

INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN
COMPUTACION

ALINEACIÓN DE ONTOLOGÍAS PARA
LA INTEGRACIÓN DE FUENTES DE
DATOS HETEROGÉNEAS

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

MARIO IVÁN MARTÍNEZ IBARRA

DIRECTOR DE TESIS:

DR. MIGUEL JESÚS TORRES RUIZ

DR. ROLANDO QUINTERO TÉLLEZ



MÉXICO D.F. NOVIEMBRE 2012



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 9:00 horas del día 26 del mes de noviembre de 2012 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Alineación de ontologías para la integración de fuentes de datos heterogéneas"

Presentada por el alumno:

MARTÍNEZ

Apellido paterno

IBARRA

Apellido materno

MARIO IVÁN

Nombre(s)

Con registro:

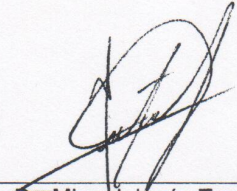
B	1	0	1	6	6	0
---	---	---	---	---	---	---

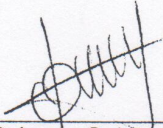
aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.


LA COMISIÓN REVISORA

Directores de Tesis

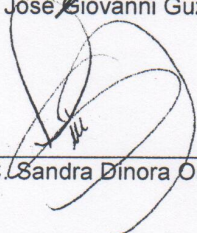

Dr. Miguel Jesús Torres Ruiz


Dr. Rolando Quintero Téllez

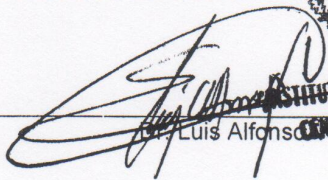

Dr. Marco Antonio Moreno Ibarra


Dr. José Giovanni Guzmán Lugo


M. en C. Roberto Eswart Zagal Flores


M. en C. Sandra Dinora Orantes Jiménez

PRESIDENTE DEL COLEGIO DE PROFESORES


Luis Alfonso Contreras Vargas
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 26 del mes de Noviembre del año 2012, el (la) que suscribe Mario Iván Martínez Ibarra alumno(a) del Programa de Maestría en Ciencias de la Computación, con número de registro B101660, adscrito(a) al Centro de Investigación en Computación, manifiesto(a) que es el (la) autor(a) intelectual del presente trabajo de Tesis bajo la dirección del (de la, de los) Dr. Miguel Jesús Torres Ruiz y Dr. Rolando Quintero Téllez y cede los derechos del trabajo titulado “Alineación de ontologías para la integración de fuentes de datos heterogéneas”, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del (de la) autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones: CIC-IPN, Unida Profesional “Adolfo López Mateos”, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, Delegación Gustavo A. Madero, CP 07738. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.


Mario Iván Martínez Ibarra.

Abstract

The increasing amount of data has led advances in computer technology and information, have produced a lot of ways of seeing and representing this information as well as data that represent them, so interoperability between systems is very difficult to achieve.

The semantic web has emerged as a solution to the problem of finding a meaning to the large amount of data used on the web, recently these techniques have claimed related to this research strength within one of these areas it occupies within the semantic web artificial intelligence are ontologies, a powerful tool which allows to represent the real world as closely as possible.

With these approaches can be developed to solve many problems, and one of them is data integration, which is a growing problem today where classical approaches applied to databases appear insufficient and lack features that ontologies can offer.

In this paper the problem of information integration is to resolve with the use of ontologies and alignment techniques, integrating two data sources, in the particular case in this paper uses databases. It is considered that these ontologies are obtained by transforming a model database, including Entity-Relationship model, using rules raised in this work to achieve these models represent in an ontology. For the alignment of ontologies apply semantic similarities to find possible similarities between ontology elements, and from a query engine, retrieve data from various sources transparently to the user (integration).

It presents a case study for the methodology in general and specific cases for the various stages that make up this.

Resumen

La creciente cantidad de datos que han provocado los avances en las tecnologías de la computación e información, han producido una gran cantidad de formas de ver y representar esta información, así como los datos que las representan, por lo cual la interoperabilidad entre los sistemas es muy complicada de lograr.

La web semántica surgió como una solución al problema de encontrarle un significado a la gran cantidad de datos que se manejan en la web, recientemente estas técnicas relacionadas a ésta han cobrado fuerza dentro de la investigación, uno de estos campos que ocupa la web semántica dentro de la Inteligencia Artificial son las ontologías, una poderosa herramienta la cual permite representar al mundo real de la forma más precisa posible.

Con ellas se pueden desarrollar acercamientos para resolver infinidad de problemas, uno de ellos es la integración de datos, el cual es un creciente problema hoy en día donde las aproximaciones clásicas aplicadas a bases de datos parecen ser insuficientes y carecen de características que las ontologías pueden ofrecer.

En este trabajo el problema de la integración de información se pretende resolver con el uso de ontologías y técnicas de alineación, integrar dos fuentes de datos, es el caso particular en este trabajo se utilizan bases de datos. Se considera que dichas ontologías se obtendrán mediante la transformación de un modelo de base de datos, en particular el modelo Entidad-Relación, usando reglas planteadas en el presente trabajo para lograr representar estos modelos en una ontología. Para la alineación de las ontologías se aplican similitudes semánticas para encontrar las posibles similitudes entre los elementos definidos en las ontologías; y a partir de un motor de consultas, recuperar datos de diferentes fuentes de forma transparente para el usuario (integración).

Se plantea un caso de estudio para la metodología en general y casos específicos para las diferentes etapas que conforman a ésta.

Índice general

Índice general	i
Índice de figuras	iv
Índice de Tablas	vi
1 Introducción	3
1.1 Generalidades	3
1.2 Motivación	4
1.3 Descripción del problema	5
1.4 Justificación	7
1.5 Objetivos	8
1.5.1 Objetivo General	8
1.5.2 Objetivos Particulares	9
1.6 Organización del documento de tesis	9
2 Estado del Arte	10
2.1 Integración de datos	10
2.1.1 Aplicación y desarrollo para la integración de bases de datos	10
2.1.2 Integración de esquemas de base de datos	11
2.1.3 Generalidades de la integración semántica en bases de datos	12
2.2 Integración de datos basada en ontologías	12
2.2.1 El rol de las ontologías en la integración de datos	12
2.2.2 Alineación semi-automática de ontologías para la integración de datos geoespaciales	13
2.2.3 Integración de fuentes de datos espaciales con base en ontologías	14
2.3 Alineación de ontologías	14
2.3.1 Alineación de ontologías con OLA	14
2.3.2 Aplicación dinámica multiestrategia para la alineación de ontologías: RiMOM	15

2.3.3	Falcon-AO: Un sistema práctico para el mapeo de ontologías	16
2.4	Tópicos relacionados	17
2.4.1	Integración semántica geoespacial	17
3	Marco Teórico	19
3.1	Base de Datos	19
3.1.1	Modelo Conceptual	20
3.1.2	Modelo Relacional	20
3.2	Integración de Información	21
3.2.1	Correspondencia Semántica	22
3.3	Ontologías	23
3.3.1	Clasificación de las ontologías	24
3.3.2	Lenguaje de ontologías Web “OWL”	25
3.3.3	Conceptos de los lenguajes en las ontologías de la Web Semántica .	25
3.4	Alineación de Ontologías	26
4	Metodología propuesta	29
4.1	Introducción	29
4.1.1	Etapa de Conceptualización	30
4.1.2	Etapa de Alineación Semántica	31
4.1.3	Etapa de Recuperación y Visualización	31
4.2	Diseño de la Conceptualización	31
4.2.1	Análisis de las fuentes de información	32
4.2.2	Transformación del modelo Entidad-Relación a una Ontología	38
4.3	Diseño de la Alineación Semántica	44
4.3.1	Detección de correspondencias semánticas	44
4.3.2	Similitud léxica	45
4.3.3	Similitud semántica	46
4.3.4	Método para la Alineación de Ontologías	46
4.4	Diseño de la Recuperación y Visualización	48
4.4.1	Motor de consultas	49
5	Pruebas y Resultados	53
5.1	Plantamiento del escenario de trabajo	53
5.2	Implementación de la metodología	54
5.2.1	Conceptualización	54
5.2.2	Transformación del modelo ER a Ontologías	56
5.3	Alineación	60
5.4	Recuperación de información	63
5.5	Visualización	65
6	Conclusiones	70

Referencias	72
A Ontologías OWL	79
A.1 Ontología OWL de la BD_1	79
A.2 Ontología OWL de la BD_2	83
B Herramientas de implementación	88
B.1 PROTEGE	88
B.2 JENA	89
B.3 SPARQL	90

Índice de figuras

1.1	Consultas individuales a las fuentes de datos	5
1.2	Consultas a las fuentes de datos mediante el uso de un sistema de integración	6
1.3	Solución del problema de integración	7
1.4	Propuesta detallada a la solución del problema de integración	8
3.1	RDF tripleta [67]	26
3.2	Proceso de mapeo	27
4.1	Estructura general de la metodología propuesta	30
4.2	Etapas generales de la metodología propuesta	30
4.3	Etapas de conceptualización	32
4.4	Ejemplo de un Dominio: Plantas	33
4.5	Ejemplo de Dominio y subdominio	33
4.6	Representación de Entidad en el modelo ER	35
4.7	Representación de atributos en el Modelo ER	36
4.8	Representación de atributos monovalorados en el modelo ER	36
4.9	Representación atributos compuestos en el modelo ER	36
4.10	Representación de Relaciones en el modelo ER	37
4.11	Entidad en el modelo E-R	37
4.12	Entidad en el modelo E-R	38
4.13	Representación de las Fuentes de Datos	39
4.14	Ejemplo de Modelo ER	41
4.15	Transformación de entidades en clases	41
4.16	Transformación de atributos simples	42
4.17	Transformación de atributos mono y multivaluados	42
4.18	Transformación de atributos mono y multivaluados	42
4.19	Transformación de atributos mono y multivaluados	43
4.20	Transformación de relaciones binarias	43
4.21	Alineación de ontologías	44

4.22	Alineamiento manual de dos ontologías	47
4.23	Proceso de Recuperación semántica de información	49
4.24	Descripción de las ontologías y su alineamiento	51
5.1	Diagrama ER de la BD_1	54
5.2	Diagrama ER de la BD_2	55
5.3	Modelo de la ontología BD_1 a partir del diagrama ER	57
5.4	Grafo de la ontología BD_1 generado en Protégé	57
5.5	Modelo de la ontología de la BD_2 : Primera Regla de transformación	58
5.6	Modelo de la ontología de la BD_2 : Segunda Regla de transformación	58
5.7	Grafo de la ontología de BD_2 generado en Protégé	59
5.8	Comparación de mapeos con las técnicas de similitud aplicadas	62
5.9	Diagrama de casos de uso de la aplicación	65
5.10	Aplicación de escritorio	66
5.11	Cuadro de selección de archivos	66
5.12	Archivos seleccionados	66
5.13	Grafo de la ontología A	67
5.14	Grafo de la ontología A	68
5.15	Grafo de la alineación	69
5.16	Archivos seleccionados	69

Índice de Tablas

3.1	Elementos de RDF/RDFS	26
4.1	Diccionario de datos de la figura 4.11	37
4.2	Descripción de las relaciones de la figura 4.12	38
4.3	Mapecto de elementos del modelo Entidad-Relación a una Ontología	39
4.4	Diccionario de datos de la figura 4.14	41
4.5	Descripción de las relaciones de la figura 4.14	41
4.6	Ponderación de la medida de similitud	47
4.7	Alineamiento de la figura 4.22	48
4.8	Archivo de alineación	49
4.9	Archivo de alineación para la figura 4.24	51
5.1	Número de elementos del diagrama ER de BD_1	55
5.2	Diccionario de datos de BD_1	56
5.3	Número de elementos del diagrama ER de BD_2	56
5.4	Diccionario de datos de BD_2	56
5.5	Relaciones del diagrama ER de BD_2	57
5.6	Distancia de Levenshtein	60
5.7	Distancia de SMOA	61
5.8	Similitud de propiedades	61
5.9	Vista parcial de resultados de la consulta SPARQL	64
B.1	Comparación de la Web Semántica y el Framework Jena	90

Agradecimientos

Agradezco infinitamente a mis padres por su formación, valores, entusiasmo, comprensión, confianza, apoyo, que me han forjado y me ha hecho ser la persona que soy, y que sin ellos no tendría las bases para lograr cualquier reto que me proponga.

A mi madre por ser una luchadora incansable y enseñarme a nunca rendirnos, a ser siempre cada día mejores y mantenernos unidos.

A mi padre que con su esfuerzo nos ha dejado esta herencia tan valiosa que nunca terminara.

Al Instituto Politécnico Nacional una vez más me abrió sus puertas para seguir formándome como un profesional con los más altos estándares y me ha dejado una herencia académica sólida y reconocida en cualquier lugar, solo me queda portar con orgullo tu nombre y que pertenezco a esta institución, en cualquier lugar al que vaya o me encuentre siempre llevarla a lo más alto

Al Dr. Miguel por la oportunidad de poder trabajar con él, el apoyo proporcionado durante este proceso, su infinita paciencia y compromiso en el desarrollo de este trabajo, por creer en mí en momentos de dificultad, por su guía y sus consejos.

Al Dr. Rolando por su valioso apoyo, su esfuerzo, paciencia, compromiso, y la oportunidad de poder realizar este trabajo con él.

Al Dr. Marco por el interés y apoyo que brinda a cualquier persona, su entrega por la enseñanza y el gran compromiso que tiene por los demás. El aliento a seguir trabajando y destacar en el día a día.

Al Dr. Giovanni por liberar el combustible necesario para concluir satisfactoriamente este trabajo.

A la Maestra Dinora por su paciencia, sus puntuales observaciones, sus acertados cuestionamientos, el tiempo otorgado a este trabajo y por brindar la luz para terminar este trabajo.

A la Maestra Roció por su apoyo para culminar esta travesía.

A la Maestra Erendira que de no ser por ella no habria comenzado esta travesia.

A la generación geos Isac, Angel, Fernando, que se forgo una buena amistad, un buen grupo de trabajo y un gran apoyo durante este tiempo. Ademas de los buenos momentos y experiencias que vivimos durante el transcurso de este esfuerzo.

Al grupo del Laboratorio de Procesamiento de Información Geográfica por ese compañerismo, apoyo y buena vibra que se genera en nuestro laboratorio todos los días.

A la Ing. Osiris por el apoyo mostrado para poder culminar este trabajo, sabiendo lo importante que es para mi.

A mis amigos y compañeros por la buena vibra y el esfuerzo constante que contagian.

A las personas que no creian en mi, por que aveces son necesarias para retomar ese coraje y orgullo que se requieren en ciertos momentos.

Al destino por ponerme a las personas adecuadas para que me enseñaran infinidad de cosas las cuales no habría podido aprender de nadie si no de ellos.

A todas las personas que en algún momento me dieron y compartieron su amistad, su apoyo, su conocimiento, algún consejo, algunas palabras, un instante, ¡Gracias!

A Dios por la vida y la fuerza para realizar esta aventura, así como tantas dichas y bendiciones que me dio en el camino.

Politecnico simpre...

Huelum, Huelum gloria a la cachi cachi porra pin pon porra pin pon porra POLITECNICO POLITECNICO GLORIA!!!

Capítulo 1

Introducción

Todas las verdades son fáciles de entender una vez que se hayan descubierto; el problema es descubrirlas.

GALILEO GALILEI

1.1 Generalidades

En los últimos tiempos, se ve marcada la tendencia de que la sociedad, la cual está cambiando rápidamente a ser una sociedad del conocimiento [33].

En la Ciencia de la Computación, el conocimiento es representado en una innumerable variedad de formas. Adicionalmente, conforme los recursos computacionales se vuelven más baratos, cada año la cantidad de información se incrementa [11]. Debido a esto, las personas están buscando formas de obtener todo este conocimiento de una forma conjunta y no solo como elementos individuales de información.

Desde el comienzo del desarrollo de sistemas computacionales, la integración de la información heterogénea siempre ha sido un tema y reto de la investigación y el desarrollo, como un importante paso podemos mencionar, que el intercambio físico de datos ya no es un problema, gracias al Internet [8].

Aunque la mayoría de las veces los esquemas de representación de la información o conocimiento, no son compatibles entre sí. En años recientes, se han desarrollado normas para representaciones de la información, las cuales permitan generar una compatibilidad entre diferentes esquemas, por ejemplo, el lenguaje XML (Extensible Markup Language

[16][62]). Pero transformar los viejos esquemas a los nuevos, aún existen grandes problemas, por ejemplo: transformar una base de datos relacional a un esquema XML, se requiere de una gran cantidad de esfuerzo humano para entender e integrar los conceptos que están representados en estos esquemas.

Considérese que, el siguiente paso hacia una mejor comprensión e integración, debe ser una representación explícita y semántica, a través de las ontologías [45], las cuales consisten de entidades, relaciones entre sí y axiomas que restringen o mejoren la representación.

1.2 Motivación

Un problema hoy en día es la cantidad de información que se tiene disponible en diferentes fuentes de datos heterogéneas en Internet, bibliotecas digitales, bases de datos antiguas, sistemas de correo electrónico, etc. Los usuarios de dichas fuentes necesitan realizar en esta información dispersa, consultas de una manera rápida, consolidada y relevante que vaya de acuerdo con las necesidades preestablecidas.

Con la aparición de la web, la gran popularización de dispositivos tecnológicos para acceder a esta información y la facilidad de agregar más, surge el problema de que la información es ambigua y no está estructurada. Por lo tanto, la manera de encontrar lo que se está buscando, es sumamente complicado.

“Un mar de información para poder encontrar un grano de arena.”

Tradicionalmente, las consultas se realizaban a una base de datos local o remota con un esquema bien conocido. En los sistemas actuales las consultas implican a diferentes fuentes de datos, que no solo pueden estar dispersas geográficamente, sino que también suelen tener esquemas diferentes.

En ciencias de la información geográfica, la heterogeneidad es el problema relacionado con el estudio de la integración de la información geográfica, la cual se divide en: la heterogeneidad sintáctica y semántica [84].

La *heterogeneidad sintáctica* se refiere a las diferencias que se pueden apreciar entre las fuentes de información como las diferentes formas de almacenar los datos o las diferencias entre las estructuras de archivos, por ejemplo: de texto plano y archivos XML. [84]

La *heterogeneidad semántica* se refiere a las diferentes formas de representar el significado de un objeto. Por ejemplo, al representar el objeto *parcela* se podrían tener los términos *lote* y *sembradío*, los cuales tienen el mismo significado pero son expresiones distintas. [84]

1.3 Descripción del problema

La gran cantidad de información a la que se puede tener acceso hoy en día, genera el problema de cómo poder recuperar esta información sin tener que consultar cada una de las fuentes por separado. (véase la figura 1.1)

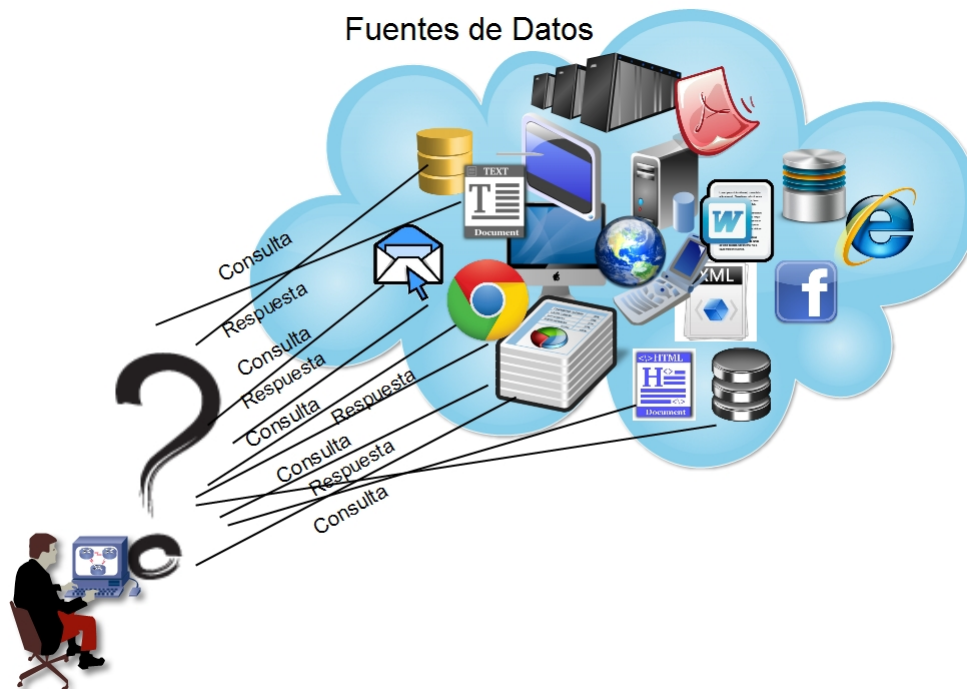


Figura 1.1: Consultas individuales a las fuentes de datos

Entonces podemos decir que el problema de integración consiste en combinar los datos que residen en diferentes fuentes y proporcionar a los usuarios una vista unificada de los datos (véase la figura 1.2). Este proceso llega a ser significativo en una variedad de situaciones de dominios comerciales y científicos, donde estas fuentes tienen la característica que son heterogéneas entre sí.

La heterogeneidad puede estar presente en múltiples niveles:

- Nivel de estructuración
- Modelo de Datos
- Plataforma de software
- Convención de Sintaxis
- Convenciones semánticas. Taxonomías

- Granularidad
- Esquemas

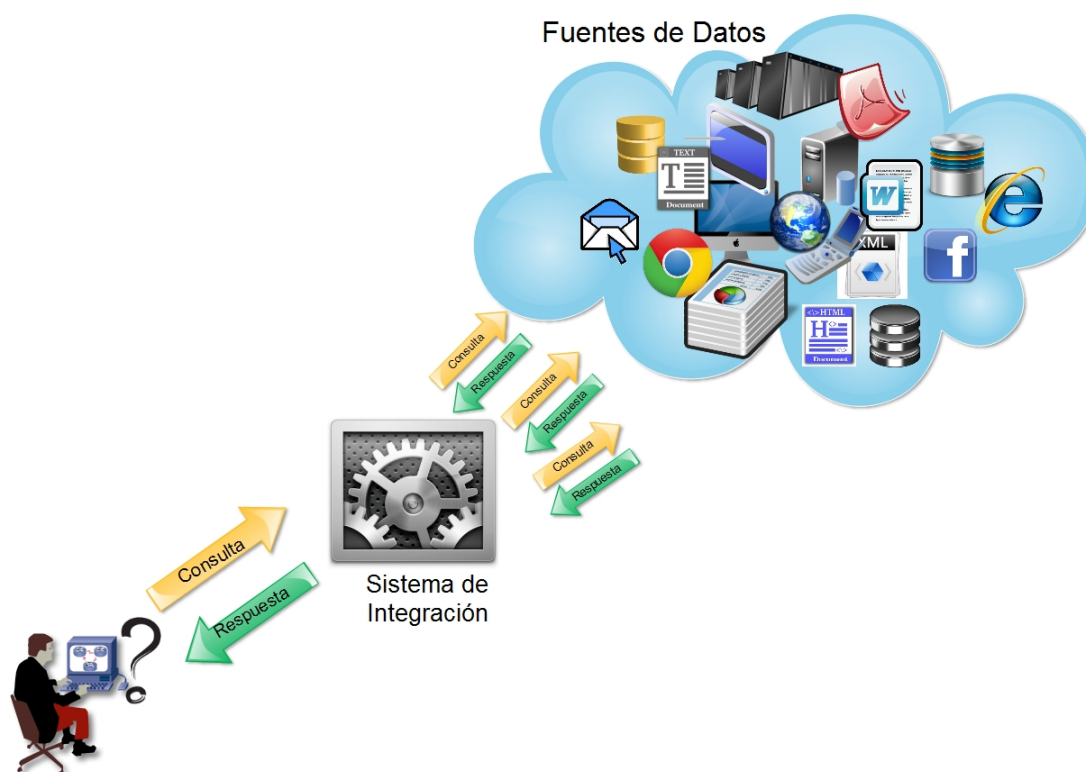


Figura 1.2: Consultas a las fuentes de datos mediante el uso de un sistema de integración

La integración de fuentes de datos aparece con mayor frecuencia a medida que el volumen y la necesidad de compartir y explotar los datos aumenta. Esta se ha convertido en un tema de gran interés en el trabajo de investigación teórico y un gran número de problemas aun quedan sin resolver. En círculos de gestión, se refieren a menudo a la integración de datos como “Integración de Información Empresarial”.

Este trabajo trata de resolver de manera parcial la integración de fuentes de datos heterogéneas por medio de un motor de integración. (véase la figura 1.3). Este motor de integración se basa en la representación de las fuentes de datos como ontologías y haciendo uso de técnicas de alineación lograr la integración de una forma semiautomática y a partir de un motor de búsquedas recuperar los datos de las diferentes fuentes de una forma unificada. (véase la figura 1.4)

Se describirá un caso de estudio para ejemplificar el funcionamiento de las etapas que conforman la metodología así como un caso de estudio para la metodología completa.

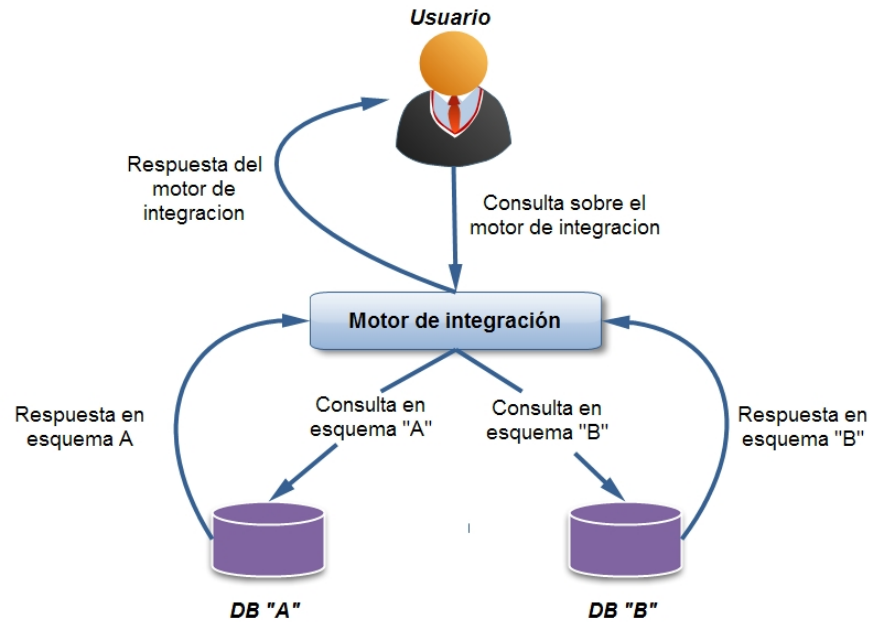


Figura 1.3: Soluci3n del problema de integraci3n

1.4 Justificaci3n

Hoy en d3a, hay una gran cantidad de datos generados acerca del mundo, no s3lo de nuevos sistemas de informaci3n geogr3fica, sino tambi3n de nuevas tecnolog3as y modelos para almacenar esta informaci3n, este escenario deja un gran n3mero de interesantes retos de investigaci3n, como lo es la integraci3n de informaci3n en fuentes de datos heterog3neas.

La integraci3n de informaci3n ha incrementado su importancia, debido a las nuevas posibilidades que surgen de la interconexi3n del mundo y la creciente disponibilidad de la informaci3n.

El uso de ontolog3as se debe a su propiedad de representar el conocimiento, de describir el significado de las cosas, a trav3s de los conceptos y las relaciones que definen un dominio en particular. Entonces las ontolog3as ayudan a la creaci3n de modelos conceptuales para facilitar la integraci3n sem3ntica de la informaci3n.

Para llevar a cabo la integraci3n de las diferentes fuentes de datos, el personal encargado debe de estar familiarizado con la terminolog3a de las fuentes, as3 como las diferentes formas en las cuales se accede a esta informaci3n. Adem3s algunos sistemas integradores basados en ontolog3as, pr3cticamente son de una forma manual, ya que el personal encargado de esta labor debe de generar la ontolog3a de integraci3n, as3 como la relaci3n de 3sta con las fuentes de datos.

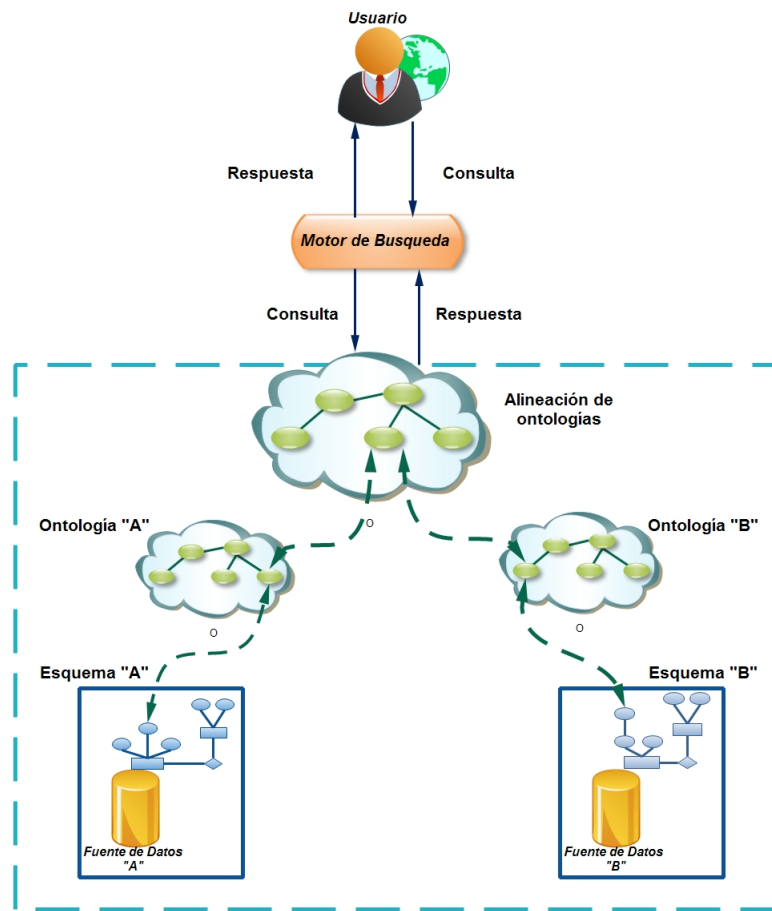


Figura 1.4: Propuesta detallada a la solución del problema de integración

En este trabajo se plantea la posibilidad de que este personal no necesariamente deba conocer tanto la estructura como la terminología de dichas fuentes, sino que el sistema sea capaz de presentarle un prototipo de integración, el cual podría o no coincidir completamente pero sí en algo muy aproximado. Dando la posibilidad de ser refinado manualmente y así obtener resultados más precisos a las consultas de los usuarios [12] la alineación de ontologías puede ser una herramienta útil para este propósito.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar una metodología basada en la aplicación de un método de alineación de ontologías para resolver parcialmente el problema de heterogeneidad semántica y sintáctica en la integración de fuentes de datos heterogéneas.

1.5.2 Objetivos Particulares

- Analizar los métodos de integración de fuentes de datos del estado de arte actual.
- Analizar los algoritmos de alineación de ontologías del estado de arte actual.
- Crear las ontologías correspondientes a las fuentes de datos mediante el estándar OWL (Ontology Web Language) que describa sus conceptos y relaciones.
- Utilizar un motor de consulta que integre dichas fuentes por medio de la alineación de ontologías.
- Implementarla visualización de resultados de manera gráfica, en forma de árbol.
- Diseñar un caso de estudio que requiera la implementación de la metodología desarrollada para solucionar los problemas de la integración de fuentes de datos.

1.6 Organización del documento de tesis

El siguiente trabajo de tesis está organizado de la siguiente manera:

En el Capítulo 2 se presenta el estado de arte, donde se ubica la presente investigación. Por otra parte el Capítulo 3 revisa las bases teóricas que fundamentan la metodología propuesta para la integración de datos espaciales, como los fundamentos para la alineación de ontologías.

En el Capítulo 4 se describe la metodología de integración semántica propuesta. Este capítulo describe detalladamente cada componente utilizado en la propuesta de solución. Las pruebas y resultados obtenidos, a partir de la implementación de la aplicación desarrollada para la integración se muestran en el Capítulo 5.

Finalmente, el Capítulo 6 describe las conclusiones a las que se llegaron con el desarrollo de este trabajo de tesis, así como las propuestas de trabajo a futuro a partir de esta metodología.

Capítulo 2

Estado del Arte

Si buscas resultados distintos, no hagas siempre lo mismo.

ALBERT EINSTEIN

2.1 Integración de datos

2.1.1 Aplicación y desarrollo para la integración de bases de datos

Roger H. L. Chiang en [24] plantea las motivaciones por las cuales se busca la integración de fuentes de datos heterogéneas, son la descentralización de la información organizacional, la fusión de empresas, la necesidad de acceso a la información externa para la toma de decisiones y la migración o actualización de sistemas de información heredados.

La integración de base de datos requiere la comprensión absoluta del significado de cada base de datos, comprensión que se adquiere en un proceso de ingeniería inversa, a través de los esquemas de implementación, aun cuando éstos tengan estructuras de nombrado inconsistentes. Así mismo, esta integración solo es posible si los dominios en la información de las bases de datos a integrar son similares. Esta integración debe ser capaz de darse bajo cinco niveles: *semántico*, *conceptual*, *lógico*, *de consulta* y *de instancia*. En estos niveles cada uno enfrenta diferentes problemas, emplea diferentes tipos de conocimiento y semántica de dominio y representa a la base de datos integrada en sus distintos niveles de abstracción.

Chiang considera que para resolver la heterogeneidades semánticas en un ambiente de base de datos múltiples, se requiere tener conocimiento acerca del dominio de aplicación de cada base de datos, del modelo conceptual de cada una, de la sintaxis de consulta empleada

por el manejador de base de datos local y global, y de los procedimientos de integración de instancias como las funciones de conversión entre tipos de datos.

2.1.2 Integración de esquemas de base de datos

Rachel Pottinger en [73] nos explica que el problema consiste en resolver el hecho de que la misma información es almacenada bajo diferentes esquemas; a lo que se le ha llamado *heterogeneidad semántica de esquemas*. Para resolver este problema, debe haber un mecanismo que permita ejecutar o procesar consultas sobre múltiples esquemas.

Esto involucra, crear un esquema de base de datos el cual es la integración de todos los esquemas originales a los cuales queremos acceder. Crear un mapeo entre los esquemas. A este proceso se le llama esquema *matching*. Tener un sistema el cual permita usar estos mapeos para poder interpretar o transformar consultas.

Algunos acercamientos que han tenido impacto para resolver este problema son:

Data Warehousing

En estos sistemas todos los datos de los N esquemas son importados a un esquema global, el cual contiene todas las propiedades necesarias para poder hacer esta importación de datos. En otras palabras, se genera una gran fuente de información, la cual contiene toda los datos dando la apariencia de estar integrada. Uno de los problemas más comunes en estos sistemas es que para crear un esquema global, éste debe ser usado para responder diferentes tipos de consultas y debe ser optimizado para esta función.

Integración de esquemas de base de datos

En éste los datos se mantienen en su esquema original, y se genera un esquema global, el cual es utilizado para generar consultas específicas para cada esquema original.

Sistemas manejadores de datos punto a punto

Un sistema punto a punto es un sistema de integración de datos distribuido que provee un acceso transparente a bases de datos heterogéneas, sin la necesidad de un esquema global. Este sistema permite que cada punto tenga su propio esquema y realizar la reformulación de las consultas a través de los mapeos entre los esquemas de la red.

En general todos estos sistemas solucionan el problema clásico de integración, pero dejan abierto un nuevo problema que se trata de resolver con diferentes acercamientos más sofisticados, que es la integración semántica tanto de los esquemas como de los datos contenidos en éstos.

2.1.3 Generalidades de la integración semántica en bases de datos

Doan y Haley [34] muestran un corto recorrido que ha llevado la investigación de la integración semántica en las bases de datos, este tipo de integración ha recibido mayor importancia en las últimas dos décadas y se está convirtiendo en una área de investigación destacada.

Las aplicaciones de bases de datos que requieren una integración semántica son las que usan estructuras de representación, por ejemplo, los esquemas relacionales, XML, definiciones de tipo de documentos (DTDs), etc., para almacenar los datos y algunas de éstas representaciones ocupan más de un tipo de representación. Como tal los sistemas deben resolver las heterogeneidades con respecto a los esquemas y los datos, la manipulación de éstos y la habilidad de traducir los datos y consultas a través de los esquemas.

Las primeras aplicaciones de integración son las relacionadas a la integración de esquemas, en la cual dado un conjunto de esquemas se fusionan para crear un esquema global que abarque a todos. Este proceso de integración requiere establecer las correspondencias semánticas entre los esquemas y luego usar estas correspondencias para fusionar los esquemas. Conforme las bases de datos se volvieron populares y más usadas, se generó la necesidad de trasladar datos entre las diferentes bases de datos. El problema se agravó cuando se trasladaban los datos de viejas bases a nuevas, por lo cual originó al surgimiento de los *datawarehousing* y la *minería de datos*.

En años recientes, con el crecimiento explosivo de la información en la red, se da la posibilidad de que muchas aplicaciones necesiten integración semántica, dando pie a los sistemas de integración de datos, los cuales proveen al usuario de una interface de consulta unificada, la cual permite recuperar información de diferentes fuentes de datos haciéndolo transparente al usuario. El principal problema que se genera con estos sistemas es la detección y eliminación de información duplicada.

2.2 Integración de datos basada en ontologías

2.2.1 El rol de las ontologías en la integración de datos

La integración de datos provee la habilidad de manipular transparentemente los datos a través de múltiples fuentes de datos [29]. Esta integración es relevante en aplicaciones como empresariales, medicas, sistemas de información geográfica, E-Commerce. Basadas en la arquitectura de integración existen dos diferentes tipos de sistemas: los sistemas de información centralizados [3] [20] [13] [5] y sistemas de integración punto a punto (peer to peer) [4] [26] [71] [9]. En los sistemas centralizados usualmente se tiene un esquema global el cual provee al usuario de una interface uniforme para acceder a la información de las fuentes de datos. En contraste los sistemas peer-to-peer, no tienen puntos globales de control en las fuentes de datos. Entonces cualquier punto puede aceptar consultas de la información

distribuida en todo el sistema.

Las ontologías han sido utilizadas para tratar de solucionar el problema de integración por que proveen una conceptualización explícita y que puede calcularse fácilmente, de un dominio. Y se usan generalmente de tres diferentes maneras [82]:

- *Uso de una ontología simple*: Todos los esquemas están directamente relacionados con una ontología global la cual provee una interface única para el usuario [27]. Este acercamiento requiere que todos las fuentes tengan casi la misma vista del dominio, con el mismo nivel de granularidad.
- *Uso de múltiples ontologías*: Cada fuente de datos es descrita por su propia ontología. En vez de usar una ontología común, cada ontología local es mapeada una a otra. Para este propósito, se requiere un representación formal para definir las relaciones de los mapeos entre las ontologías. Híbridos: La combinación de los acercamientos anteriores, primero se genera una ontología para cada esquema de las fuentes de datos, estos esquemas no se mapean entre sí, si no que se mapean a una ontología global. Una ventaja de este tipo de aplicaciones es que se pueden añadir fácilmente nuevas fuentes de datos sin la necesidad de modificar los mapeos existentes.

Con esto se pueden identificar cinco usos de las ontologías en la integración de datos [29]:

- La representación de metadatos:
- Conceptualización Global
- Soporte a consultas de alto nivel
- Soporte de mapeos
- Declaración de mediadores

2.2.2 Alineación semi-automática de ontologías para la integración de datos geoespaciales

Cruz, Sunna y Chaudhry en [28], facilitan un enfoque para la integración de datos en aplicaciones geoespaciales con bases de datos heterogéneas, la cual se basa en una jerarquía de ontologías, donde existe una ontología fundamental o global que describe el dominio, con los conceptos de las ontologías que describen a las bases de datos: se genera la alineación entre la ontología global y las demás ontologías, y al obtener esta alineación se puede tener el potencial de recuperar información de infinidad de bases de datos diferentes con una sola consulta, además de que ésta se puede extender al agregar más bases de datos y alinear su ontología con la ontología global.

La alineación de ontologías se realiza de manera semiautomática ya que generaron una herramienta que arroja sugerencias de posibles correspondencias entre los conceptos de las diferentes ontologías y éstas son aceptadas o no por el usuario, así mismo el usuario puede crear sus propias correspondencias entre los diferentes conceptos de las ontologías y la ontología global. De esta manera, se genera el archivo de mapeo correspondiente.

Al generar estos mapeos se puede recuperar la información de las bases de datos basándose en la estructura de la ontología global.

2.2.3 Integración de fuentes de datos espaciales con base en ontologías

Hernández Cardona en [21] ejemplifica un sistema de integración de esquemas mediante ontologías. En este trabajo se describe la forma en que se puede emplear y recuperar la información utilizando este tipo de enfoque.

Las ontologías se generan de sus fuentes de información en este caso bases de datos y a partir de estas ontologías se crea una ontología global que permite mapear sus ontologías entre sí, a esta técnica que utiliza se le conoce como *mediador*.

Con esto se realizan consultas al esquema mediador, las cuales son transformadas a cada esquema particular de las fuentes de datos y mediante un *wrapper* recuperan los datos de las bases de datos.

Estos datos recuperados antes de ser presentados deben pasar por una transformación al esquema global.

Esta técnica se muestra mediante un sistema geográfico con el cual recupera información con ciertas características que se prestablecen en el sistema, dando resultados exactos o resultados sugeridos (aproximados) por el sistema.

2.3 Alineación de ontologías

2.3.1 Alineación de ontologías con OLA

Euzenat en [39] presenta la herramienta OLA* para la alineación de ontologías, fue desarrollada por los grupos de DIRO, de la Universidad de Montreal e INRIA. Ola está dedicado para alineaciones de ontologías expresadas en el lenguaje OWL.

Ola es diseñado como un ambiente para manipular alineaciones y ofrece, los siguientes servicios:

*OWL Lite Aligmet

- Análisis y visualización de ontologías.
- Computación automática de las similitudes entre las entidades de las diferentes ontologías.
- Extracción automática de las alineaciones de un par de ontologías.
- Construcción manual de las alineaciones.
- Construcción de alineaciones automáticos a partir de alineamientos existentes.
- Visualización de alineaciones.
- Comparación de alineaciones.

Mediante la utilización de grafos como forma de representar las ontologías, hace que la comparación y las alineaciones sean más fáciles.

El algoritmo con el OLA emplea técnicas lingüísticas a nivel de elemento y estructura, además es iterativo, ya que computa una primera aproximación de la similitud y establece las relaciones, posteriormente vuelve a iterar. El cálculo es más preciso en cada iteración. Las correspondencias finales se establecen por medio de un umbral u optimizando las selecciones de parejas de clase.

2.3.2 Aplicación dinámica multiestrategia para la alineación de ontologías: RiMOM

RiMOM [60] es un marco de trabajo de múltiples estrategias para la alineación de ontologías basado en la minimización de riesgos de una decisión Bayesiana [81]. Emplea múltiples estrategias para la alineación de ontologías y los combina con alineaciones manuales. Una de las características principales de RiMOM es que al introducir dos ontologías para la alineación RiMOM determina automáticamente que método utilizar, que tipo de información se debe usar para calcular la similitud y como combinar los múltiples métodos si es necesario.

Las etapas de RiMOM son las siguientes:

- Preprocesamiento. Dadas dos ontologías RiMOM genera una descripción para cada entidad. Entonces calcula dos factores de similitud, que usará en las siguientes etapas.
- Alineación basada en lingüísticas: Múltiples estrategias lingüísticas son ejecutadas. Cada estrategia usa diferente información ontológica y obtiene una similitud resultante para cada par de entidades.
- Combinación de similitud. En esta etapa se combina la similitud obtenida de las estrategias de selección. La medida de similitud se obtiene de la combinación de dos factores de similitud.

- Propagación de la similitud. Esta etapa considera la similitud estructural. Utiliza tres estrategias de propagación, llamadas Concepto a Concepto, Propiedad a Propiedad y Concepto a Propiedad.
- Generación de las alineaciones y refinamiento, refina las similitudes y genera una alineación final.

Las estrategias utilizadas en RiMOM para la alineación son:

- Lingüísticas
 - Edición de distancias
 - Distancia de Vectores
- Estructura
 - Similarity Flooding. Utilizando un constructor de parejas de grafos conectadas y una propagación de la similitud representando la ontología como un grafo dirigido etiquetado.

RiMOM a sido evaluado por el Ontology Aligment Evaluation Initiative donde ha tenido buenos resultados como:

- Alto performance.
- Efectivas estrategias de selección.
- Contribución a las estrategias de Similarity Flooding.

Además RiMOM aún es ineficiente para ontologías grandes ya que requiere una gran cantidad de memoria y una gran cantidad de tiempo para encontrar las alineaciones.

2.3.3 Falcon-AO: Un sistema práctico para el mapeo de ontologías

Falcon es una infraestructura para las aplicaciones de la Web Semántica que tiene como objetivo proveer de tecnologías fundamentales para encontrar alinear o comprender ontologías, además recientemente puede usarse para captura conocimiento de la Web mediante un enfoque basado en ontologías.

Falcon-AO, un componente de Falcon, es un sistema automático para el mapeo de ontologías que ayuda a actualizar la interoperabilidad entre las aplicaciones semánticas que utilizan diferentes ontologías pero que están relacionadas entre sí [53].

La arquitectura de Falcon-AO consiste de cinco componentes:

- **Modelo Fuente:** transforma las ontologías en modelos virtuales (en memoria) mediante Jena y ajusta los modelos usando un conjunto de reglas de coordinación [55].
- **Librerías de comparadores (Matcher):** V-DOC [76] y I-Sub [80] son dos comparadores lingüísticos de bajo nivel, GMO [51] es un comprador de estructuras iterativo; PBM [52] adopta la estrategia de divide y vencerás para encontrar bloques de mapeos entre ontologías grandes.
- **Conjunto de alineaciones.** Genera alineaciones usando formatos RDF/XML [38] y evalúa las alineaciones generadas contra alineaciones basadas en las métricas de precisión y recall.
- **Controlador central,** permite la configuración manual de las estrategias de alineación. Además ejecuta a los comparadores y combina la similitud basada en lingüística y la estructural
- **Repositorio,** almacena datos reusables durante el proceso de mapeo.

Falcon-AO tiene tres fortalezas:

- Puede completar varias tareas de mapeo, especialmente en ontologías grandes.
- Puede lograr de forma estable una buena precisión y recall en diferentes pruebas.
- Es eficaz, dado que todas las pruebas pueden ser ejecutadas en un tiempo razonable y no es necesario un gran poder de cómputo.

2.4 Tópicos relacionados

2.4.1 Integración semántica geoespacial

Naijun Zhou en [85], define la integración semántica geoespacial como una vista global de diversos términos en diferentes fuentes de datos, donde la semántica geoespacial indica el significado de los términos geoespaciales como atributos de un objeto espacial.

La tarea principal de la integración semántica geoespacial es entender la semántica que los objetos espaciales representan y medir estas relaciones semánticas entre objetos. La integración semántica espacial es obtenida a través de una metodología de cuatro pasos: normalización de las definiciones, medición de la similitud semántica, representación y mantenimiento de la similitud semántica y la fusión de términos [56].

Algunas aplicaciones que tiene la integración semántica geoespacial son las:

Mapeo de ontologías geoespaciales

Las ontologías geoespaciales usualmente consisten en un conjunto de términos. La

integración semántica provee una solución a las relaciones semánticas entre los términos de las diferentes ontologías. De aquí que algunas veces la integración semántica es también llamada *ontology mapping*.

Web Semántica Geoespacial

La comunicación entre las diferentes ontologías y semánticas son la tarea principal de la Web Semántica Geoespacial. Al automatizar la integración semántica se posibilita a las máquinas par entender y compartir significados con la finalidad de ayudar a alcanzar las metas de la Web Semántica.

Portales de Datos Geoespaciales

Permiten a los usuarios buscar fuentes de datos que usan diferentes términos para los mismos objetos.

Integración de bases de datos espaciales

La integración de bases de datos necesita combinar esquemas y dominios [34]. La integración de esquemas es una tarea en la integración semántica de las bases de datos.

Capítulo 3

Marco Teórico

Vale más saber alguna cosa de todo, que saberlo todo de una sola cosa.

BLAISE PASCAL

3.1 Base de Datos

La información y los datos son dos cosas distintas. La información es comprendida por una persona, mientras que los datos son valores almacenados en un medio pasivo como un disco duro.

Una base de datos entonces puede definirse como un repositorio de datos. Una base de datos es un modelo de un sistema del mundo real. El contenido de la base de datos representa un estado de que está siendo modelado. Cambios en la base de datos representan eventos que ocurren en el entorno que cambian el estado de lo que se está modelando. Es apropiado estructurar una base de datos visualizar que es lo que se intenta modelar.

Dentro del ámbito de la informática se desarrollaron los sistemas manejadores de bases de datos los cuales son un conjunto de datos, hardware, software y usuarios que interactúan para la gestión de la información. El propósito de los sistemas de bases de datos es reducir la brecha entre la información y los datos. Es decir los datos almacenados en la memoria o disco duro deben ser convertidos a información.

Las bases de datos están compuestas por datos, y metadatos. Los metadatos valga la redundancia son datos que sirven para especificar la estructura de la base de datos.

De este modo las bases de datos están compuestas [30] por:

- **Estructura lógica:** Indica la composición y distribución teórica de la base datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente como se están almacenando.
- **Estructura física:** Es la estructura de los datos tal cual se almacenan en las unidades de almacenamiento. La correspondencia entre la estructura lógica y física se almacena en los metadatos.

Las bases de datos pueden ser analizadas a diferentes niveles:

- Modelo Conceptual*
- Modelo Lógico
- Modelo Físico

3.1.1 Modelo Conceptual

Los elementos del nivel conceptual nos permiten modelar en forma independiente a cualquier modelo de datos (modelo lógico). El modelo conceptual provee una herramienta para desarrollar un esquema de base de datos de principio a fin en el proceso de diseño de la base de datos. Dentro de estos modelos se encuentra el Modelo Entidad Relación desarrollado por Chen [22][23] el cual es uno de los elementos a los que se enfoca este trabajo en particular.

3.1.2 Modelo Relacional

El modelo entidad relación es una herramienta para analizar las características semánticas de una aplicación que es independiente de los eventos. Esta representación incluye una notación gráfica, la cual provee un método conveniente de visualizar las inter relaciones a través de los elementos de una aplicación. Además este modelo es muy utilizado para la transición de la descripción de información a un diseño físico de base de datos. Debido a esto los diagramas entidad relación se han convertido en un esquema lógico en donde la base de datos es implementada.

Los términos básicos que el modelo entidad relación utiliza para describir conceptos son:

- **Entidad.** Una entidad es un objeto del mundo real distinguible de otros objetos.
- **Atributos:** Los atributos describen propiedades de las entidades y relaciones.

*Este modelo es una elemento de análisis importante para el desarrollo de este trabajo por lo cual será el único que se explica a detalle.

- *Atributos simples y compuestos*: Un atributo simple es la unidad semántica más pequeña de datos y son atómicos, son atributos no divisibles. Los atributos compuestos pueden subdividirse en partes, en un conjunto de atributos simples o compuestos.
- *Atributos simples y multivaluados*: Los atributos simples solo tienen un único valor para una entidad en particular. Los multivaluados pueden tener múltiples valores para un atributo para una entidad en particular.
- *Dominio*: Definición conceptual de los atributos: Cada uno de los atributos tiene asociado un conjunto de valores posibles.
- **Relaciones**: Una relación es una asociación entre dos o más entidades.
- *Jerarquías (“es un”)* Es tipo especial de relación el cual permite la herencia de atributos.
- **Llaves**. Las llaves solo diferencian una instancia de todas las demás en una entidad. Las llaves son un identificador.

El modelo entidad relación también identifica ciertas restricciones de cómo se conforma el contenido de los datos. Dos de los tipos más importantes de restricciones son:

- *La cardinalidad de las relaciones*: la cual indica el número de instancias en una entidad E_1 pueden estar asociadas con instancias de otra entidad E_2 .
 - **Relaciones uno a uno**. Para cada instancia en una entidad hay a lo más una instancia asociada en otra entidad.
 - **Relaciones muchos a uno**. Una instancia de la entidad E_2 está asociada a ninguna o más instancias de la entidad E_1 . Pero cada instancia de E_1 está asociada con una instancia de E_2 máximo.
 - **Relaciones muchos a muchos**: No hay restricciones de cuantas instancias de una entidad pueden ser asociadas con una instancia de otra entidad.
- *Dependencias de existencia*: Si la existencia de una instancia x depende de la existencia de instancia y , entonces se dice que existe una dependencia de x en y . Si y no existe entonces x no puede existir.

3.2 Integración de Información

La integración de información es la combinación de datos de diferentes fuentes y que provee al usuario una vista unificada de los datos [58].

Integrar información involucra fuentes de datos de diversas plataformas y con diferentes esquemas de datos. Para determinar las diferencias entre estos esquemas de datos, no solo se necesita generar una traducción de un esquema a otro sino también un mapeo semántico, el cual vincule las entidades en los distintos esquemas basados en la correspondencia de los significados de estas entidades. [2]

3.2.1 Correspondencia Semántica

El termino “*match*” se asocia como un operador que toma dos estructuras y establece correspondencias entre los elementos que semánticamente correspondan entre ambas estructuras. En la integración de esquemas de datos (Schema Matching [77] [65] [66] [72]), se pretende automatizar los procesos para encontrar dichas correspondencias semánticas en esquemas de datos.

La integración de esquemas de datos radica en:

- Descubrir mapeos por medio del cómputo de relaciones semánticas.
- Determinar estas relaciones semánticas por el análisis de significado que son codificados en los elementos y estructuras.

Existen diferentes problemas para establecer correspondencias entre las fuentes de datos, una de ellas es la heterogeneidad entre estructuras de datos, la cual se debe al uso de diferentes representaciones o definiciones para representar la misma información (diferencias entre esquemas), o diversas expresiones, unidades, tipo de datos, que presentan los datos al describir la información (diferencias entre datos).

Diseñar métodos automáticos para establecer estas correspondencias [84] presenta los siguientes obstáculos:

- **Heterogeneidad sintáctica:** Diferencias en el lenguaje usado para la representación de modelos.
- **Heterogeneidad estructural:** Diferencias por las diversas estructuras para representar los elementos.
- **Heterogeneidad semántica:** Las mismas entidades (significado) son representadas usando diferentes términos.
- **Heterogeneidad en modelos de representación de la información:** Diferencias entre los modelos subyacentes o su representación.

Existen varias técnicas de resolución de correspondencias [19]:

- **Lingüísticas:** Se basan en el nombre y significado de los elementos, emplean comparaciones entre las letras que conforman palabras: Levenshtein, N-Gramas, SoundEx.

- **Semánticas:** Las técnicas semánticas se basan en la comprensión del significado de los elementos a ser comparados. Se usan herramientas computacionales externas, como diccionarios de sinónimos, hiperónimos e hipónimos, tesauros, ontologías, o la creación de diccionarios de términos, abreviaturas, taxonomías, etc.
- **Basadas en restricciones:** se basa en la descripción de los metadatos, como son los tipos de datos, las características de nulidad y unicidad, los rangos de valores definidos y los valores permitidos, los identificadores únicos (llaves primarias), la integridad referencial, entre otras más.
- **Basados en instancias:** Aquí se pueden aplicar diferentes técnicas de acuerdo con el tipo de dato almacenado. Por ejemplo si el contenido es de tipo texto, se puede aplicar técnicas de obtención de información (information retrieval) para obtener palabras claves y temas, basados en las frecuencias relativas de palabras y de su combinación.

3.3 Ontologías

Las ontologías han tomado gran fuerza debido al gran poder expresivo que poseen; diversas definiciones han surgido, pero el término Ontología[†] proviene de la filosofía donde una Ontología es la teoría de “*la naturaleza del ser o de la existencia*”. Los filósofos Griegos Aristóteles y Sócrates fueron los primeros en desarrollar los principios de las Ontologías. Sócrates introdujo las nociones de ideas abstractas, herencia entre ellas y las relaciones entre ellas, Aristóteles incluyó las asociaciones lógicas. El resultado fue un modelo bien estructurado para representar el mundo real [36].

Dentro del área de la computación principalmente en la Inteligencia Artificial las ontologías son muy utilizadas para representar el conocimiento, por lo cual hay un sin gran número de definiciones acerca de ellas en esta área.

- **Gruber** [44]: Define una ontología como: “*Una especificación explícita de una conceptualización*”, donde una conceptualización es una vista abstracta y simplificada de la realidad o el mundo que se quiere representar para un propósito específico [45].
- **Neches** [70]: Define que una ontología define los términos básicos y las relaciones que componen el vocabulario de un dominio, así como las reglas para combinar términos y relaciones para definir extensiones a dicho vocabulario.
- **Guarino** [46]: Una ontología puede especificar una conceptualización solo de una indirecta ya que se aproximan a un conjunto de modelos pensados de un lenguaje y tal conjunto es solo una pobre caracterización de una conceptualización que no describe realmente el significado del vocabulario. Esto se debe a que la ontología es dependiente del lenguaje, mientras que la conceptualización no lo es.

[†]Para diferenciar el término, en filosofía se utiliza “Ontología” y en computación “ontología”

- **Borst [14]:** Coincide con Gruber, pero hace una modificación en la definición y establece a las ontologías como *"una especificación formal de una conceptualización compartida"*

Formalmente es posible expresar a una ontología como una tupla de la forma:

$$O = (S, A, K, L) \quad (3.1)$$

Donde:

S es una ontología básica.

A un sistema de axiomas.

K una base de conocimiento.

L un lenguaje.

Las ontologías integran semánticas bien definidas de los modelos, a través de axiomas es posible formalizar un gran número de correlaciones, donde esta expresividad semántica distingue a las ontologías de otros esquemas como XML, esquemas de bases de datos, o esquemas de clasificación como UML.

3.3.1 Clasificación de las ontologías

Las ontologías pueden ser clasificadas de acuerdo con su generalidad y su expresividad.

La generalidad se puede definir como la amplitud de una ontología, es decir, las ontologías intentan representar todos los términos del lenguaje natural, mientras otras son muy específicas en ciertos dominios o términos.

La expresividad se relaciona en la forma de cómo se interpreta el conocimiento capturado. Cuando más relaciones y más restricciones son capturadas en la ontología comienza a ser más expresiva, con lo cual se describe que tan detallado está el dominio.

Además las ontologías se pueden clasificar según su nivel de extensión o aplicación [47] en:

- **Ontologías de alto nivel:** Aborda conceptos generales los cuales son independientes de un dominio.
- **Ontologías de dominio:** Enfocadas a cubrir alguna terminología sobre un dominio.
- **Ontologías de tarea:** Se enfocan a describir el vocabulario sobre una tarea en específico.
- **Ontologías de aplicación:** Este tipo de ontologías describe conceptos de un dominio y de tareas particulares, comúnmente especializaciones de ambos.

3.3.2 Lenguaje de ontologías Web “OWL”

Las ontologías son muy útiles para representar el conocimiento. La ontología se define como: “Una especificación formal y explícita de una conceptualización”[45], y engloba los conceptos de: clases, relaciones entre clases, propiedades de clases, restricciones de las relaciones entre las clases y propiedades de las clases. OWL es el lenguaje para publicar y compartir ontologías a través de la WWW. OWL[‡] intenta proveer de un lenguaje que pueda ser utilizado para describir clases y las relaciones entre éstas.

3.3.3 Conceptos de los lenguajes en las ontologías de la Web Semántica

La web semántica incluye el estándar de XML, XMLS, RDF, RDFS, y OWL. OWL incluye un modelado semántico de los datos que es más poderoso que el utilizado por las bases de datos convencionales.

Los lenguajes utilizados para la creación de ontologías son RDF, RDFS y OWL, todos estos se basan en XML.

Resource Description Framework “RDF”

RDF es un lenguaje estándar con el que se pueden representar sujetos S predicados P y objetos O para expresar modelos de datos que representan nodos y vértices como URI y literales como cadenas o números.

RDF es un marco de trabajo simple de representación de los metadatos, usando URIs para identificar los elementos en la web y un modelo gráfico para describir las relaciones entre los recursos.

Cada vértice en el modelo RDF es llamado tripleta $T = \{S, P, O\}$. Cada tripleta afirma un hecho acerca de cada elemento. El sujeto es el elemento del cual el vértice inicia, el predicado es la propiedad que nombra el vértice y el objeto es el elemento donde termina el vértice [67]. (Ver figura 3.1.)

Resource Description Framework Schema “RDFS”

Los esquemas RDF (RDFS) definen un número de clases y propiedades que contienen aspectos semánticos. Una clase es un conjunto de elementos, y corresponden a tipos de conceptos o categorías en otras representaciones. Los conceptos más importantes de RDF y RDFS se muestran en la tabla 3.1.

[‡]Las características principales de OWL se presentan en el Anexo B

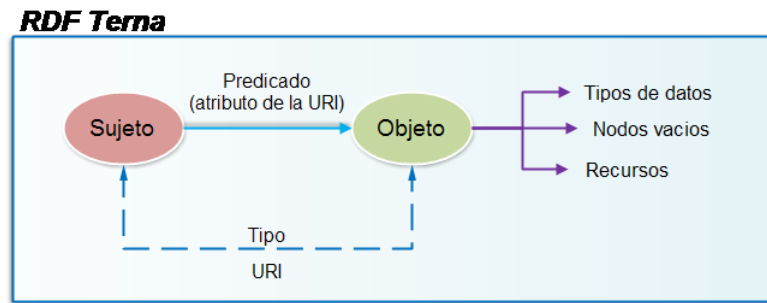


Figura 3.1: RDF tripleta [67]

Tabla 3.1: Elementos de RDF/RDFS

Sujeto	Predicado	Objeto	Representa
rdfs:Property	rfd:type	rdf:Class	Elementos Resources es un tipo de Class
rdfs:Class	rfd:type	rdf:Class	Class es un tipo de Class
rdfs:Property	rfd:type	rdf:Class	Propert es un tipo de Class
rdfs:type	rfd:type	rdf:property	type es un tipo de Property
rdfs:label	rfd:type	rdf:property	etiqueta de la propiedad
rdfs:comment	rfd:type	rdf:property	comentario de la propiedad

RDFS es un language de ontologías que utiliza RDF para representar los conceptos de elementos y propiedades, sub/super-clases, instancias y herencia [67].

3.4 Alineación de Ontologías

Puede definirse la alineación como la similitud que se puede encontrar entre los elementos que componen dos o más ontologías.

Así, es posible decir que dadas dos o más ontologías que describen un conjunto de entidades, el proceso de alineación consiste en encontrar para cada entidad de una ontología (conceptos, relaciones o instancias) una entidad que corresponda con el mismo significado en la otra ontología, donde estas correspondencias son caracterizadas por medio de relaciones de equivalencia. Por lo tanto, la alineación no es más que una relación de homogeneidad uno a uno [36].

De acuerdo con lo anterior, una función de alineación de ontologías Γ , basada en el conjunto E de todas las entidades $e \in E$ y el conjunto de posibles ontologías O es una función parcial:

$$\Gamma: E \times O \times O \rightarrow E \quad (3.2)$$

Para encontrar una alineación de la entidad e en las Ontologías O_1 y O_2 se escribe $\Gamma(e)$, una vez que es encontrada una correspondencia, puede decirse que la entidad e está alineada con la entidad f : $\Gamma(e) = f$. Un par de entidades (e, f) que no están en la función Γ , entonces este par de entidades debe ser probado y se le llama candidato de alineación (ver Figura 3.2).

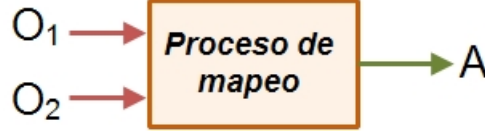


Figura 3.2: Proceso de mapeo

Al conjunto de correspondencias entre dos o más ontologías se denominan *mapeos*. Con este proceso se determina una alineación para un par de ontologías figura como se muestra en la figura 3.2.

Entonces el proceso de mapeo puede ser visto como una función μ que, a partir de un par de ontologías o y o' retorna una alineación A entre estas ontologías.

$$A' = \mu(o, o') \quad (3.3)$$

Donde la alineación resultante A' es un conjunto de tuplas [15] de la forma:

$$\langle e, e', n, R \rangle \quad (3.4)$$

Donde:

- e y e' son las entidades a las cuales se ha encontrado una relación.
- n es una medida de similitud o confianza.
- R es la relación asociada al mapeo, donde R identifica la relación que sostienen las entidades e y e' .

Algunas aplicaciones que tiene la alineación de ontologías son las siguientes:

- Evolución de ontologías: Se utiliza para encontrar los cambios que han ocurrido entre dos versiones de una ontología.
- Integración de esquemas: Se utiliza para integrar esquemas de diferentes bases de datos bajo una vista unificada.
- Integración de catálogos: Se utiliza para ofrecer un acceso integrado a catálogos en la web.

- Integración de datos: Para integrar el contenido de diferentes bases de datos a una sola base de datos.
- Compartir información por redes P2P: Se utiliza para encontrar las relaciones entre las ontologías que utilizan los diferentes nodos de la red.
- Servicios web compuestos: Se utiliza entre ontologías que describen interfaces para componer servicios web conectando las diferentes interfaces.
- Multiagentes de comunicación: Se utiliza para encontrar las relaciones entre las ontologías que utilizan dos agentes y traducir los mensajes que se intercambian dichos agentes.
- Mapeo de contexto: En el ambiente de la computación se utiliza en las necesidades de las aplicaciones y los contextos de información cuando las aplicaciones y los dispositivos fueron desarrollados independientemente uno del otro y usan diferentes ontologías.
- Transformación de consultas: Se utiliza para transformar consultas entre navegadores de web semántica, donde se buscan características en las páginas web, sobreponiendo ontologías en las páginas.

La alineación no está definida como un área especial de las ontologías, pero puede utilizarse en cualquier aplicación que utilice ontologías.

Capítulo 4

Metodología propuesta

Si hay algo difícil de hacer, entonces no vale la pena intentarlo.

HOMERO J. SIMPSON

4.1 Introducción

En la actualidad la cantidad de información que se genera mediante los diferentes medios, es enorme, por lo que surgen diversos problemas como: poder darle un formato único, la gran cantidad de formatos que existen para guardarla, o concentrarla en algún sitio o forma.

El objetivo de la metodología es recuperar información de diferentes fuentes de datos heterogéneas, es decir, el formato en la que éstas se encuentran almacenadas debe de ser igual, como por ejemplo: bases de datos o archivos XML; además de que no se tiene la necesidad de conocer la estructura de cada fuente.

Por tanto, mediante la representación de estas fuentes de información en una ontología el poder aplicar técnicas de integración, en este caso particular la alineación de ontologías para recuperar información de las diferentes fuentes de datos heterogéneas, es de vital importancia, con la finalidad de poder alimentar o visualizar la información recuperada en un sistema.

La figura 4.1 muestra de manera general la metodología propuesta y que se desarrolla a detalle en este capítulo. Como se observa en la figura se tienen entradas, las cuales se refieren a las ontologías que representan a las fuentes de información, después se encuentra el

método de integración, en el cual están las técnicas de alineación y un motor de recuperación de información, asimismo, al final como salida se obtiene la información concentrada de las diferentes fuentes de datos de la entrada para alimentar un sistema. La metodología está compuesta por tres etapas (véase figura 4.2):

- Etapa de Conceptualización
- Etapa de Alineación Semántica
- Etapa de Recuperación y Visualización



Figura 4.1: Estructura general de la metodología propuesta

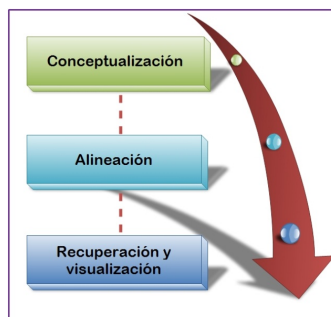


Figura 4.2: Etapas generales de la metodología propuesta

4.1.1 Etapa de Conceptualización

Es la primera etapa de la metodología, en una visión general, ésta permite modelar la entrada del sistema, es decir, nos permite generar la transformación de las fuentes de datos a ontologías.

En principio al tener diversas fuentes de datos debe considerarse que éstas pertenecen a un mismo dominio, es decir, que contienen información relacionada a un mismo tema en específico.

Esta es una parte fundamental ya que en primera instancia puede decirse que no nos interesa en que forma o formato estén representadas (diagrama ER, XML, Esquemas, entre otras representaciones) dichas fuentes, dado que la entrada del sistema son las ontologías que representan a las fuentes de datos.

Por lo tanto, debe darse a la tarea de crear la representación equivalente en una ontología de cada fuente de datos original.

En particular la metodología se ejemplifica utilizando bases de datos relacionales.

4.1.2 Etapa de Alineación Semántica

Esta es la etapa principal de la metodología ya que en ella se concentran los esfuerzos para poder integrar de manera eficaz y eficiente las diferentes fuentes de información. Mediante la aplicación a las ontologías generadas en la etapa previa de diferentes técnicas de alineación, las cuales empleen medidas de similitud, se obtiene un archivo de mapeo o una relación cuantificada entre las ontologías, según los aspectos que los algoritmos empleados consideran, el cual se usa para poder recuperar la información de una manera unificada. En otras palabras puede decirse, que este archivo sirve para consultar la información conociendo, una sola estructura o lenguaje y ésta será relacionada con las ontologías.

4.1.3 Etapa de Recuperación y Visualización

Se considera que en la etapa previa se obtuvo una salida exitosa, un archivo de alineación, el cual indica la relación entre los diferentes elementos de las ontologías.

Por lo tanto, se utilizó un motor de búsqueda que permite recuperar información conociendo la estructura o lenguaje de una fuente de datos. Se puede obtener información que sea similar en todas las fuentes de datos. Dentro de la visualización esta información que recuperada puede ser utilizada como la entrada a un sistema de información. Por ejemplo, en el escenario de trabajo que se plantea más adelante, esta información recuperada será la entrada a un sistema de visualización geográfica.

4.2 Diseño de la Conceptualización

Esta tarea es primordial para todo sistema de información, no importa cuan sofisticado, poderoso o preciso sea el sistema para resolver algún problema en específico, si no se cuentan con los datos o la información necesaria para alimentarlo o éstos presentan algún tipo de error, es probable que el sistema no funcione de manera adecuada o que arroje resultados poco confiables.

La metodología abarca la creación y validación de la información inicial, con la cual el sistema trabajará y se divide en dos procesos:

- Análisis de las fuentes de información
- Creación de ontologías.

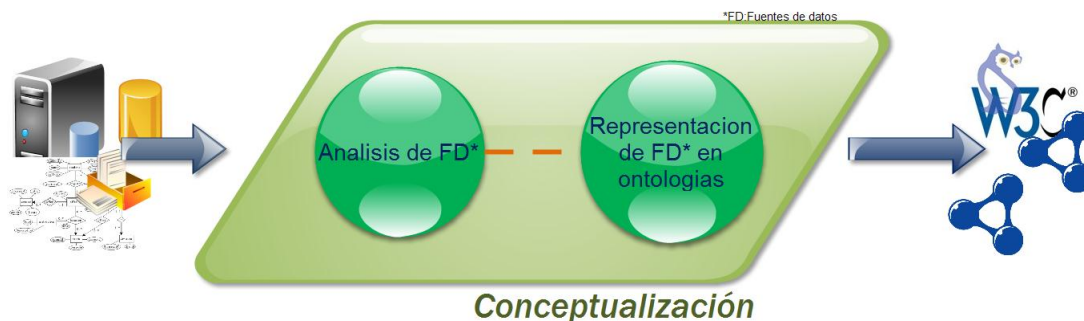


Figura 4.3: Etapa de conceptualización

La figura 4.3 muestra la etapa de conceptualización propuesta, en la cual se trabaja con los datos de entrada, en donde nuestras fuentes de datos son bases de datos; las cuales son sometidas a un proceso de análisis para poder obtener ciertas características que sirven como guía para el siguiente proceso, el cual es la creación de la ontología que representa a cada fuente de información, las cuales se obtienen al finalizar este proceso y sirven como la información de entrada a la siguiente etapa de la metodología. Ambos procesos se describen con mayor detalle más adelante en esta sección.

4.2.1 Análisis de las fuentes de información

En esta sección, se desarrollan los puntos necesarios para preparar las fuentes de información para rectificar que los datos son los adecuados para el objetivo particular que se desea, además de facilitar el proceso de transformación hacia una ontología.

El objetivo particular de esta sección es facilitar la creación de las ontologías que representan a las fuentes de datos, en nuestro caso particular Bases de Datos representadas mediante un modelo relacional. Con el análisis de estos modelos para obtener una lista de características básicas, y resaltar si se diera el caso características que sirvan para la integración. Como primer paso se debe conocer y delimitar el dominio en el cual se trabaja, es decir, de qué tema o área en particular se utiliza.

Por ejemplo en la figura 4.4: las fuentes de información pertenecen al dominio de las plantas, por lo cual si una de nuestras fuentes de datos fuera de animales, se perdería la homogeneidad semántica de la información dado que los animales no pertenecen al dominio

de las plantas.



Figura 4.4: Ejemplo de un Dominio: Plantas

Hasta cierto punto se debe tener cuidado y definir de manera clara y concreta el dominio en el cual se trabaja, ya que es fácilmente llegar a confundirse. Por ejemplo, las plantas y animales podrían pertenecer a un dominio común, el cual se puede llamar seres vivos, es decir, que las plantas y animales son un sub dominio del dominio seres vivos (véase la figura 4.5).

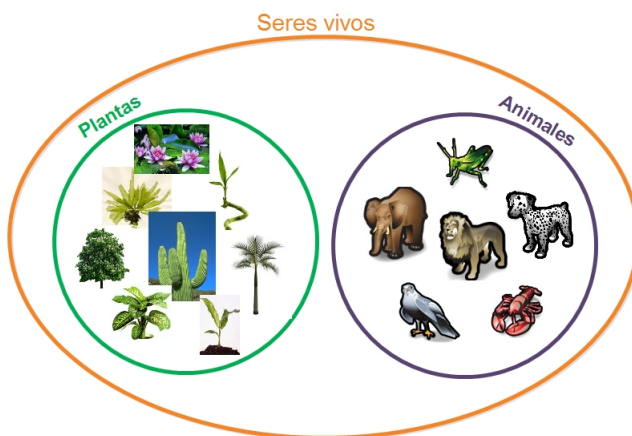


Figura 4.5: Ejemplo de Dominio y subdominio

A simple vista se da la sensación de que se puede integrar a ambos, pero todo depende sobre que dominio en el cual se integrarán, en este ejemplo para no perder la heterogeneidad en el dominio, se podrían integrar bajo el concepto de seres vivos.

Al delimitar y conocer el dominio se asegura que los datos contenidos en las bases de datos hablen de lo mismo o es la conceptualización de un mismo contexto, los datos tienen

homogeneidad semántica. Al conocer el dominio bajo el cual se desea que pertenezcan las fuentes de datos, se asegura que éstas se encuentren dentro de este dominio, es decir, hablen o se refieran a un mismo tema.

El dominio D se puede definir como:

$$D = (C, \vartheta) \quad (4.1)$$

Donde:

C es un conjunto.

ϑ es una ley de composición interna.

Por lo tanto, un dominio está formado por un conjunto de elementos y por una ley de composición interna que los relaciona. Dicha ley de composición interna indica los componentes o rasgos en común que comparten los elementos del conjunto.

Entonces, supóngase que las fuentes de datos a integrar A y B , describen o modelan el mismo dominio D cada una con su propio modelo conceptual: M_A y M_B . Donde M_A es representado por el conjunto de atributos K_A y M_B por el conjunto de atributos K_B . Entonces el conjunto de atributos puede ser descrito por $A = \{E_A, K_A\}$ y el conjunto de datos B por $B = \{E_B, K_B\}$ respectivamente, donde E_A es el conjunto de entidades del dominio D en A descritas por los atributos K_A de M_A y así como E_B es el conjunto de entidades del dominio D en A descritas por los atributos K_B de M_B [21].

Por lo tanto, el dominio de la aplicación está formado por:

$$D_\delta = \{E_A, (K_A \cap K_\Delta)\} \cup \{E_B, (K_B \cap K_\delta)\} \quad (4.2)$$

donde:

$\{E_A, (K_A \cap K_\delta)\}$ representa el conjunto de datos de A .

$\{E_B, (K_B \cap K_\delta)\}$ representa el conjunto de datos de B .

K_δ representa a los atributos del dominio indicados por la ley de composición interna.

De forma general, si se tienen n fuentes de información, que describen al mismo dominio de aplicación, al integrar estas n fuentes de datos se podría tener acceso al dominio del conjunto de datos descritos por:

$$D_\Delta = \bigcup_{i=1}^{i=n} \{E_i, (K_i \cap K_\Delta)\} \quad (4.3)$$

Los retos de la integración de fuentes de datos basadas en ontologías, es que no importa como estén representados los datos inicialmente, ya que habrá una ontología que los

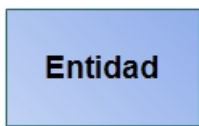


Figura 4.6: Representación de Entidad en el modelo ER

represente.

Es importante se tenga la certeza de que los datos contenidos en estas fuentes a utilizar son confiables.

En algunas ocasiones el conseguir los datos necesarios, es algo complicado debido a las miles de formas que existen. Hoy en día, una de las más comunes son las bases de datos, en particular relacionales, las cuales son fáciles de entender y tienen un gran detalle en cómo están conformadas.

En el caso de esta metodología se ejemplifica específicamente para trabajar con las bases de datos relacionales, y dentro de los modelos de representación más conocido los diagramas Entidad-Relación propuesto en [22], con el cual opera esta metodología. La mayoría de modelos posteriores a éste, se basan en las características que este nos presenta, fácilmente se puede acoplar el análisis de información de cualquier modelo de representación como UML, IEDF1X, Baker, Bachman.

Existen varios planteamientos para generar una ontología de una base de datos relacional, los cuales abarcan los lenguajes de descripción de bases de datos ya que son más expresivos en el sentido de que la mayoría de componentes y la semántica de la base de datos están definidas. Para llegar a esto en la manera tradicional conlleva un número de pasos o métodos ya establecidos para su obtención.

En los diagramas Entidad-Relación se pueden encontrar los siguientes elementos que lo conforman:

Entidad: Menor objeto con significado en una instancia. Cada registro guardado en un archivo. Se representan por medio de rectángulos (véase figura 4.6).

Atributos: Componente que determina una entidad. Cada atributo tiene asociado un dominio: Conjunto de valores que puede tomar. En otras palabras, los campos de los registros, se representan con elipses que se conectan por medio de líneas a las entidades o relaciones (véase figura 4.7)

Entre los atributos se encuentran:

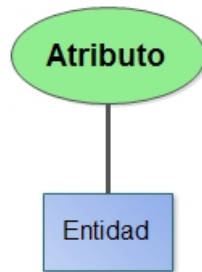


Figura 4.7: Representación de atributos en el Modelo ER



Figura 4.8: Representación de atributos monovalorados en el modelo ER

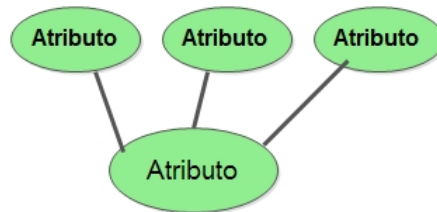


Figura 4.9: Representación atributos compuestos en el modelo ER

Atributos monovalorados y multivalorados: Se llaman atributos multivalorados a aquellos que pueden contener más de un valor simultáneamente, y monovalorados a los que sólo pueden contener un valor. Se representan por medio de una elipse con doble línea (véase figura 4.8).

Atributos simples y compuestos: Se dice que un atributo es compuesto cuando puede descomponerse en otros componentes o atributos más pequeños, y simple en otro caso. Los componentes de un atributo se representan a su vez como atributos (véase figura 4.9)

Relaciones: Conjuntos de la forma $\{(e_1, \dots, e_n) | e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$ con e_i entidades y E_i conjuntos de entidades del mismo tipo. Se representan mediante rombos conectados a las entidades que relacionan (véase figura 4.10).

Si se tiene el modelo relacional que represente a nuestra fuente de información se puede obtener la mayoría de las características de los componentes y parte de la semántica de la



Figura 4.10: Representación de Relaciones en el modelo ER

base de datos.

Por eso se genera un pequeño diccionario de datos con base en el modelo Entidad-Relación para auxiliarnos a generar la ontología y ayudarnos a descubrir más información semántica que la base de datos contenga o se nos pueda escapar. En este se definen las entidades, sus atributos, y el tipo de dato de los atributos y alguna información adicional que se consideren de importancia, como restricciones, cardinalidad, rango, etc. Además de una tabla que contenga las relaciones, así como su cardinalidad y las tablas que relaciona y el nombre con la cual se identifique, opcionalmente se puede agregar una descripción de la relación.

En la figura 4.11 se observa una entidad sencilla compuesta por 4 atributos para obtener más características acerca de éstos se genera en caso de ser necesario, un pequeño diccionario de datos descrito en la tabla 4.1.

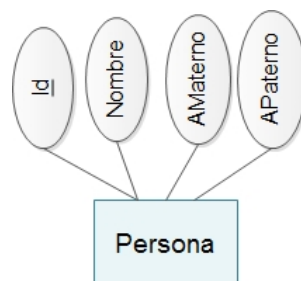


Figura 4.11: Entidad en el modelo E-R

Entidad	Atributo	Tipo de dato	Restricciones	Observaciones
Persona	Id	Numerico	Unico	Llave Primaria
Persona	Nombre	Cadena	-	-
Persona	AMaterno	Cadena	-	-
Persona	APaterno	Cadena	-	-

Tabla 4.1: Diccionario de datos de la figura 4.11

En la figura 4.12 se observa la representación de las relaciones entre diferentes entidades, la cual al analizarla generaría una tabla de relaciones como la tabla 4.2.

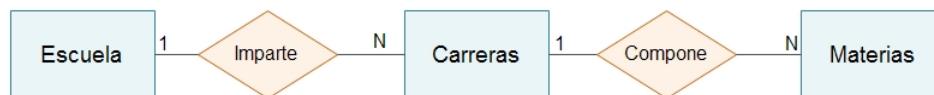


Figura 4.12: Entidad en el modelo E-R

Relacion	Entidad inicio	Entidad final	Observaciones
Imparte	Escuela	Carreras	Una escuela imparte una o varias carreras
Compone	Carrera	Materias	Una carrera se compone de varias materias

Tabla 4.2: Descripción de las relaciones de la figura 4.12

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Se debe considerar que nuestras bases de datos a utilizar deban de estar normalizadas ya que esto ayuda a encontrar la semántica de sus datos. Se debe tratar de tener una normalización hasta la tercera forma normal de preferencia, con algunas excepciones debido al posterior uso de ontologías. Aunque la metodología propuesta en este trabajo puede utilizar un modelo sin normalizar.

En conclusión se pueden definir los siguientes pasos:

- Conocer el objetivo de la aplicación.
- Establecer el dominio de trabajo.
- Validar las fuentes de información.
- Generar diccionario de datos.
- Generar tabla de relaciones.

4.2.2 Transformación del modelo Entidad-Relación a una Ontología

Las ontologías como su definición lo indica permiten representar o modelar el conocimiento. Por lo cual con ellas es posible modelar, representar o abstraer el conocimiento de cualquier lugar. El proceso de la creación de las ontologías en esta metodología es de forma manual. Se crean a partir de los modelos que representan las fuentes de información, en nuestro caso particular, modelos relacionales de las fuentes de datos descritas por bases de datos relacionales.

Por medio de una serie de reglas de transformación, se pretende representar los elementos del modelo relacional con elementos de las ontologías (ver figura 4.13).

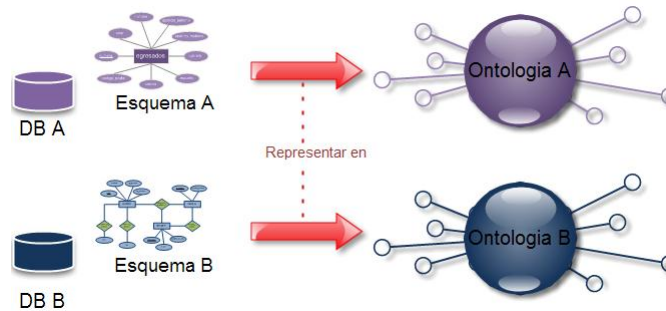


Figura 4.13: Representación de las Fuentes de Datos

Para representar las ontologías en un lenguaje computacional se ha seleccionado OWL. La 3WC considera a OWL como el estándar para desarrollar ontologías en lo que es la web semántica, debido a su gran poder expresivo que presenta este lenguaje.

Así mismo como apoyo para el desarrollo de las ontologías se eligió la herramienta Protégé [57]. Con Protégé se crean ontologías bajo el Lenguaje OWL en un marco de gráfico.

Primeramente se tienen que conocer las reglas de transformación que están basadas en las presentadas en [69], las cuales están contenidas en la tabla 4.3, donde están representados los elementos del modelo ER y el elemento con el que será representado en la ontología OWL.

Modelo ER	Ontología OWL
Entidad	Clase
Atributo	Datatype property
Atributo llave	Functional datatype property con la restricción mínima de uno
Atributo Compuesto	Clase con las propiedades correspondientes que conforman el atributo compuesto.
Relación Binaria sin atributos	Par de inverse object properties
Cardinalidad mínima	Restricción de cardinalidad mínima
Cardinalidad máxima	Restricción de cardinalidad máxima

Tabla 4.3: Mapeo de elementos del modelo Entidad-Relación a una Ontología

Primera regla de Transformación

1. Transformar cada entidad en una clase.
2. Mapear cada atributo simple a un functional datatype property.
3. Mapear cada atributo mono y multi valorado en un datatype property.
4. Mapear cada atributo compuesto en una nueva clase y en ella mapear los atributos que lo conformen en functional datatype properties. Crear un object property al cual se asocia su rango con la nueva clase creada y su dominio con la clase que contiene el atributo compuesto.
5. A cada atributo llave transformado como un functional datatype property asignarle la cardinalidad mínima en 1.

Segunda regla de Transformación

1. Mapear cada relación binaria sin atributos en un ObjectProperty el cual tendrá como dominio la clase que represente a la entidad de procedencia y como rango a la clase que represente a la entidad destino, además se genera otro object property, el cual será inverso al original, es decir, los objective properties tendrán la propiedad de inverseOf entre ellos.
2. Mapear la cardinalidad de las entidades participantes en restriction cardinality max y min o combinar las características en una functional property con una min cardinality restriction.

Ejemplificar estas dos reglas que son las más sencillas y abarcan la mayoría de elementos del modelo ER.

Por lo tanto, sea el Modelo E-R de la figura 4.14, que modela dos entidades con sus atributos y una relación entre ellas. Dentro de la entidad alumno se tiene un atributo compuesto y un atributo multivaluado.

Como primer paso hay que crear un diccionario de datos, el cual nos ayude a la transformación e identificación de elementos, así como una tabla para identificar las relaciones. Como se presenta con anterioridad en la sección 4.2.1.

Ahora con la ayuda del diccionario de datos se nota que se tienen dos entidades: Alumno y Clase, las cuales se transforman en clases de OWL (véase figura 4.15).

Se transforman los atributos simples en FuntionalDatatypeProperty y se les asigna como dominio la clase a la que pertenecen tal como se muestra en la figura 4.16.

Posteriormente se transforman los atributos mono o multivaluados haciéndolos DataTy-peProperty y se les asigna su dominio a la clase que pertenecen (ver figura 4.17)

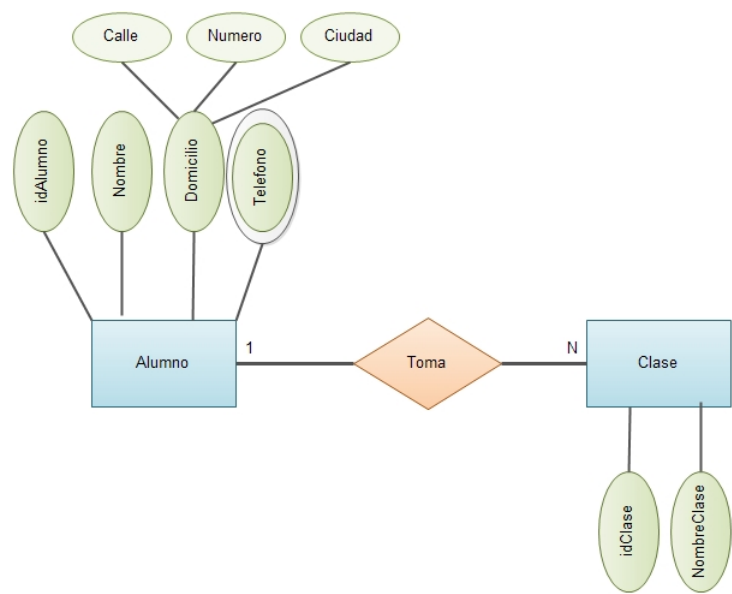


Figura 4.14: Ejemplo de Modelo ER

Entidad	Atributo	Tipo de dato	Restricciones	Observaciones
Alumno	idAlumno	Numérico	Único	Llave Primaria
Alumno	Nombre	Cadena	-	-
Alumno	Domicilio	Compuesto	-	-
Alumno	Calle	Cadena	-	Compone a Domicilio
Alumno	Numero	Numérico	-	Compone a Domicilio
Alumno	Ciudad	Cadena	-	Compone a Domicilio
Alumno	Telefono	Cadena	-	Multivaluado
Clase	idClase	Numérico	Único	Llave Primaria
Clase	NombreClase	Cadena	-	-

Tabla 4.4: Diccionario de datos de la figura 4.14

Relacion	Entidad inicio	Entidad final	Observaciones
Toma	Alumno	Clase	Un alumno toma una o varias clases

Tabla 4.5: Descripción de las relaciones de la figura 4.14

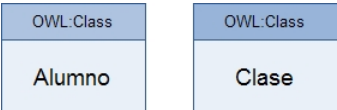


Figura 4.15: Transformación de entidades en clases

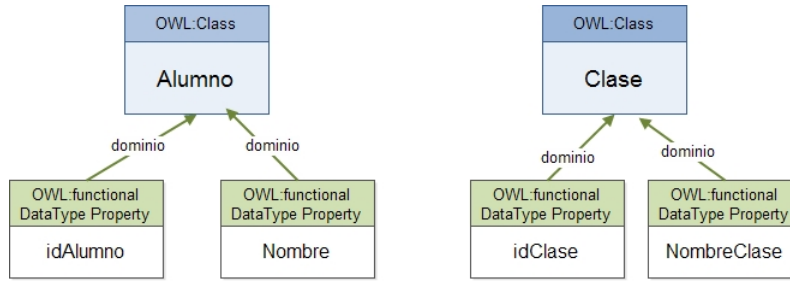


Figura 4.16: Transformación de atributos simples

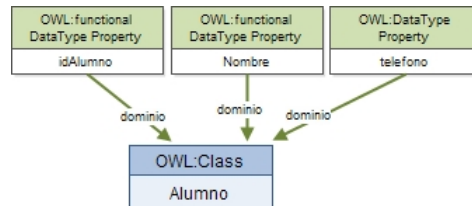


Figura 4.17: Transformación de atributos mono y multivaluados

Se transforman los atributos compuestos, los cuales generan una nueva clase y los atributos que lo componen se convierten en `functionalDataTypeProperty` y se les asigna su dominio a la nueva clase creada. Se genera un `objectProperty` "tieneDomicilio" y se asigna su rango a la nueva clase creada y su dominio a la clase que contiene al atributo compuesto (véase figura 4.18). Se recomienda elegir para la etiqueta del object property creado que sea un nombre que lo identifique con su dominio contruyendola de la siguiente manera "verbo (tiene) + nombre atributo compuesto".

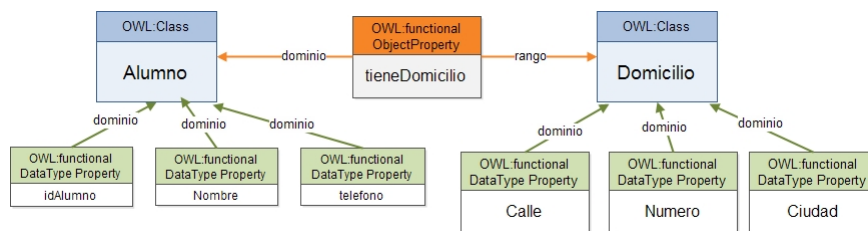


Figura 4.18: Transformación de atributos mono y multivaluados

Para terminar se asigna a los `functionalDataTypeProperty` su cardinalidad a 1 en caso de que éstos sean llaves en el modelo ER. Al hacer esto se asigna una restricción para este `functionalDataTypeProperty` (véase figura 4.19).

La segunda regla nos permite transformar las relaciones que se encuentran en el modelo Entidad-Relación, para ello se debe consultar la tabla de relaciones 4.5, la cual nos indica las relaciones presentes, así como las entidades a las cuales relaciona. Como establece la

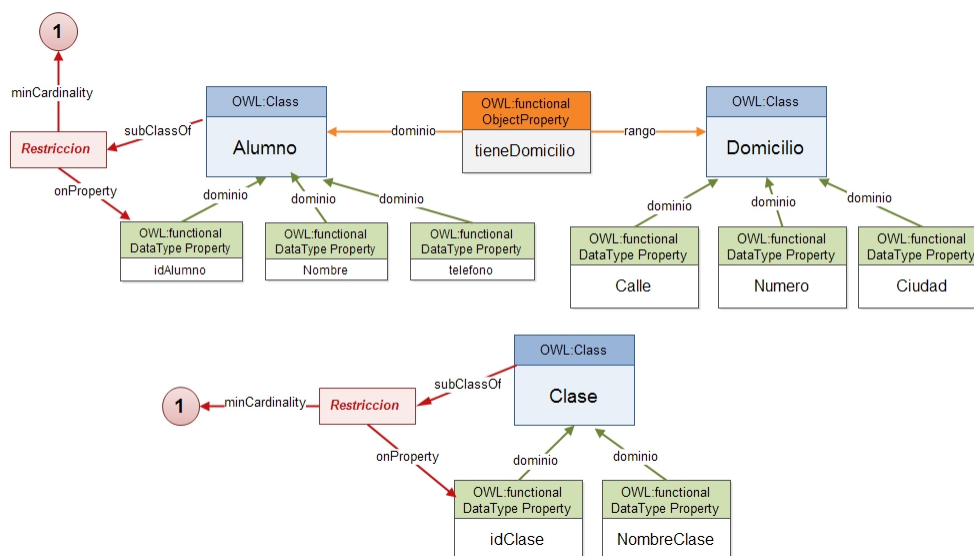


Figura 4.19: Transformación de atributos mono y multivaluados

regla de transformación tomamos la relación “Toma”y se genera el objectProperty “*tomaClase*” con rango en “Clase”y dominio en “Alumno”, además se genera una relación inversa “*tieneAlumnos*” con rango en “Alumnos”y dominio en “Clase”. Además a estos objectProperties se les asigna la propiedad de inverseOf tal como se muestra en la figura 4.20.

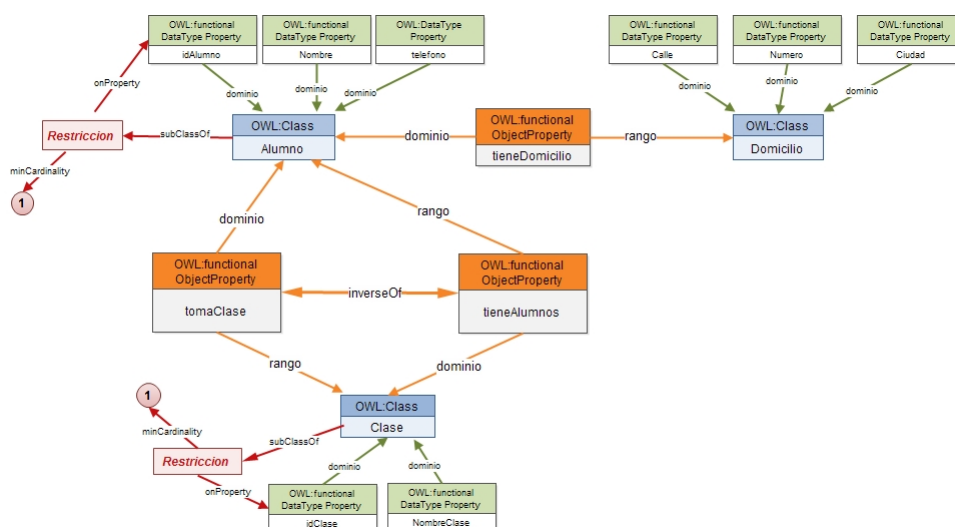


Figura 4.20: Transformación de relaciones binarias

Mediante el uso de estas reglas además se puede tener una aproximación para mapear otro tipo de modelos de representación de bases de datos como el modelo relacional, o

mediante un lenguaje de descripción como SQL.

4.3 Diseño de la Alineación Semántica

La alineación de ontologías es una técnica que se utiliza para encontrar las correspondencias de una ontología a otra, mediante un mapeo que se obtiene al ponderar diversas características establecidas o que son de nuestro interés, tal como se muestra en la figura 4.21.

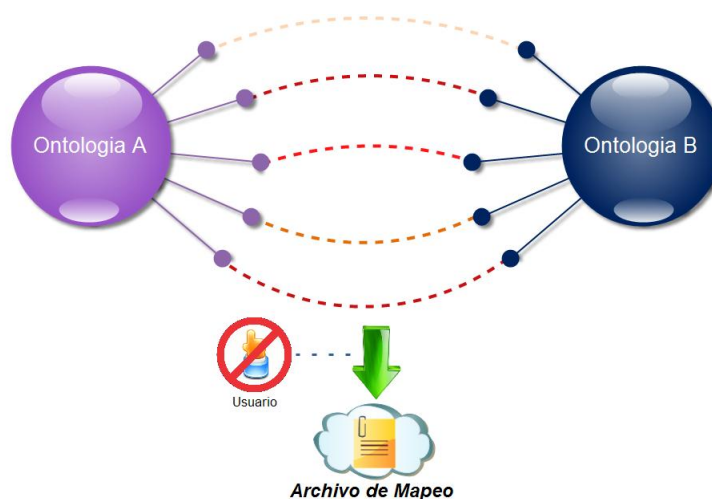


Figura 4.21: Alineación de ontologías

Realizar un método de alineación es un trabajo de investigación por sí solo, por lo que se considera utilizar un algoritmo o un marco de trabajo, seleccionado de una prueba de diferentes técnicas para así elegir el que mejor convenga según se considere.

4.3.1 Detección de correspondencias semánticas

Para encontrar las correspondencias semánticas entre los elementos de las dos ontologías de entrada, se define un conjunto de medidas de similitud a utilizar, para posteriormente evaluar el resultado obtenido del cómputo de la similitud y finalmente optimizar los valores de similitud encontrados.

Las medidas de similitud que se han considerado para la alineación en este trabajo se han dividido en dos grupos:

- **Similitud basada en términos (léxica):** Se enfoca en el nombre de las entidades de las ontologías: clases y propiedades.
- **Similitud semántica:** Se enfoca en los componentes que caracterizan a una clase.

- *Similitud entre propiedades*: considera las coincidencias existentes entre las propiedades de dos clases.

4.3.2 Similitud léxica

En los métodos definidos en la Ontology ALigment Api [31] la distancia de edición está normalizada, es decir como resultado del cálculo de similitud se obtiene un valor que va de 0 y hasta 1.

La distancia de Levenshtein

Creada e implementada por Vladimir Levenshtein en 1965 [59], con el propósito de medir la diferencia entre dos secuencias de símbolos [43].

La distancia de Levenshtein entre dos secuencias $x \in C^*$ con $n = |x|$, $m = |y|$ está definida como:

$$D(x, y) = n + m - 2\zeta(x, y) \quad (4.4)$$

Donde $\zeta(x, y)$ es la similitud de entre las secuencias x y y . Los límites de esta distancia se logran por un lado cuando la similitud entre las secuencias comparadas es nula, y en el otro extremo, cuando la similitud entre secuencias comparadas es máxima. Cuando la similitud es nula (secuencias sin símbolos comunes), la distancia es $n + m$. Cuando la similitud es máxima (se comparan secuencias iguales), la distancia es 0.

La idea general de esta distancia es que dos secuencias distan entre sí tanto como símbolos se deban borrar y símbolos se deban agregar, para hacer iguales ambas secuencias. De modo que el límite máximo de esta distancia se debe leer como: se deben borrar todos los n símbolos de x y agregar todos los m símbolos de y .

$$0 \leq D(x, y) \leq n + m$$

Métrica de caracteres para el Alinamiento de ontologías SMOA

La distancia de SMOA* se perfila para ser la métrica más utilizada para realizar la alineación de ontologías [1].

Para encontrar la similitud entre dos entidades, la distancia de SMOA toma en cuenta las diferencias y similitudes de los caracteres que las entidades poseen. Formalmente la distancia de Samoa entre dos cadenas s_1 y s_2 se define por [80]:

$$smoa(s_1, s_2) = comm(s_1, s_2) - diff(s_1, s_2) + winklerImpr(s_1, s_2) \quad (4.5)$$

*String Metric for Ontology Aligment

Donde $comm(s_1, s_2)$ es la similitud entre s_1 y s_2 definida en la expresión 4.6, $diff(s_1, s_2)$ es la diferencia definida en la expresión 4.7, y $winklerImpr(s_1, s_2)$ es el método de Jaro-Winkler [83] [54] para mejorar el resultado.

$$comm(s_1, s_2) = \frac{2 \times \sum_1 length(maxComSubString_i)}{length(s_1) + length(s_2)} \quad (4.6)$$

$$diff(s_1, s_2) = \frac{uLen_{s_1} \times uLen_{s_2}}{p + (1 - p) \times (uLen_{s_1} + uLen_{s_2} - uLen_{s_1} \times uLen_{s_2})} \quad (4.7)$$

4.3.3 Similitud semántica

Similitud entre propiedades

Este método utiliza la similitud entre los nombres de clase y propiedades, considerando la propiedad de la simetría al calcular la similitud entre propiedades [37].

El objetivo es identificar si cada propiedad p_n del conjunto de propiedades P de una clase C_1 , coincide con otra propiedad p'_M del conjunto de propiedades de P' de otra clase C_2 para realizar dicha comparación entre cada propiedad, se utilizan las etiquetas de ambas propiedades como entrada de una medida de similitud léxica.

Sean A y B los conjuntos de propiedades de las clases a y b , la función de similitud está definida por la ecuación 4.8.

$$S(a, b) = \frac{|A \cap B|}{|A \cup B|} \quad (4.8)$$

4.3.4 Método para la Alineación de Ontologías

Como se describe en la sección de este trabajo, la alineación de ontologías se describe mediante un archivo de mapeo.

Uno de los componentes de este archivo de mapeo es una medida de confianza la cual se llamará medida de similitud. Esta medida se aplica en la metodología y nos indica que tan parecidos son dos elementos de una ontología. Siendo el valor máximo de 1 que nos indica qué elementos son iguales y siendo el valor mínimo de 0 indicando que elementos son distintos entre sí. Por lo tanto, se debe considerar la ponderación de la tabla 4.6 para esta medida de similitud.

La forma más simple de conseguir este mapeo entre las ontologías, es hacerlo de forma manual, esto requiere que el usuario que realice este proceso debe ser un experto en el dominio de las ontologías. Otra desventaja es la cantidad de tiempo que se tiene que dedicar para hacer esta alineación, ya que para ontologías sencillas no es visto este esfuerzo como mucho, pero hay ontologías muy complejas o con gran cantidad de elementos, además por

Valor de similitud	Ponderación
1	Iguales o Idénticos
0.9 - .075	Muy semejantes
0.74 - 0.5	Semejantes
0.4 - 0.2	Poco semejantes
0.2.-0	Distintos

Tabla 4.6: Ponderación de la medida de similitud

la naturaleza de las ontologías éstas tienen o pueden crecer.

Entonces un ejemplo sencillo y manual de una alineación es el siguiente:

Considere dos ontologías las cuales pertenecen al dominio de los medios de transporte, para encontrar la relación que existe entre los elementos de ambas, es necesario un experto en este dominio para así asegurarnos de que estos mapeos sean confiables ya que en la alineación manual cada mapeo que se forma es una equivalencia completa entre los elementos de las ontologías.

Como se observa en la figura 4.22 se tiene la representación de las dos ontologías y los mapeos que se pueden encontrar entre los elementos de ambas, marcados por las líneas punteadas.

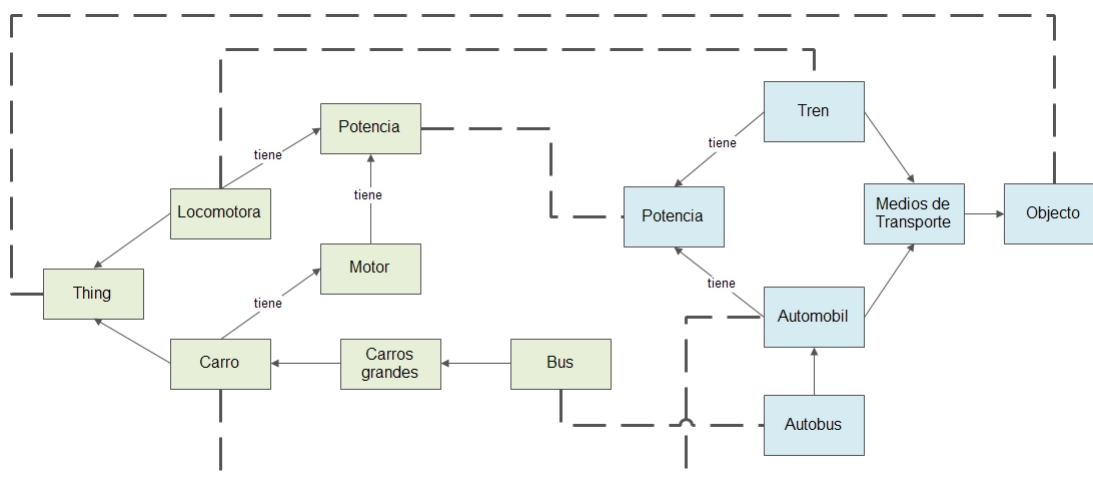


Figura 4.22: Alineamiento manual de dos ontologías

En la tabla 4.7 se aprecia el mapeo resultante de la alineación. Dado que el mapeo es manual se supone que todas las medidas de similitud entre los elementos es 1.

Ontología 1	Ontología 2
Thing	Objeto
Carro	Automovil
Locomotora	Tren
Potencia	Potencia
Bus	Autobus

Tabla 4.7: Alineamiento de la figura 4.22

Para evitar la tarea de realizar la alineación de forma manual hay aproximaciones que emplean diferentes técnicas para encontrar las similitudes entre ambas, así mismo estas aproximaciones no son del todo automáticas y en algunas veces al ser procesos computacionales no descubren todos los mapeos posibles o pueden llegar a encontrar mapeos incorrectos, actualmente se investiga y desarrollan sobre estos temas para hacer este proceso de forma automática y lo mas precisos posible.

Cualquier método o algoritmo de alineación es útil y dan como resultado un archivo de mapeo en formato XML o RDF, en este caso se utiliza uno compuesto por XML/RDF, el cual está compuesto por tres elementos: un elemento de *alineación*, un elemento de *descripción* de las ontologías y la parte más importante los *mapeos* entre los elementos.

Este último, el cual contiene la descripción de las dos entidades descritas por la URI, la relación que existe entre ambas en este caso de equivalencia y la medida de similitud que como hemos mencionado tiene un rango de 0 a 1.

La tabla 4.8 muestra parte de la estructura del archivo de alineación específicamente donde se describe la relación entre las entidades y su respectiva medida de similitud.

Con este archivo de mapeo se puede obtener en caso de ser posible, la información integrada de las ontologías que estén descritas en el archivo de alineación. Este proceso se describe en la siguiente sección.

4.4 Diseño de la Recuperación y Visualización

En la etapa anterior se obtiene un archivo de mapeo el cual nos permite relacionar ambas ontologías, se crea la relación entre conceptos para poder recuperar información de manera unificada, tal como se muestra en la figura 4.23.

En esta etapa es necesario validar los archivos de mapeo que son el resultado de la alineación, ya que esta alineación al ser semiautomática pudiera ocurrir uno de los siguientes casos:


```

<map>
  <Cell>
    <entity1 rdf:resource='http://www.example.org/ontology1#entidad1' />
    <entity2 rdf:resource='http://www.example.org/ontology2#entidad2' />
    <relation>fr.inrialpes.exmo.align.impl.rel.EquivRelation</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>
0.466666666666667</measure>
  </Cell>
</map>
<map>
  <Cell rdf:about="#veryImportantCell">
    <entity1 rdf:resource='http://www.example.org/ontology1#entidad1' />
    <entity2 rdf:resource='http://www.example.org/ontology2#entidad1' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0
</measure>
  </Cell>
</map>

```

Tabla 4.8: Archivo de alineación

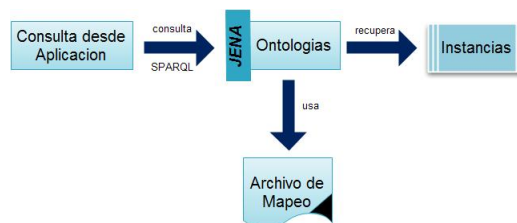


Figura 4.23: Proceso de Recuperación semántica de información

- El usuario necesita crear mapeos que no estén contenidos dentro del archivo de alineación.
- Eliminar un mapeo el cual considere erróneo.
- Modificar la ponderación de la similitud en un mapeo.

Después de validar el archivo de alineación se puede proceder a recuperar la información contenida en las ontologías a partir de un motor de consultas.

4.4.1 Motor de consultas

La recuperación de información por parte del usuario implica que el usuario debe conocer uno de los modelos de las fuentes de datos a integrar. El usuario ejecuta una consulta única acerca de la información contenida en el modelo y ésta debe de resolverse para tratar de

recuperar información de las fuentes de datos disponibles.

Con la solución propuesta, la alineación crea una interfaz para poder transformar esta consulta sobre un modelo en una consulta adecuada para poder consultar el otro u otros modelos que se estén utilizando, sí y solo sí existen mapeos descritos en el que se relacionen a los elementos solicitados del modelo de la consulta original con los elementos del otro modelo.

Por medio de la API de JENA[†] se extrae información usando SPARQL[‡] desde las ontologías, según el tipo de consulta seleccionada. La mayoría de las formas en las consultas SPARQL contienen un conjunto de patrones de tripletas llamado patrón de grafo básico. Dichos patrones se asemejan a las tripletas RDF con la excepción de que el sujeto, predicado y objeto pueden ser una variable. Estas consultas se componen de tres partes: cláusulas "**SELECT**" donde se especifican las propiedades a considerar, cláusulas "**WHERE**" para condicionar la búsqueda y cláusulas "**Filter**" para realizar un filtro sobre los resultados obtenidos con base en un parámetro.

El resultado de la consulta SPARQL consiste en una lista de soluciones posiblemente desordenada denominada secuencia de solución la cual representa las posibles formas en las que el patrón del grafo básico de la consulta coincide con los datos. Puede entonces haber cero, una o múltiples soluciones a una consulta.

Para generar una nueva consulta a partir de la consulta original es necesario identificar los componentes que se están solicitando, y la ontología a la cual se envía esta consulta original.

Después se ejecuta el siguiente algoritmo para generar la consulta:

1. Se busca que la ontología de la búsqueda original esté contenida en el archivo de alineación, es decir, el archivo de alineación nos indica a que ontologías pertenece por lo cual la ontología de la búsqueda debe estar descrita en el archivo de alineación. Si no está descrita en el archivo, detiene este proceso y se pasa al último punto.
2. Se buscan las entidades de tipo clase que se hayan detectado en el consulta para tratar de encontrar un mapeo que las relacione con un elemento de la otra ontología.
3. Se buscan los elementos restantes de la consulta en el archivo de mapeo para ver si se puede encontrar una relación con un elemento de la otra ontología.
4. Si no se encontraron relaciones entre elementos de las ontologías no se genera una nueva consulta SPARQL, o si se encontraron relaciones entonces se sustituyen los

[†]Las especificaciones de JENA se encuentran en el Anexo B

[‡]Las especificaciones de SPARQL se encuentran en el Anexo B

elementos encontrados en los elementos de la consulta original, creando así una nueva consulta.

5. Se ejecutan según sea el caso, una o ambas consultas.

Supóngase que se cuenta con dos ontologías, las cuales pertenecen al dominio de ventas de automóviles, y se desea conocer el nombre y la ubicación de cada concesionaria. Esta información esta descrita en las ontologías según la figura 4.24, y la alineación entre ambas está descrito según la tabla 4.9.

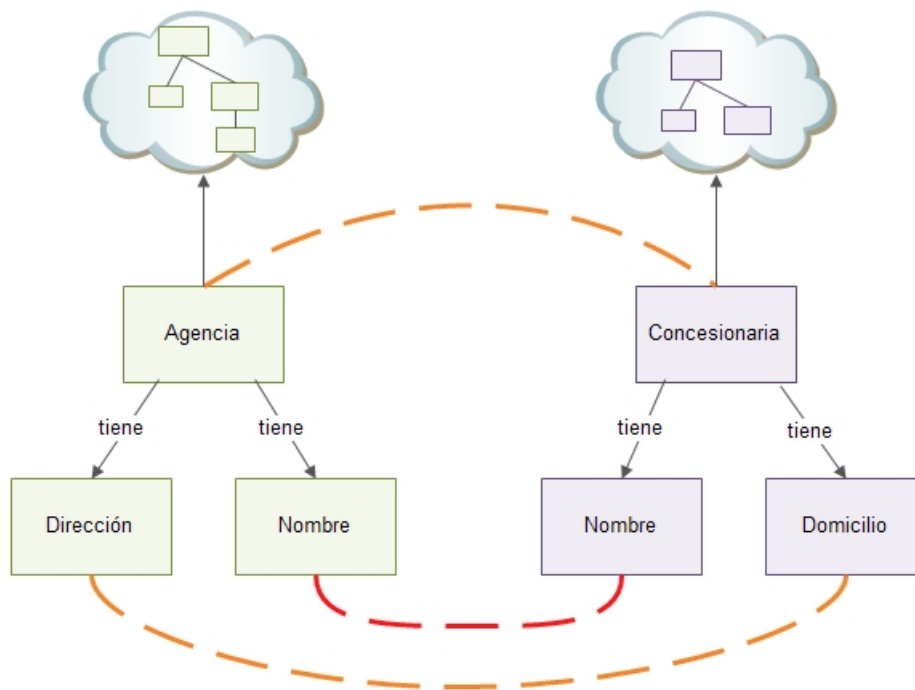


Figura 4.24: Descripción de las ontologías y su alineamiento

Ontología 1	Ontología 2	Medida Similitud
Concesionaria	Agencia	0.856352
Concesionaria:nombre	Agencia:nombre	1
Concesionaria:domicilio	Agencia:direccion	0.796521

Tabla 4.9: Archivo de alineación para la figura 4.24

El usuario solo conoce el modelo de una de las ontologías, el de la figura 4.24, por lo tanto genera una consulta resultante que se desarrolla en SPARQL es la siguiente:

```
SELECT ?nombre, ?domicilio
```

```
Where (
Concesionaria onto1:Ubicación ?domicilio
Concesionaria onto1:Nombre ?nombre
```

En este caso, todos los mapeos son aceptables (véase la tabla 4.9) para la integración, dado que su medida de similitud es mayor de 0.6. Entonces por medio de estos mapeos se genera una nueva consulta, la cual se muestra a continuación:

```
SELECT ?nombre, ?direccion
Where (
Agencia onto2: Dirección ?dirección
Agencia onto2:Nombre ?nombre
```

El resultado de estas consultas en tripletas, es información recuperada de ambas ontologías si es el caso:

```
Agencia #001 onto1:Nombre "Automotriz Lindavista"
Agencia #001 onto1:direccion "Av. IPN N.23 Col. Lindavista
Agencia #002 onto2:Nombre "Automotriz Kioto"
Agencia #002 onto2:direccion "Av. Del Trabajo Col. Progreso"
Concesionaria #001 onto1:Nombre "Mazda Vallejo"
Concesionaria #001 onto1:direccion "Av. Montevideo N.1205 Col. San Bartolo"
Concesionaria #001 onto2:Nombre "Mylsa"
Concesionaria #001 onto2:direccion "Eje central s/n Col. Industria"
```

La visualización de los datos dependerá de las necesidades del usuario, ya que estos pueden ser visualizados de diversas formas, tales como grafos, recuadros, líneas o símbolos particulares. En un caso particular estos pueden ser recuperados en forma de un objeto ResultSet en el caso de un ambiente de programación JAVA, para interactuar con diversos sistemas o exportarlos a diferentes formatos.

Capítulo 5

Pruebas y Resultados

No hay camino para la verdad, la verdad es el camino.

MAHATMA GANDHI

5.1 Plantamiento del escenario de trabajo

Para poder probar y evaluar los elementos de la metodología propuesta en este trabajo, se considera emplear un caso de estudio. Para poder comparar ciertas características de esta metodología, se ha considerado emplear un caso de estudio el cual haya sido empleado en un caso similar de integración. Por lo cual se retoma caso de estudio muy similar al planteado en [21], el cual se describe de manera general a continuación.

Se necesita proporcionar la búsqueda de profesionales egresados de una institución de educación en caso particular del Instituto Politécnico Nacional, el cual nos ha proporcionado información acerca de sus egresados de diferentes carreras y programas educativos de nivel superior y posgrado. Los criterios de búsqueda que se requieren son los siguientes: por carrera, escuela de procedencia, edad, y por nivel educativo.

Estos datos se encuentran en dos bases de datos relacionales distintas y heterogéneas, ya que el registro de estos egresados se realiza de forma separada dependiendo el nivel de la carrera de egreso es decir: si es de nivel superior o posgrado. Las estructuras planteadas en este trabajo son hipotéticas, debido a que no se cuenta con las estructuras reales para el manejo de esta información y se crearon para el desarrollo de esta propuesta con base en lo planteado en [21].

En un sentido general se requiere que la aplicación pueda servir de referencia para desarrollos posteriores que tengan por objetivo: Unificar búsquedas, de egresados de esta casa de estudios, sin ningún tipo de restricción o interoperabilidad con sistemas institucionales.

5.2 Implementación de la metodología

5.2.1 Conceptualización

En esta sección como se plantea en la sección 4.2 de este trabajo, se procede a analizar nuestras fuentes de información y posteriormente transformar nuestras ontologías.

Se tienen dos fuentes de información, Bases de datos de egresados del IPN*. Una de ellas contiene información acerca de alumnos egresados de un nivel de estudios de posgrado, con 1000 registros contenidos en ella, la cual está representada por el diagrama entidad relación de la figura 5.1 y a la cual, se refiere a ella para fines prácticos más adelante como BD_1 . Así mismo se cuenta con otra base de datos la cual representa a los alumnos egresados del nivel de estudios superior, con 500 registros contenidos en ella, la cual está representada mediante el diagrama entidad relación de la figura 5.2, y a la cual se refiere más adelante a ella como BD_2 .

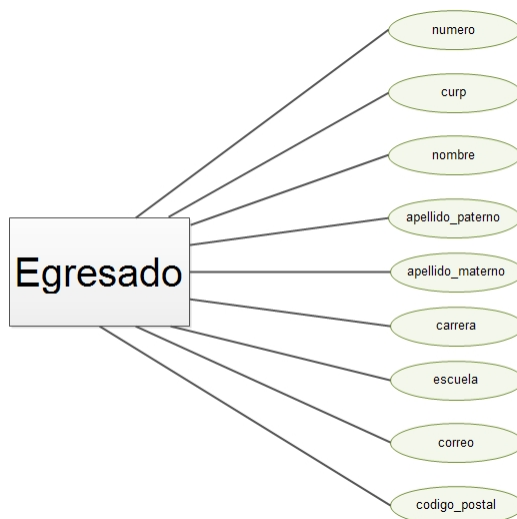
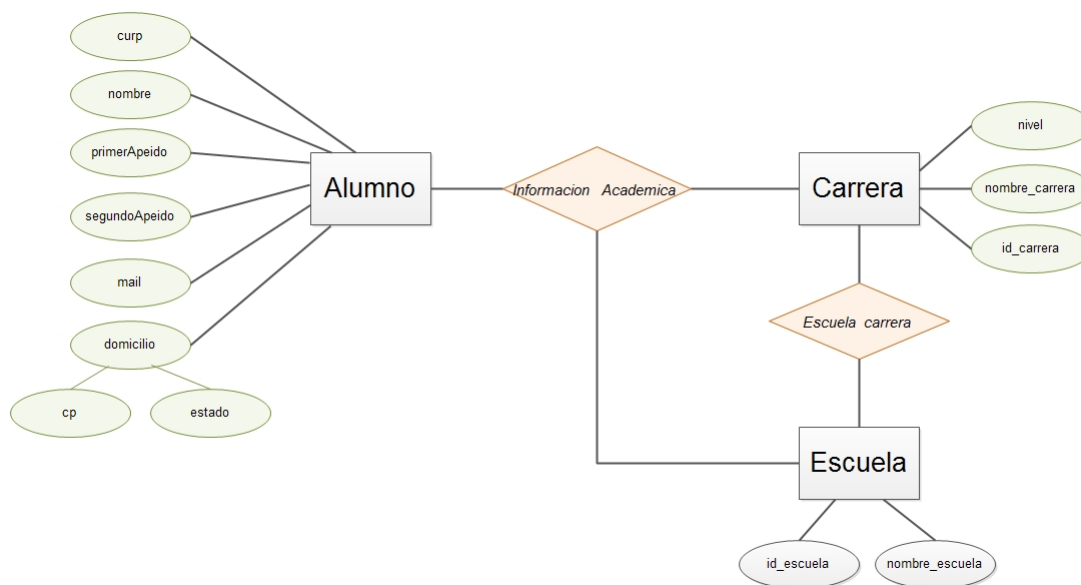


Figura 5.1: Diagrama ER de la BD_1

Análisis de las fuentes de información

Como primer paso se debe definir el dominio de nuestras fuentes de datos, el cual en el caso de estudio planteado es claro: Las fuentes de datos se encuentran en el dominio de

*Instituto Politécnico Nacional

Figura 5.2: Diagrama ER de la BD_2

alumnos egresados de una institución de educación. En otras palabras las fuentes de datos contienen información acerca de personas que han concluido alguna carrera en una escuela de una institución educativa.

Con el dominio claro se debe proceder a analizar los diagramas entidad relación para obtener características presentes en ellos.

Se comienza analizando la BD_1 , analizando el diagrama de la figura 5.1, se nota que está compuesta de elementos según la tabla 5.1. Además la tabla 5.2 nos describe cada entidad y los atributos que la componen así como sus características (diccionario de datos).

#Entidades	#Atributos	#Relaciones
1	9	0

Tabla 5.1: Numero de elementos del diagrama ER de BD_1

Como se puede apreciar en la tabla 5.1, la BD_1 no cuenta con ninguna relación.

De la misma forma se analiza la BD_2 descrita en figura 5.2 la cual está compuesta por el número de elementos descritos en la tabla 5.3. Descritos en el diccionario de datos 5.4.

Además la BD_2 cuenta con tres relaciones descritas en la tabla 5.5.

Ya que se ha terminado este análisis se procede a convertir estos diagramas entidad relación a una ontología siguiendo las reglas descritas en la sección 4.2.2.

Entidad	Atributo	Tipo de dato	Restricción	Observaciones
Egresado	Boleta	Numérico	único	Llave primaria
Egresado	Curp	Alfanumérico	-	-
Egresado	Nombre	Caracteres	-	-
Egresado	Apellido_paterno	Caracteres	-	-
Egresado	Apellido_materno	Caracteres	-	-
Egresado	Carrera	Caracteres	-	-
Egresado	Escuela	Caracteres	-	-
Egresado	Correo	Caracteres	-	-
Egresado	Código_postal	numérico	-	-

Tabla 5.2: Diccionario de datos de BD_1

#Entidades	#Atributos	#Relaciones
1	9	0

Tabla 5.3: Número de elementos del diagrama ER de BD_2

Entidad	Atributo	Tipo de dato	Restricción	Observaciones
Alumno	Curp	alfanumérico	único	Llave primaria
Alumno	Nombre	Caracteres	-	-
Alumno	primerApellido	Caracteres	-	-
Alumno	segundoApellido	Caracteres	-	-
Alumno	Mail	Caracteres	-	-
Alumno	Domicilio	compuesto	Atributo compuesto	-
Alumno	cp	numérico	-	Atributo de domicilio
Alumno	estado	caracteres	-	Atributo de domicilio
Carrera	nivel	Caracteres	-	-
Carrera	Nombre_carrera	Caracteres	-	-
Carrera	Id_carrera	Numérico	único	Llave primaria
Escuela	Id_Escuela	Numérico	único	Llave primaria
Escuela	Nombre_escuela	caracteres	-	-

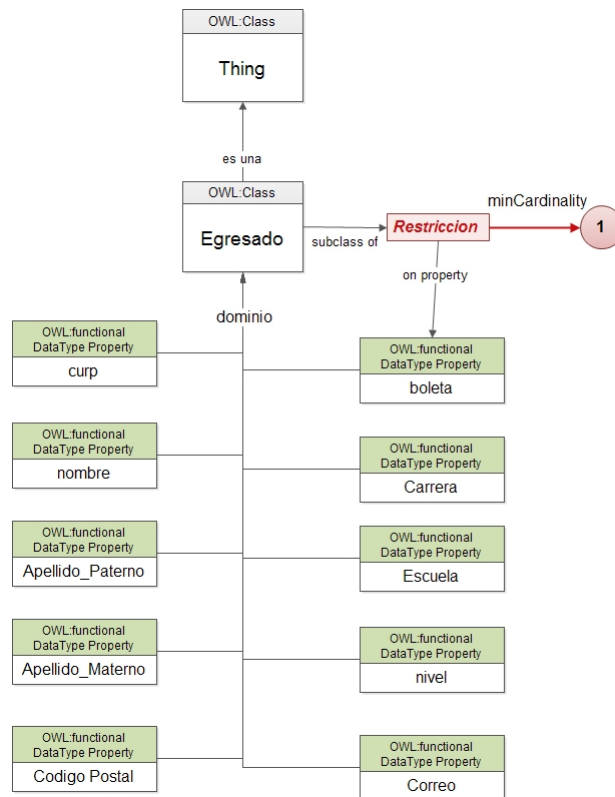
Tabla 5.4: Diccionario de datos de BD_2

5.2.2 Transformación del modelo ER a Ontologías

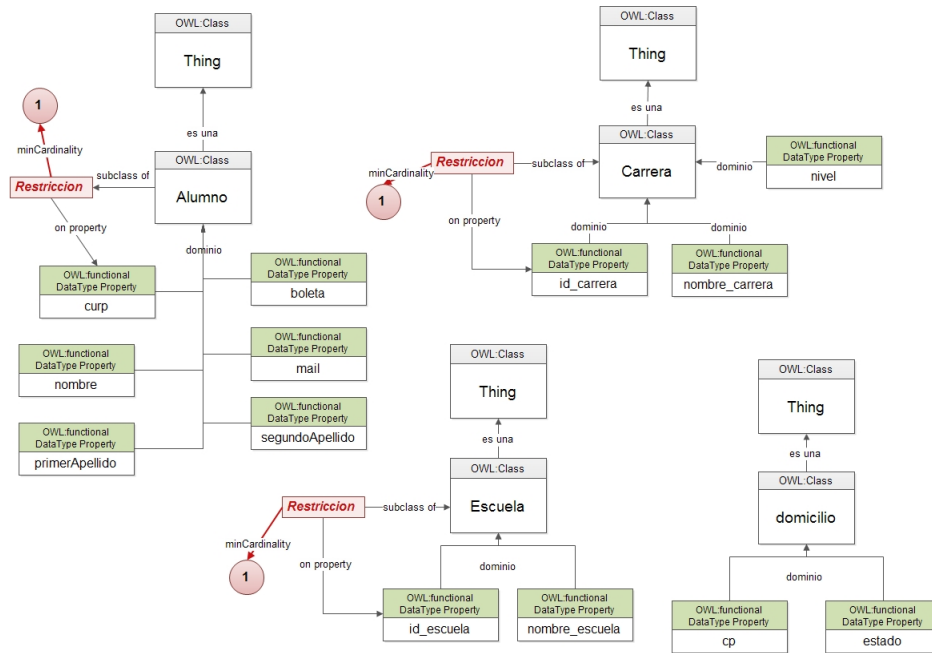
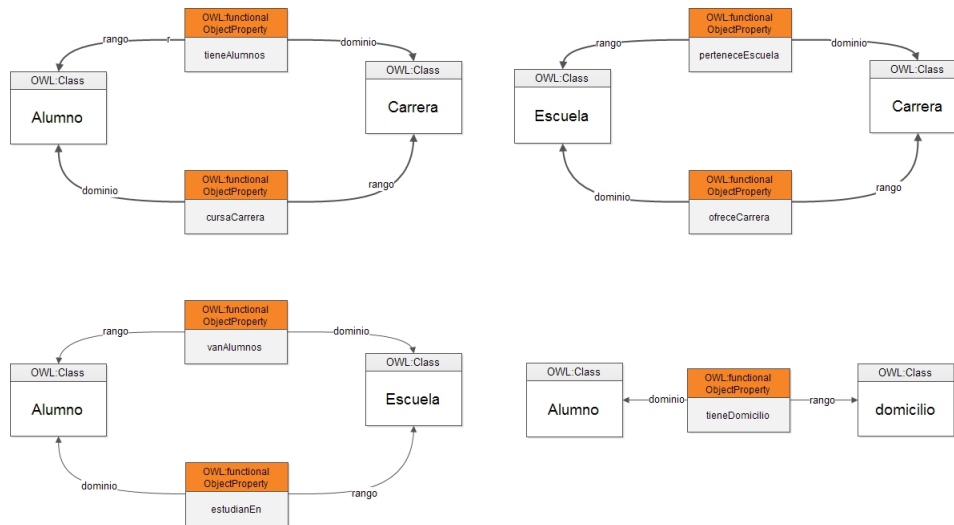
Se transforma primero el diagrama de la BD_1 aplicando los puntos de la primer regla de transformación se tiene el resultado de la figura 5.3. El archivo OWL generado se puede consultar en el apendice A.

Se procede ahora a transformar la BD_2 , después de aplicar la primer regla de transformación se obtiene lo mostrado en la figura 5.5. Así mismo aplicando la segunda regla de transformación se obtiene la figura 5.6.

Relación	Entidad inicio	Entidad final	Observaciones
Información Académica	Alumno	Carrera	Un alumno estudia una carrera
Información Académica	alumno	Escuela	Una alumno estudia en una escuela
Escuela_carrera	Escuela	Carrera	Una escuela imparte una o varias carreras

Tabla 5.5: Relaciones del diagrama ER de BD_2 Figura 5.3: Modelo de la ontología BD_1 a partir del diagrama ERFigura 5.4: Grafo de la ontología BD_1 generado en Protégé

Finalmente mediante el uso de Protégé se genera el archivo OWL el cual puede ser consultado en el apéndice A y el grafo producido se muestra en la figura 5.7

Figura 5.5: Modelo de la ontología de la BD_2 : Primera Regla de transformaciónFigura 5.6: Modelo de la ontología de la BD_2 : Segunda Regla de transformación

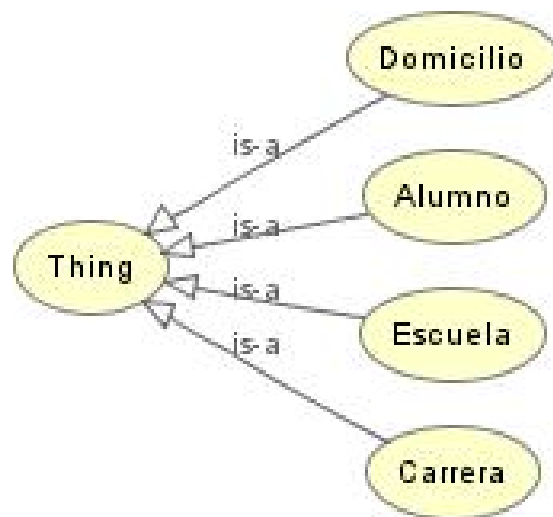


Figura 5.7: Grafo de la ontología de BD_2 generado en Protégé

5.3 Alineación

Ya construidas las ontologías que representan nuestras fuentes de información ahora se debe generar el mapeo correspondiente entre las entidades descritas en la ontología. En esta sección se seguirá usando el caso descrito anteriormente del dominio de agencias, recordando que nuestras fuentes de datos son las mostradas en la figura 5.1 y la figura 5.2.

Con el uso de la Aligment-API 4.0 [32] se generan diferentes mapeos los cuales son descritos a continuación. Además con esta API[†] se puede obtener el archivo de alineación en diferentes formatos.

El proceso para generar la alineación con esta API es el siguiente:

- Se leen dos ontologías OWL/RDF
- Crea un objeto de alineación.
- Computa la alineación entre estas ontologías.
- Despliega los resultados (aplicación, XML/RDF, HTML).

Como se describió anteriormente hay diversas técnicas para encontrar las relaciones de similitud entre cada ontología [7], en particular nos interesan la distancia de Levenshtein y SMOA para la parte lexica y la similitud de propiedades para la parte semántica.

Para la distancia de Levenshtein entre los elementos de la ontología se obtuvieron las similitudes mostradas en la tabla 5.6.

Ontología 1	Ontología 2	Distancia
Egresado	Escuela	0.25
carrera	id_carrera	0.7
curp	curp	1.0
escuela	id_escuela	0.7
apellido_paterno	estudiaEn	0.3125
apellido_materno	tieneDomicilio	0.3125
boleta	boleta	1.0
codigo_postal	boleta	0.23076
correo	id_carrera	0.4
nombre	nombre	1.0

Tabla 5.6: Distancia de Levenshtein

[†]Application Programming Interface - Interfaz de Programación de Aplicaciones

Ontología 1	Ontología 2	Distancia
Egresado	Escuela	0.05
carrera	id_carrera	0.9375
curp	curp	1.0
escuela	id_escuela	0.9375
apellido_paterno	primerApellido	0.6619
apellido_materno	segundoApellido	0.6619
codigo_postal	estado	0.4692
apellido_paterno	estudiaEn	0.1125
apellido_materno	tieneDomicilio	0.112
boleta	boleta	1.0
codigo_postal	boleta	0.0987
correo	id_carrera	0.5214
nombre	nombre	1.0

Tabla 5.7: Distancia de SMOA

La similitud de propiedades trata de descubrir correspondencias, en clases que tengan nombres diferentes pero que coincida en el mismo significado o su significado es muy similar y el caso contrario donde existan clases con el mismo nombre pero con propiedades distintas. Los valores obtenidos para esta similitud se muestran en la tabla 5.8.

Ontología 1	Ontología 2	Distancia
Egresado	Alumno	0.887
Egresado	Carrera	0.265
Egresado	Escuela	0.1368
Egresado	Domicilio	0.127

Tabla 5.8: Similitud de propiedades

En la figura 5.8 podemos apreciar la cantidad de mapeos realizados por los métodos seleccionados diferenciando los mapeos con una similitud mayor a 0.5 y menores a 0.5 así como la cantidad de mapeos encontrados por el método aplicado.

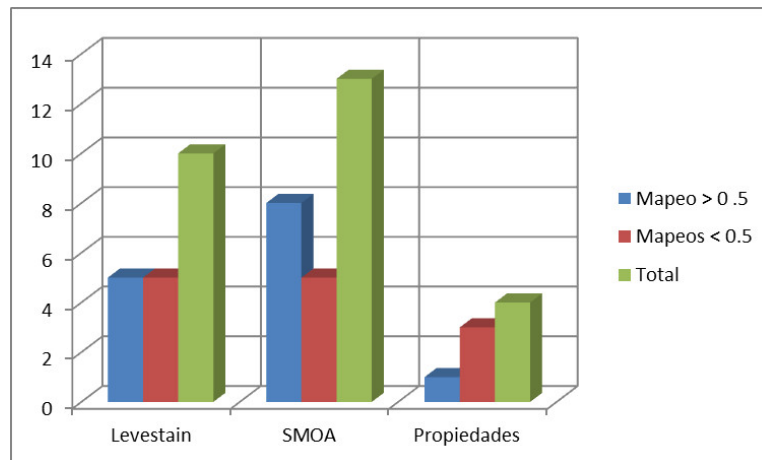


Figura 5.8: Comparación de mapeos con las técnicas de similitud aplicadas

5.4 Recuperacion de informacion

Una vez creadas nuestras ontologías es necesario acceder al contenido de ésta, lo cual se puede lograr mediante el lenguaje de consulta o recuperación SPARQL.

Las consultas implementadas en SPARQL reflejan el contenido semántico que posee la ontología; ya que realizan búsquedas directamente sobre conceptos en forma explícita y se utilizan las relaciones definidas en la ontología como criterios de búsqueda.

En la implementación, las consultas SPARQL se construyen bajo el framework de Jena a través de la QueryExecutionFactory. La clase QueryExecutionFactory proporciona un conjunto de métodos para crear objetos del tipo QueryExecution a partir de objetos Query. Se utiliza el método create (Query query, Model, model). QueryExecution es una interface para la ejecución de una consulta. El objeto Query representa la estructura de datos para la consulta en su forma externa, se obtiene a partir de la ejecución del método Query create (String queryString) de la clase QueryFactory, al cual se le da como argumento la cadena de consulta en lenguaje SPARQL. La cadena de consultas incluye los prefijos de los espacios de nombres (namespaces) utilizados para acceder a los recursos especificados. Model es la interface a través de la cual se implementa el Modelo OWL sobre el cual se hacen las consultas.

En general las consultas sobre las ontologías como ya lo hemos mencionado antes no difieren mucho de las consultas en lenguaje SQL, por lo cual la combinación de estas consultas nos podrán dar la información que se requiera específicamente. Posteriormente si se requiere, estas consultas podrían darnos información que no esté explícitamente descrita en la ontología (inferencia).

La consulta ?? es una consulta a la ontologia de las propuestas anteriormente, la cual solicita los elementos boleta, nombre, apellido Paterno, apellido Materno y carrera de la ontología que esten conformados por una carrera, además que el valor de la escuela sea la cadena "CIC" y que ordene este resultado por el elemento carrera.

```
PREFIX bd1: < http://www.owl-ontologies.com/Prueba-BD1.owl#>
PREFIX : <http://www.w3.org/2001/XMLSchema#string>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
        SELECT ?boleta ?apellido_Paterno
        ?aplleido_Materno ?nombre ?carrera

        WHERE
        ?egresado bd1: carrera ?carrera
```

```
FILTER regex(?escuela, "CIC")
```

```
ORDER BY ASC(?carrera)
```

Una vista parcial de los resultados de la consulta se muestran en la tabla 5.9

boleata	nombre	paterno	materno	carrera
B081509	Rosas	Soriano	Luis Ignacio	Doctorado en ciencias de la computación
B081494	Cadena	Martínez	Rodrigo	Maestria en ciencias de la computación
B081440	Zanatta	Juárez	Ángel Gabriel	Maestria en ciencias de la computación
B081438	Vizcarra	Romero	Julio César	Maestria en ciencias de la computación
A100358	Castelán	Romero	Juan Salvador	Maestria en ciencias de la computación

Tabla 5.9: Vista parcial de resultados de la consulta SPARQL

5.5 Visualización

Dentro de esta sección se muestra un prototipo el cual permite apreciar de forma gráfica las ontologías y la alineación generada de las etapas anteriores. Está desarrollado en el lenguaje de programación JAVA como una aplicación de escritorio.

La aplicación permite visualizar por separado las ontologías y su alineación mediante grafos generados a partir de sus respectivos archivos OWL y la alineación a partir del archivo XML que la representa. La figura 5.9 representa el diagrama de caso de la aplicación.

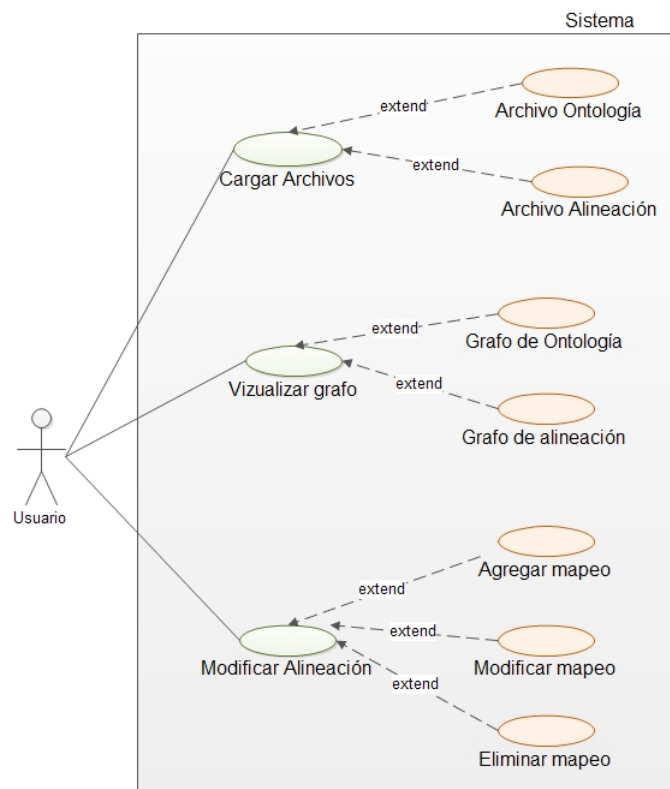


Figura 5.9: Diagrama de casos de uso de la aplicación

El funcionamiento de la aplicación es la siguiente, primero se deben de seleccionar los archivos OWL que representan las ontologías y si existe es posible cargar un archivo de alineación. (véase las figuras 5.10 y 5.11)

Ya con los archivos seleccionados (véase la figura 5.12) es posible ver su representación gráfica, mediante un grafo, seleccionando la opción grafo.

Se puede así poder apreciar tanto el grafo que representa a cada ontología (véase las figuras 5.13, 5.14) y un grafo que representa la alineación de ambas, si en el archivo de

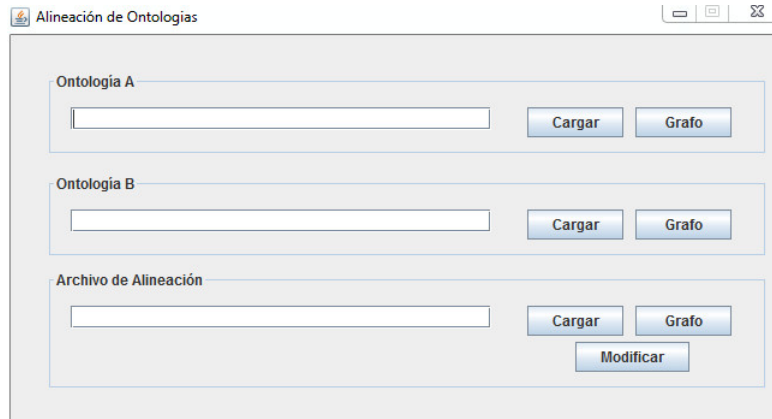


Figura 5.10: Aplicación de escritorio

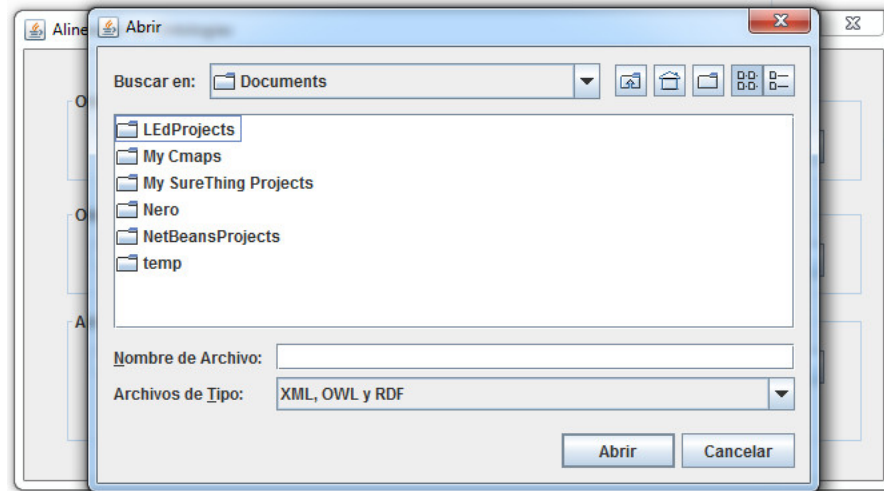


Figura 5.11: Cuadro de selección de archivos

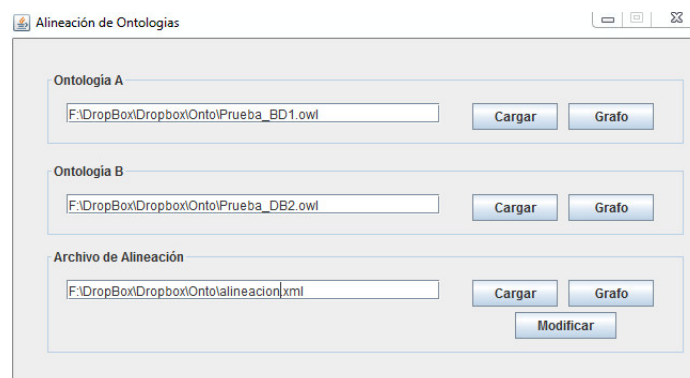


Figura 5.12: Archivos seleccionados

alineación están descritas las ontologías seleccionadas. (véase las figuras 5.15)

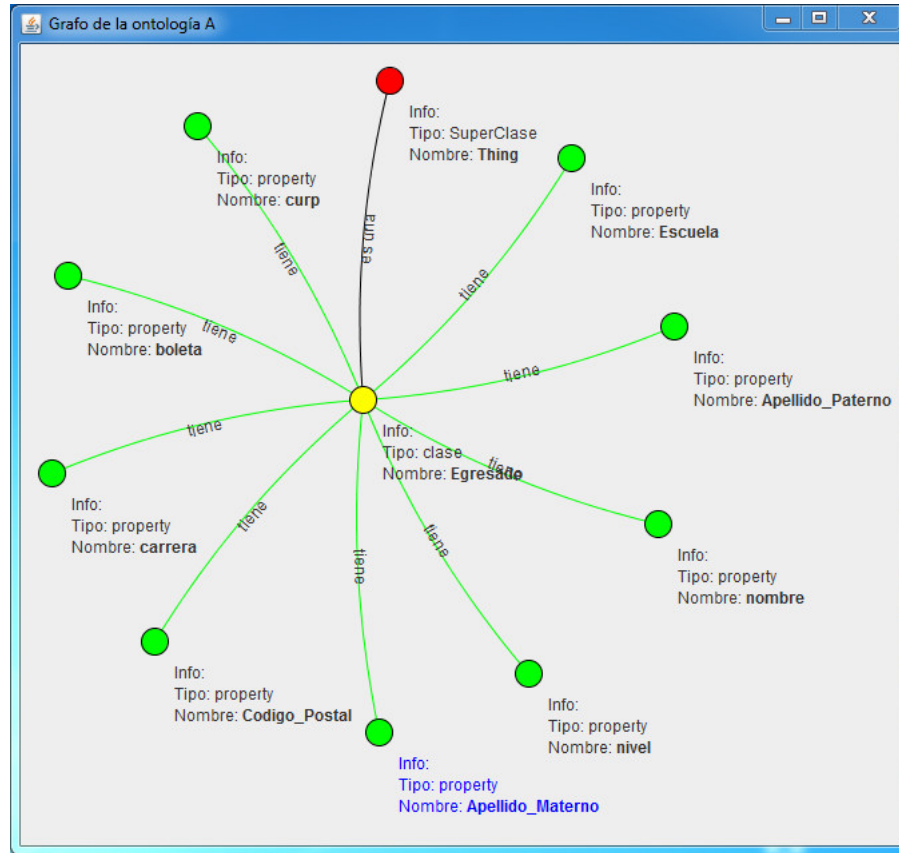


Figura 5.13: Grafo de la ontología A

c

Además se cuenta con una opción llamada *modificar* que está presente en la sección del archivo de alineación el cual permite editar el archivo de mapeo, ya sea agregando un mapeo, eliminando un mapeo o modificando el valor de similitud de un mapeo. (véase la figura 5.16)

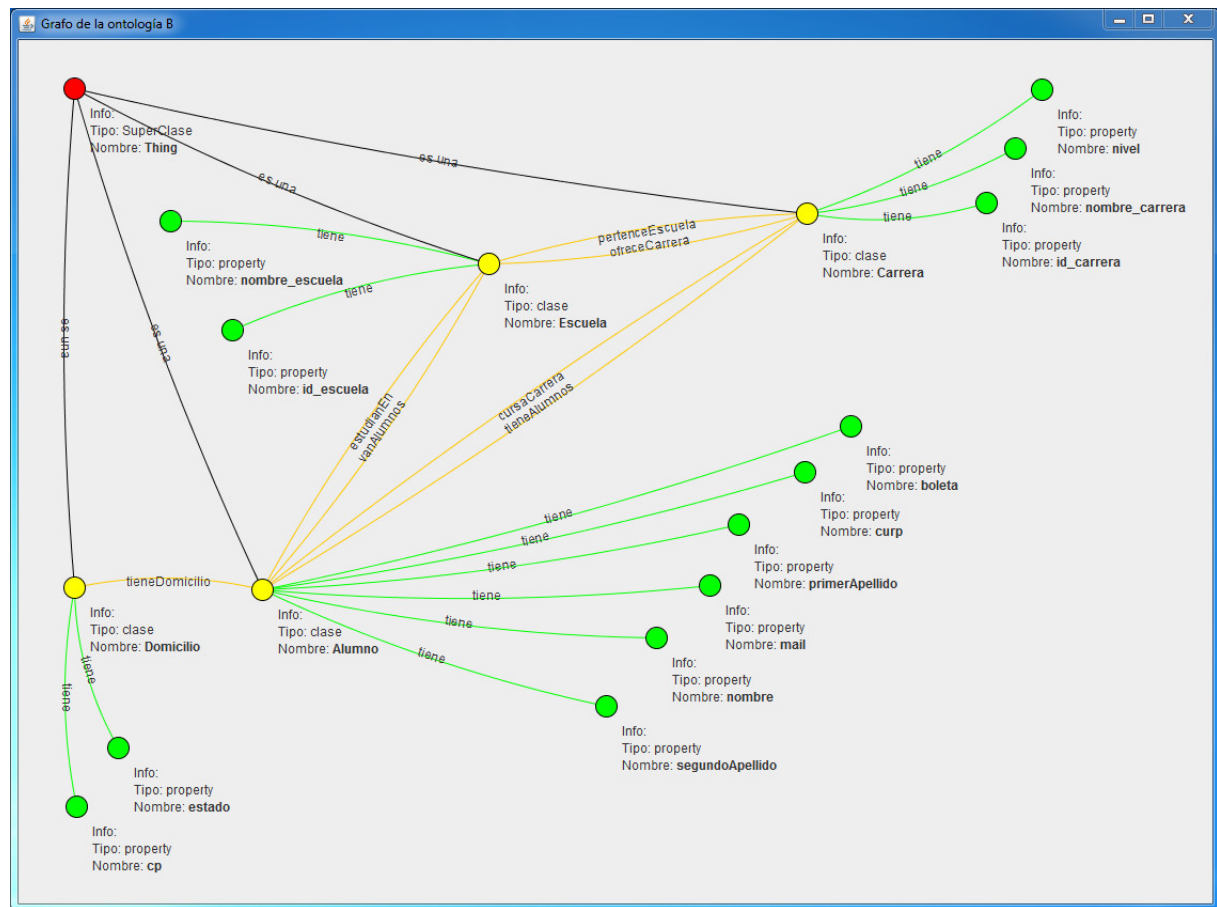
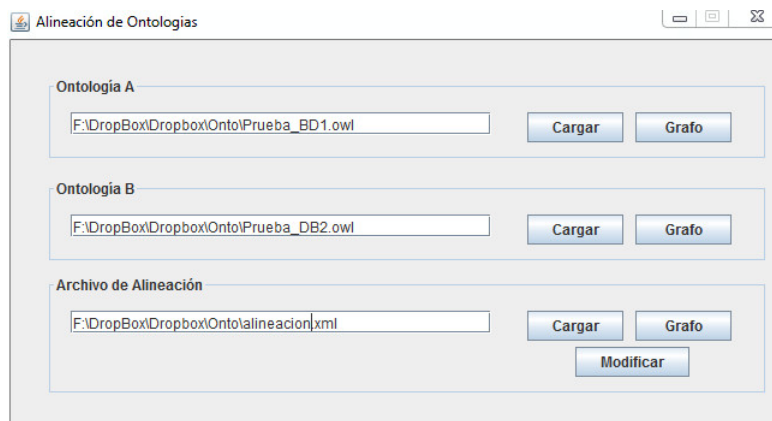
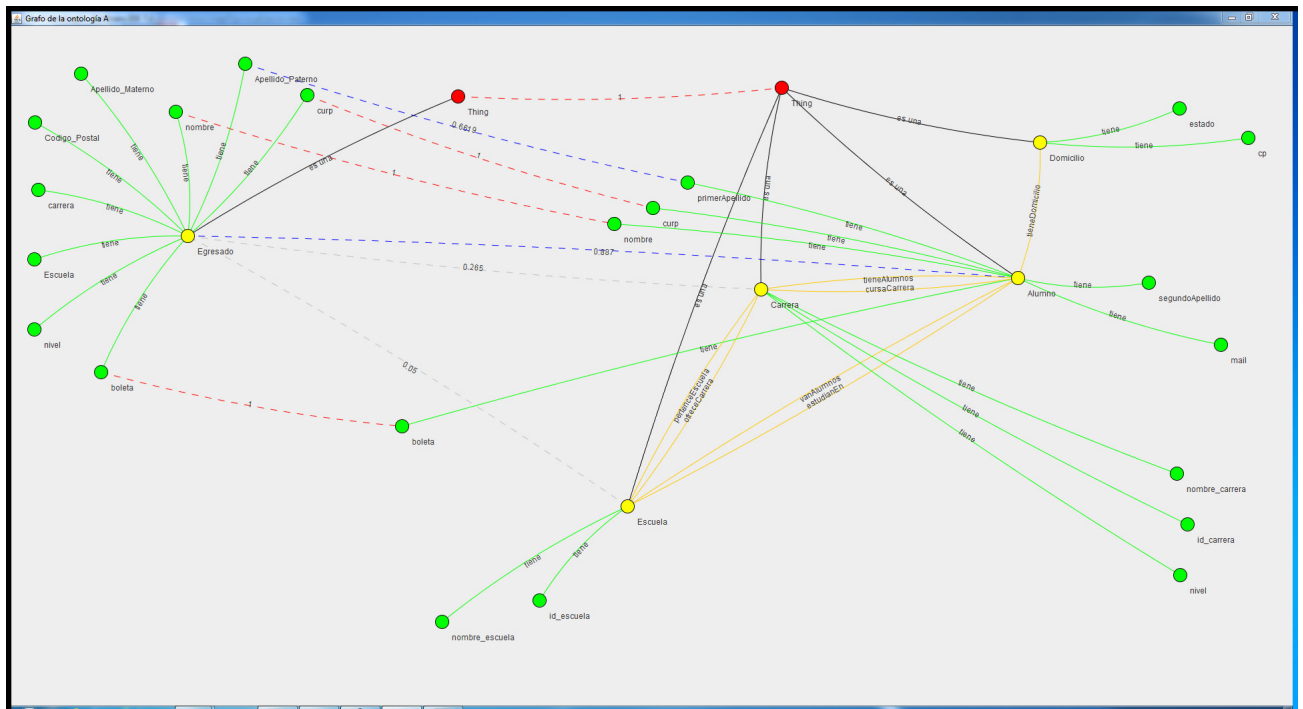


Figura 5.14: Grafo de la ontología A



Capítulo 6

Conclusiones

Se necesita poco para hacer las cosas bien, pero menos aún para hacerlas mal.

PAUL BOCUSE

Hoy en día, se maneja una gran cantidad de datos, por lo que es difícil encontrar información acerca de un tema específico en un repositorio o fuente de datos únicamente, generalmente esta información se encuentra dispersa. Esto significa que las fuentes de datos se encuentran distribuidas, son administradas por diferentes entidades y almacenadas en distintos formatos, esto es son heterogéneas en varios sentidos. Son estas diferencias en las fuentes de datos y particularmente, las que se refieren a la visión que cada una de ellas tiene del dominio que describen y el significado que dan los términos que se usan para modelar, las que dificultan la posibilidad de encontrar un método infalible para integrarlas.

En este trabajo se aborda una metodología basada en tres etapas principales, que utiliza las ontologías como herramienta principal para lograr la integración de las fuentes de datos. Esta metodología se enfoca en la utilización de fuente de datos: bases de datos relacionales, exclusivamente. Ya que actualmente la información se encuentra y se sigue trabajando con bases de datos heredadas cuya estructura, invariablemente, es relacional y dada su facilidad de implementación, seguramente la estructura de datos relacional se seguirá ocupando por algún tiempo, y que además, con esta metodología no solo sirva para integrar las fuentes de datos sino para transformarlas, en modelos de representación más complejos: ontologías.

El uso de ontologías ha cambiado y simplificado en gran medida la forma de cómo se plantea el problema de la integración de datos, a tal punto que se ha convertido en un paradigma de la integración de fuentes de datos heterogéneas. Esto se debe a que las on-

tologías cuentan con una gran capacidad de representar formalmente, con suficiente poder expresivo, cualquier dominio de aplicación a través de la definición explícita del significado de los conceptos y relaciones entre conceptos que intervienen en él.

El modelo de integración se basa en la alineación de ontologías, una técnica que sirve para encontrar correspondencias entre los elementos generalmente dos ontologías y caracterizar estas similitudes mediante un valor numérico. El uso de un conjunto de medidas de similitud como un componente en los procesos de alineación es necesario. La dificultad para expresar la importancia de estas medidas de similitud en un problema presente en los sistemas de alineación actuales. La expresividad de la importancia de las medidas de similitud, es crucial para discernir entre la heterogeneidad de las ontologías alineadas, para explotar las descripciones de un concepto y decidir si tiene relación con otros conceptos.

Que tanto se pueden conciliar las diferencias semánticas de las fuentes de datos, depende de la representación del dominio en la ontología. Por lo cual puede decirse, que la integración de datos basada en ontologías permite solucionar parcialmente el problema de la heterogeneidad semántica.

El objetivo de estas técnicas es la mínima participación del usuario, en los procesos de integración, lo cual se pretende llevar a cabo en esta metodología, pero aún es necesaria esta intervención para poder validar ciertos aspectos que permitan integrar las fuentes de información de manera confiable.

Referencias

- [1] G. Acampora, V. Loia, S. Salerno, and A. Vitiello. A hybrid evolutionary approach for solving the ontology alignment problem. *International Journal of Intelligent Systems*, 27(3):189–216, 2012. [cited at p. 45]
- [2] V. Alexiev, M. Breu, and J. de Bruijn. *Information integration with ontologies: experiences from an industrial showcase*. John Wiley & Sons, 2005. [cited at p. 22]
- [3] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Ontology-based integration of xml web resources. *The Semantic WebISWC 2002*, pages 117–131, 2002. [cited at p. 12]
- [4] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R.J. Miller, and J. Mylopoulos. The hyperion project: from data integration to data coordination. *ACM SIGMOD Record*, 32(3):53–58, 2003. [cited at p. 12]
- [5] Y. Arens, C.N. Hsu, and C.A. Knoblock. Query processing in the sims information mediator. *Advanced Planning Technology*, 32:78–93, 1996. [cited at p. 12]
- [6] Irina Astrova. *Rules for Mapping SQL Relational Databases to OWL Ontologies*, pages 415–424. Springer, 2009. [cited at p. -]
- [7] S. Bechhofer, R. Volz, and P. Lord. Cooking the semantic web with the owl api. *The Semantic Web-ISWC 2003*, pages 659–675, 2003. [cited at p. 60]
- [8] I. Benbasat and A.S. Dexter. Electronic data interchange and small organizations: adoption and impact of technology. *Management Information Systems Quarterly*, 19:465–486, 1995. [cited at p. 3]
- [9] S. Bergamaschi, F. Guerra, and M. Vincini. A peer-to-peer information system for the semantic web. *Agents and Peer-to-Peer Computing*, pages 161–182, 2005. [cited at p. 12]
- [10] T. Berners-Lee. Semantic web-xml2000. *W3C Website*, pages 10–0, 2000. [cited at p. -]
- [11] T. Berners-Lee, D. Fensel, J.A. Hendler, H. Lieberman, and W. Wahlster. *Spinning the Semantic Web: bringing the World Wide Web to its full potential*. MIT press, 2005. [cited at p. 3]
- [12] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001. [cited at p. 8]
- [13] Y. Bishr. Overcoming the semantic and other barriers to gis interoperability. *International Journal of Geographical Information Science*, 12(4):299–314, 1998. [cited at p. 12]

- [14] P. Borst, J. Benjamin, B. Wielinga, H. Akkermans, et al. An application of ontology construction. In *Proc. of ECAI-96 Workshop on Ontological Engineering. Budapest*, 1996. [cited at p. 24]
- [15] P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tessaris. Specification of a common framework for characterizing alignment. *Proc. of the 1st International Workshop on Peer-to-Peer Knowledge Management*, 2004. [cited at p. 27]
- [16] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml). *World Wide Web Journal*, 2(4):27–66, 1997. [cited at p. 4]
- [17] Agustina Buccella and Alejandra Cechich. An ontology approach to data integration. *October*, 3(2):62–68, 2003. [cited at p. -]
- [18] Guntars Bumans. Mapping between relational databases and owl ontologies : an example. *Science*, 756:99–117, 2010. [cited at p. -]
- [19] Alejandro Botello C. Aplicando técnicas de correspondencia semántica para la integración de bases de datos dispersas. CIC, Oct 2011. [cited at p. 22]
- [20] S. Camillo, C. Heuser, and R. Mello. Querying heterogeneous xml sources through a conceptual schema. *Conceptual Modeling-ER 2003*, pages 186–199, 2003. [cited at p. 12]
- [21] Alejandro Hernández Cardona. Integración de fuentes de datos espaciales con base en ontologías. Master’s thesis, IPN-CIC, Diciembre 2010. [cited at p. 14, 34, 53]
- [22] Peter Pin-Shan Chen. The entity-relationship model: toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, March 1976. [cited at p. 20, 35]
- [23] Peter Pin-Shan Chen. The entity-relationship model (reprinted historic data). In David W. Embley and Bernhard Thalheim, editors, *Handbook of Conceptual Modeling*, pages 57–84. Springer Berlin Heidelberg, 2011. [cited at p. 20]
- [24] Roger H.L. Chiang, Ee-Peng Li, and Veda C. Storey. Framework and knowledge for database integration. *Managing information technology resources and applications in the world economy*, page 218, 1997. [cited at p. 10]
- [25] I Cruz, W Sunna, N Makar, and S Bathala. A visual tool for ontology alignment to enable geospatial interoperability. *Journal of Visual Languages Computing*, 18(3):230–254, 2007. [cited at p. -]
- [26] I. Cruz, H. Xiao, and F. Hsu. Peer-to-peer semantic integration of xml and rdf data sources. *Agents and Peer-to-Peer Computing*, pages 108–119, 2005. [cited at p. 12]
- [27] I.F. Cruz and H. Xiao. Using a layered approach for interoperability on the semantic web. In *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 221–231. IEEE, 2003. [cited at p. 13]
- [28] Isabel F Cruz, William Sunna, and Anjli Chaudhry. *Semi-automatic Ontology Alignment for Geospatial Data Integration*, pages 51–66. Springer, 2004. [cited at p. 13]
- [29] Isabel F Cruz and Huiyong Xiao. The role of ontologies in data integration. *Science*, 13(4):1–18, 2005. [cited at p. 12, 13]
- [30] C.J. Date. *Introducción a los Sistemas de Bases de Datos*. Pearson Educación, 2001. [cited at p. 20]

- [31] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos. The alignment api 4.0. *Semantic Web*, 2(1):3–10, 2011. [cited at p. 45]
- [32] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment api 4.0. *Semant. web*, 2(1):3–10, January 2011. [cited at p. 60]
- [33] J. Davies, D. Fensel, and F. Van Harmelen. *Towards the semantic web*. Wiley Online Library, 2003. [cited at p. 3]
- [34] Anhai Doan and Alon Y Halevy. Semantic integration research in the database community: A brief survey. *Database*, 26(1):83–94, 2005. [cited at p. 12, 18]
- [35] Dejing Dou, Han Qin, and Paea Lependu. Ontograte: Towards automatic integration for relational databases and the semantic web through an ontology-based framework. *International Journal of Semantic Computing*, 04(01):123, 2010. [cited at p. -]
- [36] M. Ehrig. *Ontology alignment: bridging the semantic gap*. Semantic web and beyond. Springer, 2007. [cited at p. 23, 26]
- [37] Jérôme Euzenat, P. Guégan, and P. Valtchev. Ola in the oaei 2005 alignment contest. In *Integrating Ontologies Workshop Proceedings*, page 97, 2005. [cited at p. 46]
- [38] J. Euzenat. An api for ontology alignment. *The Semantic Web–ISWC 2004*, pages 698–712, 2004. [cited at p. 17]
- [39] Jérôme Euzenat, David Loup, Mohamed Touzani, and Petko Valtchev. Ontology alignment with ola. In *In Proceedings of the 3rd EON Workshop, 3rd International Semantic Web Conference*, pages 59–68. CEUR-WS, 2004. [cited at p. 14]
- [40] Muhammad Fahad. Er2owl: Generating owl ontology from er diagram. In Zhongzhi Shi, E. Mercier-Laurent, and D. Leake, editors, *Intelligent Information Processing IV*, volume 288 of *IFIP - The International Federation for Information Processing*, pages 28–37. Springer US, 2008. [cited at p. -]
- [41] Roberto E. Zagal Flores. Alineación de ontologías usando el metodo boosting. Master’s thesis, IPN-CIC, Diciembre 2008. [cited at p. -]
- [42] J.H. Gennari, M.A. Musen, R.W. Ferguson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003. [cited at p. 88]
- [43] A.E.C. González. La métrica de levenshtein. *Revista de Ciencias Básicas UJAT*, 7(2):35–43, 2008. [cited at p. 45]
- [44] T.R. Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993. [cited at p. 23]
- [45] T.R. Gruber et al. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human computer studies*, 43(5):907–928, 1995. [cited at p. 4, 23, 25]
- [46] N. Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human Computer Studies*, 43(5):625–640, 1995. [cited at p. 23]
- [47] N. Guarino. *Formal ontology in information systems: proceedings of the first international conference (FOIS’98), June 6-8, Trento, Italy*, volume 46. Ios PressInc, 1998. [cited at p. 24]

- [48] C. Gutierrez, C.A. Hurtado, A.O. Mendelzon, and J. Pérez. Foundations of semantic web databases. *Journal of Computer and System Sciences*, 77(3):520–541, 2011. [cited at p. -]
- [49] Peter Haase, Jeen Broekstra, Andreas Eberhart, and Raphael Volz. A comparison of rdf query languages. In SheilaA. McIlraith, Dimitris Plexousakis, and Frank Harmelen, editors, *The Semantic Web ISWC 2004*, volume 3298 of *Lecture Notes in Computer Science*, pages 502–517. Springer Berlin Heidelberg, 2004. [cited at p. 90]
- [50] J. Hebel, M. Fisher, R. Blace, and A. Perez-Lopez. *Semantic web programming*. Wiley, 2011. [cited at p. 89]
- [51] W. Hu, N. Jian, Y. Qu, and Y. Wang. Gmo: A graph matching for ontologies. In *Proceedings of K-CAP Workshop on Integrating Ontologies*, pages 41–48, 2005. [cited at p. 17]
- [52] W. Hu, Y. Zhao, and Y. Qu. Partition-based block matching of large class hierarchies. *The Semantic Web-ASWC 2006*, pages 72–83, 2006. [cited at p. 17]
- [53] Wei Hu and Yuzhong Qu. Falcon-ao: A practical ontology matching system. *Web Semantics Science Services and Agents on the World Wide Web*, 6(3):237–239, 2008. [cited at p. 16]
- [54] M.A. Jaro. Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7):491–498, 2007. [cited at p. 46]
- [55] N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-ao: Aligning ontologies with falcon. In *Proceedings of K-CAP Workshop on Integrating Ontologies*, pages 85–91, 2005. [cited at p. 17]
- [56] Yannis Kalfoglou, Bo Hu, Dave Reynolds, and Nigel Shadbolt. Semantic integration technologies survey. Technical report, University of Southampton, April 2005. [cited at p. 17]
- [57] Holger Knublauch, Ray W Ferguson, Natalya F Noy, and Mark A Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*, volume 3298, pages 229–243. Springer, 2004. [cited at p. 39]
- [58] M. Lenzerini. Data integration: A theoretical perspective. In *Symposium on Principles of Database Systems: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, volume 3, pages 233–246, 2002. [cited at p. 21]
- [59] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions. Technical report, and reversals. Technical Report 8, 1966. [cited at p. 45]
- [60] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8):1218–1232, aug. 2009. [cited at p. 15]
- [61] Man Li, Xiao-Yong Du, and Shan Wang. Learning ontology from relational database. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 6, pages 3410–3415 Vol. 6, aug. 2005. [cited at p. -]
- [62] P. Liu and P. Dew. Using semantic web technologies to improve expertise matching within academia. *Proceedings of I-Know (Graz, Austria)*, pages 370–378, 2004. [cited at p. 4]
- [63] Jelena Mamcenko. Introduction to data modeling and msaccess. Vilnius Gediminas Technical University, 2004. Lecture Notes on Information Resources. [cited at p. -]

- [64] F. Manola, E. Miller, and B. McBride. Rdf primer. *W3C recommendation*, 10:1–107, 2004. [cited at p. 90]
- [65] R.J. Miller, Y.E. Ioannidis, and R. Ramakrishnan. Schema equivalence in heterogeneous systems: bridging theory and practice. *Information Systems*, 19(1):3–31, 1994. [cited at p. 22]
- [66] P. Mitra, G. Wiederhold, and J. Jannink. Semi-automatic integration of knowledge sources. *Proceedings of Fusion’99, July 1999*, 1999. [cited at p. 22]
- [67] H. Mohamed, Yang Jincai, and Jin Qian. Towards integration rules of mapping from relational databases to semantic web ontology. In *Web Information Systems and Mining (WISM), 2010 International Conference on*, volume 1, pages 335 –339, oct. 2010. [cited at p. iv, 25, 26]
- [68] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between owl and relational databases. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2):74 – 89, 2009. [cited at p. -]
- [69] I Myroshnichenko and M C Murphy. Mapping er schemas to owl ontologies, 2009. [cited at p. 39]
- [70] R. Neches, R.E. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W.R. Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3):36, 1991. [cited at p. 23]
- [71] W.S. Ng, B.C. Ooi, K.L. Tan, and A. Zhou. Peerdb: A p2p-based system for distributed data sharing. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 633–644. IEEE, 2003. [cited at p. 12]
- [72] L. Palopoli, D. Sacca, and D. Ursino. Semi-automatic, semantic discovery of properties from database schemes. In *Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS’98. International*, pages 244–253. IEEE, 1998. [cited at p. 22]
- [73] Rachel Pottinger. *Database Schema Integration*, chapter D. Springer Reference Series. Springer, 2007. [cited at p. 11]
- [74] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):16, 2009. [cited at p. 91]
- [75] E. Prud’hommeaux, A. Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008. [cited at p. 90]
- [76] Y. Qu, W. Hu, and G. Cheng. Constructing virtual documents for ontology matching. In *Proceedings of the 15th international conference on World Wide Web*, pages 23–31. ACM, 2006. [cited at p. 17]
- [77] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001. 10.1007/s007780100057. [cited at p. 22]
- [78] D. Sagar and LT Tay. Gisa computing perspective. *The Photogrammetric Record*, 20(112):397–398, 2005. [cited at p. -]
- [79] Amit P Sheth. Changing focus on interoperability in information systems: From system, syntax, structure to semantics. *Interoperating Geographic Information Systems*, 5(495):529, 1999. [cited at p. -]
- [80] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. *The Semantic Web-ISWC 2005*, pages 624–637, 2005. [cited at p. 17, 45]

- [81] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using bayesian decision for ontology mapping. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(4):243–262, 2006. [cited at p. 15]
- [82] H. Wache, T. Voegelé, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 workshop: ontologies and information sharing*, volume 2001, pages 108–117. Citeseer, 2001. [cited at p. 13]
- [83] W.E. Winkler. Overview of record linkage and current research directions. In *Bureau of the Census*. Citeseer, 2006. [cited at p. 46]
- [84] M.F. Worboys and M. Duckham. *GIS: A Computing Perspective, Second Edition*. Taylor & Francis Group, 2004. [cited at p. 4, 22]
- [85] Naijun Zhou. *Geospatial Semantic Integration*, chapter D. Springer Reference Series. Springer, 2007. [cited at p. 17]

Appendices

Apéndice A

Ontologías OWL

A.1 Ontología OWL de la BD_1

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/ontologies/2012/10/Prueba_BD1.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.semanticweb.org/ontologies/2012/10/Prueba_BD1.owl">
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Annotation>
    <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
    <Literal datatypeIRI="&rdf;PlainLiteral">Una ontologia que describe a
egresados de diversas escuelas y carreras del IPN</Literal>
  </Annotation>
  <Declaration>
    <Class IRI="#Egresado"/>
  </Declaration>
```

```

</Declaration>
<Declaration>
  <DataProperty IRI="#apellido_materno"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#apellido_paterno"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#boleta"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#carrera"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#codigo_postal"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#correo"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#curp"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#escuela"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#nombre"/>
</Declaration>
<SubClassOf>
  <Class IRI="#Egresado"/>
  <DataMinCardinality cardinality="1">
    <DataProperty IRI="#boleta"/>
    <Datatype abbreviatedIRI="xsd:integer"/>
  </DataMinCardinality>
</SubClassOf>
<FunctionalDataProperty>
  <DataProperty IRI="#apellido_materno"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#apellido_paterno"/>
</FunctionalDataProperty>

```



```

<FunctionalDataProperty>
  <DataProperty IRI="#boleta"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#carrera"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#codigo_postal"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#correo"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#curp"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#escuela"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#nombre"/>
</FunctionalDataProperty>
<DataPropertyDomain>
  <DataProperty IRI="#apellido_materno"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#apellido_paterno"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#boleta"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#carrera"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#codigo_postal"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>

```

```

<DataPropertyDomain>
  <DataProperty IRI="#correo"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#curp"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#escuela"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#nombre"/>
  <Class IRI="#Egresado"/>
</DataPropertyDomain>
<DataPropertyRange>
  <DataProperty IRI="#apellido_materno"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#apellido_paterno"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#boleta"/>
  <Datatype abbreviatedIRI="xsd:integer"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#carrera"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#codigo_postal"/>
  <Datatype abbreviatedIRI="xsd:integer"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#correo"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>

```

```

        <DataProperty IRI="#curp"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DataPropertyRange>
    <DataPropertyRange>
        <DataProperty IRI="#escuela"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DataPropertyRange>
    <DataPropertyRange>
        <DataProperty IRI="#nombre"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DataPropertyRange>
</Ontology>
<!-- Generated by the OWL API (version 3.2.3.1824) http://owlapi.sourceforge.net -->

```

A.2 Ontología OWL de la BD_2

```

<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
    xml:base="http://www.semanticweb.org/ontologies/2012/10/Prueba_DB2.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.semanticweb.org/ontologies/2012/10/Prueba_DB2.owl">
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Declaration>
        <Class IRI="#Alumno"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Carrera"/>
    </Declaration>

```

```

<Declaration>
  <Class IRI="#Domicilio"/>
</Declaration>
<Declaration>
  <Class IRI="#Escuela"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#cursoCarrera"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#estudianEn"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#ofreceCarrera"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#pertenceEscuela"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#tieneAlumnos"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#tieneDomicilio"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#vanAlumnos"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#boleta"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#cp"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#curp"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#estado"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#id_carrera"/>

```

```

</Declaration>
<Declaration>
  <DataProperty IRI="#id_escuela"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#mail"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#nivel"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#nombre"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#nombre_carrera"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#nombre_escuela"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#primerApellido"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#segundoApellido"/>
</Declaration>
<InverseObjectProperties>
  <ObjectProperty IRI="#cursoCarrera"/>
  <ObjectProperty IRI="#tieneAlumnos"/>
</InverseObjectProperties>
<InverseObjectProperties>
  <ObjectProperty IRI="#vanAlumnos"/>
  <ObjectProperty IRI="#estudianEn"/>
</InverseObjectProperties>
<InverseObjectProperties>
  <ObjectProperty IRI="#ofreceCarrera"/>
  <ObjectProperty IRI="#pertenceEscuela"/>
</InverseObjectProperties>
<FunctionalObjectProperty>
  <ObjectProperty IRI="#tieneDomicilio"/>
</FunctionalObjectProperty>
<ObjectPropertyDomain>

```

```

    <ObjectProperty IRI="#cursoCarrera"/>
    <Class IRI="#Alumno"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#estudianEn"/>
    <Class IRI="#Alumno"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#ofreceCarrera"/>
    <Class IRI="#Escuela"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#pertenceEscuela"/>
    <Class IRI="#Carrera"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#tieneAlumnos"/>
    <Class IRI="#Carrera"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#tieneDomicilio"/>
    <Class IRI="#Alumno"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#vanAlumnos"/>
    <Class IRI="#Escuela"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#cursoCarrera"/>
    <Class IRI="#Carrera"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#estudianEn"/>
    <Class IRI="#Escuela"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#ofreceCarrera"/>
    <Class IRI="#Carrera"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#pertenceEscuela"/>

```

```
        <Class IRI="#Domicilio"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#tieneAlumnos"/>
        <Class IRI="#Alumno"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#tieneDomicilio"/>
        <Class IRI="#Domicilio"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#vanAlumnos"/>
        <Class IRI="#Alumno"/>
    </ObjectPropertyRange>
</Ontology>
<!-- Generated by the OWL API (version 3.2.3.1824) http://owlapi.sourceforge.net -->
```

Apéndice B

Herramientas de implementación

B.1 PROTEGE

Protégé es una herramienta de desarrollo libre desarrollada por la universidad de Stanford. El desarrollo de Protégé inició como una utilidad para aplicaciones biomédicas [42], el sistema es de dominio independiente y puede ser utilizado por otras aplicaciones y áreas.

Como otras herramientas de modelado la arquitectura de Protégé está separada en un modelo y en una vista. El modelo es la representación interna de los mecanismos para ontologías y bases de conocimiento. Los componentes de la vista proveen al usuario de una interfaz para visualizar y manipular el modelo.

La plataforma soporta dos maneras principales de modelar ontologías:

El editor *Protégé-Frames* permite construir y compartir ontologías que están basadas en marcos, de acuerdo con el protocolo de conectividad de bases de conocimiento (OKBC). En este modelo la ontología consiste en un conjunto de clases organizadas en una jerarquía que representa el dominio de los conceptos, un conjunto de características asociadas a las clases que describen sus propiedades y relaciones, y un conjunto de instancias de individuos de clases que son ejemplos de los conceptos que contienen valores específicos para sus propiedades.

El editor *Protégé-OWL* permite al usuario crear ontologías para la Web Semántica, en particular en el Lenguaje de Ontologías WEB (OWL) de la W3C, acceder a la descripción lógica de resonadores y adquirir instancias.

B.2 JENA

Es un marco de trabajo para construir aplicaciones de Web Semántica. Provee una extensa librería basada en Java para ayudar a los desarrolladores a diseñar código que maneje RDF RDFS RDFa OWL y SPARQL. Jena incluye un motor de inferencia para ejecutarse en ontologías OWL y RDF y una gran variedad de estrategias de almacenamiento para guardar en memoria o disco las tripletas RDF.

La implementación ha sido diseñada para permitir la fácil integración de módulos de procesamiento como parsers, serializadores, almacenes y procesadores de consultas.

Jena consiste de una colección de interfaces que representan recursos propiedades literales contenedores sentencias y modelos. Un conjunto de clases implementa estas interfaces de forma que se pueden sub clasificar o reemplazar para optimizar implementaciones particulares.

Además mantiene un consistente tratamiento de la Web Semántica a través del uso de estas clases y variables. La tabla B.1 muestra las clases, interfaces y las correspondencias entre éstas y el componente de la Web Semántica [50].

Jena emplea principalmente las siguientes clases:

- *Resource*: Representa un elemento que contiene una sentencia como puede ser un sujeto un predicado o un objeto. Es análogo a un recurso RDF. También existe un recurso de Jena que se refiere a un reified statement que considera una tripleta como un recurso simple.
- *Statement*: Una tripleta de Web Semántica contiene un sujeto un predicado y un objeto, la clase statement permite saber si estos componentes están contenidos en la sentencia.
- *Graph*: Un método básico para mantener los datos de la Web Semántica. Un grafo permite básicamente añadir, borrar, encontrar o contener operaciones. Típicamente una aplicación no permite lidiar directamente con el grafo. La interface Graph permite la instanciación de diferentes tipos de mecanismos de almacenamiento. Esto permite un flexibilidad de bajo nivel en el almacenamiento de la Web Semántica.
- *Model*: Un modelo se construye con base en el grafo y ofrece interacción con los datos de la Web Semántica. Las aplicaciones leen, escriben, razonan y consultan datos a través del acceso de los modelos de Jena. El modelo forma la base de conocimiento.
- *Query* y *ResultSet*: El objeto query emplea SPARQL y regresa resultados en forma de un resultSet.

Componente	Web Semántica	Jena	Notas
Sujeto, predicado, objeto	URI	Resource, Property	Un recurso puede ser un sujeto un objeto o un predicado
Sentencia	Statement	Statement	Consideraciones especiales para sentencias
Dato	Ontología e instancias de datos	Graph y Modelo	Graph son la estructura básica de los modelos y pueden contener la ontología e instancias de datos.
Consultas y resultados	SPARQL y Datos de la Web Semántica	Query y ResultSe	Análogos a los usados con bases de datos relacionales.
Razonador	Razonador	Reasoner	Permite múltiples razonadores internos o externos.
Reglas	SWRL	Reasoner	Reglas determinadas por un determinado razonador
Notificación de Evento	No aplica	ObjectListener	Habilita los procesos de eventos generados.

Tabla B.1: Comparación de la Web Semántica y el Framework Jena

- *Reasoner*: Contiene el proceso de razonamiento que emplea un razonador ya sea internó o externo.

B.3 SPARQL

RDF [64] es un modelo de datos para representar información acerca de los recursos que se encuentran en la web y fue desarrollado en 1998 como una recomendación de la W3c, el problema que se generó a partir de entonces fue el poder consultar este modelo de datos. Desde entonces han surgido un gran número de diseños e implementaciones de lenguajes de consulta para RDF [49]. En 2004 el grupo de trabajo de acceso de datos RDF realizó la primera introducción de un lenguaje de consulta para RDF llamado SPARQL [75] y en el año 2006 SPARQL se volvió un candidato de recomendación para la W3C.

SPARQL es un lenguaje de consulta basado en un mapeo de grafos. Dado una fuente de datos D , una consulta consiste en un patrón que es mapeado contra D , y los valores obtenidos de este mapeo son procesados para obtener una respuesta.

La fuente de datos D que puede ser consultada podría estar compuesta de múltiples

fuentes. Una consulta SPARQL consiste de tres partes. La parte de comparación de patrones, que incluye varias características de comparación de los grafos, como partes opcionales, unión de patrones, anidamiento, filtros o restricciones en valores, y la posibilidad de elegir la fuente de datos que corresponde a un patrón. Los modificadores de la solución, una vez que la salida de los patrones ha sido calculada, se permite modificar estos valores aplicando operadores clásicos como proyección, diferencia, orden, límites y desplazamiento (offset). Por último la salida de una consulta SPARQL puede ser de diferentes tipos, consultas booleanas de si o no, selecciones de valores de variables q coincidan con los patrones, construcción de nuevas tripletas a partir de estos valores y descripciones de recursos [74].