollo.

1.1 Almacenes de datos y OLAP

Un almacén de datos no es más que una base de datos que recoge la información relacionada con un determinado tema. Está diseñado para obtener los datos de las áreas de interés de una organización, transformarlos e integrarlos. Una vez almacenada la información se convierte en solo lectura, convirtiéndose en un expediente, sobre el cual se pueden realizar análisis, haciendo uso de OLAP, para la toma de decisiones. Los almacenes de datos están compuestos por hechos, métricas y dimensiones, las que contienen atributos que permiten establecer jerarquías entre ellos.

En (Molina, 2005) se plantea una estructura para el modelo multidimensional que expresa:

Una dimensión es una tupla $d = (l, \leq_d, l_{\perp}, l_{\top})$ donde $l = \{l_1, \dots, l_n\}$ tal que cada l_i es un conjunto de valores y $l_i \cap l_j = \emptyset$ si $i \neq j$, y \leq_d es una relación de orden parcial entre los elementos de l . Además l_{\perp} y l_{\top} son dos elementos de l tal que $\forall l_i \in l, l_{\perp} \leq_d l_i$ y $l_i \leq_d l_{\top}$. Cada elemento l_i se denomina nivel. Para identificar el nivel l_i de la dimensión d se denota como $d.l_i$. Los niveles especiales l_{\perp}, l_{\top} serán nivel base y nivel superior respectivamente. Mediante la relación de orden parcial se define una jerarquía entre los niveles existentes en la dimensión.

Luego considerando un conjunto de atributos A_1, \dots, A_n con dominios D_1, \dots, D_n , se denomina hecho a cualquier $h = (x_1, \dots, x_n)$ tal que $x_i \in D_i \forall i = 1, \dots, n$, es decir, cualquier n-tupla definida sobre los dominios de los atributos que interesa estudiar.

Entonces un cubo de datos es una tupla de la forma $C = (D, l_b, F, H, A)$. Donde $D = d_1, \dots, d_n$ es un conjunto de dimensiones $l_b = l_{1b}, \dots, l_{nb}$ es un conjunto de niveles tal que l_{ib} pertenece a d_i , $F = R \cup \emptyset$ donde R es el conjunto de hechos y \emptyset es un símbolo especial, H es un objeto de tipo historia (ver (Molina, y otros, 2006) para detalles de este elemento) y A es una aplicación definida

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

sobre los niveles de las dimensiones que le hace corresponder a cada conjunto de valores el hecho con el que está relacionado.

En la implementación de un AD se debe tener en cuenta que este es un proceso de reingeniería de la información que se va formando con la organización sobre la que se está construyendo. Lo que conlleva a que muchos autores utilicen el término como proceso «*Data Warehousing*» en vez del estático «*Data Warehouse*» (Molina, 2005).

“OLAP es una tecnología de software que permite a los analistas, directivos y ejecutivos acceder a los datos de forma rápida, consistente e interactiva a través de una amplia variedad de vistas de la información que han sido obtenidas de datos sin procesar para reflejar la dimensionalidad real de la empresa como la entiende el usuario” (OLAP Council).

A continuación se muestran en la Tabla 1 las diferencias entre OLAP y OLTP, a las que se hacía mención anteriormente:

Tabla 1. Comparación entre OLAP y OLTP

Aspecto	Base de dato clásica	Almacén de datos
Usuarios	Diseñadores, DBAs, Operadores de entradas de datos	Decisores, ejecutivos
Función	Operaciones diarias (online)	Soporte de decisiones, procesamiento analítico
Diseño	Orientado a aplicaciones	Orientado al usuario
Datos	Actuales, atómicos, relacionales, aislados	Históricos, resumidos, dimensionales, integrados
Uso	Repetitivo, rutinario	ad-hoc ²

² Según la Real Academia de Lengua Española es una expresión latina que significa:

- Que es apropiado, adecuado o especialmente dispuesto para un determinado fin.
- Que está hecho especialmente para un fin determinado o pensado para una situación concreta.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

Acceso	Lectura/escritura, transacciones simples	Lectura, consultas complejas
Necesidades	Gestión de transacciones, datos consistentes	Gestión de consultas, datos ajustados

La información contenida en los AD se gestiona por medio de uno o varios servidores OLAP, mostrando así la relación entre ambos conceptos. Dichos servidores presentan vistas multidimensionales de los datos a una gran variedad de interfaces: interfaces de consulta directa, herramientas para generar informes, de análisis exploratorio (gráficos, etc.) y herramientas de minería de datos propiamente dichas.

1.2 Minería de datos (MD)

La MD según (Triantaphyllou, 2010) permite la extracción de conocimiento y el análisis de los datos del sistema de interés permitiendo incrementar el conocimiento que del mismo se tiene. Incluye varias tareas:

1. Clasificación
2. Estimación
3. Predicción
4. Determinar grupos afines o reglas de asociación
5. Clustering (*agrupamiento*)
6. Descripción y visualización

Entre las técnicas de MD más populares están las siguientes:

- Inducción de árboles de decisión (*ID3*) (Carbonell, 1990) (Kodratoff, y otros, 1990)
- Determinación automática de clusters o grupos (Berry, 2000)
- Redes neuronales (*RN*) (Freeman, y otros, 1993)
- Ajuste de un modelo por medio de Algoritmos Genéticos u otras metaheurísticas (Goldberg, 1989)
- Construcción de modelos estadísticos a partir de los datos (Larrañaga, y otros, 1999)
- Reglas de asociación (Agrawal, y otros, 1993)

De las técnicas mencionadas se escoge la relacionada con las reglas de asociación para que los especialistas introduzcan sus conocimientos del dominio al modelo de forma sencilla y entendible, formal y no formalmente.

1.2.1 Reglas de asociación

Dentro de las técnicas de extracción de conocimiento procesable implícito en las bases de datos se encuentran las reglas de asociación. Estas se utilizan para descubrir elementos que ocurren en común dentro de una misma transacción a lo largo de todo el conjunto de transacciones, cada vez que sucede un antecedente se producirá un consecuente con cierta medida de certeza (Agrawal, y otros, 1994).

Sea $I = \{i_1, i_2, i_3, \dots, i_n\}$ un conjunto de literales y D un conjunto de transacciones definidas sobre I . Cada transacción T contenida en D tiene un conjunto de elementos X si y solo si $X \subseteq T$. Una regla de asociación es una implicación de la forma $X \rightarrow Y$, donde $X \subset I$, $Y \subset I$ y $X \cap Y = \phi$.

El soporte no es más que la proporción de transacciones en la base de datos que contiene dicha regla:

$$sop(X \rightarrow Y) = \frac{|X \cup Y|}{|D|}$$

La confianza puede ser interpretada como la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda. Cuya definición matemática sería:

$$conf(X \rightarrow Y) = \frac{sop(X \cup Y)}{sop(X)}$$

Una regla $X \rightarrow Y$ tendrá en D una *confianza* con valor c si el $c\%$ de las transacciones en D que contienen X también contienen Y . La regla tendrá un *soporte* con valor s en el conjunto de transacciones D si el $s\%$ de las transacciones en D contienen a los elementos $X \cup Y$. La confianza da una medida de la calidad de la regla (la fuerza de la implicación).

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

Normalmente se desea encontrar asociaciones entre elementos con un soporte razonablemente alto. Las reglas con alto valor de estas medidas son denominadas reglas fuertes (Agrawal, y otros, 1993). La extracción de reglas de asociación sigue un procedimiento de dos pasos (Agrawal, y otros, 1993), (Agrawal, y otros, 1994):

- Encontrar todos los ítemsets frecuentes (soporte mayor que un umbral de evaluación)
- Generar todas las reglas de asociación fuertes, que son aquellos que satisfacen de igual manera un umbral de confianza.

Uno de los principales problemas que ha sido objeto de investigación en el área, radica en la gran cantidad de reglas que son generadas y en la dificultad de utilizar una gran parte de las mismas dentro del proceso de toma de decisiones ya sea porque son obvias, demasiado generales, demasiado específicas o porque no tienen interés para el usuario final. Cuando se hace uso de OLAP dicha agravante se acentúa debido a que los almacenes poseen estructura jerárquica, por lo que surgen reglas multiniveles que implican un aumento de la cantidad de ítemset frecuentes según el nivel de granularidad en el que se esté analizando y por tanto la generación de más reglas de asociación. En presencia de redundancia esta situación es aún más compleja. Una regla redundante no es más que una regla que cubre la misma información que otras, o el conocimiento que aporta es conocido de antemano.

Para paliar dicha dificultad se han hecho varias propuestas (Balcazar, 2010), (Lee, y otros, 2006), (Bonchi, y otros, 2004), (Li, y otros, 2009) y (Berrado, y otros, 2007), destacándose en ellas por los buenos resultados alcanzados, la integración del conocimiento previo del dominio dentro del proceso de generación de reglas. Los principales resultados se han alcanzados en variantes de minado con restricciones que utilizan las expectativas del usuario final para limitar el espacio de búsqueda a través de poda o generalización de elementos (Baralis, y otros, 2009) o limitar las reglas que forman parte del conjunto final; ya sea generalizando alguna de las reglas para que sean más comprensibles (Marin, y otros, 2008), (Sriphaew, y otros, 2002) o eliminando reglas que no parezcan significativas al usuario (De Carvalho, y otros, 2007).

Zhu (Zhu, 1998) diferencia entre tres tipos de asociaciones posibles:

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- Asociaciones intra-dimensionales. En este caso se buscan asociaciones entre los elementos de una dimensión (*item dimension*), considerando otra dimensión para obtener el soporte de los elementos (*transaction dimension*).
- Asociaciones inter-dimensionales. Las relaciones se buscan entre elementos de diferentes dimensiones.
- Asociaciones híbridas. En este caso las relaciones que se buscan son entre elementos de una misma dimensión y entre elementos de diferentes dimensiones. Se trata de la unión de asociaciones intra-dimensionales e inter-dimensionales. Es decir, buscan y fusionan conjuntos de elementos frecuentes intra-dimensionales e inter-dimensionales

El proceso de minado de datos ha demostrado ser de utilidad para las organizaciones ya que trae consigo el aumento del conocimiento que se posee sobre la fuente de información en análisis. Una de las técnicas sobre las que se apoya la MD es la extracción de reglas de asociación, que usualmente genera grandes cantidades, mayor si se realiza sobre un AD, dificultando el análisis de las mismas. La introducción de conocimiento previo es una de las soluciones a este problema.

1.3 Ontología

Desde la aparición de la inteligencia artificial, las ontologías cobran una especial importancia en este campo, ya que con ellas se puede dotar de información a los sistemas, pero no una información cualquiera, sino una que representa de forma conceptual el conocimiento, donde los conceptos están relacionados unos con otros.

La palabra ontología se deriva del griego *ontos* (estudio del ser) y *logos* (palabra). Filosóficamente, ontología es la ciencia de qué es, es una explicación sistemática de la existencia, de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad (Sánchez, 2007).

En la informática y la inteligencia artificial, una ontología corresponde a un artefacto de conocimiento que es utilizado para representar un universo de discurso particular y que puede ser interpretado tanto por el sistema computacional como por los seres humanos. La definición más difundida expresa que “es una especificación formal y explícita de una conceptualización compartida” (Studer, y otros, 1998). Por *conceptualización* se refiere a una vista simplificada y abstracta del mundo que se desea representar para algún propósito específico, definiendo un

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

vocabulario controlado. Siendo una *especificación explícita* los tipos de conceptos utilizados y las restricciones sobre su uso se definen explícitamente. *Formal* se refiere al hecho de que la ontología debe ser legible por máquina. *Compartida* refleja la noción de que una ontología captura conocimiento consensual que no es privado para un individuo, sino que es aceptada por un grupo de personas (Hendler, 2008).

En (Guarino, N., 1998) se define matemáticamente, una conceptualización del dominio como $C = \langle D, W, \mathfrak{R} \rangle$, siendo D el dominio, W el máximo conjunto de estados de casos del dominio (también llamado posibles mundos) y \mathfrak{R} es un conjunto de relaciones conceptuales en el espacio de dominio $\langle D, W \rangle$.

Luego se considera un lenguaje lógico L con vocabulario V . Se define un modelo para L con una estructura $\langle S, I \rangle$, donde $S = \langle D, R \rangle$ es una estructura del mundo e $I: V \rightarrow D \cup R$ es una función de interpretación asignando elementos de D a constantes de V , y elementos de R (conjunto de relaciones relevantes en D a predicados de V .

Entonces el compromiso ontológico K sobre el lenguaje lógico L tiene la estructura $\langle C, \mathfrak{S} \rangle$ donde $C = \langle D, W, \mathfrak{R} \rangle$ es una conceptualización y $\mathfrak{S}: V \rightarrow D \cup \mathfrak{R}$ es una función que asigna elementos de D a constantes de V y elementos de \mathfrak{R} a predicados de V .

Por lo tanto una ontología es $O_k = \langle C, L, V, K \rangle$ siendo C una conceptualización, y L un lenguaje lógico con vocabulario V y compromiso ontológico K . Una ontología O_k para C con vocabulario V y compromiso ontológico K es una teoría lógica compuesta de un conjunto de fórmulas de L , diseñado de forma que el conjunto de sus modelos aproxima tanto como es posible el conjunto de modelos deseados de L de acuerdo a K .

Las propiedades a cumplir por una ontología según (Lamarca, 2007) son:

- **Claridad:** para comunicar el significado de los términos definidos.
- **Coherencia:** para sancionar inferencias que son consistentes con las definiciones.
- **Extensibilidad:** para anticipar el uso de vocabulario compartido.
- **Especificidad:** debe especificar al nivel de conocimiento sin depender de una codificación particular a nivel de símbolo.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- **Precisión:** debe de hacer la menor cantidad de “pretensiones” acerca del mundo modelado.

Según (Gruber, 1993), las ontologías se componen de:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como: asignar-fecha, categorizar-clase, etc.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.
- **Reglas de restricción o axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

Existen cuatro tipos de ontologías en función de su alcance y posibilidad de aplicación según (Guarino, N., 1998):

- **Ontología de la aplicación:** usadas por la aplicación.
- **Ontología del dominio:** específicas para un tipo de artefacto, generalizaciones sobre tareas determinadas en algún dominio concreto del conocimiento.
- **Ontologías técnicas básicas:** describen características generales de artefactos.
- **Ontologías genéricas:** describe la categoría de más alto nivel, exponiendo conceptos generales (*tiempo, espacio, objeto, etc.*).

1.3.1 Lenguajes para la representación de ontologías

Los lenguajes usados para la representación de las ontologías son (OWL Working Group, 2009):

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- El **Resource Description Framework** es un framework para metadatos en la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C).
- El **DARPA Agent Markup Language (DAML)** utilizado en la capa de descripción de la pila tecnológica de Web Services.
- **Ontology Inference Layer (OIL)**, puede considerarse como una infraestructura de ontologías para la Web Semántica. Se basa en conceptos desarrollados en Descripción Lógica (DL) y los sistemas basados en cuadros y es compatible con RDF Schema.
- **Web Ontology Language (OWL)**, lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML. Tiene como antecedente DAML+OIL, en los cuales se inspiraron los creadores de OWL para crear el lenguaje. Junto al entorno RDF y otros componentes, estas herramientas hacen posible el proyecto de web semántica.

OWL o Lenguaje de Ontologías para la Web es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque con más poder expresivo. Se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL surge como una revisión al lenguaje DAML-OIL pero es mucho más potente. Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que se precise y a los distintos tipos de aplicaciones existentes (motores de búsqueda, agentes, etc.).

OWL proporciona tres sub-lenguajes diseñados para ser utilizados por comunidades específicas de desarrolladores y usuarios. La característica que define a cada lenguaje es su expresividad.

- **OWL Lite:** es el sub-lenguaje con sintaxis más simple, su intención es ser utilizado en situaciones donde se requiera una jerarquía de clases y restricciones simples.
- **OWL DL:** es mucho más expresivo que OWL Lite y está basado en lógica descriptiva. Proporciona la máxima expresividad posible sin perder la completitud computacional

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

(todas las conclusiones pueden ser deducidas) y la posibilidad (todos los cálculos se realizan en un tiempo finito).

- **OWL Full:** es el sub-lenguaje más expresivo, su intención es ser utilizado en situaciones donde una alta expresividad es más importante que la capacidad de garantizar la completitud computacional y la posibilidad.

OWL DL es una extensión de OWL Lite y OWL Full es una extensión de OWL DL. Las afirmaciones siguientes se cumplen (sus inversas no) (McGuinness, y otros, 2004):

- Una ontología OWL Lite válida, es una ontología OWL DL válida.
- Cualquier conclusión válida de una ontología OWL Lite, será una conclusión válida en OWL DL.
- Una ontología OWL DL válida, es una ontología OWL Full válida.
- Cualquier conclusión válida de una ontología OWL DL será una conclusión válida en OWL Full.

En OWL DL y OWL Lite, sólo ciertas construcciones están permitidas, y estas solamente pueden combinarse de algunas maneras. Los beneficios de permanecer dentro de esas limitaciones incluyen la capacidad para realizar las inferencias y la posibilidad de pensar en OWL de una forma más estándar, esencialmente como una expresiva lógica descriptiva. Por otro lado en OWL Full, se permiten todos los grafos RDF. Los beneficios de este incluyen compatibilidad total con RDF y un mayor poder expresivo aunque por esto se limita su capacidad para realizar inferencias. (Horrocks, y otros, 2003)

En la Figura 3 se muestra un ejemplo de representación ontológica haciendo uso del lenguaje OWL y en la Figura 4 se puede observar el equivalente al código haciendo uso de elementos visuales.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.inf.unibz.it/swt/ontologies/2009/11/Puzzle.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  ontologyIRI="http://www.inf.unibz.it/swt/ontologies/2009/11/Puzzle.owl">
  <Prefix name="" IRI="http://www.inf.unibz.it/swt/ontologies/2009/11/Puzzle.owl#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Prefix name="Puzzle" IRI="http://www.inf.unibz.it/swt/ontologies/2009/11/Puzzle.owl#"/>
  <Prefix name="owl2xml" IRI="http://www.w3.org/2006/12/owl2-xml#"/>
  <Declaration>
    <Class IRI="#IndiceAudiencia"/>
  </Declaration>
  <Declaration>
    <Class IRI="#Pelicula"/>
  </Declaration>
  <Declaration>
    <Class IRI="#PeliculaGatos"/>
  </Declaration>
  <Declaration>
    <Class IRI="#Persona"/>
  </Declaration>
  <Declaration>
    <ObjectProperty IRI="#tieneNivelViolencia"/>
  </Declaration>
  <SubClassOf>
    <Class IRI="#IndiceAudiencia"/>
    <Class IRI="#PeliculaGatos"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#PeliculaGatos"/>
    <Class IRI="#Pelicula"/>
  </SubClassOf>
  <DisjointClasses>
    <Class IRI="#Pelicula"/>
    <Class IRI="#Persona"/>
  </DisjointClasses>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#tieneNivelViolencia"/>
    <Class IRI="#Pelicula"/>
  </ObjectPropertyDomain>
</Ontology>
```

Figura 3. Código OWL de una ontología

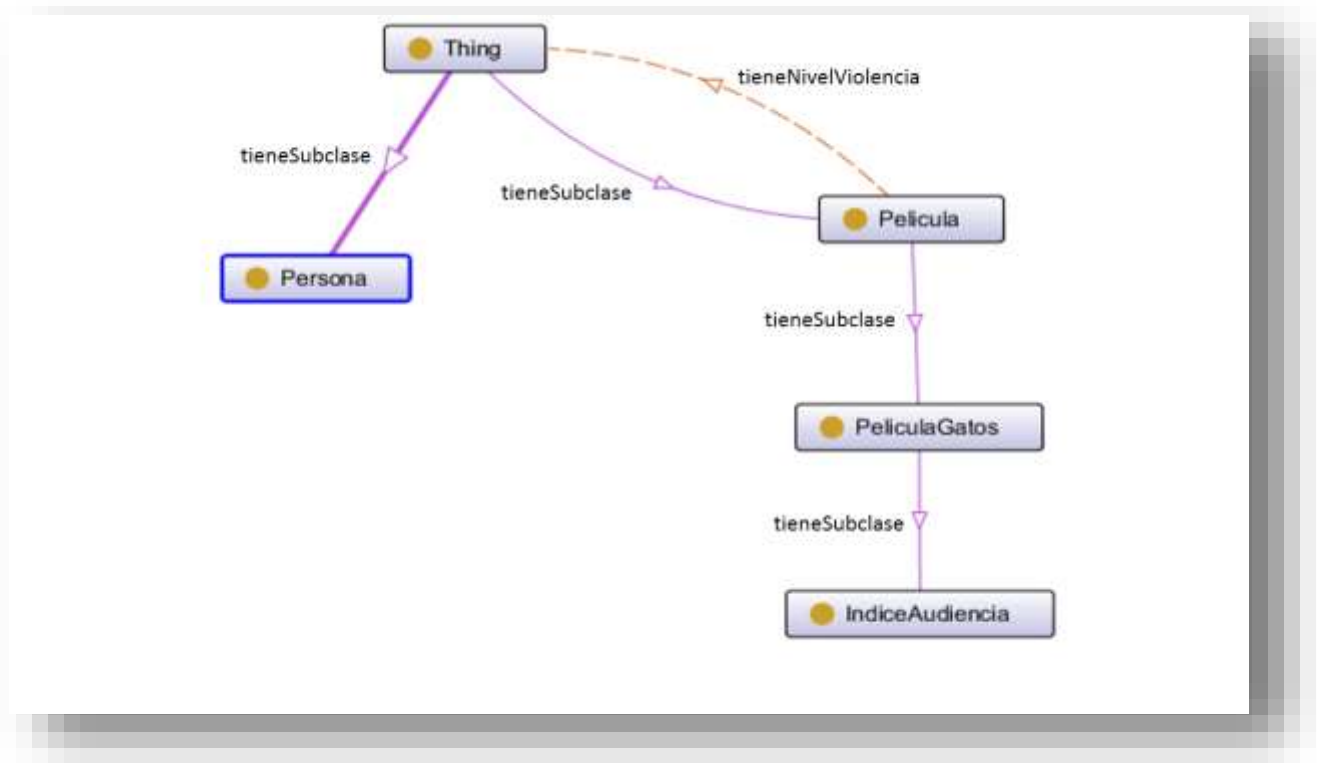


Figura 4. Representación visual equivalente al código OWL presentado, utilizando Protégé

OWL es una forma eficiente de mostrar una ontología, siendo además un lenguaje diseñado para ser procesado fácilmente por ordenadores (OWL Working Group, 2009). No obstante para los seres humanos se torna complicado el manejo de este, no siendo así el equivalente visual, donde fácilmente se puede apreciar la estructura de la ontología.

1.4 Esquemas conceptuales

Existen diversas formas de representación de ideas, entre ellas destacan los esquemas conceptuales que según (Robins, y otros, 1997) “*son modelos internos sobre aspectos del mismo y del mundo, que los individuos utilizan para percibir, codificar y recuperar la información. Se van desarrollando a través de las numerosas experiencias particulares, y son adaptativos, desde el momento en que facilitan un procesamiento de la información más eficaz.*”

1.4.1 Distintas aproximaciones para la representación de esquemas conceptuales

Desde épocas antiguas las representaciones gráficas tienen un papel determinante en la comunicación humana. Las pinturas rupestres encontradas muestran la utilización del dibujo para representar una idea, un conocimiento o un mensaje. Con el paso del tiempo y el surgimiento de la lengua escrita la expresión pictográfica fue dejada de lado. Aunque, la comunicación visual a través de figuras no ha perdido su sentido humano como medio de expresión del hombre.

La sociedad actual está fuertemente influenciada por mensajes visuales. El creciente uso de la televisión, la variedad de revistas, los anuncios publicitarios, el acceso cada vez mayor al internet, etc., propicia que grandes volúmenes de información esté expresada en imágenes.

Durante mucho tiempo los aspectos visuales y gráficos fueron considerados elementos de apoyo. Ahora se estudian e incluso se han desarrollado teorías explicatorias del aprendizaje visual. Durante los primeros años de la década del 70 se presentan estudios sobre las “representaciones mentales” (*mental imagery*) (Otero, 1999). Se establecía que si la persona representaba mentalmente lo que estaba leyendo la retención sería más efectiva.

No obstante las herramientas o técnicas gráficas usadas tienen sus orígenes en el “cuadro sinóptico”. Es la más antigua pero también la más limitada para la organización interna de un tema. Generalmente consiste en una lista de conceptos o enunciados presentados de una manera vertical.

Fueron (Barron, y otros, 1974) quienes introducen el concepto de “organizador gráfico” basado en la idea de “organizador de avanzada” de David Ausubel. El organizador de Ausubel era textual, brindaba soporte entre el nuevo conocimiento y el ya adquirido por el individuo, pero la información contenida en él era muy general. Barron plantea que para que el organizador cumpla su función debe ser gráfico ya que se evidencian las relaciones de mejor forma. Esta idea, no entendida inicialmente, ha sido retomada con mucha fuerza en estos últimos años, es el nombre más usado y general para incluir todas las maneras gráficas de representación del conocimiento.

Joseph Novak, colaborador de Ausubel en la propuesta de aprendizaje significativo, presenta el “mapa conceptual” como herramienta para hacer efectivo el aprendizaje (Novak, 1998). El trabajo sobre los mapas se inicia a mediados de los años 60 y es en los 80 cuando se difunden y ganan

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

aceptación. Mediante esta técnica se relacionan conceptos de un mismo dominio de conocimiento mediante conectivos verbales. Son versátiles y poco complejos, se les conoce además como grafos o esquemas conceptuales.

En (Buzan, 1993) se propone el uso del “mapa mental”, representación del conocimiento haciendo uso de los dos “lados del cerebro”. Incluye texto, figuras, iconos, color, etc. Es una gráfica más elaborada y compleja que argumenta grandes beneficios en la retención y comprensión.

Pearson y Johnson propusieron, a fines de los años 70, el “mapa semántico” que ha sido difundido por Heimlich y Pittelman (Heimlich, y otros, 2001). Al igual que los mapas anteriores relaciona conceptos pero de una manera más libre, conocido también como grafo léxico.

También aparecen los “mapas cognitivos” mencionados por Tolman en 1948, difundidos y experimentados por Kevin Lynch en los años 60. Estos, al igual que los anteriores, son representaciones del conocimiento (Moreira, 2002).

Donald Norman en (Norman, 2002) introduce la denominación de “herramientas cognitivas” a las representaciones del conocimiento (esquemas, gráficos, tablas) que tratan de poner en comunicación lo perceptivo con lo cognitivo.

Los “mapas de pensamiento” fueron desarrollados por David Hyerle (Thinking, 2012) como un lenguaje para el aprendizaje por cuanto permiten transformar la información en verdadero conocimiento. Sostiene Hyerle que mediante los gráficos los alumnos crean marcos conceptuales que permiten organizar, retener y entender la información.

También se está llamando a estas técnicas como “modelos visuales” (Concari, 2001). Es decir, representaciones gráficas de los modelos mentales (ideas, pensamientos, razonamientos, conceptos, opiniones, etc.). Los modelos mentales usan modelos visuales para expresarse a través de diferentes soportes, formatos o estilos visuales: diagramas, mapas, gráficos, cuadros, matrices, etc. La premisa en esta propuesta es que existe una base de conocimiento (memoria) con información (la cual puede ser procesada y representada en un modelo mental). Esto último puede ser representado gráfica-visualmente en un determinado soporte. Los modelos visuales permiten presentar significados, aspectos destacables y la estructura interna de un contenido. Estos modelos deben ser susceptibles de ser manejados visual y cognitivamente (nivel de

claridad y abstracción apropiados). La visualización es una construcción mental que supera la simple percepción y está más relacionada con el conocimiento.

Asimismo, se mencionan a las “técnicas de aprendizaje visual” como las formas de trabajar con ideas, clarificar el pensamiento, organizar, presentar y priorizar información, establecer relaciones e interdependencia, integrar conocimiento e identificar errores (Cobos, y otros, 2010). También, se nombran a las “herramientas viso-verbales” para mostrar relaciones, integración de conceptos y significados y como ayudas eficientes de aprendizaje. En el área de negocios y capacitación se habla de “herramientas de solución de problemas”.

Win y Snyder mencionan que (Mayer, 1989) lista siete características que debe tener un material visual, en este caso, modelo visual o diagrama, para ser efectivo. Un buen modelo visual debe ser:

1. completo, contener todos los elementos correspondientes,
2. conciso, tener el detalle necesario,
3. coherente, presentar una representación que “tenga sentido”,
4. concreto, tener un nivel adecuado de familiaridad,
5. conceptual, ser potencialmente significativo,
6. correcto, los elementos y sus relaciones deben corresponder a la situación y,
7. considerado, usar vocabulario adecuado y organización apropiada.

Los estudios realizados acentúan que la utilización de representaciones gráficas del conocimiento permite a las personas de todas las edades entender de manera efectiva y eficiente un determinado tema. Así como estimula el pensamiento creativo y reflexivo. El nivel de procesamiento de un diagrama se ve afectado e influenciado por las características personales de su usuario. Además, la preferencia por presentaciones y aprendizaje visual depende no sólo de estilos de aprendizaje o tipo de inteligencia dominante, sino también del contexto cultural.

1.4.2 Tipos de esquemas conceptuales

➤ Mapas conceptuales

Dentro de estos esquemas se encuentran los mapas conceptuales, utilizados como herramientas gráficas para organizar y representar conocimiento (Cañas, y otros, 2009). Incluyen conceptos,

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

usualmente encerrados en figuras, y relaciones entre conceptos indicados por una línea conectiva que los enlaza. Las palabras sobre la línea, denominadas palabras de enlace o frases de enlace, especifican la relación entre ellos. El concepto es una regularidad percibida en eventos u objetos, o registros de eventos u objetos, designados por una etiqueta. La etiqueta para la mayoría de los conceptos casi siempre se describe en una palabra. Las proposiciones son afirmaciones sobre un objeto o evento en el universo, ya sea que ocurra naturalmente o sea construido. Contienen dos o más conceptos conectados mediante palabras o frases de enlace para formar una afirmación con significado. En ocasiones son llamadas unidades semánticas o unidades de significado.

Los tipos de mapas conceptuales son (TiposDe.Org, 2014):

- **Mapas panorámicos:** son aquellos con características descriptivas, los cuales abarcan la temática a tratar de forma general o global.
- **Mapas detallados o desarrollados:** son analíticos, es decir que incluyen gran cantidad de detalles respecto al tema tratado.

➤ **Mapas cognitivos**

Otro de los tipos de esquemas conceptuales son los mapas cognitivos que expresan estrategias que hacen posible la representación gráfica de una serie de ideas, conceptos y temas con un significado y sus relaciones, enmarcando estos en un esquema o diagrama. En la elaboración de mapas cognitivos no sólo se utilizan aspectos visuales, si no que se incluyen otros aspectos sensoriales y motores. Estos mapas representan un rol importante en el desarrollo de la capacidad para resolver problemas de espacio, poseyendo un valor adaptativo. Indicando hacia donde ir para satisfacer necesidades individuales y como llegar ahí.

Poseen las características siguientes:

- Sirven para la organización de cualquier contenido.
- Auxilian a los usuarios para enfocar el aprendizaje a actividades específicas.
- Ayudan a construir significados más precisos
- Permiten hacer diferencias, comparar, clasificar, categorizar, secuenciar, agrupar y organizar una gran serie de documentos.

Existen diferentes tipos (Pimienta, 2013):

- **De sol:** para introducir u organizar un tema, se colocan las ideas que se tienen respecto a un tema o concepto.
- **De telaraña:** para organizar contenidos señalando sus características.
- **De nubes:** para organizar características.
- **De aspectos comunes:** para comparar temas que comparten características, y para distinguir sus diferencias.
- **De ciclos:** para representar información ciclada como circuito.
- **De secuencias:** para jerarquizar información que tiene una continuidad lógica o consecutiva en la solución de problemas.
- **De agua mala:** para organizar los contenidos o temas y subtemas.
- **De panel:** para organizar o clarificar cualquier tipo de información.
- **De comparaciones:** para comparar características semejantes y diferentes de dos temas.
- **De categorías:** para clasificar la información de un tema, unidad u organización.
- **De escalones:** para organizar o clasificar los contenidos.
- **De cadena:** para organizar información de manera jerárquica, con referencias de apoyo.
- **De arco iris:** para clarificar el inicio, final y/o conclusión de un tema o bien para la solución de problemas.
- **De cajas:** para organizar información de manera puntual y sintética.

De los esquemas conceptuales antes mencionados se propone hacer uso de los mapas conceptuales ya que están diseñados para organizar y representar el conocimiento. Además propone relacionar conceptos de un mismo dominio de conocimiento mediante conectivos verbales y se torna poco complejo su entendimiento. Características deseables para lograr mayor entendimiento de la representación visual de una ontología.

1.4.3 Representación de esquemas conceptuales usando ontologías

La gran presencia de las ontologías en la literatura profesional y académica actuales se origina al apreciarlas como uno de los recursos significativos para la Web Semántica, pues resulta

evidente la necesidad de modelar semánticamente los conceptos en un sistema de organización del conocimiento (Antoniou, y otros, 2004).

Al aplicarse las ontologías en y por medio de internet han tenido que habilitar elementos para mejorar su interoperabilidad y reutilización, y han adaptado su expresión a lenguajes web (XML o basados en XML). Funcionalmente, han heredado los frutos de las investigaciones hechas en los Sistemas de Organización del Conocimiento o KOS (*Knowledge Organization Systems*) ahora comprendidos dentro de las ontologías (Zeng Lei, y otros, 2004). Las ontologías constituyen un ingenio práctico para muchos sistemas de los que se espera que dispongan de una estructura que relacione sus elementos y que cuente con cierto grado de inferencia o razonamiento.

Las ontologías tienen como misión representar el conocimiento a partir de la organización taxonómica en cuanto modo de clasificación o categorización jerárquica de los conceptos pertenecientes a un conjunto temático. Siguiendo, por lo general, una configuración que establece entre los conceptos que integra una relación de *generalización-especialización*, es decir, asociando los términos por *subclase-de* o *subclasificación-de*. De este modo aplican una semántica simple de acuerdo con algunas de las propiedades que caracterizan a las ontologías (Daconta, y otros, 2003). Por este motivo, se trata de clasificar o categorizar un conjunto de conceptos con semántica más compleja que varían desde taxonomías a las ontologías más completas y formalizadas.

1.5 Tecnologías y herramientas empleadas para el desarrollo

Para el desarrollo de aplicaciones informáticas es necesario el uso de tecnologías y herramientas. Los marcos de trabajo, compuestos por lenguajes de programación, son algunas de las tecnologías que facilitan el trabajo. Por otro lado se hace necesario utilizar herramientas de apoyo para la implementación, tratamiento de imágenes y visualización de las aplicaciones en caso de ser web (navegadores).

1.5.1 Marcos de trabajo

Un marco de trabajo es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática y provee una metodología de trabajo que sistematiza y facilita la

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación (ALEGSA, 1998).

Se tiene en cuenta para la selección el grado de dificultad y la familiaridad del autor, de lo contrario el tiempo para su aprendizaje. También, la portabilidad y cualidades multiplataforma. Se selecciona:

- **ExtJS**, es biblioteca o conjunto de bibliotecas de JavaScript para el desarrollo de aplicaciones web interactivas que usa tecnologías AJAX (*JavaScript Asíncrono And XML*), DHTML (*HTML Dinámico*) y DOM (*Document Object Model*). Incluye un API (*Application Programming Interface*) fácil de usar y está sujeta a Licencias Open Source (*GPL*) y comerciales. ExtJS permite realizar completas interfaces de usuario de alto performance, personalizables, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio. Permitiéndole a los desarrolladores web concentrarse en la funcionalidad de las aplicaciones (García, 2009). Para la implementación del software se utilizará la versión 3.3.
- **Draw2D**, en la versión 0.9.14, es un marco de trabajo JavaScript que funciona en todos los navegadores modernos, desde Internet Explorer 6 a la última versión de Chrome. Brinda la opción de crear aplicaciones de diagramas de plataforma cruzada usando nada más que un navegador, y una API fenomenal. Es altamente sensible y está cuidadosamente optimizado para ofrecer una experiencia excepcional en una amplia gama de dispositivos, sistemas operativos y navegadores (Herz, 2014).
- **Graphics2D**, es una clase de Java que extiende de la clase Graphics para proporcionar un control más sofisticado sobre geometría, transformaciones de coordenadas, la gestión del color y la disposición del texto. Esta es la clase fundamental para la prestación de formas de 2 dimensiones, el texto y las imágenes en la plataforma Java (Oracle, 1993). No se elige debido a la restricción que representa el estar diseñada para la plataforma especificada.
- **Raphaël.js**, es una pequeña biblioteca JavaScript que simplifica el trabajo con gráficos vectoriales en la web (Baranovskiy,D., 2008). Es web como se menciona, pero su enfoque vectorial no se ajusta a las necesidades de la HEOMM.

Se elige Draw2D debido a la portabilidad que presenta pues está basado en tecnologías web que aportan características multiplataforma. Ofrece en su API ejemplos de aplicaciones usadas para representación gráfica, funcionalidad en la que se basa la HEOMM, facilitándole el trabajo al autor.

1.5.2 Lenguajes de programación

Los lenguajes que a continuación se enuncian se encuentran de manera implícita en el marco de trabajo propuesto, Draw2D:

- **HTML** es el lenguaje de programación utilizado para crear las páginas web de Internet. Tanto las herramientas para crear dichas páginas como los navegadores utilizan HTML para escribir o interpretar su contenido respectivamente. A pesar de ser usado por las computadoras es muy fácil de aprender por parte de las personas. Es un estándar reconocido en todo el mundo y cuyas normas define W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de la misma manera en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como “un lenguaje reconocido universalmente y que permite publicar información de forma global”. Desde su creación ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas: buscadores, tiendas online, banca electrónica, etc.
- **JavaScript** es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Se define como orientado a objetos (ECMA, 2011), basada en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente para crear páginas web dinámicas que son aquellas que incorporan efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java.

1.5.3 Otras herramientas

De forma auxiliar se hace uso de otras herramientas:

- **Entorno Integrado de Desarrollo NetBeans:** es un proyecto exitoso de código abierto con una gran base de usuarios. El proyecto cuenta con una comunidad en constante crecimiento. En la actualidad hay disponibles dos productos: el NetBeans IDE y NetBeansPlatform. NetBeans IDE se considera una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para otros lenguajes de programación: PHP, JavaScript, HTML, C, C++, Groovy y JSP. Existen además un número importante de módulos para extender las funcionalidades del entorno de desarrollo. Es un producto libre y gratuito sin restricciones de uso (Oracle Corporation). Para la implementación de la aplicación se empleará Netbeans IDE en su versión 7.3.
- **Mozilla Firefox**, en su versión 28, es un navegador web que define como sus características fundamentales: la confiabilidad, la flexibilidad y la rapidez. La confiabilidad que ofrece se basa en la privacidad condicionada por tres criterios fundamentales: no ser rastreado, innovación que permite indicar como se prefiere que se recopile o utilice la información personal en línea; navegación privada que permite proteger el historial de navegación; por último el haz de luz permite descubrir quién te observa en línea. Este navegador ha sido recientemente nombrado “rey de la velocidad” en unas pruebas independientes de referencia que miden el rendimiento respecto a los otros navegadores (Mozilla, 1998). Teniendo en cuenta las características expuestas anteriormente se utilizará Mozilla Firefox como navegador web, para visualizar la implementación de la aplicación.
- **Photoshop**, en la versión 7, es un programa de edición de imágenes que puede ser utilizado además para la realización de animaciones digitales sencillas. Es muy utilizada en el retoque fotográfico y en multitud de disciplinas del campo del Diseño y fotografía, como Diseño web, composición de imágenes Bitmap, Estilismo digital, Fotocomposición, edición y grafismos de Vídeo y básicamente en cualquier actividad que requiera el tratamiento de Imágenes digitales. (Adobe Systems Incorporated)

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

Los lenguajes HTML (*Hyper Text Markup Language* o *Lenguaje de Marcas de Hipertexto*) y JavaScript fueron usados como parte del marco de trabajo Draw2D que engloba además el marco de trabajo ExtJS. Las herramientas escogidas por el autor para la obtención del resultado esperado son: NetBeans para la implementación, Photoshop para el tratamiento de imágenes y Mozilla Firefox para la visualización de la aplicación. Fueron seleccionadas por comodidad y dominio del autor pues, usando otras similares se obtendrían los mismos resultados.

1.6 Conclusiones parciales

A partir del estudio realizado en el capítulo se puede plantear que las principales definiciones de los conceptos tratados fueron presentadas por: Inmon para almacén de datos, Triantaphyllou para minería de datos, Agrawal para reglas de asociación y Studer para ontología.

La evolución histórica del hombre ha demostrado que el uso de representaciones visuales ha facilitado la comunicación de los individuos. Las investigaciones muestran que el aprendizaje a través de representaciones gráficas del conocimiento adquirido es efectivo y eficiente en personas de todas las edades.

Se ha constatado que los modelos de representación del conocimiento son herramientas muy útiles pero se tornan complejos cuando se hace uso de bases de datos multidimensionales. Esta dificultad obstaculiza a los especialistas cuando desean la introducción de sus conocimientos del dominio a dicho modelo.

2 CAPÍTULO 2: HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

En este capítulo se presenta un software que soporta la edición de ontologías del modelo multidimensional. Se describe la arquitectura de la HEOMM así como la metodología utilizada en su desarrollo. Además se explican los módulos y funcionalidades del sistema.

2.1 Ontología del modelo multidimensional

Varios investigadores han trabajado en proponer ontologías del modelo multidimensional. Una en particular destaca (Díaz, 2013), que plantea una ontología “core” y describe el modelo multidimensional además de estar orientada a representar reglas de asociación en dicho modelo.

Partiendo de los cinco elementos propuestos por Molina en la definición de un cubo de datos (ver epígrafe 1.3), hace una selección y considera que no son relevantes el objeto de tipo historia H pues representa cada una de las transformaciones por las que ha pasado el almacén y por lo tanto no es significativo para el minado. El conjunto de hechos F es descartado también pues el objetivo principal es representar las asociaciones que existen entre los elementos de las dimensiones, dadas por un experto con conocimiento del negocio.

Para la presente investigación se toman como relevantes los siguientes elementos:

D , l_b y A , usados para agrupar los registros en jerarquías (contenidos en l_b). También se utilizan los valores que toman los elementos de dimensión (contenidos en A), pertenecientes a cada dimensión d_i que se encuentran en el conjunto de todas las dimensiones representado por D . De manera general solo se utiliza para formar la ontología el nombre del cubo de datos, el nombre de cada dimensión, el nombre de cada atributo de dimensión y los valores que toma cada atributo.

Se define esta estructura formal (Figura 5) basada en los conceptos: cubo, dimensión, elemento de dimensión, medida y operador de agregación. Todos ellos agrupados en la ontología del modelo multidimensional permiten crear un marco de trabajo estructural para representar y organizar ontologías de dominio específico a partir de puntos de extensión. También es necesario establecer relaciones entre estos elementos, que se presentan como clases en la ontología, para definir los nexos entre los elementos siguientes:

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- Cubo de datos y dimensiones: *tieneDimensión*.
- Cubo de datos y medidas: *tieneMedida*.
- Cubo de datos y operadores de agregación: *tieneOperadorAgregación*.
- Dimensión y elementos de dimensión: *tieneElementoDimensión*.
- Entre elementos de dimensión, entre sus instancias o entre ellos: *relaciónConocida*.
- Entre elementos de dimensión: *generaliza*.

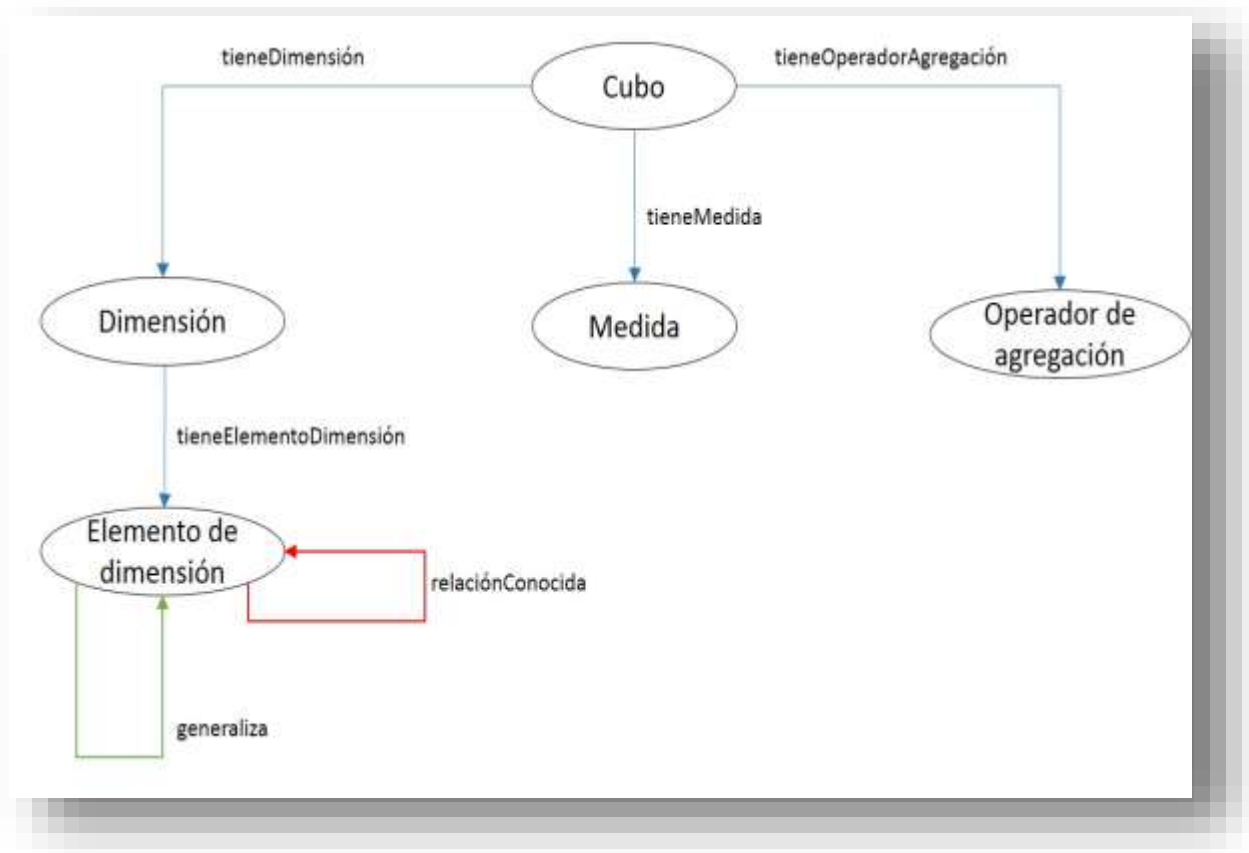


Figura 5. Ontología del modelo multidimensional

2.2 Metodología de desarrollo

Las metodologías de software son un conjunto de estrategias, procedimientos, técnicas, métodos y actividades intencionadas. Las actividades deben estar organizadas, secuenciadas e integradas de manera tal que permitan el logro de aprendizajes significativos y de calidad. Las metodologías son capaces de añadir y optimizar las prácticas de desarrollo de software (Canós, y otros, 2003).

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

Las metodologías de desarrollo de software permiten llevar a cabo actividades para transformar los requisitos de los usuarios en un sistema de software. No existen metodologías estándar, ni mejores. Las mismas se eligen en función de las características de cada proyecto, de su equipo de desarrollo, y de los recursos y tiempo del que se dispone para su elaboración (Morgade, 2009).

Existen dos grandes grupos de metodologías: las tradicionales y las ágiles. De las primeras se puede mencionar a RUP (*Rational Unified Process*) y MSF (*Microsoft Solution Framework*). Este tipo de metodología presenta resistencia a los cambios durante el desarrollo. Además, estos cambios son controlados mediante numerosas normas y políticas. Dada esas características, no son consideradas adecuadas para el desarrollo de la HEOMM.

Dentro del segundo grupo se ubican algunas metodologías reconocidas, tales como: XP (*eXtreme Programming*), DSDM (*Dynamic Systems Development Method*), ASD (*Adaptive Software Development*). Están especialmente preparadas para proyectos con requisitos cambiantes. El cliente forma parte del equipo de desarrollo. Además se recomienda para equipos de desarrollo de entre dos a diez personas en total (Riverón, 2013).

- ASD es una técnica para construir software y sistemas complejos. Se enfoca en la colaboración humana y la organización propia del equipo. Highsmith, quien propone su uso, define un ciclo de vida para esta metodología compuesto por tres fases (Pressman):
 - Especulación: en esta fase se inicia el proyecto y se conduce el ciclo adaptativo de planeación. Para ello se tiene en cuenta el enunciado de la misión del cliente, las restricciones del proyecto y los requisitos básicos.
 - Colaboración: las personas motivadas trabajan juntas de una forma que multiplica su talento y sus salidas creativas más allá de sus números absolutos.
 - Aprendizaje: como miembros de un equipo de ASD se comienza a desarrollar los componentes integrantes de un ciclo adaptativo, la importancia radica en el aprendizaje y el progreso a través de un ciclo completo.

Dada las características de la HEOMM, la presente metodología se adaptaría ya que al formar parte de las ágiles hereda sus ventajas. Pero su enfoque en el equipo de trabajo no se ajusta ya que está conformado por una persona. Además, los errores y cambios que no son detectados con anterioridad afectan la calidad del producto.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- El DSDM proporciona un marco de trabajo para construir y mantener sistemas con restricciones de tiempo muy estrechas. Esto ocurre mediante la construcción de prototipos incrementales en un ambiente de proyecto controlado. Al igual que ASD y XP sigue un proceso iterativo de software. Además, tiene en cuenta el enfoque en cada iteración de la regla del 80 por ciento. El ciclo de vida del DSDM define tres ciclos iterativos diferentes. A estos les preceden dos actividades adicionales del ciclo de vida: estudio de factibilidad, estudio de negocios, iteración de modelo funcional, iteración de construcción y diseño e implementación (Pressman).

Como la anterior esta también hereda las ventajas de las metodologías ágiles de las que se beneficia el proceso de desarrollo de la HEOMM. Pero, al no ser una metodología de uso común, el proceso es un tanto difícil de comprender, razón que implicaría un mayor consumo de tiempo.

- XP, considerada una metodología ágil, pretende minimizar el riesgo de fallos del proceso. Para ello cuenta de forma permanente de un representante competente del cliente a disposición del equipo de desarrollo. El mismo debe estar apto para contestar de forma rápida y concreta cualquier duda que pueda surgir por parte del equipo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo y la comunicación fluida entre todos los participantes, utiliza además las historias de usuario para especificar los requisitos del software. En (Beck, y otros, 2004) se plantea que para la realización de las historias de usuario basta con enunciar y describir los requisitos funcionales. Otros artefactos que genera son las tarjetas CRC (*Clase, Responsabilidad y Colaboración*) y el plan de iteraciones. Se define XP como especialmente adecuada para proyectos con requisitos altamente cambiantes y donde existe un alto riesgo técnico.

Pasos para llevar a cabo el ciclo de desarrollo de un software bajo la metodología XP:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo a sus prioridades y las restricciones del tiempo.
4. El programador construye ese valor de negocio.

5. Vuelve al primer paso.

Teniendo en cuenta las características analizadas se selecciona la metodología ágil XP para el desarrollo de la aplicación. Es la que más se ajusta a las condiciones del equipo de trabajo y a las peculiaridades de la aplicación. Fue seleccionada para guiar el proceso de construcción del software, entre otros factores, por sus bondades para equipos pequeños y un ciclo de desarrollo corto (Beck, y otros, 2004), dos características presentes en la HEOMM. Dada la naturaleza de herramienta, diseñada para la edición de ontologías del modelo multidimensional, se decide generar solamente las historias de usuario.

2.3 Arquitectura

La arquitectura utilizada es N-Capas (Arquitecturas, 2012), ya que ofrece una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan.

Ofrece ventajas como:

- Abstracción, los cambios se realizan a alto nivel y se puede incrementar o reducir el nivel de abstracción que se usa en cada capa del modelo.
- Aislamiento, se pueden realizar actualizaciones en el interior de las capas sin que esto afecte al resto del sistema.
- Rendimiento, distribuyendo las capas en distintos niveles físicos se puede mejorar la escalabilidad, la tolerancia a fallos y el rendimiento.
- Testeabilidad, cada capa tiene una interfaz bien definida sobre la que realizar las pruebas y la habilidad de cambiar entre diferentes implementaciones de una capa.
- Independencia, elimina la necesidad de considerar el hardware y el despliegue así como las dependencias con interfaces externas.

Establece los siguientes principios:

- Muestra una vista completa del modelo y a la vez proporciona suficientes detalles para entender las relaciones entre capas.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

- No realiza ninguna suposición sobre los tipos de datos, métodos, propiedades y sus implementaciones.
- Separa de forma clara la funcionalidad de cada capa.
- Cada capa contiene la funcionalidad relacionada solo con las tareas de esa capa.
- Las capas inferiores no tienen dependencias de las capas superiores.
- La comunicación entre capas está basada en una abstracción que proporciona un bajo acoplamiento entre capas.

A continuación se describen brevemente las capas que componen la HEOMM:

1. **Capa de datos:** en esta capa se encuentra el archivo que contiene la ontología del modelo multidimensional con la que se desea trabajar.
2. **Capa de acceso a datos:** encargada de la escritura y lectura del archivo de la ontología, según la necesidad. Se sirve de la capa inferior de datos. Hace uso de las clases *Abrir* y *Exportar* para las operaciones de la capa.
3. **Capa controladores:** es la que recibe los eventos de entrada desde la vista. Luego procesa y devuelve los datos solicitados por el usuario en la capa presentación. Se vale de las clases *Cubo*, *Dimensión*, *ElementoDimensión*, *Individuo*, *RelaciónConocida* y *Conector*, que brindan funcionalidades que permiten dar solución a los eventos de entrada junto a la clase *Pintar*.
4. **Capa presentación:** contiene las clases encargadas de garantizar la interacción del usuario con el sistema. Se vale de ExtJS y Draw2d.

2.4 Usuarios y funcionalidades

La herramienta podrá ser utilizada por personas que tengan interés en la representación de ontologías del modelo multidimensional. La misma podrá accederse sin hacer usos de servidores web ya que las operaciones se llevan a cabo aprovechando las potencialidades del lenguaje JavaScript sobre el que se basa el marco de trabajo Draw2d. La HEOMM tiene tres módulos (Figura 6) que permiten solucionar el problema planteado. A continuación se hace una descripción de las funcionalidades de cada uno.



Figura 6. Módulos de la HEOMM

2.4.1 Subsistema de carga de la ontología

El subsistema de carga de la ontología tiene como objetivo la lectura del archivo que contiene la ontología que permite al usuario la observación del equivalente visual al código OWL introducido. Dicha acción encuentra soporte en las siguientes funcionalidades:

1. **Seleccionar archivo que contiene a la ontología:** esta funcionalidad le permite a los usuarios buscar donde se encuentra el archivo que contiene a la ontología en su ordenador y lo carga en el sistema para un posterior procesamiento. En la Figura 7 se evidencia dicho proceso.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

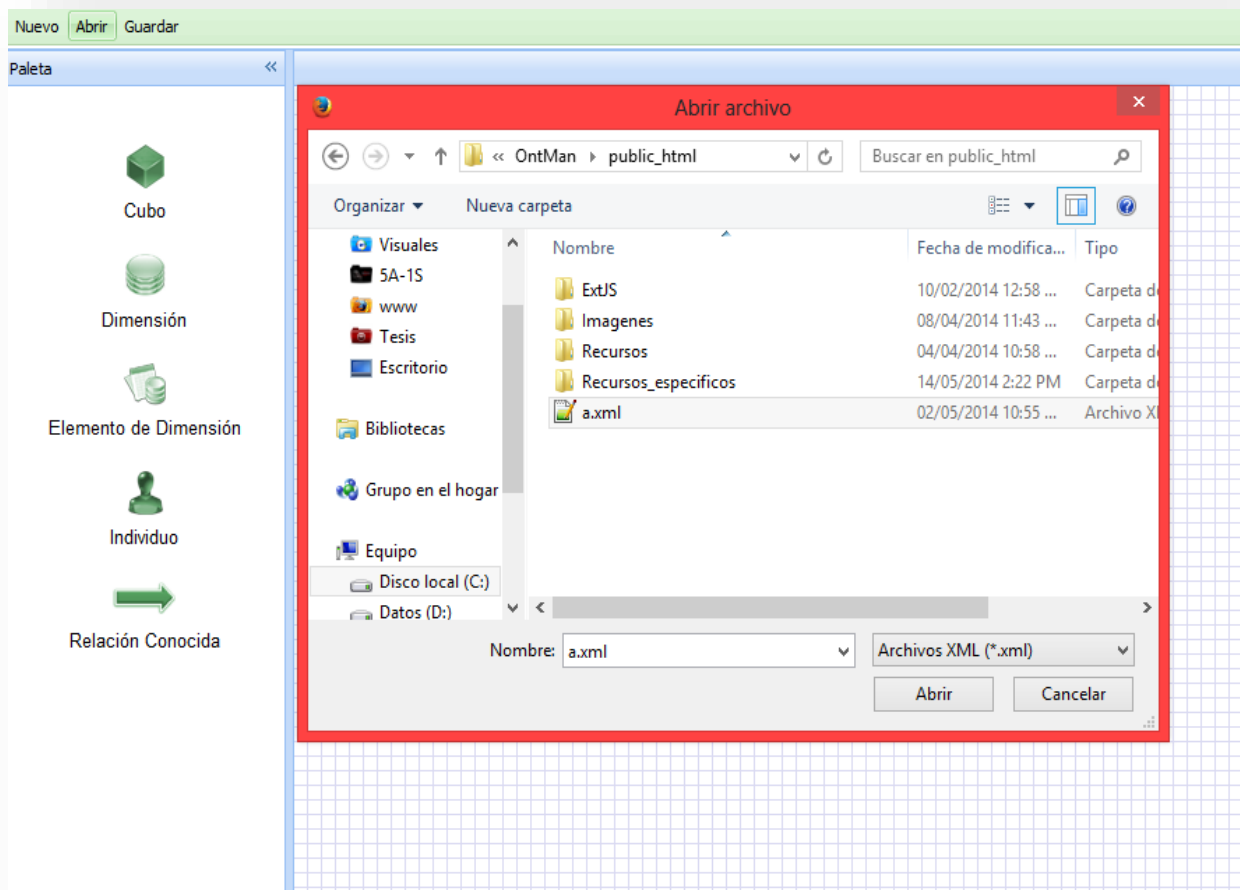


Figura 7. Selección de una ontología

- 2. Crear los elementos de la ontología cargados previamente:** permite al sistema crear los objetos que equivalen a los elementos de las ontologías que se encuentran en el archivo cargado (cubos, dimensiones, elementos de dimensión, individuos y relaciones que se establecen entre ellos).
- 3. Representar los elementos de la ontología introducida al sistema:** representa visualmente los objetos creados a partir del archivo introducido que contiene la ontología (Figura 8), mediante los nodos equivalentes a los elementos de la ontología que se definen en la HEOMM.

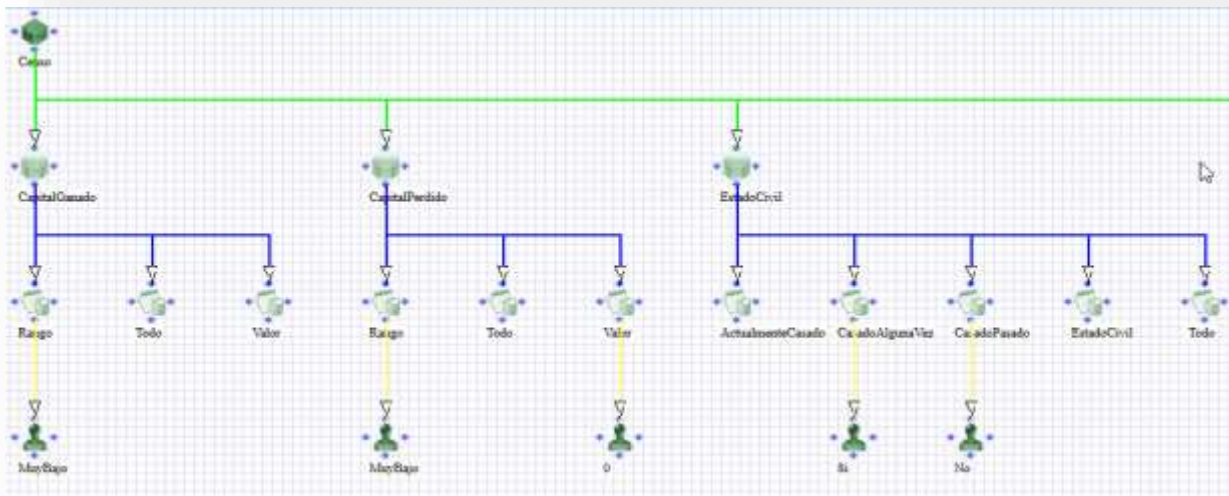


Figura 8. Representación visual de la ontología

2.4.2 Subsistema de gestión de la ontología

El subsistema de gestión de la ontología permite añadir, modificar o eliminar los nodos definidos en la HEOMM así como las relaciones que entre ellos se establecen. A continuación se desglosan:

1. **Adicionar nodo cubo:** permite añadir cubos al área de trabajo.
2. **Modificar nombre del nodo cubo:** otorga la posibilidad de modificar el nombre de los cubos representados (Figura 9). Ocurre de igual forma para los requisitos 7, 12 y 17.

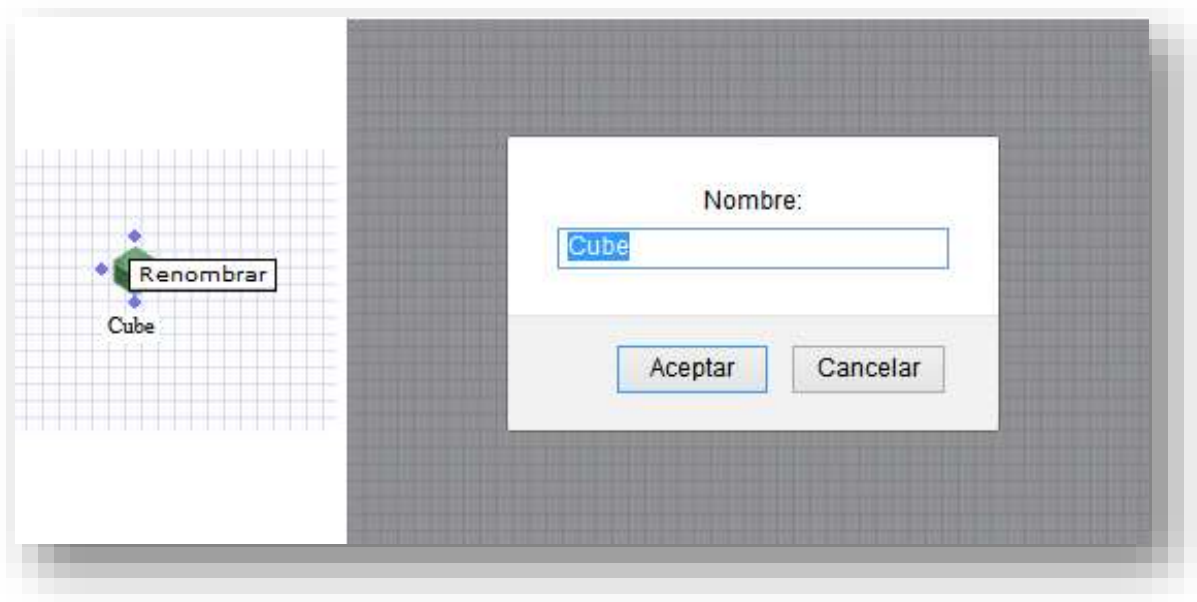


Figura 9. Cambiando el nombre del nodo de tipo cubo

3. **Modificar posición del nodo cubo:** permite modificar la posición de los cubos representados.
4. **Modificar visibilidad de los hijos del nodo cubo:** muestra u oculta las dimensiones asociadas al cubo seleccionado (Figura 10). Ocurre de igual forma para los requisitos 9, 14 y 19.

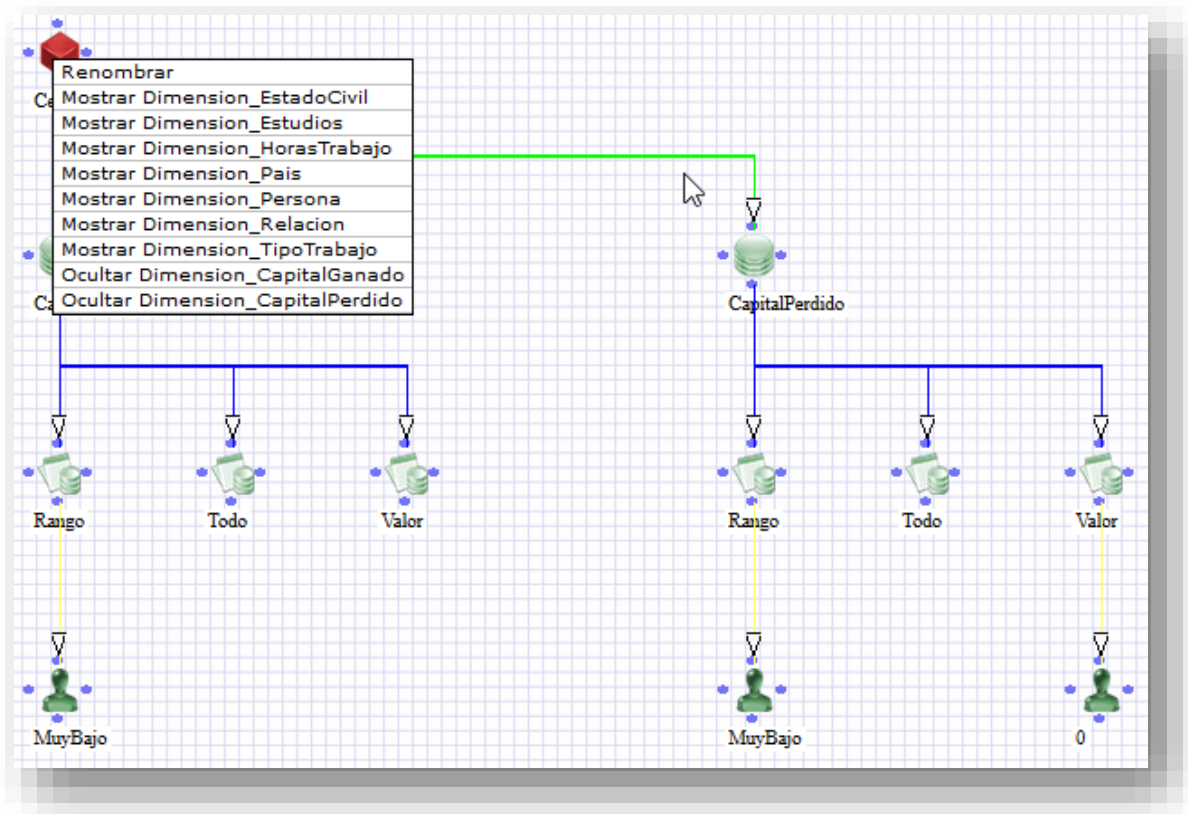


Figura 10. Ocultando o mostrando nodos hijos

5. **Eliminar nodo cubo:** elimina el cubo seleccionado.
6. **Adicionar nodo dimensión:** permite añadir dimensiones al área de trabajo.
7. **Modificar nombre del nodo dimensión:** otorga la posibilidad de modificar el nombre de las dimensiones representadas.
8. **Modificar posición del nodo dimensión:** permite modificar la posición de las dimensiones representadas.
9. **Modificar visibilidad de los hijos del nodo dimensión:** muestra u oculta los elementos de dimensión asociados a la dimensión seleccionado.
10. **Eliminar nodo dimensión:** elimina la dimensión seleccionada.
11. **Adicionar nodo elemento de dimensión:** permite añadir elementos de dimensión al área de trabajo.
12. **Modificar nombre del nodo elemento de dimensión:** otorga la posibilidad de modificar el nombre de los elementos de dimensión representados.

13. **Modificar posición del nodo elemento de dimensión:** permite modificar la posición de los elementos de dimensión representados.
14. **Modificar visibilidad de los hijos del nodo elemento de dimensión:** muestra u oculta los individuos asociados al elemento de dimensión seleccionado.
15. **Eliminar nodo elemento de dimensión:** elimina el elemento de dimensión seleccionado.
16. **Adicionar nodo individuo:** permite añadir individuos al área de trabajo, nodo que representa una instancia de un elemento de dimensión.
17. **Modificar nombre del nodo individuo:** otorga la posibilidad de modificar el nombre de los individuos representados.
18. **Modificar posición del nodo individuo:** permite modificar la posición de los individuos representados.
19. **Modificar visibilidad de los hijos del nodo individuo:** muestra u oculta las relaciones conocidas asociadas al individuo seleccionado.
20. **Eliminar nodo individuo:** elimina el individuo seleccionado.
21. **Adicionar nodo relación conocida:** permite añadir relaciones conocidas al área de trabajo, nodo especial que se define para representar las reglas de asociación.
22. **Modificar posición del nodo relación conocida:** permite modificar la posición de las relaciones conocidas representadas.
23. **Eliminar nodo relación conocida:** elimina la relación conocida seleccionada.
24. **Establecer conexiones entre los nodos:** se crea una línea conectora entre los nodos y la misma recibirá el nombre y color según dichos nodos (Figura 11).

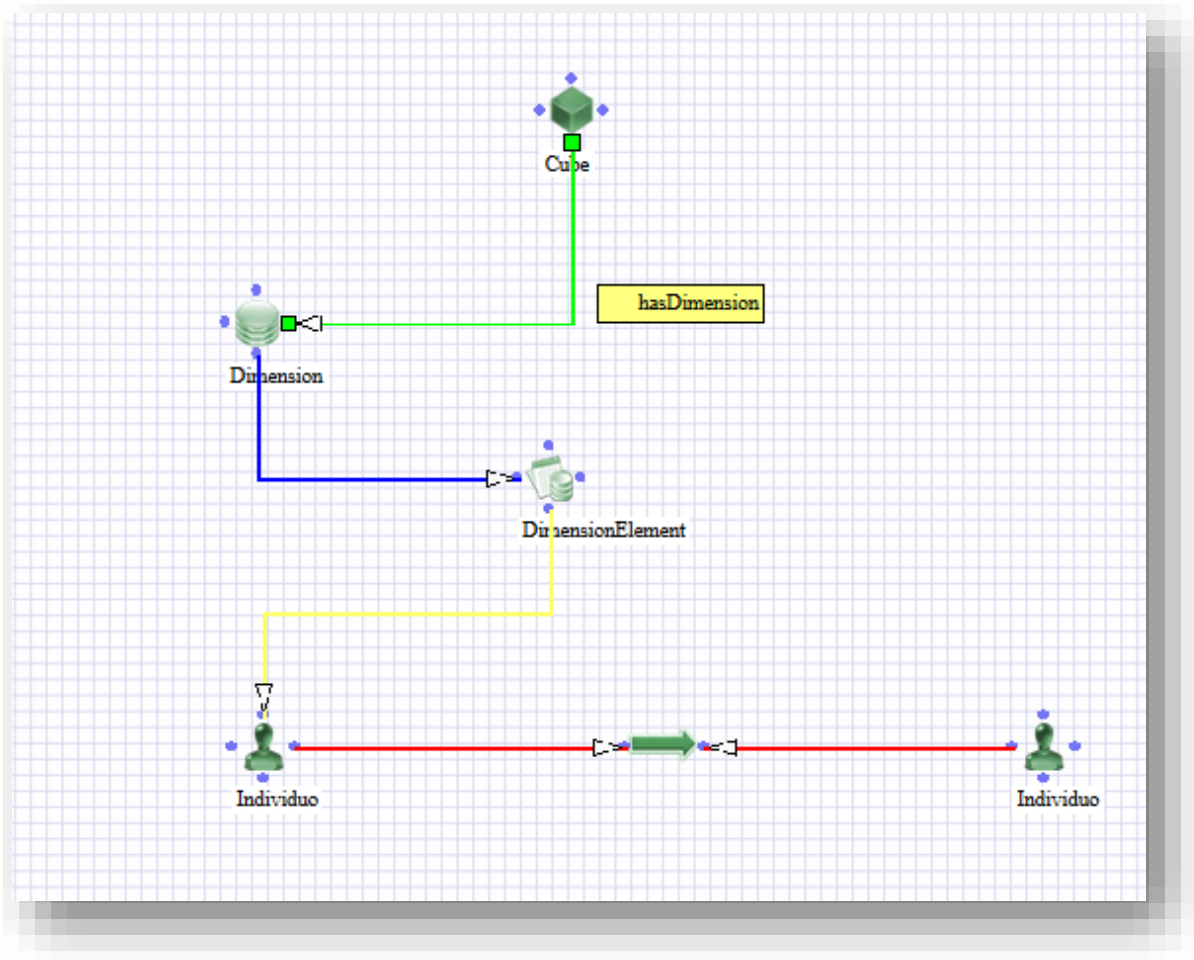


Figura 11. Estableciendo conexiones

La Tabla 2 representa todos los tipos de conexiones y los colores que cada una de ellas toma, o sea:

Tabla 2. Tabla de correspondencia de color según nombre de conexión

Nombre de conexión	Color
hasDimension	verde
hasDimensionElement	azul
hasIndividual	amarillo
hasRelacionConocida	rojo

25. Modificar visibilidad de las conexiones entre nodos: cambia a oculto o visible el estado de la conexión en dependencia de la visibilidad de los nodos asociados y su nivel jerárquico.

26. Eliminar conexiones entre los nodos: se suprimen las conexiones que se seleccionen.

2.4.3 Subsistema de guardado

Terminados todos los cambios que el usuario desea hacer sobre la ontología se hace necesario que los mismos sean persistentes para usos posteriores. En el área de trabajo estarán todos los elementos de la ontología, visibles o no, permitiendo la lectura de ellos y la escritura del código equivalente de cada uno, incluidas las relaciones establecidas. Dan soporte al subsistema las siguientes funcionalidades:

- 1. Leer la ontología representada en el área de trabajo:** se encarga de la lectura de todos los elementos que se han representado en el área de trabajo.
- 2. Crear código OWL equivalente a la ontología representada:** básicamente lee cada elemento de la ontología representada y convierte en el código OWL respectivo (Figura 12).

```
<ClassAssertion>
  <Class IRI="#ElementoDimension_Continente_Pais"/>
  <NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ElementoDimension_Trabajo_TipoTrabajo"/>
  <NamedIndividual IRI="#Private_Trabajo_TipoTrabajo"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ElementoDimension_CasadoAlgunaVez_EstadoCivil"/>
  <NamedIndividual IRI="#Si_CasadoAlgunaVez_EstadoCivil"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ElementoDimension_Raza_Persona"/>
  <NamedIndividual IRI="#White_Raza_Persona"/>
</ClassAssertion>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#Censo_hasDimension_CapitalGanado"/>
  <ObjectProperty IRI="#hasDimension"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#CapitalGanado_hasDimensionElement_Rango"/>
  <ObjectProperty IRI="#hasDimensionElement"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#CapitalGanado_hasDimensionElement_Todo"/>
  <ObjectProperty IRI="#hasDimensionElement"/>
</SubObjectPropertyOf>
```

Figura 12. Ejemplo de código que genera la HEOMM

3. **Exportar ontología representada:** esta funcionalidad se encarga de la creación del archivo y la exportación del mismo (Figura 13).

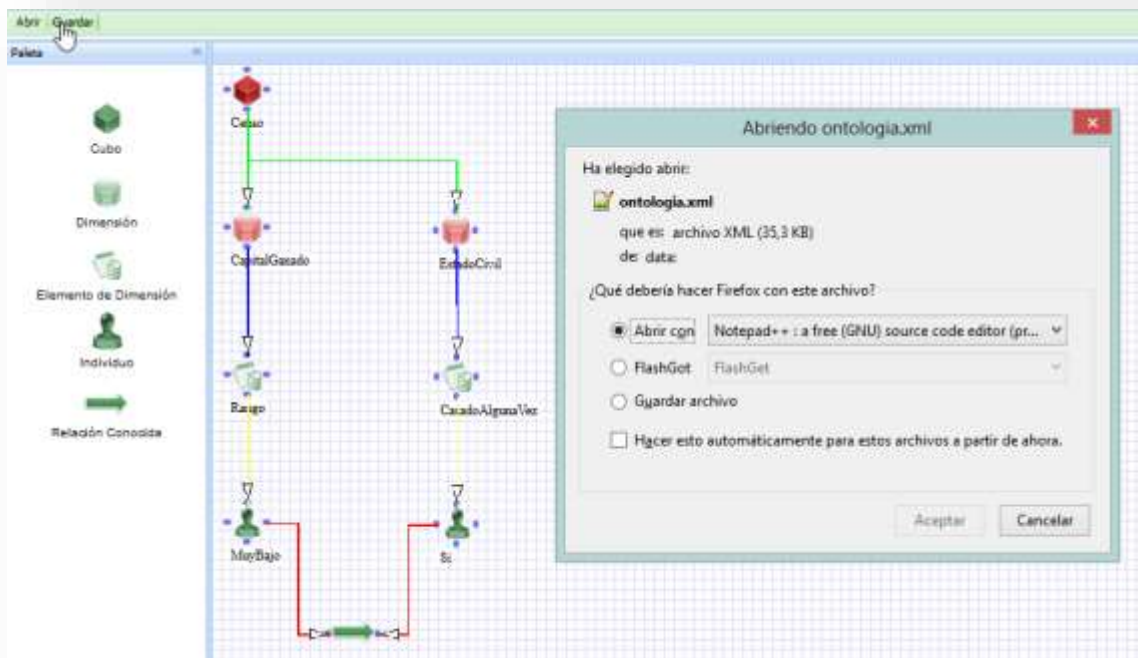


Figura 13. Guardando cambios realizados

2.5 Conclusiones parciales

Las ontologías “core” manejan la introducción de reglas de asociación a los modelos multidimensionales por lo que será su estructura la usada para la implementación de la HEOMM. La utilización de la metodología XP para el desarrollo se adecua perfectamente a las condiciones del equipo de desarrollo pues está compuesto por una persona, cuenta con poco tiempo para la implementación y el constante intercambio con el cliente. El uso del patrón arquitectónico de 4-Capas ha permitido que el desarrollo se haga de forma organizada, logrando un mejor aprovechamiento de los recursos, dando como resultado las capas: de datos, de acceso a datos, controladores y presentación. El API de Draw2D contribuye de manera significativa al proceso de desarrollo, por la existencia de ejemplos con características similares.

El funcionamiento de la aplicación está basado en la lectura de un archivo que contiene la ontología, la representación de la misma, así como la edición de los elementos ontológicos contenidos, guardado y obtención del código equivalente a la representación gráfica. La HEOMM brinda varias ventajas pues los usuarios que harán uso de ella no necesariamente tienen que

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

poseer un alto nivel de conocimiento sobre lógica descriptiva. Permite además, introducir el conocimiento de los especialistas a los modelos multidimensionales representados.

3 CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Existen diferentes patrones o métodos que permiten evaluar y validar los resultados alcanzados en una investigación. Se componen de la demostración, experimentación, simulación, uso de métricas, evaluación comparativa, razonamiento lógico y los modelos matemáticos (Vaishnavi, y otros, 2008). A continuación se enuncian algunos de estos métodos.

3.1 Métodos de validación

- Demostración (Vaishnavi, y otros, 2008).
- Experimentación. (Sampieri, y otros, 1991)
- Simulación. (Vaishnavi, y otros, 2008).
- Razonamiento lógico. (Vaishnavi, y otros, 2008).
- Modelos matemáticos (Vaishnavi, y otros, 2008).

Se puede decir que el razonamiento lógico depende en gran medida de la precisión de sus argumentos y suposiciones a pesar de su certeza. Las pruebas matemáticas constituyen una de las formas más precisa de validación. La experimentación y simulación son útiles cuando el problema es complejo y es inviable efectuar una demostración matemática. El uso de métricas y la evaluación comparativa, mecanismos para cuantificar afirmaciones respecto a una solución, son generalmente útiles cuando se utilizan en la experimentación y la simulación.

La demostración intenta mostrar que la solución es factible y válida para una o varias situaciones predefinidas. Es especialmente relevante cuando la demostración de una solución en sí misma se considera una contribución. Consta de dos momentos importantes, construir la solución o prototipo de solución que demuestre que esta es factible y demostrar que la solución construida es razonable para un conjunto predefinido de situaciones (Vaishnavi, y otros, 2008). La demostración muestra las deficiencias de la solución y también que la solución es viable y aceptable. Si las situaciones de prueba están diseñadas apropiadamente, entonces la construcción de la solución y sus pruebas para estas situaciones pueden demostrar su validez, a pesar de que teóricamente constituyen el patrón de validación menos formal o débil. Ha sido utilizada en innumerables artículos científicos de la especialidad hasta el punto de ser el tipo de validación que más se utiliza desde el año 1999 (Shaw, 2002).

Las especificidades del problema estudiado y las pautas aceptadas por la comunidad de investigadores como elecciones correctas de comprobación en la temática son factores claves a la hora de hacer uso de uno o varios métodos de validación. La presente investigación utiliza el método de “**Demostración**” como mecanismo de validación. Siendo ideal para corroborar su validez pues basta con comprobar las salidas obtenidas con las esperadas para obtener una respuesta positiva o negativa. La demostración en sí misma tiene valor práctico al estar compuesta por un prototipo funcional que permite construir modelos descriptivos aun cuando es necesario perfeccionar algunos detalles de usabilidad en dicho prototipo.

3.2 Recursos computacionales utilizados para la demostración

Los recursos computacionales en esta investigación se refieren a las características de hardware y software de la computadora utilizada en el proceso representación visual de ontologías del modelo multidimensional. Para un funcionamiento correcto de la aplicación basta con un ordenador con procesador Dual Core, una memoria RAM con capacidad mayor a 512 MB y no hace uso de una plataforma específica por lo que puede ser usado en cualquier sistema operativo.

Para efectuar el proceso de “Demostración” se utilizó una computadora de ASUSTek Computer Inc. Este ordenador cuenta con una motherboard Intel(R) Pentium(R) que incorpora un procesador CPU B960 a 2.2 GHz y memoria RAM DDR3 con 4 GB de capacidad. Los algoritmos propuestos fueron probados en una plataforma Microsoft Windows 8 Pro 64 bits (6.2, compilación 9200).

3.3 Conjuntos de datos utilizados y resultados obtenidos

Para mostrar la aplicación del modelo se desarrolla un ejemplo basado en el cubo de datos con información del censo presentado en (Molina, y otros, 2006) y en el de la bolsa de (Nuñez, 2007). La estructura del cubo de datos con información del censo es presentado en la Figura 14 y la de la bolsa en la Figura 15.

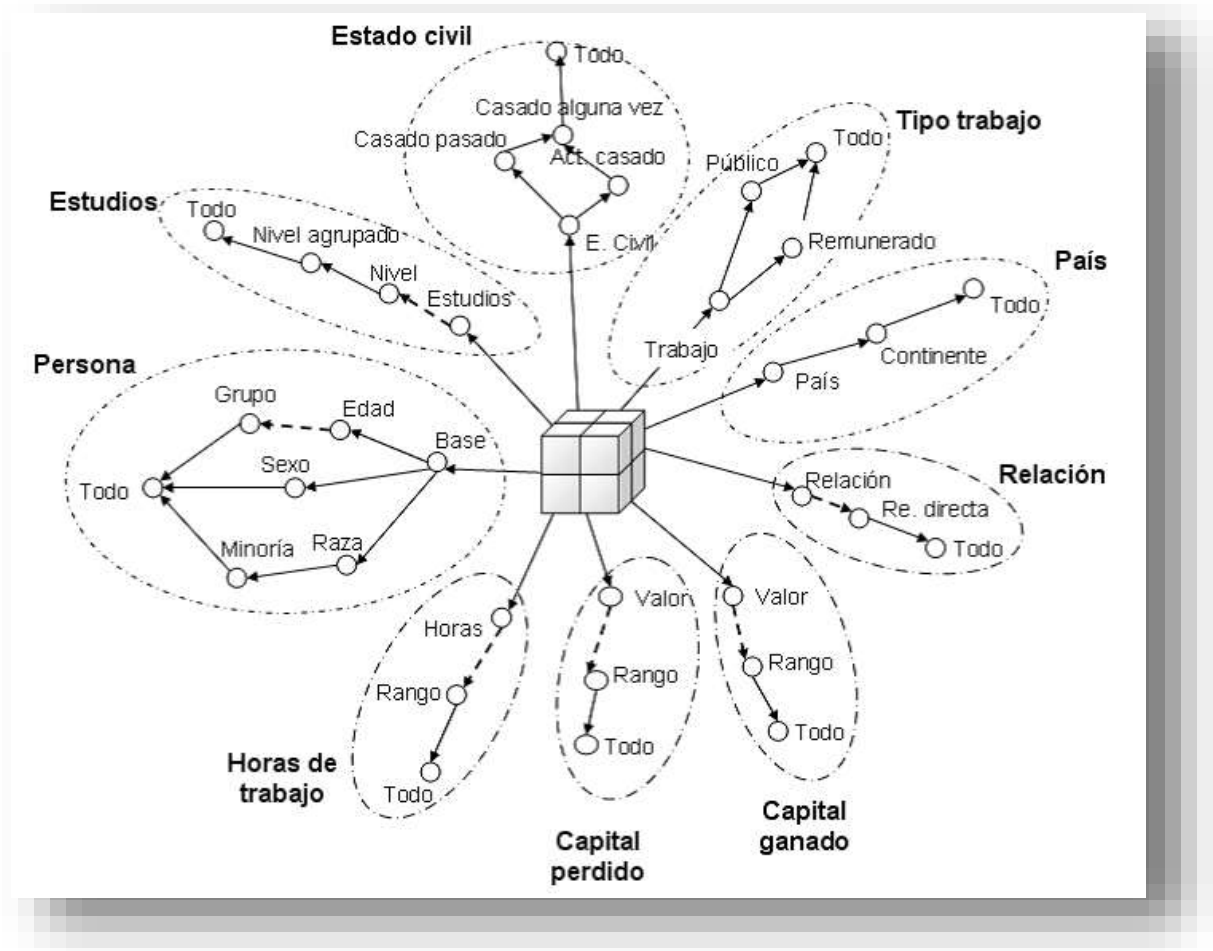


Figura 14. Cubo de datos con información del censo

Puede observarse que el cubo de datos está formado por nueve dimensiones que a su vez las conforman elementos de dimensión que en total son 35 y además existen individuos para los elementos de dimensión, ya que son los posibles valores que pueden tomar.

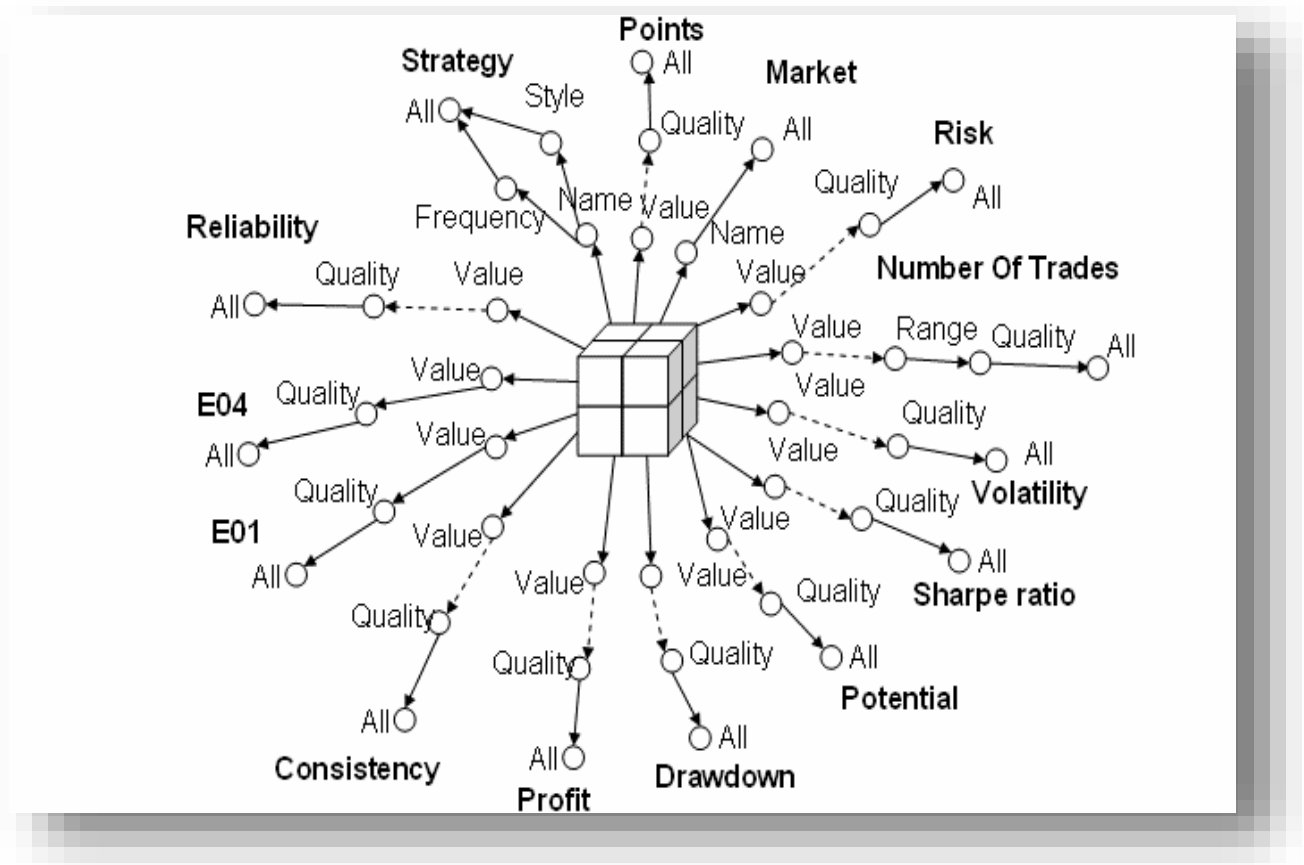


Figura 15. Cubo de datos con información de la bolsa

Este cubo de datos está formado por 14 dimensiones que a su vez las conforman elementos de dimensión que en total son 43 y como en el anterior cada elemento de dimensión posee instancias (individuos).

Se carga en la herramienta la ontología del modelo multidimensional que se usará, basada en los cubos de datos presentados. A continuación en la Figura 16 se muestra cómo queda la representación de la ontología.

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

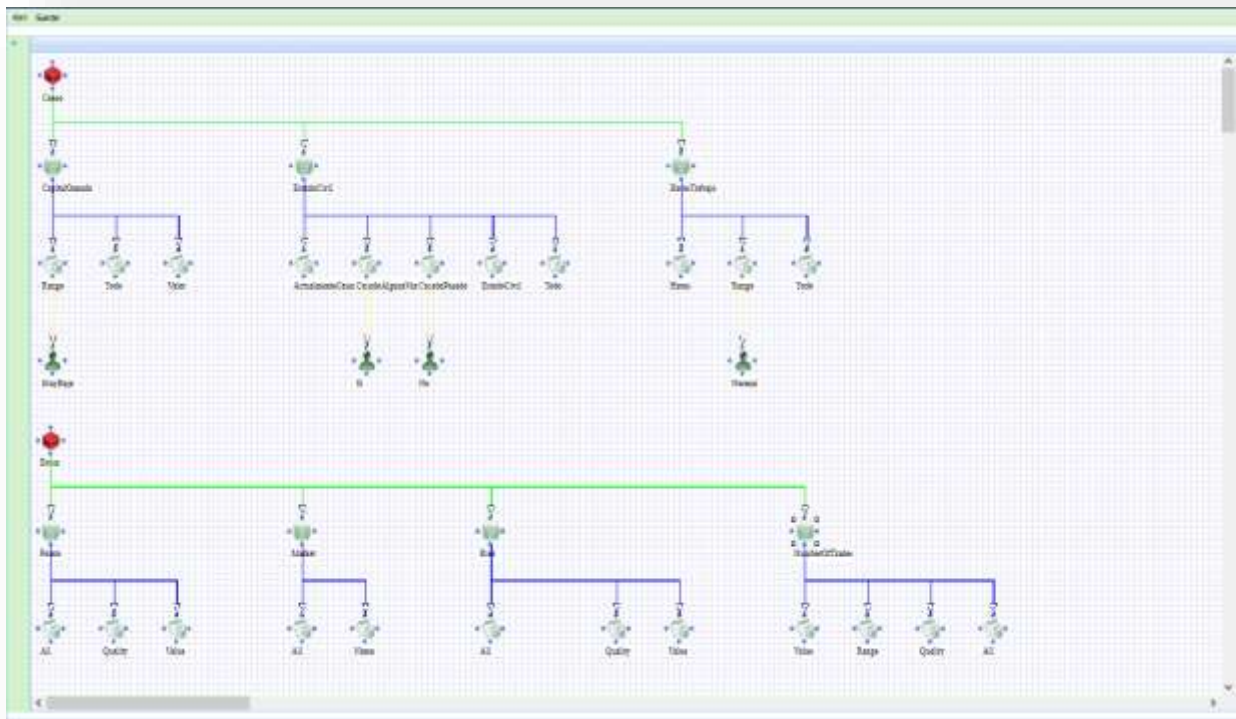


Figura 16. Ontología del modelo multidimensional correspondiente a los cubos de datos: Censo y Bolsa

Luego se introducen reglas de asociaciones obtenidas del almacén de datos Censo mediante un algoritmo de extracción:

$\{[Pais]. [Continente]. [Norte America]\} \rightarrow \{[Persona]. [Minoria]. [No]\}$ Sup: 0.72457457 CF: 0.14969045

$\{[Persona]. [Minoria]. [No]\} \rightarrow \{[Pais]. [Continente]. [Norte America]\}$ Sup: 0.72457457 CF: 0.21771474

$\{[CapitalGanado]. [Rango]. [Muy bajo], [Pais]. [Continente]. [Norte America]\} \rightarrow \{[EstadoCivil]. [Casado en el pasado]. [No]\}$ Sup: 0.5547108 CF: -0.034695998

$\{[Pais]. [Continente]. [Norte America]\} \rightarrow \{[EstadoCivil]. [Casado en el pasado]. [No], [CapitalGanado]. [Rango]. [1. Muy bajo]\}$ Sup: 0.5547108 CF: -0.029112488

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

{[CapitalGanado]. [Rango]. [1. Muy bajo]} -> {[EstadoCivil]. [Casado en el pasado]. [No], [Pais]. [Continente]. [Norte America]} Sup: 0.5547108 CF: -0.024011334

Mediante la herramienta se obtiene una representación para cada regla como la que se muestra en la Figura 17 y un código OWL como en la Figura 18 y 19.

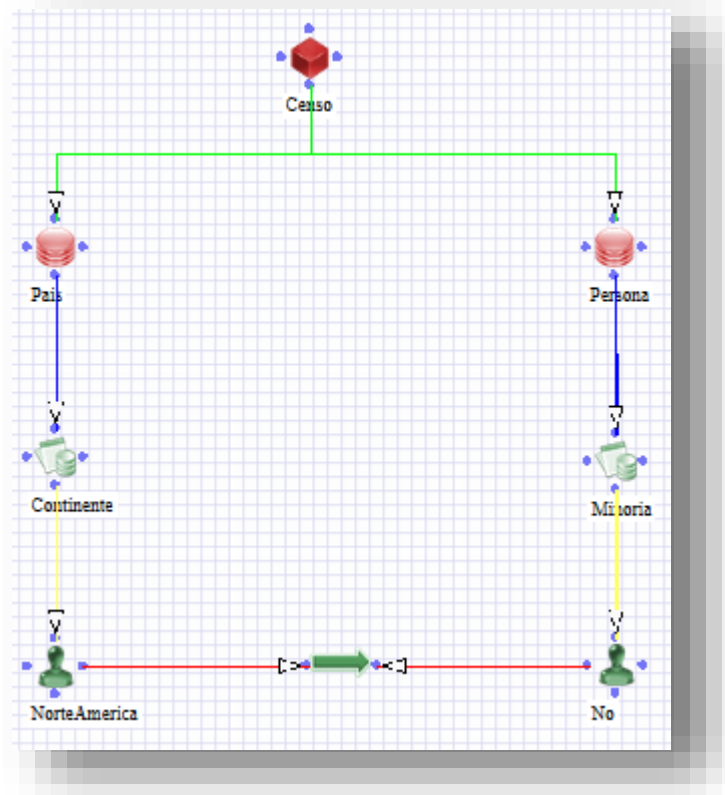


Figura 17. Representación visual de la primera regla mostrada

HERRAMIENTA PARA LA EDICIÓN DE ONTOLOGÍAS DEL MODELO MULTIDIMENSIONAL

```
<ObjectPropertyDomainAssertion>
  <ObjectProperty IRI="#NorteAmerica_Continente_Pais_relacionConocida_No_Minoria_Persona"/>
  <NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ObjectPropertyDomainAssertion>
<ObjectPropertyDomainAssertion>
  <ObjectProperty IRI="#MuyBajo_Rango_CapitalGanado_relacionConocida_No_CasadoPasado_EstadoCivil_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#MuyBajo_Rango_CapitalGanado"/>
</ObjectPropertyDomainAssertion>
<ObjectPropertyDomainAssertion>
  <ObjectProperty IRI="#0_Valor_CapitalPerdido_relacionConocida_NorteAmerica_Continente_Pais_No_CasadoPasado_EstadoCivil_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#0_Valor_CapitalPerdido"/>
</ObjectPropertyDomainAssertion>
<ObjectPropertyDomainAssertion>
  <ObjectProperty IRI="#No_Minoria_Persona_relacionConocida_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#No_Minoria_Persona"/>
</ObjectPropertyDomainAssertion>
<ObjectPropertyDomainAssertion>
  <ObjectProperty IRI="#NorteAmerica_Continente_Pais_relacionConocida_No_CasadoPasado_EstadoCivil_MuyBajo_Rango_CapitalGanado"/>
  <NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ObjectPropertyDomainAssertion>
<ObjectPropertyRangeAssertion>
  <ObjectProperty IRI="#NorteAmerica_Continente_Pais_relacionConocida_No_Minoria_Persona"/>
  <NamedIndividual IRI="#No_Minoria_Persona"/>
</ObjectPropertyRangeAssertion>
```

Figura 18. Código equivalente a las reglas de asociación introducidas. Parte 1


```
<ObjectPropertyRangeAssertion>
  <ObjectProperty IRI="#MuyBajo_Rango_CapitalGanado_relacionConocida_No_CasadoPasado_EstadoCivil_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#No_CasadoPasado_EstadoCivil"/>
<NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ObjectPropertyRangeAssertion>
<ObjectPropertyRangeAssertion>
  <ObjectProperty IRI="#0_Valor_CapitalPerdido_relacionConocida_NorteAmerica_Continente_Pais_No_CasadoPasado_EstadoCivil_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
<NamedIndividual IRI="#No_CasadoPasado_EstadoCivil"/>
<NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ObjectPropertyRangeAssertion>
<ObjectPropertyRangeAssertion>
  <ObjectProperty IRI="#No_Minoria_Persona_relacionConocida_NorteAmerica_Continente_Pais"/>
  <NamedIndividual IRI="#NorteAmerica_Continente_Pais"/>
</ObjectPropertyRangeAssertion>
<ObjectPropertyRangeAssertion>
  <ObjectProperty IRI="#NorteAmerica_Continente_Pais_relacionConocida_No_CasadoPasado_EstadoCivil_MuyBajo_Rango_CapitalGanado"/>
  <NamedIndividual IRI="#No_CasadoPasado_EstadoCivil"/>
<NamedIndividual IRI="#MuyBajo_Rango_CapitalGanado"/>
</ObjectPropertyRangeAssertion>
```

Figura 19. Código equivalente a las reglas de asociación introducidas. Parte 2

3.4 Conclusiones parciales

Usando como conjuntos de datos de entrada ontologías basadas en los cubos de datos Censo y Bolsa se demuestra que para las personas es más sencillo manejar la representación visual de una ontología que el código OWL asociada a la misma. Esto ocurre debido a que para la gestión del lenguaje cada acción que se realice repercute en varios cambios. Por lo que la representación tiene la ventaja de que la persona solo tiene que encargarse de crear el concepto y establecer sus relaciones, siendo inherente a ella el código generado por las acciones. También se tiene en cuenta la introducción de conocimiento previo en forma de reglas de asociación, ya que se introduce un mecanismo para su representación en forma visual y en código OWL.

4 CONCLUSIONES

En el presente trabajo, después de recopilar, seleccionar y analizar diferentes fuentes bibliográficas, se eligieron definiciones para los conceptos de autores reconocidos: Inmon para almacén de datos, Triantaphyllou para minería de datos, Agrawal para reglas de asociación y Studer para ontología. Permitiendo establecer una base teórica que facilita el entendimiento de la investigación. Se constata además la evolución y el uso de esquemas conceptuales para representar conocimiento.

Se propusieron cinco equivalentes gráficos para los elementos que conforman la ontología del modelo multidimensional. Los mismos posibilitaron la construcción de los artefactos necesarios para plasmar sus representaciones.

Se implementó una aplicación web que facilita la edición de ontologías de dominio específico, creadas a partir de cubos de datos. Ofreciéndole a los especialistas del negocio una forma de interactuar de manera intuitiva, logrando incorporar el conocimiento dentro de los cubos. Con ello se satisface el objetivo propuesto para esta investigación.

5 RECOMENDACIONES

Para el mejoramiento de este trabajo es importante optimizar un grupo de elementos que no fueron tomados en cuenta por cuestiones de tiempo. Dentro de ellos se señalan los siguientes, debido a su importancia:

1. Crear un mecanismo para la obtención de una ontología en lenguaje OWL a partir de un almacén de datos o integrar de existir alguno.
2. Representar en forma matricial las reglas de asociación que se introduzcan en forma de conocimiento previo.

6 REFERENCIAS

Adobe Systems Incorporated. Photoshop.com. [En línea] <http://www.photoshop.com/>.

Agrawal, R. y Srikant, R. 1994. 1994. Proc. 20th Int. Conf. Very Large Data Bases. págs. 487–499.

Agrawal, R., Imielinski, T. y Swami, A. 1993. *Mining association rule between sets of items in large databases.* En Proceedings of ACM SIGMOD. 1993. págs. 207-216.

ALEGSA. 1998. Diccionario de Informática. [En línea] 1998.
<http://www.alegsa.com.ar/Dic/framework.php>.

Antoniou, G. y van Harmelen, F. 2004. *A Semantic Web Primer.* London : s.n., 2004.

Arquitecturas. 2012. GUIA DE ARQUITECTURA EN N-CAPAS. *GUIA DE ARQUITECTURA EN N-CAPAS.* [En línea] 2012. <http://arquitecturancapas.blogspot.com/>.

Balcazar, J.L. 2010. *Redundancy, Deduction Schemes, and Minimum-Size Bases for Association Rules.* 2010.

Baralis, E., y otros. 2009. 2009, Springer, págs. 50–57. Knowledge-Based and Intelligent Information and Engineering Systems. Context-aware user and service profiling by means of generalized association rules.

Baranovskiy, D. 2008. Raphaël-JavaScriptLibrary. [En línea] 2008. <http://raphaeljs.com/>.

Barron, R. F. y Stone, V.F. 1974. *The effects of student-constructed graphic post-organizers upon learnin vocabulary relationships.* 1974.

Beck, K. y Cynthia, A. 2004. *Extreme programming explained: embrace change.* s.l. : Addison-Wesley Professional, 2004.

Berrado, A. y Runger, G.C. 2007. *Using metarules to organize and group discovered association rules.* 2007. págs. 409–431.

Berry, L. 2000. *Mastering Data Mining.* Wiley : s.n., 2000.

Bonchi, F. y Lucchese, C. 2004. 2004. Fourth IEEE International Conference on Data Mining. págs. 35–42.

Buzan, T. 1993. *Cómo utilizar su mente.* Bilbao : Destuo, 1993.

Canós, J. H., Letelier, P. y Penadés, C. 2003. *"Metodologías Ágiles en el desarrollo de Software."* Valencia : Universidad Politécnica de Valencia, 2003.

Cañas, A. J. y Novak, J. D. 2009. *¿Qué es un mapa conceptual?* Institute for Human and Machine Cognition. 2009.

Carbonell, J. 1990. *Machine Learning - Paradigms and Methods.* The MIT Press. Cambridge, Massachusetts : s.n., 1990.

Cobos, J. A. y Tufiño, L. J. 2010. *TÉCNICAS DE APRENDIZAJE VISUAL PARA DESARROLLAR APRENDIZAJES SIGNIFICATIVOS EN LOS ESTUDIANTES.* 2010.

Concari, S. B. 2001. *Las teorías y modelos en la explicación científica: implicancias para la enseñanza de las ciencias.* 2001.

Daconta, M. C., Obrst, L. J. y Smith, K. T. 2003. *The Semantic Web. A guide to the future of XML, Web Services, and Knowledge Management.* Indianapolis : s.n., 2003.

De Carvalho, V.O., Rezende, S.O. y Castro, M. 2007. 2007. Enterprise Information Systems. págs. 310–315.

Díaz, J. C. 2013. *Ontología del modelo multidimensional.* 2013.

ECMA. 2011. [En línea] 2011. <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>.

Fayyad, U. M., y otros. 1996. *Advances in Knowledge Discovery and Data Mining*. 1996.

Freeman, J. A. y Skapura, D. M. 1993. *Redes Neuronales- Algoritmos, aplicaciones y técnicas de programación*. 1993.

García, J. 2009. *Ext JS in Action*. s.l. : Manning Publications, 2009.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1989.

Gruber, T. R. 1993. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". Knowledge Systems Laboratory, Stanford University. CA : s.n., 1993. Technical Report.

Guarino, N. 1998. *Formal Ontology in Information Systems*. Trento, Italia : s.n., 1998. págs. 3-15.

Han, J. 1997. *OLAP mining: An integration of OLAP with data mining*. 1997. págs. 1–9.

Heimlich, J. y Pittelman, S. 2001. *Elaboración de mapas semánticos como Estrategia de aprendizaje*. México : s.n., 2001.

Hendler, J. 2008. *Agents and the Semantic Web*. s.l. : IEEE Intelligent Systems, 2008. págs. 30-37. 2.

Hernández, R. A. y Coello, S. 2011. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana : Editorial Universitaria, 2011. ISBN 978-959-16-1307-3.

Herz, A. 2014. Draw2D touch. [En línea] 2014. <http://www.draw2d.org/draw2d/>.

Horrocks, I., Patel-Schneider, P. F. y van Harmelen, F. 2003. *From SHIQ and RDF to OWL: The Making of a Web Ontology Language*. Department of Computer Science, University of Manchester. Manchester : s.n., 2003.

Inmon, B. 2002. *Building the Data Warehouse*. New Jersey : John Wiley & Sons, 2002.

Kodratoff, Y. y Michalski, R. 1990. *Machine Learning - An Artificial Intelligence Approach*. San Mateo, California : Morgan Kaufmann Publishers, Inc., 1990.

Lamarca, M. J. 2007. *HIPERTEXTO: EL NUEVO CONCEPTO DE DOCUMENTO EN LA CULTURA DE LA IMAGEN*. Dpto. de Biblioteconomía y Documentación, Universidad Complutense de Madrid. . 2007. Tesis doctoral.

Larrañaga, P., y otros. 1999. "A Review of the Cooperation between Evolutionary Computation and Probabilistic Graphical Models", . 1999. Proceedings of 2nd International Symposium on Artificial Intelligence, ISAS'99.

Lee, A.J.T., Lin, W. y Wang, C. 2006. 2006, Journal of Systems and Software, págs. 79–92.

Li, J., y otros. 2009. *Enhancing rule importance measure using concept hierarchy*. Faculty of Computer Science and Engineering, York University. 2009.

Marin, N., y otros. 2008. *A Complexity Guided Algorithm for Association Rule Extraction on Fuzzy DataCubes*. *IEEE Transactions on Fuzzy Systems*. 2008. págs. 693–714.

Mayer, R. E. 1989. *Models for understanding*. *Review of Educational Research*. 1989.

McGuinness, D. y Van Harmelen, F. 2004. OWL Web Ontology Language Overview. [En línea] 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3>.

Medina, P. y José, E. 2007. *Generación de conjuntos de items y reglas de asociación*. Dpto. Minería de Datos, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV). La Habana : s.n., 2007.

Molina, C. 2005. *Imprecisión e Incertidumbre en el Modelo Multidimensional: Aplicación a la Minería de Datos.* Dpto. de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada. Granada : s.n., 2005. Tesis Doctoral.

Molina, C., y otros. 2006. *A New Fuzzy Multidimensional Model.* *IEEE Transactions on Fuzzy Systems.* 2006. págs. 14, 897–912.

Moreira, R. 2002. Los mapas cognitivos. [En línea] 2002.
<http://members.fourtunecity.com/teoría1/imagen.htm>.

Morgade, G. 2009. *ClioBD. Sistema de control de versiones para bases de datos.* La Habana : Universidad de las Ciencias Informáticas, 2009.

Mozilla. 1998. Navegador Firefox. [En línea] 1998. <http://www.mozilla.org/es-ES/firefox/desktop/>.

Norman, D. 2002. *Emotional Design: Why We Love (or Hate) Everyday Things.* 2002.

Novak, J. D. 1998. *Conocimientos y Aprendizaje: Los Mapas Conceptuales como herramientas facilitadoras para escuelas y empresas.* Madrid : Alianza Editorial, 1998.

Nuñez, J. F. 2007. *Empleo de Fuzzy OLAP para Obtener Reglas que Caractericen Estrategias de Inversión.* 2007.

OLAP Council. OLAP Council. *OLAP Council.* [En línea] <http://www.olapcouncil.org>.

Oracle Corporation. NetBeans IDE. [En línea] <https://netbeans.org/>.

Oracle. 1993. Graphics2D. [En línea] 1993.
<http://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html>.

Otero, M. R. 1999. *Psicología Cognitiva, representaciones mentales e investigación en enseñanza de las Ciencias.* s.l. : Investigaciones en Enseñanza de Ciencias, 1999.

OWL Working Group. 2009. W3C. [En línea] 2009. <http://www.w3.org/2007/OWL>.

Pimienta, J. 2013. *Estrategias para aprender a aprender*. 2013.

Pressman, R. *Ingeniería de software. Un enfoque práctico*. s.l. : McGraw-Hill.

Riverón, K. 2013. *Sistema basado en casos para apoyar el proceso de toma de decisiones en las organizaciones frente a una mejora de procesos de software*. LaHabana : Universidad de las Ciencias Informáticas, 2013.

Robins, A. y Hayes, A. 1997. Una valoración de la terapia cognitiva. *Psicoterapias cognitivistas y constructivistas*. Desclée de Brower, Bilbao : s.n., 1997, pág. 64.

Sampieri, C. R., Fernández, C. y Baptista, P. 1991. *Metodología de la investigación*. MÉXICO : McGRAW - HILL INTERAMERICANA DE MÉXICO, 1991.

Sánchez, S. E. 2007. *Modelo de indexación de formas en sistemas VIR basado en ontologías*. Departamento de Computación, Electrónica y Mecatrónica , Escuela de Ingeniería y Ciencias. Cholula, Puebla, México : s.n., 2007. Tesis de Maestría.

Shaw, M. 2002. *What makes good research in software engineering*. SPRINGER BERLIN / HEIDELBERG : FOR TECHNOLOGY TRANSFER (STTT), 2002.

Sriphaew, K. y Theeramunkong, T. 2002. 2002. Computers and Communications. Proceedings. págs. 1040–1045.

Studer, R., Benjamins, R. y Fensel, D. 1998. *Knowledge engineering: Principles and methods. Data & Knowledge Engineering*. 1998.

Thinking. 2012. Thinking Maps » Designs for Thinking. [En línea] 2012. http://dft.designsforthinking.com/?page_id=17.

TiposDe.Org. 2014. TiposDe.Org. [En línea] 2014. <http://www.tiposde.org/escolares/196-tipos-de-mapas-conceptuales/>.

Triantaphyllou, E. 2010. *Data Mining and Knowledge Discovery via Logic-Based Methods.* 2010.

Vaishnavi, V. K. y Kuechler Jr., W. 2008. *Design Science Research Methods and Patterns Innovating Information and Communication Technology.* NewYork : Auerbach Publications is an imprint of the , an informa business, 2008.

Zeng Lei, M. y Chan, L. M. 2004. *Trends and issues in establishing interoperability among knowledge organization systems, en Journal of the American Society for Information Science and Technology.* 2004. págs. 377-395. 5.

Zhu, H. 1998. *On-Line Analytical Mining of Association.* Tesis doctoral. Simon Fraser University. 1998.