



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SECCIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN

Tema

“Modelo ubicuo de integración de información”

Que para obtener el grado de

**Maestro en Ciencias en Sistemas
Computacionales Móviles**

Presenta

Ing. Kevin Ricardo Verástegui Pérez

Directores

M. en C. Chadwick Carreto Arellano

Dr. Jorge Cortés Galicia





INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:20 horas del día 30 del mes de noviembre del 2018 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI - ESCOM para examinar la tesis titulada:
"Modelo ubicuo de integración de información"

Presentada por el alumno:

Verástegui **Pérez** **Kevin Ricardo**
Apellido paterno Apellido materno Nombre(s)
Con registro:

A	1	7	0	3	3	1
---	---	---	---	---	---	---

aspirante de:

Maestría en Ciencias en Sistemas Computacionales Móviles

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

M. en C. Chadwick Carreto Arellano
Vocal 1

Dr. Jorge Cárdenas Galicia
Vocal 2

Dr. José Alfredo Jiménez Benítez
Presidente

Dra. Elena Fabiola Ruiz Ledesma
Secretaría

Dra. Jazmin Adriana Juárez Ramirez
Vocal 3

Dra. Lorena Chavarría Báez
Suplente

PRESIDENTE DEL COLEGIO DE PROFESORES

Lic. Andrés Ortíz Campos
S.E.P.
ESCUELA SUPERIOR DE COMPUTACIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la **Ciudad de México, CDMX** el día **11** del mes de **diciembre** del año **2018**, el que suscribe **Ing. Kevin Ricardo Verastegui Pérez** alumno del Programa de **Maestría en Ciencias en Sistemas Computacionales Móviles** con número de registro **A170331**, adscrito a la **Escuela Superior de Cómputo**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de **M. en C. Chadwick Carreto Arellano** y **Dr. Jorge Cortés Galicia** y cede los derechos del trabajo intitulado **Modelo ubicuo de integración de información**, 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 autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **ikevin.verastegui@gmail.com** Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Ing. Kevin Ricardo Verastegui Pérez

Resumen

La creciente cantidad de información que producimos hoy en día ha provocado un particular interés en avances tecnológicos para determinar la manera de administrar y explotar dichos datos, lo antes mencionado apunta hacia la búsqueda de métodos para integrar datos de distintas fuentes de forma congruente y que resulte útil a un propósito específico, es decir tener información de manera personalizada. Sin embargo, dicha cantidad de datos ha provocado una gran cantidad de formas de ver y representar esta información, por lo cual la interoperabilidad y la transparencia de integración entre los sistemas es muy complicada de lograr.

En ese sentido, se han elaborado trabajos significativos sobre motores de integración de información, todos caracterizados por la intención de trabajar con las fuentes originales de datos, evitando la duplicidad, esto se traduce en eliminar la actualización de terceros, así como el presentar al usuario en una vista única el contenido de manera personalizada. Como se mencionó anteriormente la tarea no es sencilla, ya que las fuentes de datos son heterogéneas en más de un sentido.

En este trabajo el problema de la integración de información se pretende resolver con la propuesta de un modelo a capas, en el que a través de este se permita tener un control en el proceso de búsqueda, integración y clasificación de la información que se encuentre en distintos nodos, sin embargo por lo ambicioso que resulta el desarrollo de todas las capas que conforman al modelo, los resultados estarán enfocados únicamente en la capa integradora de información, dejando para trabajo futuro el planteamiento de la capa clasificadora de los datos obtenidos. Para esta capa integradora se propone el uso de una ontología como una capa intermedia entre la interfaz de usuario para el manejo de la información y las fuentes originales de datos. Con el uso de la ontología, las consultas se elaboran solo en términos de esta y la integración de cada fuente de datos es independiente de la estructura particular de las otras fuentes que participan en la integración.

Finalmente, los resultados obtenidos en el presente documento dan lugar a la apertura de múltiples posibilidades de investigación sobre la computación móvil centralizada en la gestión de información, asimismo se promueven nuevas implementaciones enfocadas al aprovechamiento y explotación de las herramientas tecnológicas de vanguardia.

Abstract

The increasing amount of information we produce today has caused a particular interest in technological advances to determine how to manage and exploit such data, the aforementioned points to the search for methods to integrate data from different sources in a consistent manner and resulting useful for a specific purpose, that is, having information in a personalized way. However, this amount of data has caused a large number of ways to view and represent this information, so interoperability and transparency of integration between systems is very complicated to achieve.

In this regard, significant work has been done on information integration engines, all characterized by the intention to work with the original data sources, avoiding duplication, this translates into eliminating the update of third parties, as well as presenting the user in a single view the content in a personalized way. As mentioned earlier, the task is not simple, since the data sources are heterogeneous in more ways than one.

In this work, the problem of integrating information is to be solved with the proposal of a layered model, in which through this it is possible to have a control in the process of search, integration and classification of the information found in different nodes, however for the ambitious results of the development of all the layers that make up the model, the results will be focused only on the information integrating layer, leaving for future work the classification layer approach of the data obtained. For this integrative layer, the use of an ontology is proposed as an intermediate layer between the user interface for the management of information and the original sources of data. With the use of the anthology, the consultations are elaborated only in terms of this and the integration of each data source is independent of the particular structure of the other sources that participate in the integration.

Finally, the results obtained in this document give rise to the opening of multiple research possibilities on mobile computing centralized in information management, and new implementations are promoted that focus on the exploitation and exploitation of cutting-edge technological tools.

Agradecimientos

A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos.

Kevin Verastegui

Índice General

CAPÍTULO I. INTRODUCCIÓN	11
1.1	Introducción..... 12
1.2	Antecedentes 13
1.3	Problemática..... 18
1.4	Solución propuesta..... 20
1.5	Objetivos 23
1.5.1	General..... 23
1.5.2	Específicos..... 23
1.6	Justificación 24
1.7	Motivación 25
1.8	Organización del documento 26
CAPÍTULO II. ESTADO DEL ARTE.....	27
2.1	Integración de bases de datos heterogéneas 28
2.2	Integración esquemática de bases de datos 38
2.3	Integración semántica de fuentes de información..... 40
2.3.1	Web Semántica..... 41
2.3	Integración de información con ontologías 42
CAPÍTULO III. MARCO TEÓRICO	47
3.1	Computo ubicuo..... 48
3.2	Base de datos..... 50
3.2.1	Modelo conceptual..... 51
3.2.2	Base de datos relacional 52
3.3	Integración de información..... 55
3.3.1	Factores que impiden la integración de datos 55
3.3.2	Técnicas para abordar el problema de la integración..... 57
3.3.3	Enfoque centralizado vs. Distribuido..... 59
3.4	Ontologías 61
3.4.1	Componentes principales de las ontologías 62
3.4.2	Clasificación de las ontologías..... 64
3.4.3	Lenguajes de ontologías 65
CAPÍTULO IV. METODOLOGÍA PROPUESTA	68
4.1	Etapas de la metodología..... 71
4.1.1	Etapas de conceptualización y dominio de aplicación..... 71
4.1.2	Creación de la ontología 72
4.1.3	Mapeo de instancias 76
4.1.4	Etapas de recuperación y despliegue 79
CAPÍTULO V. CASO DE ESTUDIO.....	82
5.1	Metodología caso de estudio 83
5.1.1	Caracterización del perfil de usuario 84

5.1.2	Análisis y procesamiento de la información	86
5.1.3	Visualización de resultados.....	87
5.2	Desarrollo caso de estudio.....	88
5.2.1	Tarea 1: Glosario de términos.....	88
5.2.2	Tarea 2: Taxonomía de conceptos.....	95
5.2.3	Tarea 3: Diagrama de relaciones binarias ad hoc.....	95
5.2.4	Tarea 4: Diccionario de conceptos.....	96
5.2.5	Tarea 5: Relaciones binarias ad hoc.....	99
5.2.6	Tarea 6: Atributos de instancia.....	99
5.2.7	Tarea 7: Describir los atributos de clase	100
5.2.8	Tarea 8: Describir las constantes.....	101
5.2.9	Tarea 9: Definición de axiomas formales.....	102
5.2.10	Tarea 10: Definir reglas	102
5.2.11	Tarea 11: Describir instancias	103
5.3	Transformación y recuperación de información original.....	110
5.3.1	Mapeo de instancias (Hospitales)	111
5.3.2	Consulta mediante SPARQL	113
5.3.3	Visualización de marcadores en mapa	116
CAPÍTULO VI. PRUEBAS Y RESULTADOS.....		119
6.1	Creación capa de integración de información.....	120
6.2	Arquitectura aplicación móvil	125
6.2.1	Requerimientos funcionales.....	125
6.2.2	Requerimientos no funcionales	126
6.2.2.1	Restricciones.....	126
6.2.2.2	Propiedades	127
6.2.3	Reglas de negocio	127
6.2.3.1	Definición.....	127
6.2.3.1	Restricción	128
6.2.2	Modelo de arquitectura 4 + 1	129
6.2.2	Vista de escenarios.....	130
6.2.3.1	Casos de uso aplicación móvil.....	130
6.2.3.1.1	CUA01 Configurar aplicación.....	131
6.2.3.1.2	CUA02 Ingresar datos personales	132
6.2.3.1.3	CUA03 Agregar contacto favorito	133
6.2.3.1.4	CUA04 Consultar servicio médico	134
6.2.3.1.5	CUA05 Confirmar hospital	135
6.2.3.1.6	CUA06 Visualizar resultado.....	136
CAPÍTULO VII. ANÁLISIS Y DISCUSIÓN DE RESULTADOS		139
CAPÍTULO VIII. CONCLUSIONES		150
	Limitaciones.....	153
	Trabajo Futuro	153
	Bibliografía	155

Índice de figuras

Figura 1.1	Esquema conceptual global	16
Figura 1.2	Enfoque basado en múltiples esquemas virtuales	17
Figura 1.3	Consulta de distintas fuentes de manera individual.....	18
Figura 1.4	Esquema de propuesta de solución.....	20
Figura 1.5	Esquema de red de ontologías	22
Figura 2.1	Estructura para la integración de bases de datos.....	29
Figura 2.2	Arquitectura de un Sistema de Bases de Datos Heterogéneas dentro de un proceso KD...	31
Figura 2.3	Ubiquitous Integration	33
Figura 2.4	Arquitectura en 5 niveles para la integración de bases de datos.....	37
Figura 2.5	Arquitectura BUSTER.....	39
Figura 2.6	Adquisición de datos	40
Figura 2.8	Modelo de Hakimpour	46
Figura 3.1	Integración de dispositivos inteligentes en el ambiente	48
Figura 3.2	Correspondencia de cardinalidades	54
Figura 3.3	Distintos enfoques de integración de acuerdo al nivel de integración	57
Figura 3.4	Enfoque centralizado.....	59
Figura 3.5	Enfoque distribuido.....	60
Figura 3.6	RDF Tripleta	66
Figura 4.1	Arquitectura ontológica	70
Figura 4.2	Etapas generales del modelo propuesto.....	71
Figura 4.3	Proceso de transformación de las fuentes de información	72
Figura 4.4	Methontology	73
Figura 4.5	Grafo virtual D2RQ.....	79
Figura 4.6	Despliegue	81
Figura 5.1	Metodología caso de estudio.....	84
Figura 5.2	Primer bloque.....	85
Figura 5.3	Segundo bloque	86
Figura 5.4	Tercer bloque	87
Figura 5.5	Taxonomía de conceptos	95
Figura 5.6	Relaciones binarias ad hoc	95
Figura 5.7	Recuperación de fuentes originales.....	110
Figura 5.8	Mapeo D2RQ	113
Figura 5.9	Agregar anotaciones	118
Figura 6.1	Activar opciones Protégé	120
Figura 6.2	Subclases principales	121
Figura 6.3	Red ontológica servicios médicos.....	121
Figura 6.4	Agregar instancias.....	122
Figura 6.5	Vincular atributos.....	122
Figura 6.6	Vincular instancias en clases	123
Figura 6.7	Relaciones de clases.....	123
Figura 6.8	Modelo de Kruchten.....	129
Figura 6.9	Diagrama de casos de uso	130
Figura 6.10	Métricas en la ontología	137
Figura 7.1	Red ontológica.....	140
Figura 7.2	Ontología de servicios médicos	143
Figura 7.3	Grafo hospital.....	143

Índice de tablas

Tabla 3.1	Elementos de RDF.....	67
Tabla 5.1	Glosario de términos caso de estudio.....	67
Tabla 5.2	Diccionario de conceptos.....	96
Tabla 5.3	Descripción de relaciones binarias ad hoc.....	99
Tabla 5.4	Descripción atributos de instancia.....	100
Tabla 5.5	Atributos de clase.....	100
Tabla 5.6	Reglas en ontología.....	102
Tabla 5.7	Atributos de instancia.....	103
Tabla 5.8	Resultados consulta sparql.....	115
Tabla 6.1	CUA01 Configurar aplicación.....	123
Tabla 6.2	CUA02 Ingresar datos personales.....	124
Tabla 6.3	CUA03 Agregar contacto favorito.....	125
Tabla 6.4	CUA04 Consultar servicio médico.....	126
Tabla 6.5	CUA05 Confirmar hospital.....	127
Tabla 6.6	CUA06 Visualizar resultado.....	128
Tabla 6.1	CUA01 Configurar aplicación.....	123

CAPÍTULO I. INTRODUCCIÓN



1.1 Introducción

“No cabe duda de que vivimos en una era donde la generación y consumo de información ha visto una aceleración estrepitosa. El ritmo con el que ésta ha crecido puede describirse como exponencial y ciertamente eso no nos exime de administrar dicha información” [1].

En la actualidad nos encontramos en un mundo inmerso en la tecnología donde los sistemas computacionales forman parte fundamental en la vida cotidiana de cada uno, es tan sencillo que resulta bastante difícil encontrar personas que no cuenten con un teléfono o algún dispositivo electrónico a la mano, es por ello que durante los últimos años se ha producido un aumento exponencial de la información que producen las personas y organizaciones, con lo cual se ha impulsado el desarrollo de soluciones cada vez más eficientes para su almacenamiento, nuevos formatos digitales, protocolos de comunicación entre otros. Aunque en la mayor de las veces la representación de información o conocimiento no son compatibles entre si. Es por ello que se han desarrollado normas para la representación de la información, las cuales permiten generar una compatibilidad entre distintos esquemas, por ejemplo, el lenguaje XML (eXtensible Markup Language) [2].

Gracias a ellos, se ha permitido el acceso a grandes cantidades de datos, situados en distintos lugares, con múltiples formatos, contextos, plataformas y también con diferente semántica. Sin embargo, el conocimiento utilizado en los procesos de toma de decisiones no ha experimentado un incremento proporcional al de la información disponible.

No obstante, un problema en cuanto al uso de sitios integrados de esta manera es la enorme cantidad de información que los motores de búsqueda deben manipular, lo cual ha originado que cada vez sea mas complejo su tratamiento y mas complicado sea la generación de resultados personalizados para un perfil en específico, ya que no solamente se trata de una cantidad lo bastante considerable de datos si no que también nos encontramos con la cuestión de la diversidad del origen de estos.

En consecuencia, las investigaciones sobre esta integración sobre los datos provenientes de fuentes heterogéneas y su recuperación han cobrado una enorme importancia. Una manera de atacar este problema es mediante el uso de ontologías, las cuales modelan formalmente un dominio de la realidad, entendiéndose por dominio un agregado de objetos que comparten relación entre sí. Una ontología proporciona tanto las características como los estándares necesarios para lograr la integración que se pretende, además de ello se tiene la ventaja de poder aplicar inferencias en el conocimiento almacenado, es decir, generar nuevo conocimiento a partir de la información concentrada en la ontología [3][4].

En virtud de ello, estas características serán utilizadas en este trabajo para la propuesta de un modelo integrador de información sobre datos de diferentes orígenes, así como conocimiento generado a partir de dichos datos. Este conocimiento es representado en una innumerable variedad de formas, sin embargo, la heterogeneidad y descentralización de las fuentes ha provocado que cuanto más información halla accesible, más difícil sea localizar lo que se busca; y que una vez encontrado, integrarlo con lo que ya se dispone no sea un proceso inmediato, debido a esto se pretende obtener todo este conocimiento de una forma conjunta y no solo como elementos individuales de información [5]. Por tanto, el siguiente paso hacia una mejor comprensión, debe ser la representación explícita y semántica, a través de ontologías.

1.2 Antecedentes

Con la aparición del internet, y el gran número de dispositivos que surgen día a día, la manera en que se accede a la información y la facilidad de agregar más surge el problema de que la información es ambigua y a su vez no esta estructurada, por lo tanto, la manera de encontrar lo que se esta buscando, es sumamente complicado.

La forma en que comúnmente se accede a los datos, es a través de consultas a una base de datos local o bien esta puede ser remota con un esquema previamente conocido. En contraste con los sistemas actuales las consultas implican diferentes fuentes de datos, que no solo estén dispersos geográficamente, si no que a su vez presenten esquemas diferentes [6][7].

La tarea no es sencilla, y no solamente debido a la dispersión geográfica de las fuentes de información si no que, en algunos casos, se tienen datos no estructurados. Otras en cambio si presentan esta estructura en sus datos, pero almacenan y manipulan con diferentes sistemas manejadores de bases de datos ya sean relacionales u orientado a objetos, sumándole a esta tarea el hecho de que no existe un estándar respecto a la forma de presentar la información sea cual sea, ni en cuanto a que términos son los adecuados para referirse a un tema en particular.

En este escenario, los objetivos que se persiguen en el panorama actual de la tecnología de la información apuntan hacia la generación de métodos para integrar información no redundante con un propósito en específico, o bien una personalización de dicha información.

La diferencia que presentan los almacenes de datos actuales se conoce como heterogeneidad, la cual es un problema relacionado con el estudio de la integración de la información, la cual se divide en heterogeneidad sintáctica y semántica.

En primer lugar, la heterogeneidad sintáctica se refiere a las diferencias en la representación de los datos. Un claro ejemplo de este tipo de heterogeneidad se da cuando se busca integrar datos cuyos tipos no son iguales, por ejemplo, en una fuente el dato es entero y en la otra es un flotante, así como las diferencias entre las estructuras de los archivos, por ejemplo, texto plano y archivos XML [8].

Por otra parte, se presenta la heterogeneidad semántica que es la diferencia de la información en el contexto. En un sistema de información heterogéneo, el significado de la información se intercambia ya que tiene que ser entendido a través de los sistemas. Los conflictos semánticos ocurren siempre que dos contextos no utilicen la misma interpretación de la información y se da por 3 razones principalmente [9]:

- a) Expresiones con un mismo significado en diferentes contextos.
- b) Empleo de sistemas de referencias diferentes.
- c) Existencia de homónimos y sinónimos.

Por ejemplo, al representar el elemento *camino*, su interpretación se puede representar con los objetos *trayecto* y *sendero*, las cuales relacionan el mismo significado sin que ello implique que su comportamiento sintáctico sea el mismo.

En este sentido, en los últimos años, se han realizado distintas investigaciones enfocadas en este campo de estudio sobre integradores de información, los primeros esfuerzos se orientaron en resolver principalmente la heterogeneidad tecnológica de las distintas fuentes dando así, los sistemas middleware como RMI (Remote Method Invocation) y CORBA (Common Object Request Broker Architecture)[10].

Sin embargo, los avances que se dieron en estos sistemas apuntaron hacia otro problema que se denominó como la heterogeneidad lógico-semántica de dichas fuentes de información, el cual resultó ser el principal obstáculo de la integración.

A pesar de ello, la solución parcial a esta dificultad se presenta en tres esquemas principales, los cuales se han trabajado para tratar de resolver la integración lógico-semántica [11]:

- Enlazado de información.
- Traducción de datos.
- Traducción de consultas.

Enlazado de información

Las entidades de información (sitios web, instancias de una BD entre otros) relacionados están enlazados a través de enlaces estáticos a las diferentes piezas de información que se desea encontrar, su principal ventaja es la sencillez de uso por parte del usuario ya que solo utiliza los enlaces, sin embargo, estos enlaces son unidireccionales y estas consultas están limitadas o predefinidas por los mismos. Un ejemplo claro de ellos es la World Wide Web, otro ejemplo para este tipo de enlace se presenta en la bases de datos en la web como Medline y PDB (Protein data bank)[12][13].

Traducción de datos

Los datos provenientes de distintas fuentes de información se transforman a un formato común y se almacenan en un repositorio centralizado accediendo a este de una manera unificada, este esquema tiene la ventaja de tener un alto rendimiento y control de datos (seguridad, confiabilidad entre otros), pero lamentablemente se debe de considerar que cuando los datos de la fuente original se modifican, es necesario actualizar el repositorio centralizado, otra inconveniente son las traducciones almacenadas en los algoritmos, lo que produce problemas de mantenimiento y verificación. Este esquema tiene como ejemplo a los almacenes de datos o bien “DataWareHouses” [14].

Traducción de consultas

Este último esquema tiene un agente intermedio el cual se encarga de recibir y transformar las consultas a un formato que las distintas fuentes de información puedan interpretar. Del mismo modo, este agente recibe y transforma la respuesta para así presentarla de manera unificada para el usuario. Este esquema se puede dividir a su vez en tres clases distintas:

- **Mediación Pura**

Los mediadores y agentes se utilizan para ejecutar las consultas de los usuarios, donde se contiene los datos necesarios para recuperar la información y presentarla al usuario, sin embargo, este enfoque es poco intuitivo para los usuarios ya que la información de integración esta codificada en los mediadores.

- **Esquemas conceptuales simples**

Se utiliza una sola conceptualización global que contiene virtualmente toda la información de las fuentes de datos, a su vez las consultas se construyen según este esquema conceptual global. Los resultados se devuelven al usuario a nivel conceptual,

es decir, como instancias de entidades que pertenecen a un esquema conceptual. Cualquier cambio en el sistema (la inclusión, o borrado de una BD) implica un cambio en el modelo global, este enfoque solo se puede aplicar si todos los datos provienen de un dominio en común [15].

Los enfoques de esquemas conceptuales simples utilizan un esquema global enlazado a las distintas fuentes y que suministra un vocabulario común, este se emplea en la especificación de la semántica de las fuentes y que suministra un vocabulario en común.

La Figura 1.1 muestra este esquema global:

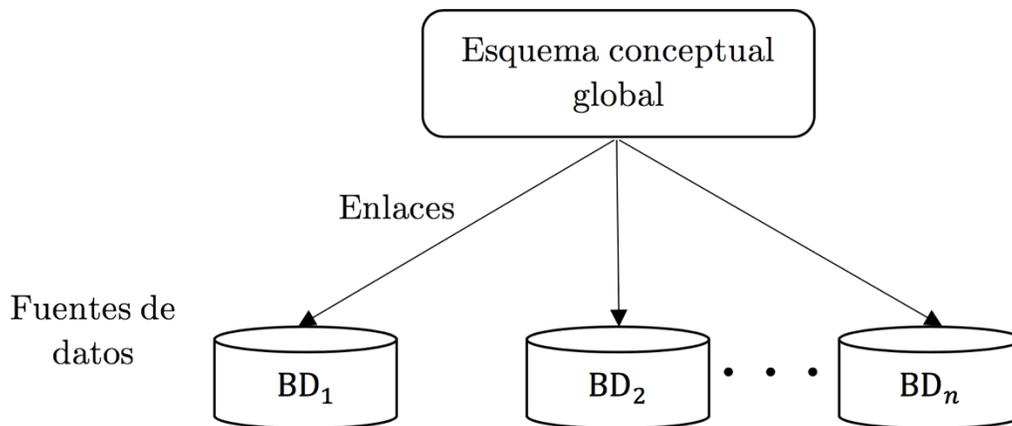


Figura 1.1 Esquema conceptual global [15]

En contra de este enfoque se ha de comentar que la creación de este esquema conceptual global no es la mejor opción ya que estas fuentes pueden ofrecer distintas vistas de un mismo dominio.

- Esquemas conceptuales múltiples

Las desventajas de los esquemas globales han llevado al desarrollo de enfoques con múltiples esquemas conceptuales. En los sistemas basados en múltiples esquemas conceptuales, cada una de las fuentes de datos está descrita por su propio esquema conceptual pero no tiene por qué compartir el mismo vocabulario. Es decir, cada uno de los esquemas conceptuales locales puede desarrollarse independientemente de los demás, lo que facilita futuras modificaciones [4][12]. Para poder comparar los diferentes esquemas conceptuales, se introduce un nuevo mecanismo de enlazado entre los mismos. Estos enlaces inter-esquemas identifican relaciones esquemáticas entre conceptos.

La Figura 1.2 muestra la descripción del esquema mencionado.

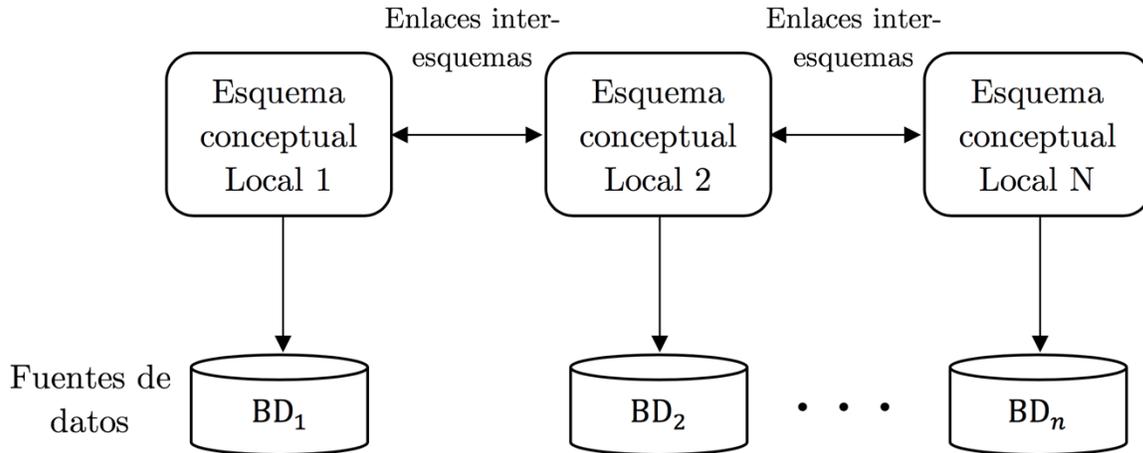


Figura 1.2 Enfoque basado en múltiples esquemas virtuales [4]

Distintos esquemas virtuales describen la semántica de cada base de datos integrada en el sistema, el cual resuelve el principal problema del enfoque conceptual simple, las consultas pueden expresarse con términos de ontologías específicas de dominio. A pesar de ello no se puede asumir que esquemas individuales comparten el mismo vocabulario. En la práctica, sin embargo, los enlaces inter-esquemas son muy difíciles de definir debido a la gran heterogeneidad semántica que se pueden dar dentro de los sistemas de bases de datos heterogéneas [15].

De los tres esquemas mencionados anteriormente, el esquema de traducción de consultas ha recibido mayor atención en investigaciones recientes, acerca de la integración de información de fuentes heterogéneas, así mismo se han desarrollado varios trabajos que se diferencian del resto por implementar el uso de esquemas de conceptualización para describir la semántica de los datos manejados en las distintas fuentes de información y a partir de ellos lograr la unificación buscada.

Este tipo de esquemas se conoce mas comúnmente como ontologías, con estos esquemas se ha logrado dar los primeros avances en la integración de fuentes de datos estructuradas como las bases de datos relacionales y no estructuradas como los documentos de texto plano [15]. El motivo por el cual resulta oportuno el uso de ontologías es que proporcionan un lenguaje compartido común a nivel conceptual para representar la información incluida en las fuentes originales, además de esto se permiten la definición de relaciones entre los conceptos que describen, facilitando de esta forma al usuario realizar consultas mas complejas y recuperar con mayor personalización los datos en los que esta interesado. En especial esta ultima manera de emplear las ontologías para integrar distintas fuentes de información es en la que esta enfocado este trabajo.

1.3 Problemática

La gran cantidad de información a la que se tiene acceso hoy en día genera el inconveniente de cómo poder recuperar la información de manera personalizada sin tener que consultar cada una de las fuentes de información por separado.

La manera en que habitualmente se accede a la información se muestra en la Figura 1.3.



Figura 1.3 Consulta de distintas fuentes de manera individual

Siendo así, se determina que el problema de integración consiste en combinar los datos que residen en diferentes fuentes y proporcionar al usuario una vista personalizada de los datos de manera unificada.

Este problema de integración se puede ver con más frecuencia a medida que el tamaño y la necesidad de compartir y explotar conocimiento aumenta. Este punto se ha convertido en un foco de interés en el trabajo de investigación teórico y un gran número de casos de estudio en particular que quedan sin resolver.

De acuerdo con la revisión de antecedentes, se han detectado los siguientes puntos:

1. La heterogeneidad y descentralización de información.

Dentro de este punto, se tienen en la actualidad sistemas dispersos que son heterogéneos, los cuales poseen características físicas y operativas distintas entre sí, y aunque la gran mayoría de datos de corporaciones, gobierno e instituciones educativas residen en grandes equipos, migrar estos datos entre plataformas no es un proceso inmediato, variando en complejidad dependiendo del hardware, la red y los sistemas operativos que se ejecutan en cada uno de los sistemas, nos encontramos ante un conjunto de objetos no comunes los cuales requieren una interconexión tanto para la migración de datos como para la compartición de los mismos.

2. El proceso de integración de información dispersa en distintos nodos no resulta inmediato.

Retomando el punto anterior en los sistemas heterogéneos cada elemento de la red cuenta con requerimientos muy particulares que le exigen que posea características estándar para poder integrarse con otro sistema. Si bien este con este trabajo no se pretende interconectar cada sistema, acceder a la información con la que cuenta cada uno de ellos no resulta un proceso inmediato ni mucho menos sencillo.

3. La gran diversidad de equipos heterogéneos, inconexos y de difícil integración desde el punto de vista en la toma de decisiones.

Existe una gama muy amplia de posibilidades de centralización y descentralización de los sistemas de información, desde un único ordenador mainframe con pantallas en el mismo local del centro de procesamiento de datos hasta un sistema formado por varios ordenadores no conectados. En un principio, la descentralización era obligada ya que los ordenadores y dispositivos apenas podían manipular la carga de trabajo asignada, como consecuencia de ello, se trasladó las bases de datos y los nodos que contienen información a departamentos especializados, por lo cual al día de hoy su máximo inconveniente radica en que tomar decisiones con base en la información de distintas fuentes genera retrasos y pérdidas de tiempo, lo cual tiene un gran impacto cuando se requiere una respuesta de acción inmediata.

Dada la naturaleza de la tarea que implica unificar datos de distintas fuentes sin relación expresa alguna, es preciso mencionar el obstáculo en las diferencias de conceptualización en el esquema de cada fuente de información y en los términos que se emplean para definirlo; atributos, entidades, relaciones, etc. Cada diseñador hace la abstracción del dominio de acuerdo con la realidad que lo circunscribe, lo cual evidentemente puede ser distinta para cada caso en particular.

1.4 Solución propuesta

La solución que se propone en el siguiente trabajo, es diseñar un modelo a capas, que permita tener un control en el proceso de búsqueda, integración y clasificación de la información que se encuentre en distintos nodos.

Cada capa dentro de este modelo tiene como propósito una acción en específico. Si bien todas estas capas están conectadas y relacionadas a través de distintas interfaces, la información que comparte cada una de estas se utiliza en la capa siguiente. En el siguiente esquema (figura 1.4) se muestra de manera general lo que se propone, así como la organización de las distintas capas que conforman este modelo.



Figura 1.4 Esquema de propuesta de solución

Proceso de la solución del modelo propuesto:

Nodos con datos almacenados: Son los puntos de acceso llamados nodos, de los cuales se obtiene la información, esta información se puede recabar en distintos formatos, y a su vez estos nodos pueden ser representados por bases de datos, servidores, sensores, equipos móviles, etc.

Integración: Dentro de este apartado una vez identificados los distintos nodos de donde se obtendrá la información, se procede a buscar los datos e información que se estén solicitando para un perfil en específico. Dentro de este punto se intenta obtener la mayor cantidad de información posible.

Filtrado de información: Una vez recabada la información de los distintos nodos, lo siguiente es hacer una clasificación de esta información, almacenando únicamente lo que realmente sea útil y de vital importancia para el perfil específico.

Perfil específico: Se define como el despliegue y la visualización de esta información recabada y clasificada, dentro de este nivel el perfil determina los criterios del filtrado de la información.

Si bien con el modelo presentado anteriormente se pretende resolver de manera parcial la integración de fuentes de datos heterogéneas, y la generación de conocimiento para perfiles específicos, este trabajo se centra únicamente en la capa de integración de datos; excluyendo así el desarrollo de las capas de búsqueda, filtrado y el análisis de los datos integrados. Esto debido a la complejidad y al periodo de tiempo que se tiene para el desarrollo de este trabajo.

Una vez delimitado el enfoque que se aborda en este trabajo lo que sigue a continuación es la solución al desarrollo de esta capa de integración de datos. Para ello se propone una metodología de integración de fuentes de datos, la cual se basa en la representación de las fuentes de dichos datos como ontologías, para así recuperar los datos de las distintas fuentes de una forma unificada.

La metodología de integración propuesta plantea el uso de una red de ontologías con una *meta-ontología* que conceptualice el dominio que abordan todas las ontologías que están integradas en esta red, esto es una capa intermedia entre la aplicación o interfaz para la manipulación de la información y las fuentes originales de datos. Cada uno de los elementos de cada esquema identificado para su integración a la ontología global se enlaza o mapea a una clase o elemento dentro de esta, ocupando el lugar que le asigna la jerarquía definida en la red de ontologías.

Con el uso de la ontología, los elementos de los esquemas ya no forzosamente necesitan guardar entre si una relación explícita o entrelazarse unos con otros al momento de realizar una consulta. La consulta por lo tanto ya no se crea en los términos de los esquemas que participan en la integración a través del lenguaje SQL. La consulta se elabora solo en los términos de la ontología, por lo tanto, la integración de una nueva fuente de datos es independiente de las otras fuentes de información [16].

En el modelo que se propone el principal motivo de emplear una *meta-ontología* a nivel de conceptualización es proporcionar al usuario un conjunto de características que deben expresar los conceptos relacionados con el dominio deseado. Estos conceptos se utilizan para representar la abstracción de dicho dominio, es decir, el modelo de la realidad. En este caso la *meta-ontología* cumplirá el rol de una ontología terminológica. Partiendo del concepto de que la abstracción es la idea que solo existe en la mente de acuerdo con cada persona, la conceptualización de la abstracción se puede llevar a cabo mediante esta meta-ontología propuesta, mientras que para poder representar la abstracción del dominio es necesario contar con un lenguaje de modelado. De esta forma se permite representar los elementos que conforman dicha red de ontologías. Con la consecuencia de que así se tendrían ontologías de dominio que conceptualizan la información almacenada en las BD facilitando la integración semántica de las mismas.

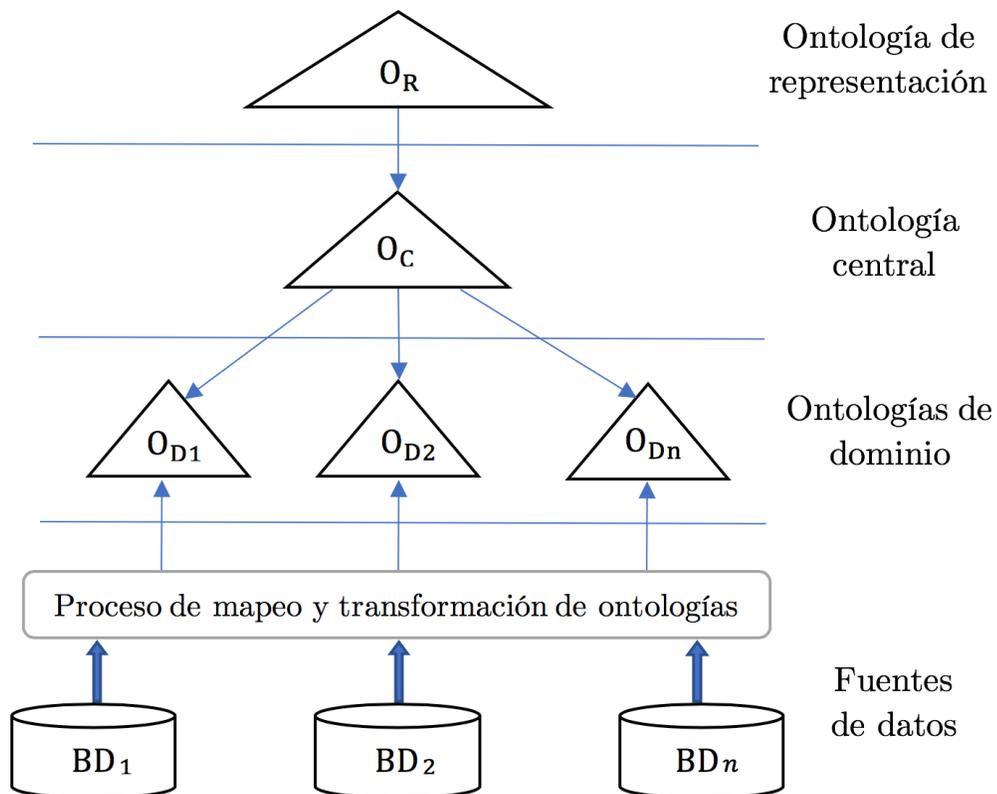


Figura 1.5 Esquema de red de ontologías

1.5 Objetivos

A continuación, se describe el objetivo general y los objetivos específicos que se pretenden cumplir tras finalizar el desarrollo de este trabajo.

1.5.1 General

Definir un modelo a capas empleando una metodología con base en ontologías, para la integración parcial de fuentes de datos heterogéneas.

1.5.2 Específicos

- Definir las capas que conforman el modelo propuesto.
- Desarrollar la capa de integración de información del modelo definido con anterioridad a partir de una metodología propuesta utilizando una red de ontologías para la organización, integración de datos provenientes de fuentes heterogéneas.
- Crear las ontologías correspondientes con base en el estándar OWL (Ontology Web Language) para la representación y descripción de las fuentes de datos que presenten heterogeneidad semántica.
- Diseñar un caso de estudio que implemente el modelo propuesto.

1.6 Justificación

Actualmente la gran cantidad de datos presenta un gran número de retos de investigación, en donde la integración de información es uno de estos, lo que ha impulsado al desarrollo de nuevas tecnologías, nuevos formatos de comunicación y de almacenamiento. Como se mencionó al principio de este capítulo, tenemos que aceptar la heterogeneidad en sus diferentes manifestaciones, por el momento y probablemente por mucho tiempo hacia el futuro.

La integración de información ha tenido mucha mayor importancia debido a las nuevas posibilidades que aparecen en la interconexión del mundo y la creciente disponibilidad de los datos. No obstante que la heterogeneidad es un gran problema en los sistemas actuales, ya que causan algunas restricciones en la eficiencia y rendimiento del sistema como conjunto.

Todo trabajo de investigación acerca del tratamiento de la integración de datos conlleva la intención implícita de tener un panorama global acerca del dominio de interés particular.

Con este trabajo se pretende plantear una metodología basada en la utilización de ontologías como herramienta vital para lograr dicha integración. La metodología planteada se centra en el uso de bases de datos relacionales únicamente, ya que la mayoría de la información se encuentra y se continúa accediendo a través de bases de datos heredadas cuya estructura, es relacional y dada su fácil implementación, seguramente la estructura de datos relacional se seguirá desempeñando por algún tiempo, y que además este trabajo no solo sirva para integrar las fuentes de datos, sino transformarlas, en modelos de representación semánticos: ontologías.

La aplicación de las ontologías se ha modificado y simplificando en gran medida, a tal punto que se ha transformado en un paradigma en integración de fuentes de datos heterogéneas. Esto como resultado a que las ontologías cuentan con una gran capacidad de mapear formalmente con suficiente poder expresivo cualquier dominio de aplicación a través de una definición explícita del sentido de los conceptos y las relaciones entre conceptos que intervienen en el.

Por lo cual puede decirse que con esta metodología propuesta basada en ontologías permite solucionar parcialmente el problema de heterogeneidad semántica, existiendo sin embargo el principal obstáculo en la conciliación de las diferencias semánticas de las fuentes de datos.

1.7 Motivación

La cantidad de información que se encuentra disponible hoy en día sigue creciendo continuamente a un ritmo elevado, siendo este un problema que enfrentan los desarrolladores de software debido a la complejidad de los crecientes sistemas de información. Sin embargo, este reto a superar se traduce por parte del usuario en más contenido para consumir, pero también en más información a descartar antes de encontrar lo que realmente es de interés.

Si bien el objetivo que se persigue en la integración de esquemas basados en ontologías es resolver las heterogeneidades estructurales y semánticas entre los esquemas de las fuentes de datos. Estas ontologías representan la estructura de la información contenida en una base de datos a nivel conceptual, es decir, cada concepto en una base de datos física se enlaza a un concepto dentro de la ontología específica. Con la finalidad de facilitar una posterior unificación en una red de ontologías con los demás repositorios en un entorno homogéneo, produciendo nuevos esquemas unificados, a los que puedan acceder los usuarios a través de una interfaz uniforme e intuitiva basada en ontologías. Sin embargo, el proceso de mapeo de los repositorios de datos a ontologías se realiza de forma manual, existen distintas investigaciones acerca de la realización de este proceso de manera automática, tomando en consideración que siempre será indispensable la supervisión de un experto en el dominio de los datos mapeados.

La motivación para la realización de este trabajo surge a partir de la tendencia a la creación de la próxima evolución de la WWW, denominada Web Semántica y con ontologías, las cuales están siendo utilizadas como soporte en el proceso de obtención de conocimiento. A pesar de la relevancia que tiene esta fase de integración, se han llevado a cabo relativamente pocos esfuerzos científicos en este campo de aplicación. Lo cual da como consecuencia que esta sea una de las principales razones por las que el aumento de datos disponibles hoy en día, no se vea reflejado proporcionalmente en un incremento de conocimiento.

Es así como la aplicación de las ontologías en la solución de este y otros problemas de integración da sustento por su capacidad de representar e inferir conocimiento, de describir el significado de las cosas a través de los conceptos y las relaciones que definen un área o fragmento de la realidad así como de su capacidad de que en un mismo modelo o representación, unificar criterios o conceptos antes sin relación alguna entre sí y darles a cada uno su lugar en la jerarquía del dominio del conocimiento que representan. Para finalmente incrementar la experiencia del usuario proporcionando mayor precisión y personalización en los resultados.

1.8 Organización del documento

El resto del documento de tesis se encuentra organizado de la siguiente manera:

- El capítulo 2 aborda el estado del arte, donde se presentan los trabajos que sirven como base para el desarrollo de esta investigación.
- El capítulo 3 presenta el marco teórico, sobre el cual se encuentra desarrollado este trabajo; computo ubicuo, bases de datos, ontologías, su fundamento técnico y como se incorporan en la integración de información.
- En el capítulo 4 se presenta la metodología para abordar el modelo propuesto para la integración de información, y las etapas que conforman a dicha metodología
- El capítulo 5 aborda el caso de estudio, en el cual se desarrolla una aplicación móvil enfocada en los servicios médicos, empleando el modelo desarrollado en el capítulo anterior.

CAPÍTULO II. ESTADO DEL ARTE



Estado del arte

Por medio de una investigación realizada, en la que se buscaron trabajos y artículos con propuestas similares, se destacan algunos los cuales incluyen modelos en donde se toma referencia para la realización de este trabajo, así como de los trabajos mas recientes en el ámbito de integración de datos de manera general.

2.1 Integración de bases de datos heterogéneas

Framework and Knowledge for Database Integration

De acuerdo con este trabajo [16] tradicionalmente las investigaciones enfocadas en la integración de datos se centran en tratar de entender los problemas desde la perspectiva de los esquemas de datos y sus instancias. Sin embargo, cuando se habla de integración de fuentes de datos heterogéneas, se pasa por alto la semántica de estas y esto lleva a tener una integración sin ninguna optimización.

Un punto importante dentro de este trabajo es la propuesta de una framework que permita la integración de forma semiautomática de distintas bases de datos, esto debido a que en la gran mayoría los trabajos centrados en dicha integración se realizan de forma manual. En cambio, con este framework se realiza una buena aproximación de acuerdo con el dominio semántico y de conocimiento necesarios para una correcta integración.

Este autor proporciona 4 aspectos fundamentales para el framework propuesto los cuales son:

- Analizar cuales son los problemas de la integración para tener un alcance mucho mas amplio.
- El impacto que tiene el realizar dicha integración de bases de datos.
- Distinguir la diferencia entre un data warehouse y la instancia de base de datos federadas en el sistema.
- Identificar el tipo de conocimiento generado al realizar una integración.

Finalmente se debe considerar que la resolución de heterogeneidades semánticas, se debe tener conocimiento de acuerdo con el dominio de aplicación que tiene cada una de ellas. Dando como resultado una integración semántica, conceptual, lógica de consulta y de instancia.

En el siguiente esquema (figura 2.1) se muestra la propuesta de solución:

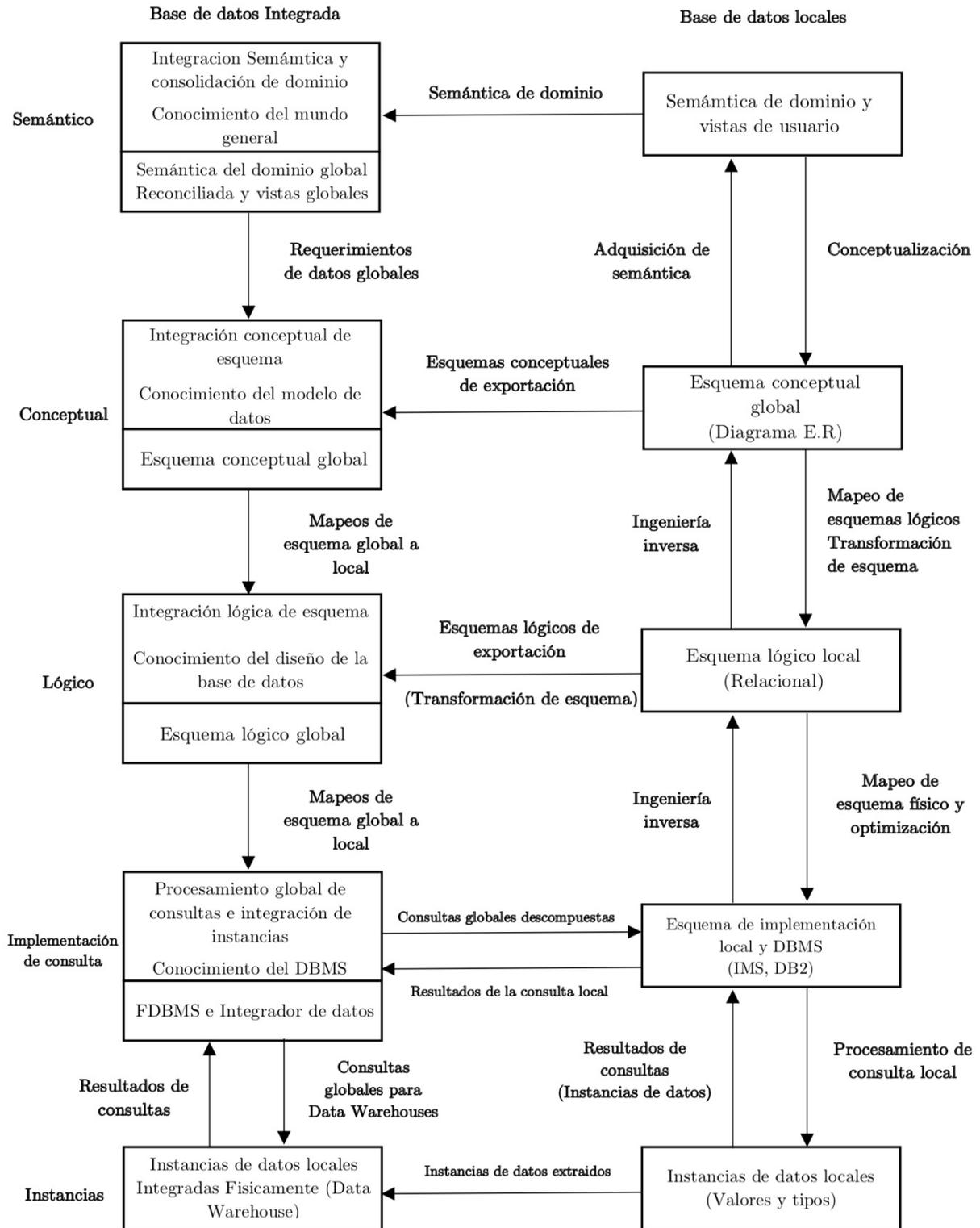


Figura 2.1 Estructura para la integración de bases de datos [16]

Los niveles que describe este autor de acuerdo con el modelo anterior se describen de la siguiente manera:

- **Nivel semántico:** Las distintas formas de modelar y representar un mismo objeto en las distintas fuentes de información y se necesita un conocimiento de manera general de la realidad a la que pertenece.
- **Nivel conceptual:** Se consigue un esquema conceptual global partiendo de los esquemas locales del subconjunto de datos provenientes de cada fuente que se ponen a disposición del usuario de la base de datos integrada. Estos esquemas se conocen comúnmente como entidad relación (E.R).
- **Nivel lógico:** En este nivel se tiene un esquema lógico de manera global partiendo de los esquemas lógicos locales partiendo del relacional.
- **Nivel de consulta:** En este nivel se integran instancias que están contenidas en resultados de las consultas locales de tal forma, en donde se representan a los mismos objetos se igualan, a su vez estas instancias representan objetos reales, así como sus propiedades.
- **Nivel de instancia:** Se refiere a la forma en que son consultadas, como se obtienen y se organizan las instancias en la base de datos integrada. Para este nivel se tienen dos opciones de implementación; Base de datos federada y Datawarehousing.

Modelo de integración y pre procesamiento de información distribuida

En este trabajo se enmarca un modelo propuesto por estos dos autores dentro del área de la informática la cual es la integración de las bases de datos y teniendo como dominio de aplicación la biomedicina [17].

Este trabajo ha sido desarrollado en el marco de varios proyectos europeos y ha generado diversas publicaciones en revistas científicas y congresos internacionales.

El enfoque de este trabajo esta denominado como el Descubrimiento de Conocimiento en Bases de Datos o KDD (“Knowledge Discovery in Databases”) y también conocido en ocasiones por minería de datos engloba las distintas metodologías existentes que describen el proceso de obtención de conocimiento a partir de los datos. Ante unos repositorios de datos heterogéneos y descentralizados, la política de integración es el primer tema que estos autores abordan en donde se debe de tratar el proceso de descubrimiento de conocimiento, ya que determina la metodología del proceso KDD.

El caso de estudio que se aborda es una aplicación en la medicina, que a su vez puede ser escalable en entornos empresariales en donde se suele optar por un enfoque centralizado en el que se cuenta con un repositorio central con una copia de los datos objeto de análisis. Estos ejemplos tienen como similitudes en que los almacenes de datos o “Data Warehouses” son la implementación más común de este enfoque, que cuenta con importantes ventajas en cuanto a seguridad, accesibilidad y rendimiento. Sin embargo, de acuerdo con este trabajo no todas las áreas de trabajo disponen de una jerarquía entre los nodos que asegure una correcta gestión de la información, lo que unido a su principal inconveniente que es la falta de actualización continua, hace que el enfoque distribuido o federado sea el elegido por otros campos como el de la investigación biomédica.

Para entender de una mejor manera este trabajo se debe de recordar lo que son los sistemas de bases de datos heterogéneas (SBDH) son los encargados de llevar a la práctica el enfoque distribuido y han sido el objetivo de numerosos esfuerzos científicos durante los últimos años. Retomando que una arquitectura típica de SBDH, dentro de un proceso de KDD, se puede observar en la siguiente figura:

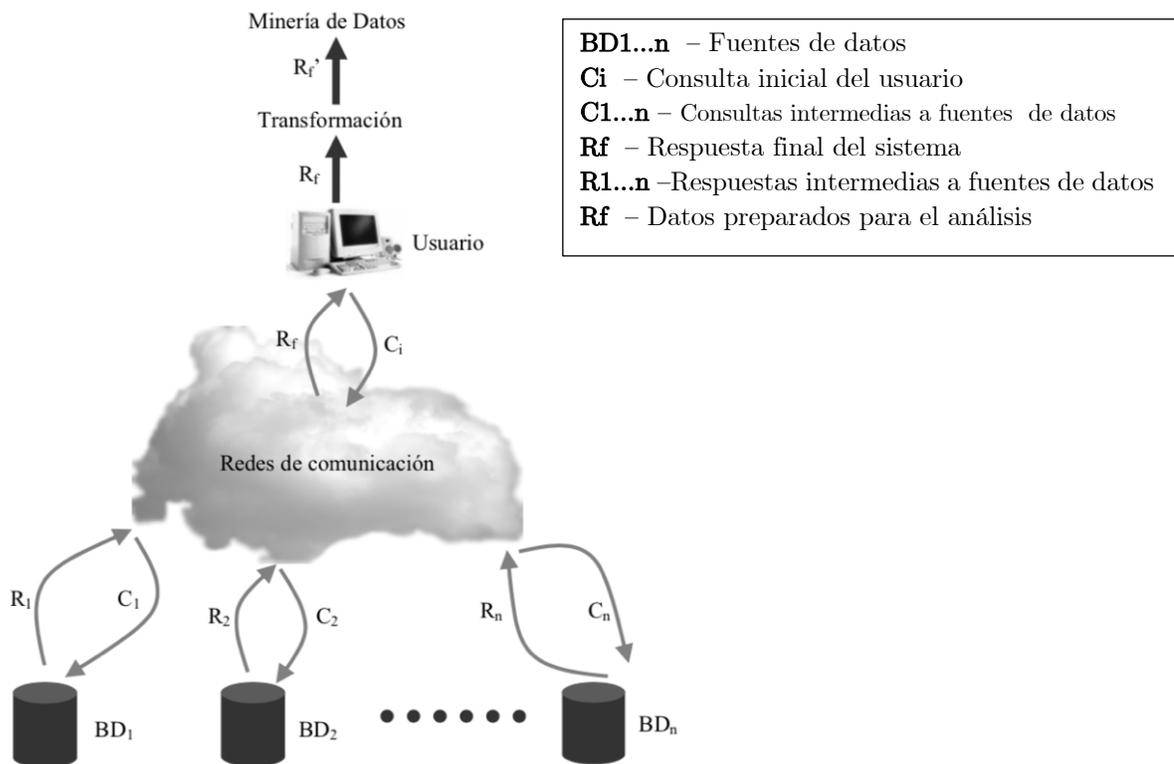


Figura 2.2 Arquitectura de un Sistema de Bases de Datos Heterogéneas dentro de un proceso KDD [17]

Las primeras fases del proceso KDD, encargadas de la integración y pre procesamiento de datos, son cruciales para asegurar tanto la calidad como una correcta unificación de las fuentes, preparando los datos para la obtención del conocimiento asociado

Para facilitar el proceso KDD, los investigadores están trabajando con una próxima evolución de la WWW, denominada Web Semántica, y con ontologías. Estas son utilizadas como soporte de metadatos en el proceso de obtención de conocimiento.

Ubiquitous Integration Architectural Issues

Dentro del trabajo se aborda el tema en el que los sistemas sufren distintos fallos cuando se intentan unificar en uno solo [18].

La integración de estos sistemas es una actividad que requiere interoperabilidad entre diferentes entidades. Con este trabajo se tiene la hipótesis que integrar sistemas reduce costos de mantenimiento y de operación. Otro punto que destacar es el hecho de la optimización de actividades dentro del ambiente de negocios y dentro de las organizaciones.

Un término que se destaca dentro de este trabajo es la interoperabilidad, la cual se define de la siguiente manera:

- La interoperabilidad es la propiedad que se refiere a la capacidad que tienen diferentes sistemas y organización para trabajar juntos (inter-operar). El término se usa a menudo en el sentido técnico de la ingeniería de sistemas. Alternativamente se usa en un sentido general, que tiene en cuenta factores sociales, políticos y de organización los cuales impactan en el funcionamiento sistema a sistema.

Aplicada al aspecto tecnológico, y en particular de una arquitectura, cuando se habla de interoperabilidad, se habla de la capacidad de un tipo de equipo o sistema lógico, de interactuar de forma correcta con otro, generalmente de otro fabricante o proveedor.

Desde un enfoque real se tiene que para pueda darse la interoperabilidad, y que por sí sola no es suficiente, es que exista detrás un documento o documentos que especifiquen totalmente la forma en que los sistemas tienen que interactuar, es por ello que los sistemas que se encuentran interactuando entre sí, generan inconsistencias al momento de realizar la transacción en el intercambio de información, el cual es el siguiente punto que se aborda dentro de este trabajo.

Es entonces que se presenta el caso en donde la integración de sistemas no resulta como se espera de manera inmediata, estos dos autores intentan reducir en lo posible las dificultades con este tipo de situaciones. Desde una perspectiva empresarial, se hace una distinción entre integración de aplicación a aplicación, de empresa a empresa y de empresa a consumidor.

Los conceptos de integración fundamentales tratados en este trabajo incluyen Enterprise Integration y Enterprise Service Bus, middleware y mensajería, para admitir la transferencia de datos.

Las arquitecturas de integración más recientes se basan en conceptos tales como arquitectura orientada a eventos, computación grid o procesamiento extremo de transacciones, lo cual se deja para un posible trabajo futuro dentro de la arquitectura que estos dos autores proponen.

La arquitectura que se propone se muestra en la siguiente figura llamada “Ubiquitous Integration model”. Los componentes de dicha arquitectura son: servidor, cliente, descriptor, y un interprete.

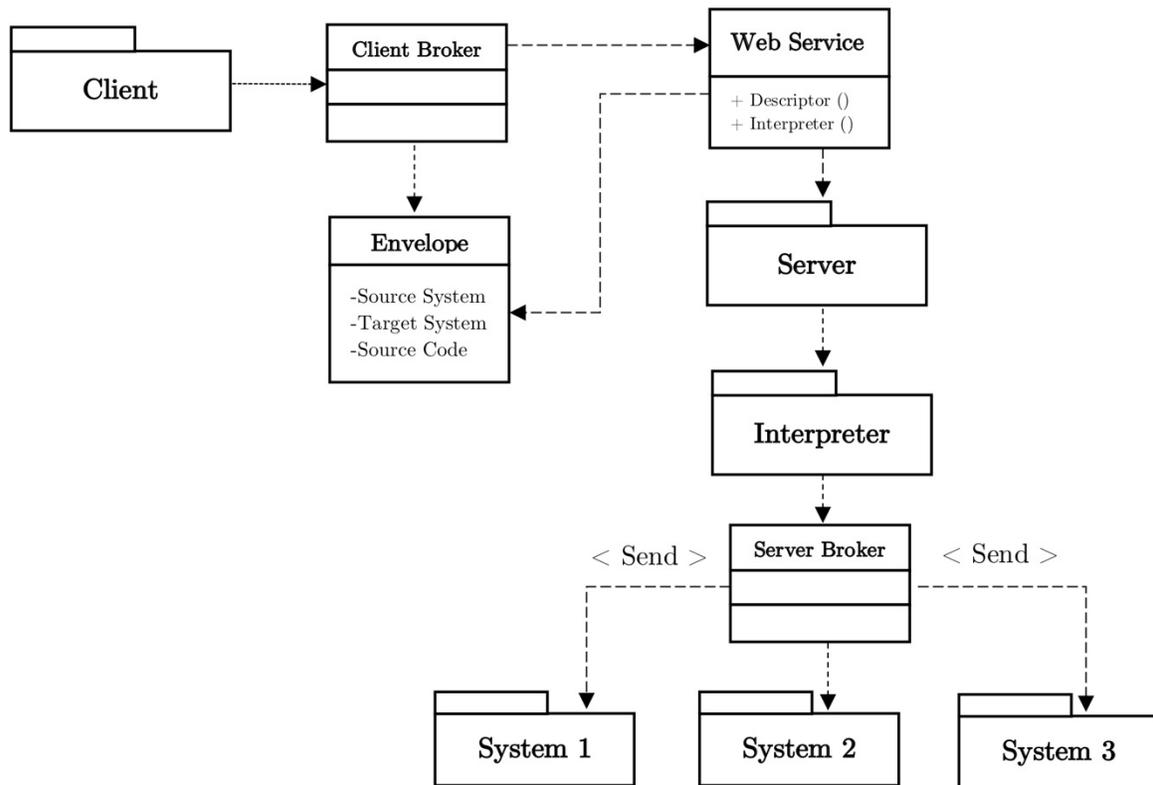


Figura 2.3 Ubiquitous Integration model [18].

De forma detallada se describen los componentes de dicho modelo:

- **System server:** es responsable de la publicación de las funcionalidades y proporciona el mecanismo para procesar las instrucciones recibidas en un lenguaje de programación XML, que el sistema entiende.

Las instrucciones delegadas provienen del sistema del cliente. El servidor recibe y procesa las órdenes. El servidor valida el cumplimiento de las instrucciones enviadas, en términos de estructura del lenguaje de programación XML y el uso correcto de las funciones publicadas de la biblioteca.

Después de la interpretación de las instrucciones, el resultado se devuelve al sistema del cliente. El éxito o los errores se devuelven al sistema del cliente.

- **Server Adapter System:** este componente no tiene, de forma nativa, las características que necesita el modelo de Integración ubicua. La integración es implementada y soportada por un adaptador de servidor que llevará a cabo la adaptación a este modelo. Por lo tanto, no es necesario realizar cambios en el código fuente del sistema base, donde se encuentran las funcionalidades.
- **Client System:** las aplicaciones del sistema cliente acceden a los sistemas del servidor, utilizando las características publicadas por ellos. Este modelo no requiere la validación local de las instrucciones para delegar en el sistema del servidor. Esta validación es realizada por el sistema del servidor. El sistema del cliente debe conocer el lenguaje de programación basado en XML, para estructurar los conjuntos de instrucciones. El sistema del cliente también debe conocer, de antemano, las características publicadas por el sistema del servidor.
- **Descriptor System:** este componente describe el sistema, con características detalladas permitidas por el sistema que implementa el modelo de Integración ubicua. A través de un mecanismo de descripción que se encuentra en un archivo independiente, para enumerar las clases y los métodos, los parámetros de entrada, el resultado y el tipo correspondiente. La descripción de las funciones disponibles en el sistema se realiza mediante un archivo, que enumera las clases y métodos disponibles para otros sistemas.
- **Client Broker:** es un componente del cliente con la tarea de proporcionar el canal para enviar el sobre, que contiene las instrucciones para ser procesado por el servidor. Esta interfaz es responsable de la compresión y el cifrado de las funcionalidades de datos, antes de transmitir las al sistema del servidor y también mediante el uso de credenciales. En este componente es posible intervenir para garantizar la eficiencia y la seguridad del uso de los canales de transmisión.

- **Server Broker:** es responsable del canal del receptor y de la entrada del sobre recibido del sistema del cliente, para ser procesado internamente. También es responsable de las actividades complementarias de validación de credenciales, descompresión y descifrado de los datos recibidos del sistema del cliente y también contribuye a la eficiencia y seguridad involucradas en el uso del canal de comunicación.
- **Interpreter:** El intérprete está ubicado en el sistema del servidor. Este componente principal es responsable de la interpretación del conjunto de instrucciones contenidas en el sobre del mensaje, delegado del sistema del cliente.

Para concluir con las aportaciones de este trabajo se tiene la propuesta de un modelo que permite la adaptación e implementación del sistema existente bajo este modelo, respaldado por la transmisión de instrucciones XML, organizado y recopilado por el sistema cliente y ejecutado por el servidor. Los principales beneficios y diferencias del estado del arte residen en aprovechar los servicios web y los mensajes SOAP, para proporcionar al canal el mecanismo de transferencia de instrucciones y recopilación de resultados, basado en un lenguaje de programación XML simple y gratuito, contrario al enfoque o modelos simples de transferencia centrada en datos reales y lenguajes de programación restringidos utilizados.

Las instrucciones y los datos se presentarán simultáneamente en el mismo mensaje, permitiendo la evolución y escalabilidad de los sistemas sin intervención en su código fuente central. Este modelo también permite el desarrollo de soluciones sin la necesidad de recurrir a un intérprete local, ya que las instrucciones se interpretan en el lado del servidor. Las limitaciones de este modelo pueden estar relacionadas con el número y el volumen de instrucciones que se pueden transferir entre los sistemas y el tráfico generado en la red y la duplicación de las instrucciones en varias secuencias de solicitudes.

Hacia la Integración de Bases de Datos

Este trabajo [19] presenta una arquitectura para realizar una integración de distintas bases de datos. La característica es que son bases de datos federadas de datos espaciales (SFBDE). Este tipo de bases de datos tienen características muy puntuales en donde se integran características geográficas, geométricas y alfanuméricas.

Sin embargo, con este trabajo se puede apreciar los inconvenientes que existen al intentar integrar este tipo de fuente de información, estos inconvenientes incluyen la especialización de las bases componentes, las unidades de medida, el idioma, la forma

de representar las coordenadas, el significado que representan cada columna, su representación gráfica, así como en la topología.

Como se mencionó al principio el resultado de este trabajo es la propuesta de una arquitectura, dicha arquitectura describe a continuación:

- **Proceso de transformación:** La etapa inicial de esta arquitectura se lleva a cabo realizando un mapeo de un esquema local de la base de datos a uno canónico, dicho modelo se puede representar con un esquema o bien una ontología.
- **Esquema de exportación:** Esta capa se encuentra en un nivel superior respecto a la anterior, en donde se realiza un proceso de filtrado el cual sirve para limitar las operaciones permitidas que se envían al esquema componente.
- **Esquema federado:** Esta capa abarca la integración de los esquemas de exportación, sin embargo, entre estos esquemas se encuentra un procesador de construcción el cual transforma los comandos para el esquema de federación en comandos para el esquema de exportación.
- **Esquema externo:** Como última instancia se encuentra este esquema, el cual define los esquemas de perfiles para los usuarios y para las aplicaciones. Si bien esta arquitectura se puede tener un esquema de federación por cada perfil de usuario que se cuente.

No obstante, esta arquitectura propuesta, no se cierra a la posibilidad de emplear tecnología con XML, si bien se puede emplear para el modelado de los esquemas o bien como un metalenguaje para la realización de consultas a las distintas fuentes de información. Esta posibilidad trae consigo la ventaja de poder representar los datos en forma de documentos XML dando la posibilidad de utilizar XML Query Language sobre los datos espaciales previamente representados, lo que permite tener una interoperabilidad.

La arquitectura detallada anteriormente se muestra a continuación (figura 2.4):

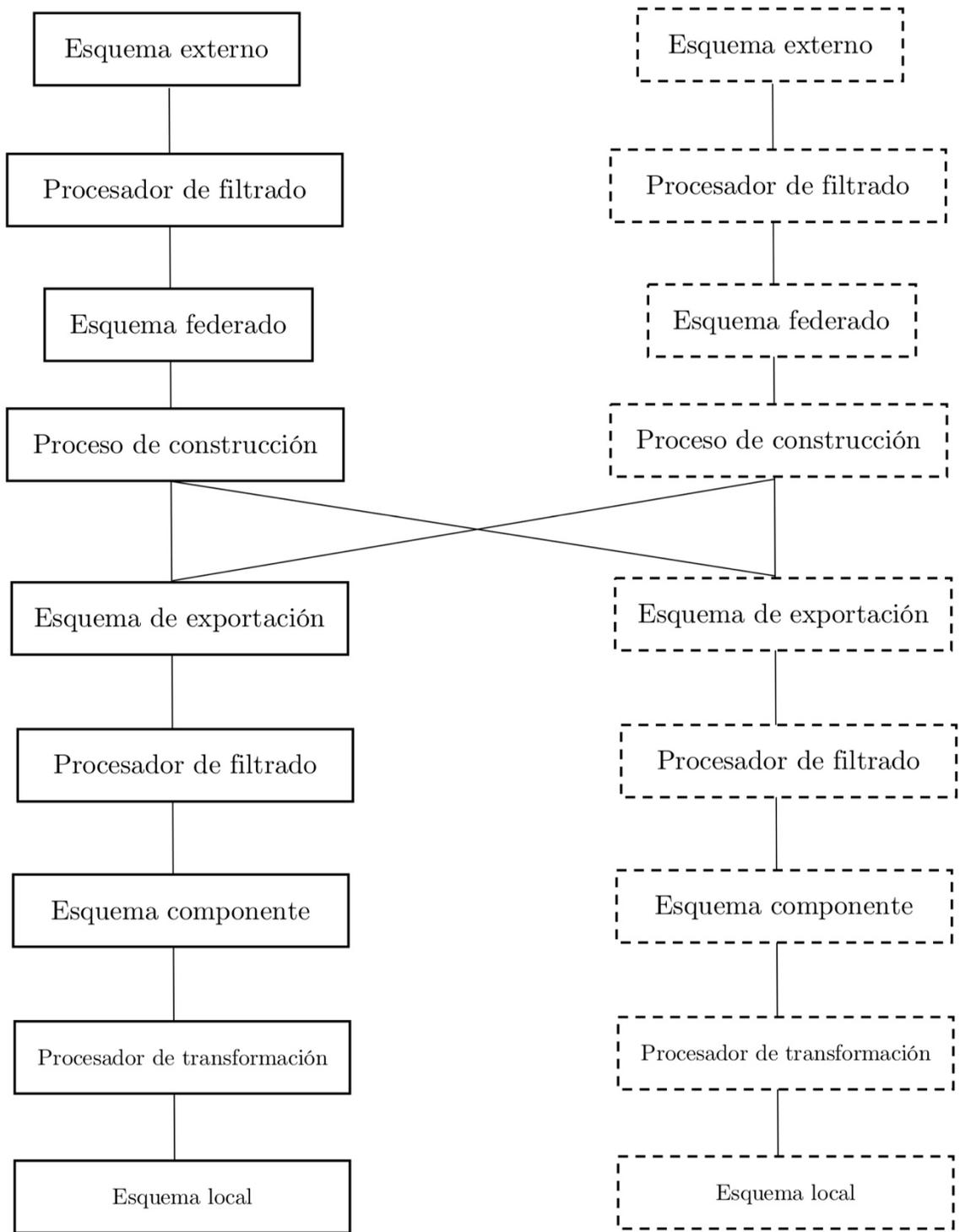


Figura 2.4 Arquitectura en 5 niveles para la integración de bases de datos [19].

2.2 Integración esquemática de bases de datos

Para esta integración, existe un problema denominado heterogeneidad semántica de esquemas, sin embargo, se han desarrollado mecanismos que permiten la integración que permiten realizar consulta de datos provenientes de distintas fuentes.

Retomando el trabajo realizado por Roger H. L. Chiang [16] en donde en el esquema propuesto se tiene un nivel de instancia en el cual se incluyen distintos mecanismos para implementar esta instancia, los cuales se detallan a continuación:

Data WareHousing: El concepto de data warehouse se originó en 1988 con el trabajo de los investigadores de IBM, Barry Devlin y Paul Murphy aunque el término data warehouse fue acuñado por William H. Inmon, el cual es conocido como el padre de Data Warehousing. Inmon describió un data warehouse como una colección de datos orientada a un tema específico, integrado, variante en el tiempo y no volátil, que soporta el proceso de toma de decisiones [20]

La arquitectura de un data warehouse puede ser dividida en tres estructuras simplificadas: básica, básica con un área de ensayo y básica con área de ensayo y data marts.

- **Con una estructura básica,** sistemas operativos y archivos planos proporcionan datos en bruto que se almacenan junto con metadatos. Los usuarios finales pueden acceder a ellos para su análisis, generación de informes y minería.
- **Al añadir un área de ensayo** que se puede colocar entre las fuentes de datos y el almacén, ésta proporciona un lugar donde los datos se pueden limpiar antes de entrar en el almacén. Es posible personalizar la arquitectura del almacén para diferentes grupos dentro de la organización.
- **Se puede hacer agregando data marts,** que son sistemas diseñados para una línea de negocio en particular. Se pueden tener data marts separados para ventas, inventario y compras, por ejemplo, y los usuarios finales pueden acceder a datos de uno o de todos los data marts del departamento.

Integración de esquemas de bases de datos: En este mecanismo de integración se plantea la idea de que no hay modificación en los esquemas originales, por el contrario, se genera un nuevo esquema global con el cual se utiliza para realizar las consultas específicas para cada esquema original [21].

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 [22].

Aunque todos estos sistemas proveen una solución parcial al problema de integración, los mecanismos anteriores no resuelven el problema de forma mas profunda y con mayor complejidad la cual es la integración semántica y de dominio de los datos que contienen dichos esquemas.

BUSTER

BUSTER (Bremen University Semantic Translation for Enhanced Retrieval) es una arquitectura presentada en el año 2000 por Heiner Stuckenschmidt [23]. Dicha arquitectura presenta una integración de manera inteligente de información, cabe destacar que esta arquitectura presenta un enfoque hibrido. Para lograrlo se apoya de tecnologías como son CORBA y el estándar de ontologías OIL. Dicha arquitectura se presenta a continuación (figura 2.5):

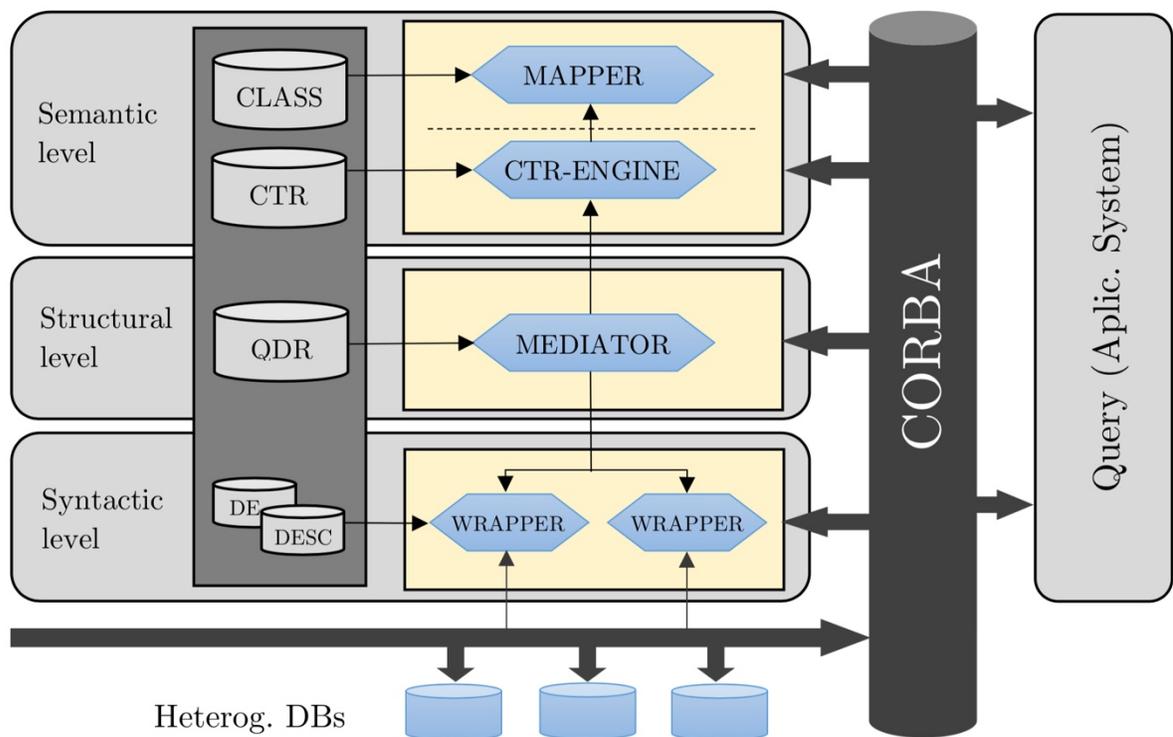


Figura 2.5 Arquitectura BUSTER [23]

Si bien la figura 2.5 detalla la propuesta de manera general, esta arquitectura contempla una fase de adquisición de datos la cual se muestra a continuación (figura 2.6):

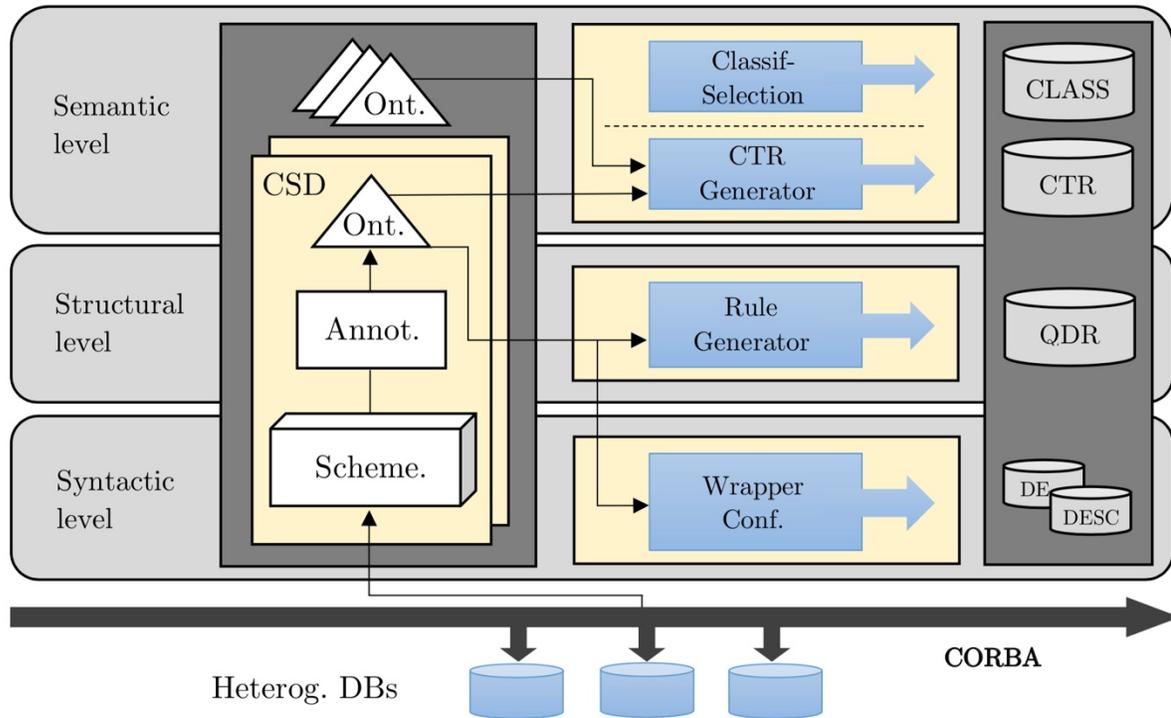


Figura 2.6 Adquisición de datos [23]

Como se muestra en la figura 2.6 de adquisición de datos los esquemas conceptuales se crean a partir de una ontología de dominio.

2.3 Integración semántica de fuentes de información

Las características comunes clave que tiene las aplicaciones de bases de datos que requieren integración semántica son que utilizan representaciones estructuradas (por ejemplo, esquemas relacionales DTD, XML, etc.) para codificar los datos, y que emplean más de una representación. Como tal, las aplicaciones deben resolver heterogeneidades con respecto a los esquemas y sus datos, ya sea para permitir su manipulación (por ejemplo, fusionar los esquemas o calcular las diferencias o para habilitar la traducción de datos y consultas a través de los esquemas). Muchas de esas aplicaciones han surgido a lo largo del tiempo y han sido estudiadas activamente por la comunidad científica [24].

2.3.1 Web Semántica

La Web Semántica ha sido uno de los movimientos tecnológicos dominantes en las últimas décadas que ha tenido un alto impacto para los usuarios de internet. El concepto de la Web Semántica, definido como “...una extensión de la Web actual, en la cual la información tiene un significado bien definido, facilitando a las computadoras trabajar mejor en cooperación con los humanos” fue introducido por Tim Berners-Lee en 2001[25] y su objetivo principal ha sido permitir que los datos almacenados en la Web puedan ser procesados por las máquinas de manera inteligente, facilitando a las personas la búsqueda, integración y análisis de la información disponible.

Con el apoyo de la W3C, las ideas de Berners-Lee acerca de la Web Semántica derivaron en un conjunto de estándares y tecnologías que conforman la base de su implementación. Entre estos estándares destacan RDF [26], OWL [27] y SPARQL[28], que han sido utilizados en proyectos incluso de dominios distintos al de la Web, por ejemplo: proyectos de ciudades inteligentes, telemedicina, investigación, colaboración científica, entre otras.

Siguiendo la visión de la Web Semántica y anticipando su evolución, el Fondo de Información y Documentación para la Industria (INFOTEC) planteó en 2008 la construcción de una plataforma y un framework para el desarrollo acelerado de aplicaciones, aprovechando la tecnología y estándares disponibles con el objetivo de lograr que la información en las aplicaciones desarrolladas contara con una estructura y significado bien definidos. Dicha plataforma, denominada SemanticWebBuilder Platform [29] y [30], implementa las ideas del desarrollo dirigido por ontologías y permite obtener, de manera semiautomática, el código fuente de una aplicación a partir de su definición en una ontología específica. Esto permite contextualizar la información de las aplicaciones, usando RDF y OWL, para mejorar la búsqueda e integración de información, además de aprovechar las ventajas de las ontologías, como son la reutilización de conceptos, la extensión y las reglas de inferencia.

En años recientes, con el crecimiento exponencial 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 resultando transparente al usuario. El principal problema que se genera con estos sistemas es la detección y eliminación de información duplicada, es por ello que estos sistemas se apoyan de ontologías y distintos mecanismos, los cuales se detallan a continuación.

De manera general la web semántica se constituye principalmente de los siguientes componentes [29]:

- **XML:** Sintaxis superficial para documentos estructurados.
- **XML SCHEMA:** Lenguaje para definir la estructura de los documentos XML.
- **RDF:** Modelo de datos para los recursos y las relaciones que se pueden establecer entre ellos.
- **RDF SCHEMA:** Vocabulario para describir las propiedades y las clases de los recursos RDF con una semántica para establecer jerarquía de generalización entre dichas propiedades y clases.
- **SPARQL:** Lenguaje de consulta sobre RDF que permite hacer búsquedas sobre los recursos de la web semántica utilizando distintas fuentes de datos.

2.3 Integración de información con ontologías

Las ontologías tienen vital importancia desde diferentes puntos de vista. Esto se debe a que las ontologías surgen a partir de diferentes ramas interdisciplinarias en donde destacan la filosofía y la lingüística, las cuales integran un papel fundamental en el análisis de la estructura de una determinada realidad y por otra, a partir del uso que se hace de estas en el campo de la informática, en donde sobresale el área de la Inteligencia Artificial [30].

Retomando las arquitecturas mostradas la integración existe en dos diferentes tipos de sistemas: los sistemas de información centralizados y sistemas de integración punto a punto (peer to peer). Si bien en los sistemas centralizados generalmente se tiene un esquema global el cual provee al usuario de una interface uniforme para acceder a la información de las fuentes de información. En contraparte los sistemas peer-to-peer, no tienen puntos globales de control en las fuentes de información, por lo tanto, cualquier punto puede aceptar consultas de la información distribuida en todo el sistema [31].

Retomando la idea inicial, las ontologías se han utilizado para tratar de solucionar el problema de integración debido a que proveen una conceptualización explícita y su cálculo es relativamente sencillo en un dominio. Generalmente se emplean de dos maneras distintas [32]:

Ontología simple: Todos los esquemas se encuentran directamente relacionados con una ontología global, de la cual se cuenta con solamente una interfaz para el usuario. Sin embargo, para que se pueda hacer uso de esta ontología simple se necesita que todas las fuentes pertenezcan a casi el mismo dominio incluyendo de igual forma el mismo nivel de granularidad.

Múltiples ontologías: A diferencia del esquema anterior, teniendo múltiples ontologías cada fuente es descrita por su propia ontología. En contraste en vez de utilizar únicamente una ontología en común, cada una de ellas es mapeada una a otra. Para llevar a cabo esto, se requiere una representación formal para definir relaciones de los mapeos entre las ontologías.

La manera general del empleo de las ontologías se identifica en 5 rubros principalmente [33]:

- Representación de metadatos
- Soporte de mapeos
- Declaración de mediadores
- Conceptualización global
- Soporte a consultas de alto nivel

A continuación, se presentan cuatro de los principales proyectos que usan ontologías para la integración de bases de datos que se centran en cuestiones semánticas. El modelo propuesto presentado en este trabajo se basa en parte en el resultado de estos proyectos.

InfoSleuth Project

El proyecto InfoSleuth se basa en los resultados de Carnot, ampliando sus capacidades [34]. Carnot [35] es uno de los proyectos pioneros en el dominio de la integración y aborda la integridad semántica. Carnot usa Cyontology (llamada base de conocimiento de sentido común) además del conocimiento extraído de las definiciones de esquema para la integración semántica.

InfoSleuth es una solución basada en varios agentes para la recuperación de información y servicios de fuentes autónomas y cambiantes, como las de Internet. Su arquitectura se basa en diferentes tipos de agentes, como agentes de usuario, agentes de ontología, agentes de consulta, etc. Un usuario puede seleccionar una ontología (a través de un

agente de usuario) de una lista de ontologías ofrecidas por un agente de ontología. Las ontologías se crean para cada grupo de usuarios. InfoSleuth ofrece la capacidad de generar consultas basadas en una ontología seleccionada y, en consecuencia, combina información o servicios con las consultas del usuario.

KRAFT Project

Es un proyecto para la integración de información heterogénea [36], que utiliza ontologías para resolver problemas de semántica. Extraen el vocabulario de la comunidad y la definición de términos de documentos existentes en un dominio de aplicación. KRAFT utiliza la ontología compartida [37] como una base para el mapeo entre las definiciones de ontología y la comunicación entre agentes. En la ontología compartida es "elegida para hacer que la ontología compartida sea tan expresiva como la 'unión' de las ontologías". Sin embargo, no se establece la definición de la unión de las ontologías y sus similitudes o diferencias con la ontología compartida. KRAFT detecta un conjunto de discrepancias ontológicas (como se describe en [36]) y establece asignaciones entre la ontología compartida y las ontologías locales. Un resultado importante de este proyecto para nuestro trabajo es la metodología para construir ontologías presentada en [37].

COIN Project

El proyecto COIN [38] presenta una arquitectura adecuada para la interoperabilidad semántica y partes inspiradas del trabajo presentado en esta tesis. El rol del Modelo de Dominio en la arquitectura COIN se puede comparar con el de una ontología. Los componentes de la arquitectura se adaptan a un enfoque basado en la ontología. La manipulación de datos basada en el modelo de dominio (ontología) y la relación de los datos con el modelo de dominio son hechos importantes considerados en su arquitectura. Sin embargo, un Modelo de Dominio está más cerca de un esquema conceptual que una ontología.

En un ejemplo en [38], se puede ver que "cantidad de dinero" se considera un subtipo de "número semántico" mientras que el número es solo un tipo primitivo para representar el valor -o "tipo de moneda" es un subtipo de "cadena semántica". Sin embargo, según nuestra definición de ontología, se basa en las conceptualizaciones de las personas en una comunidad. Por lo tanto, "cantidad de dinero" es un monto o una cantidad. Tratar el "número semántico" como un súper tipo es el resultado de la influencia del desarrollo de la aplicación. Si bien, "cantidad de dinero" o "tipo de moneda" están relacionados con un valor de tipo de letra o cadena, respectivamente, solo para fines de representación.

OBSERVER

OBSERVER [39] utiliza ontologías para permitir consultas en fuentes heterogéneas. Reemplaza términos en consultas de usuarios con términos adecuados en ontologías objetivo, mediante relaciones Inter-Ontología.

Este trabajo usa la descripción lógica ya que tanto el lenguaje de definición de ontología como el de consulta y el método tradicional de integración.

Un módulo interesante en la arquitectura OBSERVER es el Administrador de Relaciones Inter-Ontológicas. Mantiene la relación entre las definiciones ontológicas de términos en diferentes ontologías. Por medio de tales relaciones inter-ontológicas, OBSERVER reemplaza términos en consultas de usuarios con términos adecuados en ontologías de objetivos. Este trabajo también propone el uso de relaciones de similitud entre ontologías para la integración de esquemas. Las relaciones Inter-Ontology son sinónimo y homónimo: las mismas tres relaciones utilizadas en [37]. Las mismas relaciones también se utilizan para construir cada ontología en OBSERVER. [39] se refiere a la traducción de términos en una ontología de usuario mediante el uso de relaciones inter-ontológicas como integración ontológica.

Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata

A continuación, se presenta un trabajo de tesis [40] donde hace mención en la interoperabilidad de los sistemas y la integración de datos se están convirtiendo en cuestiones cada vez más importantes, ya que tanto la cantidad de datos disponibles como la cantidad de productores de datos están creciendo.

Así mismo menciona que La reutilización de datos producidos por otras fuentes es la principal motivación de la integración. Las empresas tienden a reducir su inversión para producir datos mediante la integración de datos subcontratados. Sin embargo, la integración como una condición previa para la reutilización de datos exige sus propios costos. La integración de datos debe resolver las diferencias en las estructuras de datos, así como también, resolver las heterogeneidades semánticas. Según Hakimpour la semántica se refiere al significado de los datos en contraste con la sintaxis, que únicamente define la estructura de los elementos del esquema (por ejemplo, clases y atributos) en las bases de datos.

Esta tesis contribuye a manejar la heterogeneidad semántica durante la integración del esquema de la base de datos.

Hakimpour propone una metodología para la integración de bases de datos espaciales a por medio de ontologías. Las ontologías se utilizan para crear un esquema global que define de forma explícita, a partir de las conceptualizaciones de los miembros

de la comunidad, los términos empleados para nombrar a los elementos de cada esquema de base de datos en su vocabulario, con la mínima influencia del dominio de aplicación, lo que lo hace independiente del nombrado de los elementos de esquema.

El resultado de la fusión es utilizado por un integrador de esquema para construir un esquema global integrado a partir de esquemas locales. Por ejemplo, el integrador de esquema sugiere una clase en el esquema global, así como el mapeo de esta clase y sus atributos a una o más clases en los esquemas locales subyacentes. Por lo tanto, el enfoque no solo sugiere un esquema global, sino que también trata de encontrar todos los posibles mapeos significativos entre el esquema global generado y el esquema del componente.

La figura 2.8 ejemplifica el conjunto de acciones a seguir para conseguir la integración de datos entre esquemas con vocabularios distintos según Hakimpour:

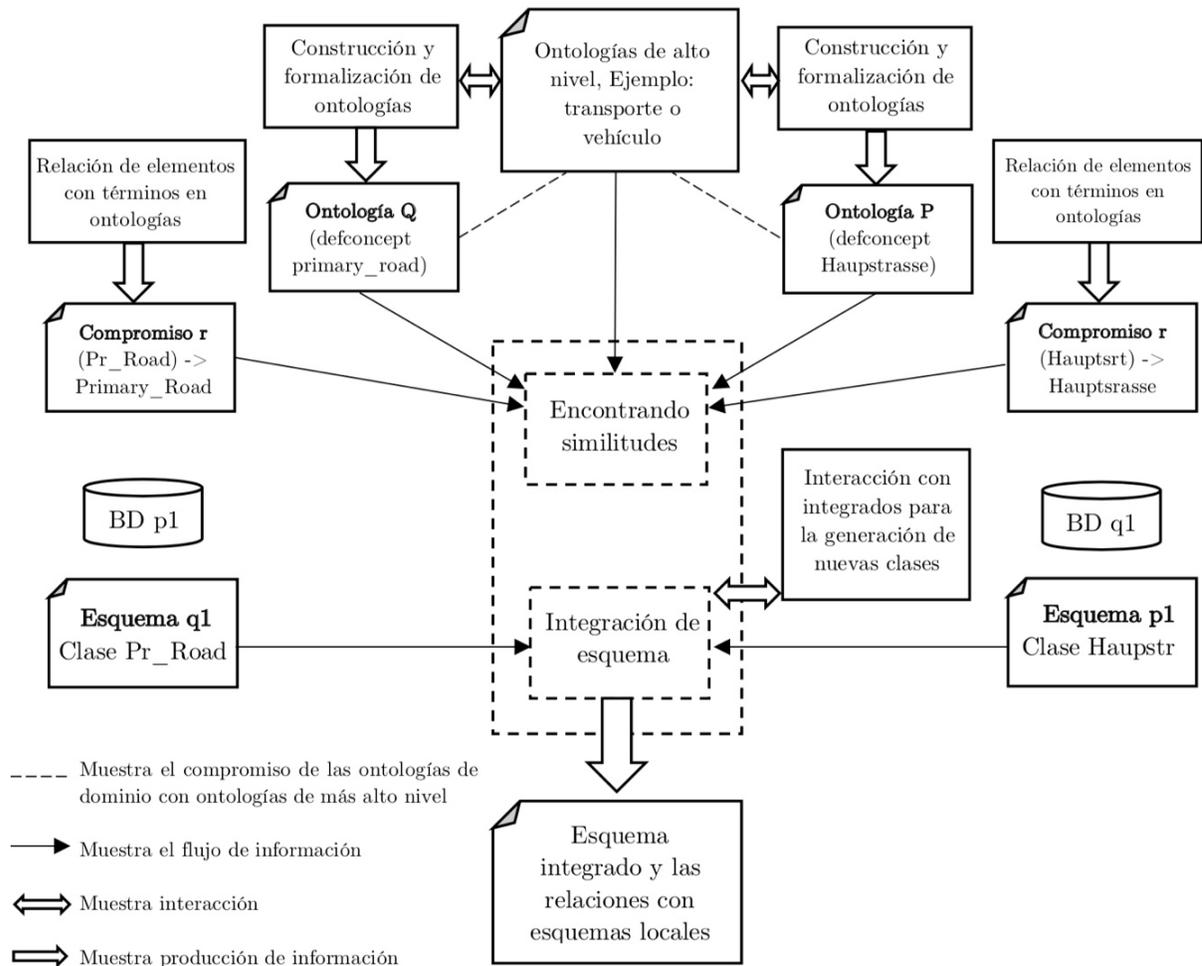


Figura 2.8. Modelo de Hakimpour [40]

CAPÍTULO III. MARCO TEÓRICO



Por tanto, los objetos cotidianos en los que se integra la tecnología de computación tienen una serie de características que permiten y delimitan la creación del entorno ubicuo buscado: Comunicación entre dispositivos, ya que los elementos del sistema disponen no solo de capacidad de computación, sino también de comunicación, tanto con el usuario como con los demás elementos a su alrededor mediante Wifi, Bluetooth, GPRS/UMTS, UWB, RFID, etc.

De este modo, la Computación Ubicua, incorpora cuatro nuevos conceptos:

- **Uso eficaz de espacios "perspicaces":** Se basa, en la detección del estado de un individuo y de sus necesidades, deducidas de dicho estado, ya sea en la oficina, sala de reuniones, clase, domicilio, coche, etc. El espacio perspicaz surge cuando varios dispositivos inteligentes coinciden en el mismo espacio físico e interactúan colaborativamente para dar soporte a los individuos que se encuentren en el [4]. La domótica, computación ubicua en el domicilio, es la aplicación más popular.
- **Invisibilidad:** Actualmente, se está lejos de la propiedad expuesta por Weiser para los sistemas ubicuos, la completa desaparición de la tecnología de la consciencia del usuario. Una buena aproximación es tener presente, en el diseño de estos sistemas, la idea de mínima distracción del usuario. La invisibilidad va a requerir del cambio drástico en el tipo de interfaces que nos comunican con los computadores. Reconocimiento de voz y de gestos, comprensión del lenguaje natural y del texto manuscrito, en la dirección hombre-máquina y en el sentido contrario, síntesis de lenguaje hablado y escrito y de representaciones gráficas.
- **Escalabilidad local.** El concepto de localidad de servicios en computación ubicua es fundamental frente a la universalidad de servicios de Internet. Los usuarios disponen de capacidades asociadas al contexto en el que se encuentran, careciendo de sentido, por ejemplo, que las aplicaciones domóticas situadas en el domicilio particular tengan que estar escrutando las necesidades del usuario que se encuentra trabajando en ese momento en la oficina. Al igual que la mayoría de las interacciones en la naturaleza, la proporcionada por estos sistemas, decrece con la distancia al usuario.
- **Ocultación de los desniveles de acondicionamiento:** Dependiendo de la infraestructura y del desarrollo tecnológico disponible, la distribución de los servicios ofrecidos puede ser muy poco uniforme, en esta situación el principio de invisibilidad puede no cumplirse ya que el usuario detectaría desagradables transiciones. Este requisito es hoy día el más alejado respecto de la situación ideal, los sistemas que incorporan computación ubicua están aislados, sin continuidad entre unos y otros.

Aplicaciones

La Computación Ubicua permite abarcar muy diversos campos:

- Almacenamiento de información, búsqueda y tratamiento.
- Visualización de información, enriqueciendo el contenido que se presenta.
- Simulación y realidad aumentada, permitiendo ofrecer entornos virtuales o superponer información extra
- Construcción y modelado, haciendo uso de modelos en la realidad que estén asociados con modelos digitales virtuales.
- Gestión, configuración y control de sistemas complejos, como plantas industriales
- Sistemas de programación basados en el uso de objetos físicos para la codificación de los algoritmos.
- Trabajo colaborativo.

De los campos anteriores este trabajo esta especialmente centrado en 4; Almacenamiento de información, tratamiento, construcción y modelado de los cuales se derivan los puntos a tratar a continuación.

3.2 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.

Aclaro este punto se puede definir una base de datos como una colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones [42].

No obstante, si se habla de la definición de una base de datos no se puede pasar por alto definir lo que es un sistema gestor de base de datos (SGBD) el cual consiste en una colección de datos inter-relacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto practica como eficiente [15].

Es claro que la forma física como estén almacenados los datos es independiente del concepto que tengamos de ellos. Esto es el conjunto de programas que saben como traer,

unir y mostrar los datos, así como aquellos encargados de almacenarlos, los que le dan coherencia al concepto base de datos.

Entonces se dice que existen dos visiones de la base de datos definido por:

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

Digamos que es como la diferencia entre harina, levadura, sal y agua por separado y una pieza de pan. Quién le da coherencia a esa pieza de pan es el proceso que se sigue para elaborarlo.

Es importante conceptualizar esto, porque del diseño de la estructura lógica depende toda la funcionalidad del sistema. Almacenar datos en una base de datos aprovechando solamente la estructura física no ofrece, relativamente, ninguna ventaja. En cambio, un buen diseño de acuerdo con la naturaleza de los datos y a la forma como serán explotados hace toda la diferencia.

3.2.1 Modelo conceptual

Si bien en la sección de antecedentes de este trabajo se abordó lo que es un esquema conceptual, en esta sección concretamente se puede retomar nuevamente este tema. Un esquema también conocido como modelo de dominio es la descripción de cómo se relacionan los conceptos en un problema. El modelo conceptual sirve para representar un problema de manera gráfica a través de diagramas entidad relación, diccionarios/glosarios y diagrama de clases, por mencionar algunos [15].

¿Para qué modelar?

Es importante para abstraer un problema e identificar como interactúa el sistema en el cual se desenvuelve la solución.

Al modelar un problema se identifica su funcionamiento y es realizado para solucionar problemas, es aquí donde entra la esencia de este trabajo y es por ello por lo que la propuesta de solución es un modelo.

3.2.2 Base de datos relacional

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de una tabla representa una relación entre un conjunto de valores. De manera informal, cada tabla es un conjunto de entidades, y cada fila es una entidad. Dado que cada tabla es un conjunto de relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación del que toma su nombre el modelo de datos relacional [42].

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. Está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos.

Entidad

Representa una "cosa", "objeto" o "concepto" del mundo real con existencia independiente, es decir, se diferencia únicamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad.

- Una persona. (Se diferencia de cualquier otra persona, incluso siendo gemelos).
- Un teléfono. (Aunque sean de la misma marca, el mismo modelo, tendrán atributos diferentes, por ejemplo, el número de serie).
- Una casa (Aunque sea exactamente igual a otra, aún se diferenciará en su dirección).

Una entidad puede ser un objeto con existencia física como: una persona, un animal, una casa, etc. (entidad concreta); o un objeto con existencia conceptual como: un puesto de trabajo, una asignatura de clases, un nombre, etc. (entidad abstracta).

Atributos

Los atributos son las características que definen o identifican a una entidad. En un conjunto de entidades del mismo tipo, cada entidad tiene valores específicos asignados para cada uno de sus atributos, de esta forma, es posible su identificación unívoca

- **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.

Clave

Es un subconjunto del conjunto de atributos comunes en una colección de entidades, que permite identificar inequívocamente cada una de las entidades pertenecientes a dicha colección. Asimismo, permiten distinguir entre sí las relaciones de un conjunto de relaciones.

Dentro de los conjuntos de entidades existen los siguientes tipos de llaves:

- **Superclave:** Es un subconjunto de atributos que permite distinguir unívocamente cada una de las entidades de un conjunto de entidades. Si se añade un atributo al anterior subconjunto, el resultado seguirá siendo una superclave.
- **Clave candidata:** Se trata de superclave mínima, es decir, cualquier subconjunto de atributos de la misma no puede ser una superclave.
- **Clave primaria:** Es una clave candidata, elegida por el diseñador de la base de datos, para identificar unívocamente las entidades en un conjunto de entidades.

Los valores de los atributos de una clave no pueden ser todos iguales para dos o más instancias.

Relaciones

Una relación es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente López con el préstamo P-15. Esta relación especifica que López es un cliente con el préstamo número P-15.

Se llama *grado* del conjunto de relaciones a la cantidad de conjuntos de entidades participantes en la relación.

Restricciones

Un esquema de desarrollo Entidad Relación puede definir ciertas restricciones a las que los contenidos de la base de datos se deben adaptar. Las cuales son:

Correspondencia de cardinalidades

Expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B, la correspondencia de cardinalidades debe ser una de las siguientes:

- **Relaciones uno a uno:** Para cada instancia en una entidad hay a lo más una instancia asociada en otra entidad.
- **Relaciones uno a varios:** Una entidad en A se asocia con cualquier número de entidades en B (ninguna o varias).
- **Relaciones varios a varios:** Una entidad en A se asocia con cualquier número de entidades (ninguna o varias) en B, y una entidad en B se asocia con cualquier número de entidades (ninguna o varias) en A.

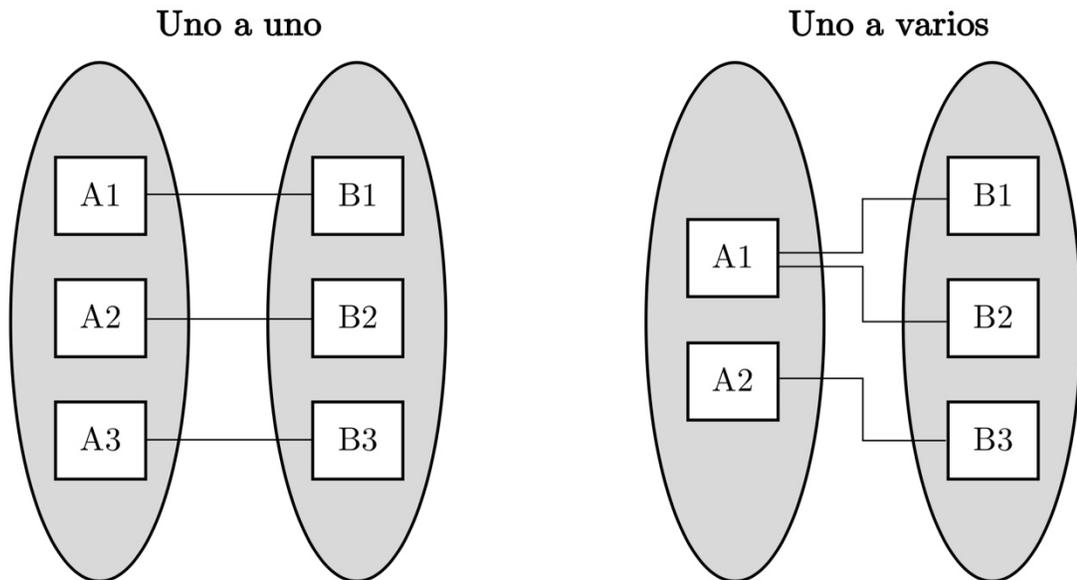


Figura 3.2 Correspondencia de cardinalidades [41]

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.

3.3 Integración de información

La integración de múltiples sistemas de información tiene como objetivo combinar sistemas seleccionados de manera que formen un nuevo todo unificado y ofrezcan a los usuarios la ilusión de interactuar con un sistema único de información. La razón para la integración es doble: primero, dado un conjunto de sistemas de información existentes, una vista del sistema integrado se puede crear para facilitar el acceso a la información y la reutilización de esta, a través de una sola información de punto de acceso. En segundo lugar, dada una cierta necesidad de información, los sistemas de información se combinan para obtener una base más integral para satisfacer la necesidad [43].

Estas soluciones al almacenamiento y tratamiento de datos han creado a su vez nuevos retos, como es el acceso integrado a datos almacenados en distintos sistemas de bases de datos heterogéneos, generalmente localizados en emplazamientos remotos. En una primera clasificación, podemos diferenciar los tipos de integración en dos:

- **Integración Vertical.** Donde se unifica información semánticamente similar y proveniente de distintas fuentes, por ejemplo, integrar datos de pacientes con cáncer de mama de distintos países en un solo repositorio, incluyendo los campos comunes y todos los pacientes.
- **Integración Horizontal.** En la que se agrega información adicional o complementaria a una determinada fuente, por ejemplo, incluir datos genéticos a los datos clínicos de pacientes con cáncer de mama.

3.3.1 Factores que impiden la integración de datos

Existen varios factores por los cuales la integración se vuelve compleja y no es fácil de realizar, esto es por mencionar respecto a los SGBD utilizados, aunque todos siguen el enfoque entidad / relación (E/R) pueden existir conflictos de representación. Así pues, las dificultades en la integración de bases de datos no solo se dan desde el punto de vista técnico al utilizar distintos SGBD en distintas localizaciones. El mayor problema se debe a que la misma información puede ser representada de diversas formas,

incluso utilizando el mismo SGBD. Los problemas técnicos de integrar varias bases de datos descentralizadas, se afronta a través de Internet utilizando estándares de comunicación como ODBC (“Open Database Connectivity”), que permiten el acceso desde distintos lenguajes de programación a la mayoría de los SGBD del mercado. Pueden aparecer, sin embargo, inconsistencias más complejas (o incluso imposibles) de resolver en lo referente a la representación de la información. Estos conflictos se pueden clasificar en [44]:

- **Conflictos estructurales.** Aquellos referentes al esquema o estructura de la base de datos. Estos conflictos se dan cuando, en el diseño de una base de datos, se pueden utilizar distintas alternativas para representar la misma información, por ejemplo, la utilización de distintos tipos de datos, múltiples representaciones de una relación, diferente disposición de atributos en las tablas y distinto número de tablas, utilización de valores enumerados frente a texto libre.
- **Conflictos de nombre.** La utilización de distintos sinónimos para un mismo concepto o de distintas escalas para una misma medida hace que una computadora interprete los valores como distintos cuando en realidad no lo son, provocando un conflicto de nombre o escala.
- **Conflictos semánticos.** Este tipo de conflictos se dan cuando los valores en la base de datos son similares, pero no exactamente equivalentes, por ejemplo, conflictos debidos a la utilización de distintos rangos de valores o solapamiento de conceptos.
- **Conflictos de contenido.** Se producen cuando al integrar distintas fuentes, parte de la información de alguna de ellas no está representada. Ya sea porque esté implícita, que se pueda derivar de los datos que sí están representados o simplemente porque falten esos datos.

Antes de analizar los problemas de las bases de datos que se deseen integrar, se ha de abordar la localización dispersa de las fuentes y su política de acceso y actualización. En la siguiente sección se discuten los enfoques centralizado y distribuido, sus características, ventajas e inconvenientes y como afecta cada uno a las distintas metodologías para el modelo que se propone en este trabajo.

A continuación, se enlistan algunos factores que impiden la inmediata integración:

- Hardware y sistemas operativos.
- Software de gestión de datos.
- La semántica de datos, de modelos y esquemas.
- Middleware.
- Interfaces de usuario.
- Las reglas de negocio y las restricciones de integridad [44]

3.3.2 Técnicas para abordar el problema de la integración

La Figura 3.3 muestra la clasificación que distingue los distintos enfoques de integración de acuerdo con el nivel de abstracción donde se realiza la integración.

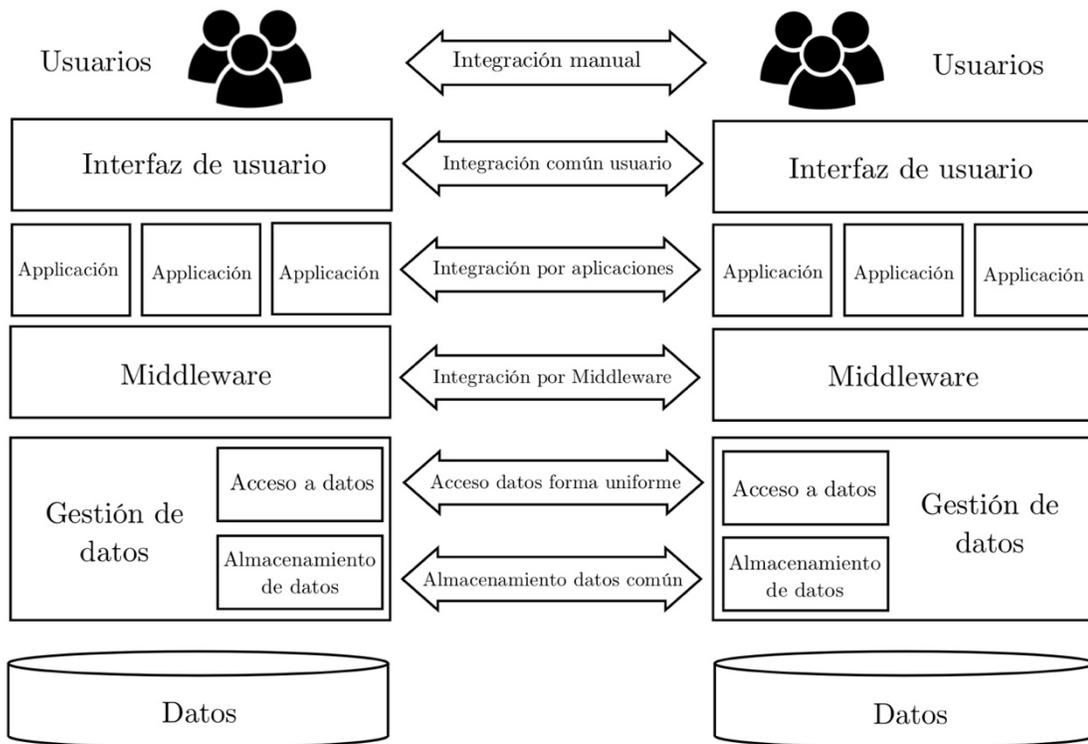


Figura 3.3. Distintos enfoques de integración de acuerdo al nivel de Integración [45]

Integración manual: Usuarios tienen que lidiar con distintas interfaces y lenguajes de consulta. Además, deben saber con detalle la representación, ubicación, lógica de los datos, así como su semántica.

Interfaz Común para usuario: El usuario es provisto por una interfaz de aspecto igual. Los datos aun se tienen por separado por ende la integración es de forma manual por el usuario.

Integración por Aplicaciones: Este enfoque utiliza aplicaciones de integración que acceden a diversas fuentes de datos y devuelven resultados integrados al usuario. Esta solución es practica para un pequeño número de sistemas de componentes. Sin embargo, las aplicaciones se han vuelto cada vez más grandes, debido al número de interfaces del sistema y el formato de datos para homogeneizar, en consecuencia, la integración crece.

Integración por Middleware. Middleware proporciona funcionalidad reutilizable que se utiliza generalmente para resolver aspectos específicos del problema de la integración, como se ha hecho por aplicaciones middleware para SQL (como ejemplo característico tenemos ODBC y JDBC). Mientras que las aplicaciones están exentas de la funcionalidad común de integración, todavía se necesitan esfuerzos de integración en aplicaciones. Un ejemplo, es el middleware para SQL que proporciona un único punto de acceso para enviar consultas SQL para todo componente conectado sistemas. Sin embargo, los resultados de la consulta no se integran en un conjunto de resultados único y homogéneo.

Acceso a datos de forma uniforme: En este caso, una integración lógica de los datos se lleva a cabo en el nivel de acceso a datos. Aplicaciones globales están provistas de un sistema unificado de visión global a los datos distribuidos físicamente, aunque los datos sólo están disponibles de forma virtual. Sin embargo, el suministro de datos integrados físicamente consume mucho tiempo ya que el acceso de datos, homogeneización, y la integración tienen que ser en tiempo de ejecución.

Almacenamiento de datos común. Aquí, la integración de datos físicos es realizada por la transferencia de datos a un nuevo almacenamiento de datos; fuentes locales o bien pueden ser retiradas o seguir funcionando. En general, la integración de datos física proporciona rápido acceso a los datos. Sin embargo, si se retiran las fuentes de datos locales, las aplicaciones que acceden a ellos, también tienen que ser migradas al nuevo almacenamiento de datos. A su vez, si se actualiza periódicamente la información, este tipo de integración no debe tomarse en cuenta [42].

3.3.3 Enfoque centralizado vs. Distribuido

El primer problema que se ha de abordar en la integración de bases de datos es la localización remota de los repositorios de información.

Generalmente las instituciones que producen datos los almacenan en bases de datos locales que no se encuentran en el mismo lugar, empleándose tecnologías de transmisión de datos para acceder a esta información. En la actualidad existen estándares de comunicación que permiten realizar conexiones a bases de datos remotas de forma inmediata. Sin embargo, la política de acceso a estas bases de datos no es un tema trivial y determina en gran medida las características de un sistema de integración.

Los dos enfoques que se pueden adoptar son: centralizado (Figura 3.4) y distribuido (Figura 3.5).

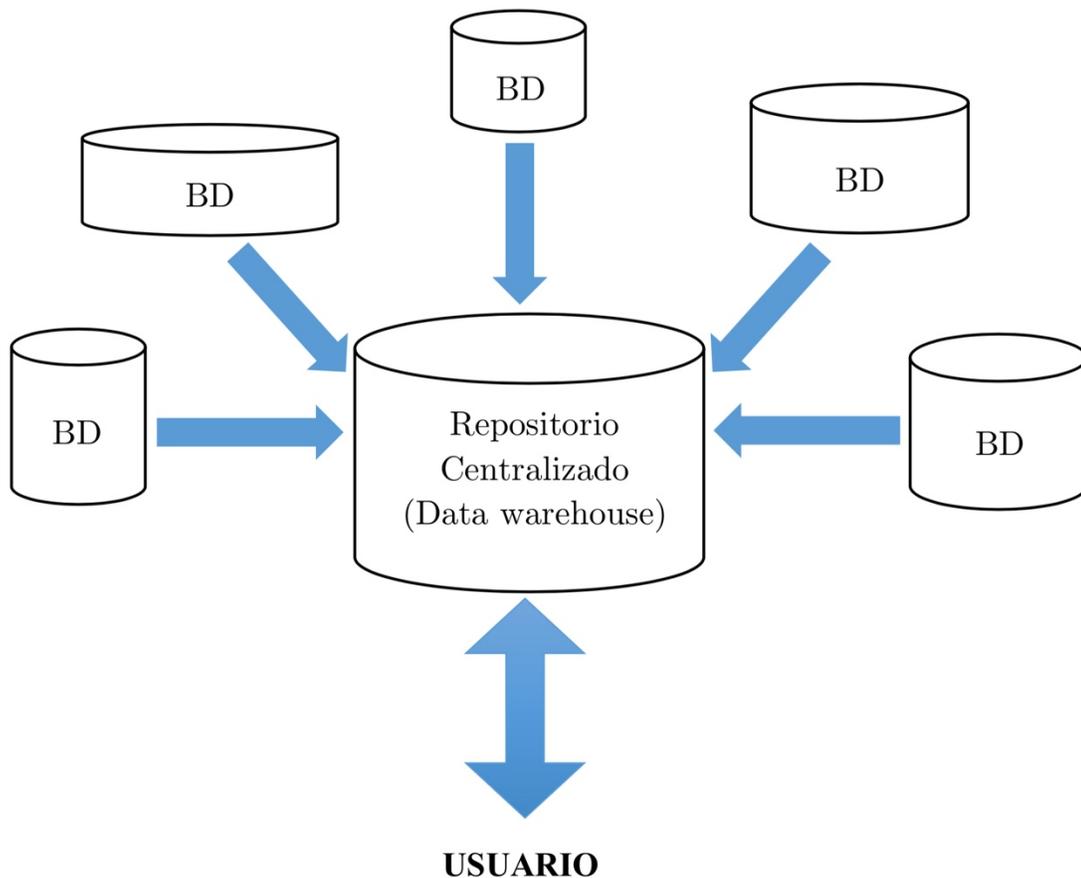


Figura 3.4 Enfoque centralizado

En el modelo centralizado los datos se obtienen desde diversas localizaciones y son almacenados en un sistema central, normalmente llamado almacén de datos ("Data

warehouse”) que muestra capacidades específicas en contraste con las bases de datos relacionales clásicos.

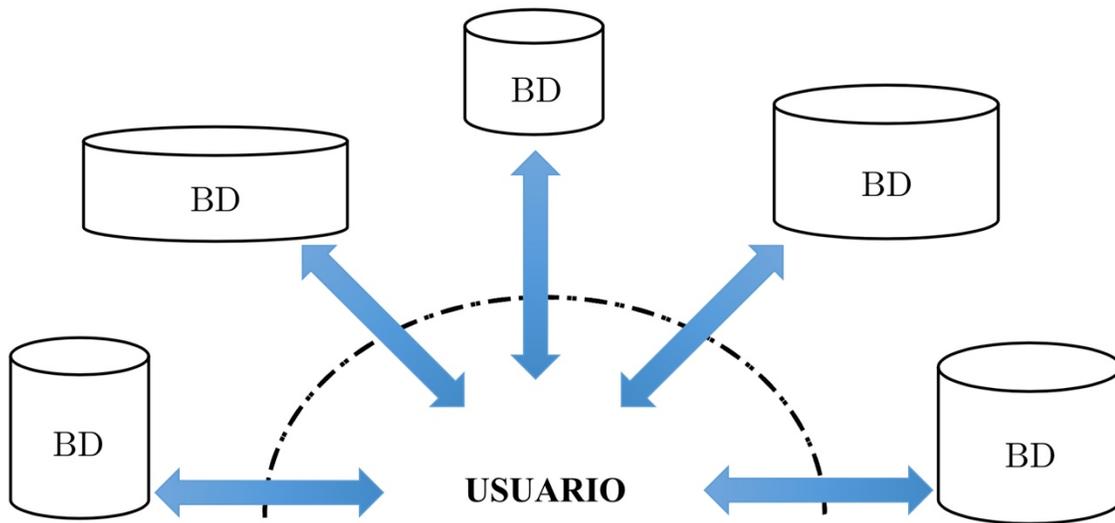


Figura 3.5 Enfoque distribuido

En este segundo enfoque distribuido (Figura 3.5), las bases de datos se mantienen en diferentes puntos remotos bajo control de la organización que las produce. A través de redes de ordenadores se conectan las distintas fuentes entre ellas y con el usuario final y los datos se recuperan cada vez que son requeridos.

Almacenar toda la información en una sola localización tiene ventajas en cuanto a rendimiento. No obstante, teniendo en cuenta la gran cantidad de datos que se manejan actualmente, su actualización periódica puede no ser posible en muchos casos

Aunque las arquitecturas distribuidas presentan varias desventajas tales como coste, disponibilidad, rendimiento, seguridad y fiabilidad, poseen características fundamentales que en ciertos entornos las hacen más apropiadas. Los sistemas que siguen el enfoque distribuido son capaces de aportar los mecanismos necesarios para que organizaciones independientes compartan su información sin perder su autonomía local y evitando una redundancia de datos. Por el contrario, la visión centralizada que aportan los almacenes de datos los hace más apropiados dentro de grandes empresas e instituciones unificadas, en las que habitualmente se opta por una política de gestión de datos que aporte homogeneidad, un acceso rápido a los datos y procesamiento analítico en línea. Esto sin embargo no es posible dentro de organizaciones con nodos al mismo nivel y no jerarquizadas.

Con los puntos mostrados anteriormente la propuesta de este trabajo se centra en un enfoque distribuido, tomando consigo todas las ventajas y desventajas que esto conlleva.

3.4 Ontologías

Las ontologías han tomado gran fuerza debido al gran poder expresivo que poseen; diversas definiciones han surgido, pero el termino Ontología proviene de la filosofía donde una Ontología es la teoría de la naturaleza del ser o de la existencia.

Las ontologías suponen una conceptualización del conocimiento, definiendo formalmente conceptos de diferentes dominios, sus relaciones y atributos. Con capacidad añadida de realizar inferencias sobre estas conceptualizaciones. El termino Ontología [49] es un termino clásico en filosofía. Desde esta perspectiva, “ουτολογία” significa “tratado del ser”, o también, “tratado de todo lo que existe”. Para la Inteligencia Artificial (IA), lo que existe es todo aquello que puede ser representado en un lenguaje comprensible por un sistema computarizado. Así pues, tiene sentido hablar de Ontologías en IA, que puedan ser utilizadas tanto por humanos como por ordenadores. Desde esta perspectiva, varios grupos y autores proporcionan distintas definiciones de lo que es una ontología en informática.

- **Gruber** [45]: 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.
- **Neches** [46]: Define que una ontología establece 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** [47]: 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.
- **Borst** [48]: Coincide con Gruber, pero hace una modificación en la definición y establece a las ontologías como una especificación formal de una conceptualización compartida.

De acuerdo con las definiciones anteriores se dice que las ontologías también pueden incluir la capacidad de generar inferencias lógicas definiendo reglas y axiomas, que además pueden controlar los objetos introducidos en la ontología.

Como se puede observar, las ontologías son más que un simple vocabulario controlado o una taxonomía, ya que han sido concebidas para representar el significado subyacente a un dominio científico. En cualquier caso, estos modelos se han de almacenar en un medio físico y para ello se han implementado distintos lenguajes de representación de ontologías. El desarrollo de estos lenguajes ha sido paralelo al de las ontologías, por lo que en la siguiente sección se describen cronológicamente, junto a sus principales características, para mostrar la evolución de las ontologías hasta la actualidad.

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

$$O = (S, A, K, L)$$

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.

3.4.1 Componentes principales de las ontologías

La comunidad de las ontologías distingue entre ontologías que son principalmente taxonomías, y las ontologías que modelan un dominio de una manera profunda y proporcionan mayores restricciones en la semántica del dominio. A las primeras las llaman ontologías ligeras (lightweight ontologies) y a las segundas ontologías pesadas (heavyweight ontologies). Las ontologías ligeras incluyen conceptos, taxonomías de conceptos, relaciones entre conceptos y propiedades que describen conceptos. Por otra parte, las ontologías pesadas agregan axiomas y restricciones a las ontologías ligeras. Axiomas y restricciones clarifican el significado pretendido de los términos incluidos en la ontología [49].

Las ontologías pesadas y ligeras pueden ser modeladas con diferentes técnicas de modelado de conocimiento y pueden ser implementadas en varios tipos de lenguajes.

A su vez estas pueden ser:

- **Altamente informales:** si son expresadas en lenguaje natural.
- **Semi-informales:** si son expresadas en un lenguaje natural estructurado y restringido.
- **Semi-formales:** si son expresadas en un lenguaje artificial formalmente definido.
- **Rigurosamente formales:** si proporcionan términos definidos meticulosamente con una semántica formal, teoremas y pruebas de propiedades como la completitud.

A principio de los 90's, las ontologías fueron construidas utilizando principalmente técnicas de modelado de la Inteligencia Artificial basadas en lógica de primer orden (first order logic, FOL). En años recientes, se han empleado otras técnicas de representación de conocimiento basadas en lógica de descripción (description logics, DL) en el contexto de la Web Semántica, nuevos lenguajes de etiquetas (markup *languages*) de lógica de descripción para la creación de ontologías como OIL [37], DAM+OIL [24] y OWL [44].

Otras técnicas ampliamente usadas en Ingeniería de Software como UML, (Unified Modeling Language) y Bases de Datos (como el modelo E-R) para modelar conceptos, relaciones entre conceptos y atributos de conceptos pueden también ser apropiadas para construir ontologías ligeras, ya que estas técnicas imponen una estructura al conocimiento del dominio y restringen las interpretaciones de los términos.

No todas las técnicas para modelar ontologías pueden representar el mismo conocimiento con el mismo nivel de formalidad y granularidad. Las ontologías pesadas pueden ser modeladas tomando técnicas basadas en Inteligencia Artificial que combinan estructuras y lógica de primer orden o empleando lógica de descripción.

Otras técnicas basadas en Ingeniería del Software y Bases de Datos sólo permiten la representación de ontologías ligeras. En general, las ontologías construidas con técnicas de Inteligencia Artificial restringen las posibles interpretaciones de los términos mucho más que las ontologías creadas con técnicas de Ingeniería de Software y Bases de Datos.

3.4.2 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 como se interpreta el conocimiento capturado. Cuando mas relaciones y mas restricciones son capturadas en la ontología comienza a ser mas expresiva, con lo cual se describe que tan detallado esta el dominio.

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

- **Ontologías de representación de conocimiento:** (knowledge representation, KR). Capturan las primitivas de representación usadas para formalizar el conocimiento bajo un paradigma de KR. Ejemplos de tales primitivas son: clases, relaciones, atributos, etc. Más adelante en este apartado se revisarán las ontologías de KR: RDF(S) y OWL.
- **Ontologías generales o comunes.** Utilizadas para representar conocimiento de sentido común, reutilizable entre dominios diferentes. Incluyen vocabulario en relación a cosas, eventos, tiempo, espacio, casualidad, comportamiento, funciones, etc.
- **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:** Describe el vocabulario sobre una tarea en específico.
- **Ontologías de dominio-trabajo:** Son ontologías de tarea reutilizables en un dominio dado, pero no entre dominios.
- **Ontologías de aplicación:** Este tipo de ontologías describe conceptos de un dominio y de tareas particulares, comúnmente especializaciones de ambos.

3.4.3 Lenguajes de ontologías

A finales de los 90 aparecieron los primeros lenguajes de representación de ontologías basado en XML (“eXtensible Markup Language”), manteniéndose desde entonces este lenguaje como base del almacenamiento de ontologías. Los lenguajes basados en XML representaron una primera aproximación a la web semántica, y aunque XML no está expresamente pensado para definir ontologías, es el estándar más extendido en la actualidad para aplicaciones de la web. XML permite estructurar datos y documentos en forma de arboles de etiquetas con atributos.

En XML Schema (XMLS) se establecen las estructuras que se van a utilizar, así como los tipos de datos primitivos y derivados. Por su parte, con XSLT se pueden definir plantillas asociadas a las estructuras XML, que pueden describir por ejemplo como generar el correspondiente código HTML para visualizar los contenidos en un navegador web.

En la actualidad, todos los lenguajes de representación de ontologías modernos se basan en RDF y RDF Schema, que a su vez se basan en XML. Por tanto, heredan sus ventajas de comprensión tanto por personas como por computadoras, pero también las dificultades cuando los modelos son extensos—complejidad de gestionar archivos excesivamente extensos. Para manejar estos ficheros se precisan, en un primer paso, librerías que gestionen su estructura desde los distintos lenguajes de programación. Y posteriormente herramientas de alto nivel adecuadas para editarlos de forma amigable para el usuario. Librerías como DOM o Xerces facilitan la gestión de estructuras XML desde código Java o C++, y existen herramientas para facilitar la compatibilidad de XML con bases de datos como JavaBeans [50].

Resource Description Framework “RDF”

RDF son las siglas de Resource Description Framework; una recomendación del W3C (World Wide Web Consortium) desarrollado para describir recursos Web con metadatos. 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.

El modelo de datos RDF consiste de tres componentes:

- **Recursos:** los cuales son cualquier tipo de datos descrito por RDF. Los recursos son descritos con expresiones RDF y referenciados como URIs (Identificadores uniformes de recursos, Uniform Resource Identifiers) más identificadores opcionales.
- **Propiedades:** las cuales definen atributos o relaciones empleados para describir un recurso.
- **Sentencias o tripletas:** 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 [50]. Como se muestra en la figura 3.6. Por ejemplo, en la sentencia “Juan compra un boleto”, Juan es el sujeto, compra es el verbo y boleto es el objeto. Si representamos la sentencia en RDF, Juan y boleto son recursos; sujeto y objeto respectivamente, representados gráficamente por nodos, mientras, compra es una propiedad representada gráficamente por un arco.

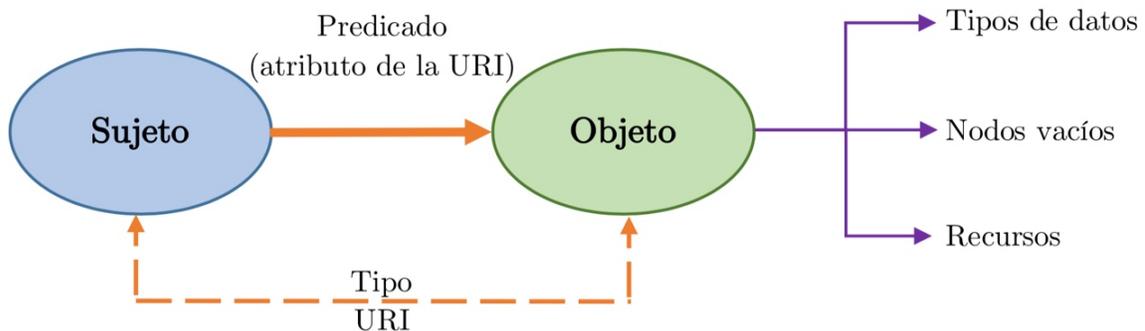


Figura 3.6 RDF Tripletas [49]

No solo los recursos pueden ser objetos en una sentencia RDF, las sentencias RDF pueden ser objetos en sí mismos. Por ejemplo, en la sentencia “Juan dice que Pedro compró un boleto”, Juan es el sujeto, dice es la propiedad y Pedro compró un boleto es el objeto, el cual puede ser descompuesto a su vez, como se describió anteriormente. En general, cada tripleta RDF se representa como un enlace nodo-arco-nodo en donde el arco es un arco dirigido que siempre apunta hacia el objeto, como lo muestra la figura 3.6. Un conjunto de tales tripletas es llamado grafo RDF.

El modelo de datos RDF no hace suposición alguna acerca de la estructura de un documento que contiene información RDF por lo que, en una ontología RDF, las sentencias pueden aparecer en cualquier orden.

Una ontología RDF de KR se codifica en RDFS y contiene las siguientes primitivas de modelado:

Tabla 3.1 Elementos de RDF

SUJETO	PREDICADO	OBJETO	REPRESENTA
rdfs: Resource	rdf: type	rdf: Class	Elementos resources es un tipo de class
rdfs: Class	rdf: type	rdf: Class	Class es un tipo de Class
rdfs: Property	rdf: type	rdf: Class	Property es un tipo de class
rdfs: Type	rdf: type	rdf: Property	Type es un tipo de property
rdfs: Label	rdf: type	rdf: Property	Etiqueta de la propiedad
Rdfs: comment	rdf: type	rdf: Property	Comentario de la propiedad

Las ontologías son muy útiles para representar el conocimiento. La ontología se define como: Una especificación formal y explícita de una conceptualización [44], 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 estas.

CAPÍTULO IV. METODOLOGÍA PROPUESTA



Como en todo esfuerzo encaminado a resolver un problema, en la integración de datos se debe tener en claro cual es el objetivo de la integración, por lo que surgen diversas cuestiones antes de poder realizar de manera inmediata una integración como: poder darle un formato único, la gran cantidad de formatos que existen para guardarla, o concentrarla en algún sitio o forma.

El reconocer claramente el objetivo de la integración suele ser una pieza clave en el desarrollo de una buena aplicación de integración [51].

El objetivo que se persigue en este caso es dar acceso unificado a través de una sola interfaz a conjuntos de datos de estructura relacional, independientes. El acceso unificado implica que a partir de una misma interfaz de usuario (la interfaz de la aplicación), a través de sus campos y elementos, se tendrá acceso a conjuntos de datos heterogéneos, independientes entre sí, pero que a la vez son parte del mismo dominio o “porción de la realidad” que se desea mostrar a través de la propia aplicación. Se pretende que con esta metodología se recuperare 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.

Con el acceso unificado, la petición de información a través de la aplicación involucra la respuesta conjunta de las fuentes de datos que participan de la integración, ya que las preguntas que se formulan lo hacen en atención a la clase de información que se puede obtener del total de los conjuntos de datos, vistos como un todo. El acceso unificado implica además trabajar con las fuentes originales de datos evitando la duplicidad, la pérdida de consistencia y la falta de actualización de la información.

En general, es posible decir que es condición de integración que los datos se encuentren distribuidos en fuentes (almacenes o repositorios) independientes y que se refieran al mismo dominio de aplicación.

En el caso del presente trabajo se propone la integración de fuentes de datos independientes acerca emergencias medicas (el dominio de la aplicación) para acceder a un conjunto más amplio hospitales, doctores, clínicas, etc. descritos bajo un número fijo de atributos o características comunes como se verá en el siguiente capítulo, concretamente en el caso de estudio.

Retomando el esquema propuesto de una arquitectura ontológica mostrada en la Figura 4.1 se describe a continuación la estructura de las ontologías que lo integran, para ello se propone una metodología que se desarrolla a detalle en el punto siguiente.

Como se observa en la arquitectura (Figura 4.1), se tienen entradas las cuales se vinculan a las ontologías que representan las fuentes de información (Bases de datos), pasando primeramente por un proceso de mapeo, posteriormente se tiene una etapa de alineación semántica, la cual esta representada por una ontología central, para finalmente tener como resultado una ontología de representación especificando el conocimiento de un dominio determinado.

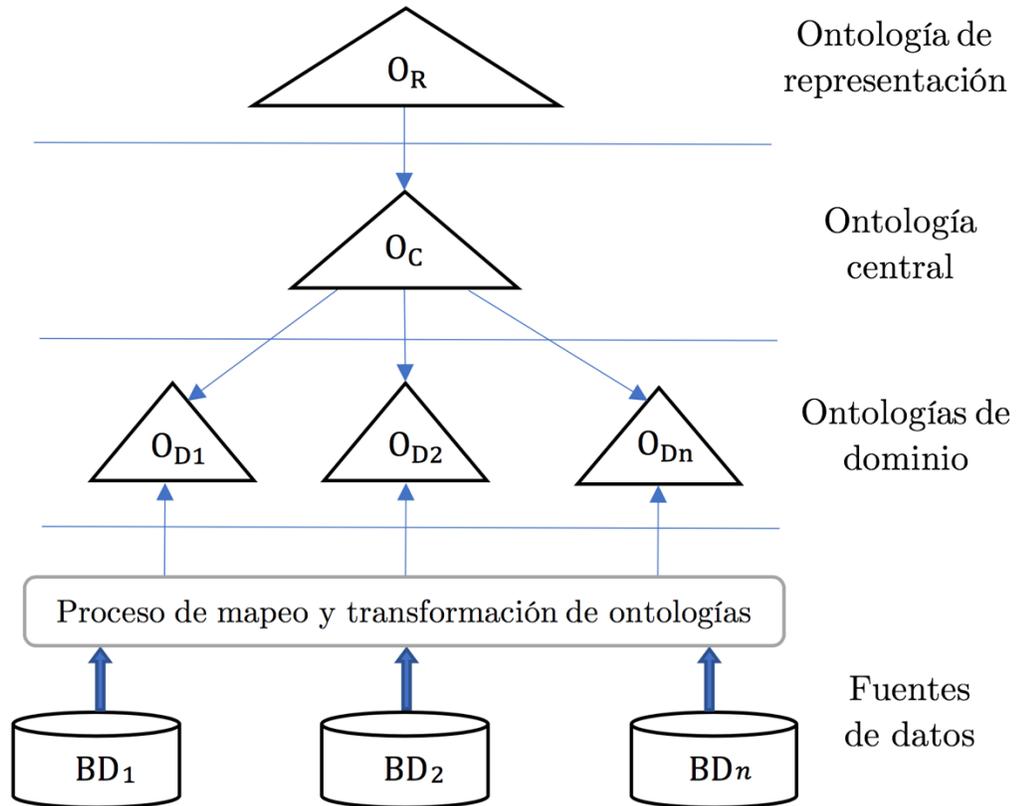


Figura 4.1 Arquitectura ontológica

Para conseguir este objetivo planteado en la descripción de la arquitectura, se propone 5 etapas principales las cuales conforman una metodología presentada a continuación.

- Etapa de conceptualización y dominio de aplicación
- Creación de la ontología
- Mapeo de instancias
- Etapa de recuperación y despliegue

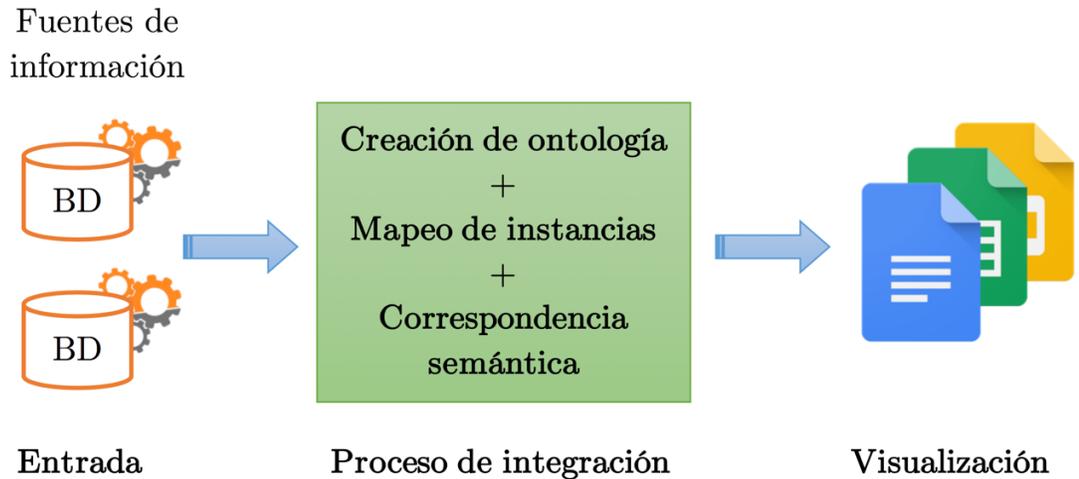


Figura 4.2. Etapas generales del modelo propuesto

4.1 Etapas de la metodología

A continuación, se explica en detalle los pasos de la metodología propuesta, así como el diseño respectivo de cada etapa.

4.1.1 Etapa de conceptualización y dominio de aplicación

La primera fase de esta metodología consiste en determinar las entradas al sistema, es decir que fuentes de información serán mapeadas en ontologías de dominio. Cabe aclarar que estas fuentes deben de tener algún dominio concreto que compartan es decir una relación en cuanto a información un 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, esta creación se realiza en una etapa posterior, sin embargo, las diferencias entre las distintas fuentes se resuelven mediante la construcción de ontologías que actúan como repositorios virtuales homogéneos.

En resumen, esta etapa se ejemplifica seleccionando las bases de datos relacionales a integrar posteriormente, 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 mas adelante en este apartado.

4.1.2 Creación de la ontología

El hecho de tener un modelo conceptual ideal del dominio de la aplicación se ajusta perfectamente al uso de ontologías como parte de la solución propuesta para la integración de datos. La ontología permite representar o modelar el dominio de la aplicación a través de sus conceptos y las relaciones entre los mismos. El proceso de creación de la ontología en este trabajo es un proceso manual. Se crea a partir del conocimiento que se tiene del dominio que se modela, del conocimiento intuitivo o general de la estructura de las fuentes a integrar y de los requerimientos de información de la propia aplicación. En resumen, la ontología se crea convenientemente a partir del modelo conceptual ideal del dominio de la aplicación.

De esta forma con la ontología se tendrá la “visión” completa del dominio de la aplicación y todo lo que se necesite saber de él deberá estar representado en la misma.

Por medio de una serie de reglas de transformación, se pretende representar los elementos del modelo relacional con elementos de las ontologías:

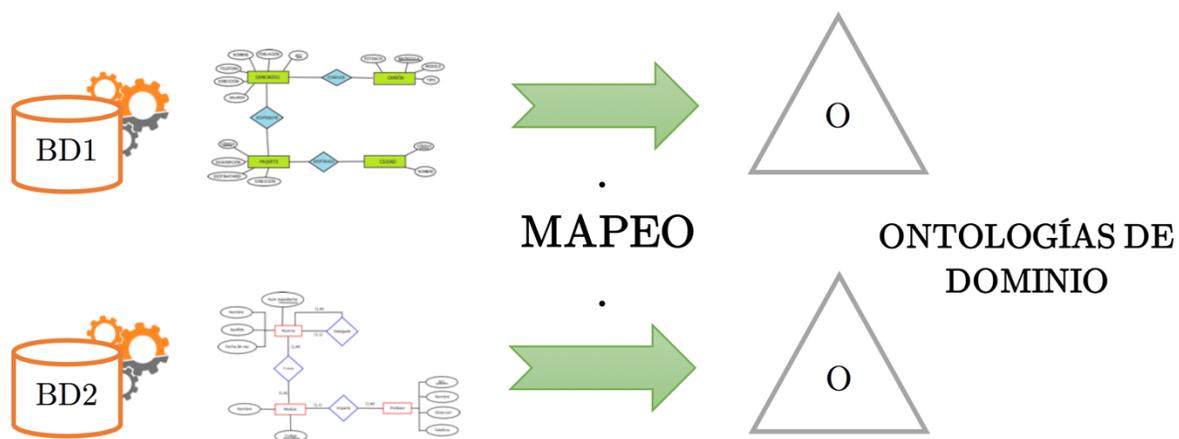


Figura 4.3 Proceso de transformación de las fuentes de información

Para representar las antologías 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.

Para la creación de cada ontología, se utilizará dentro de este trabajo una metodología desarrollada por la Universidad Politécnica de Madrid denominada **Methontology**, esta metodología se basa en once actividades mostradas en la siguiente imagen (Figura 4.4):

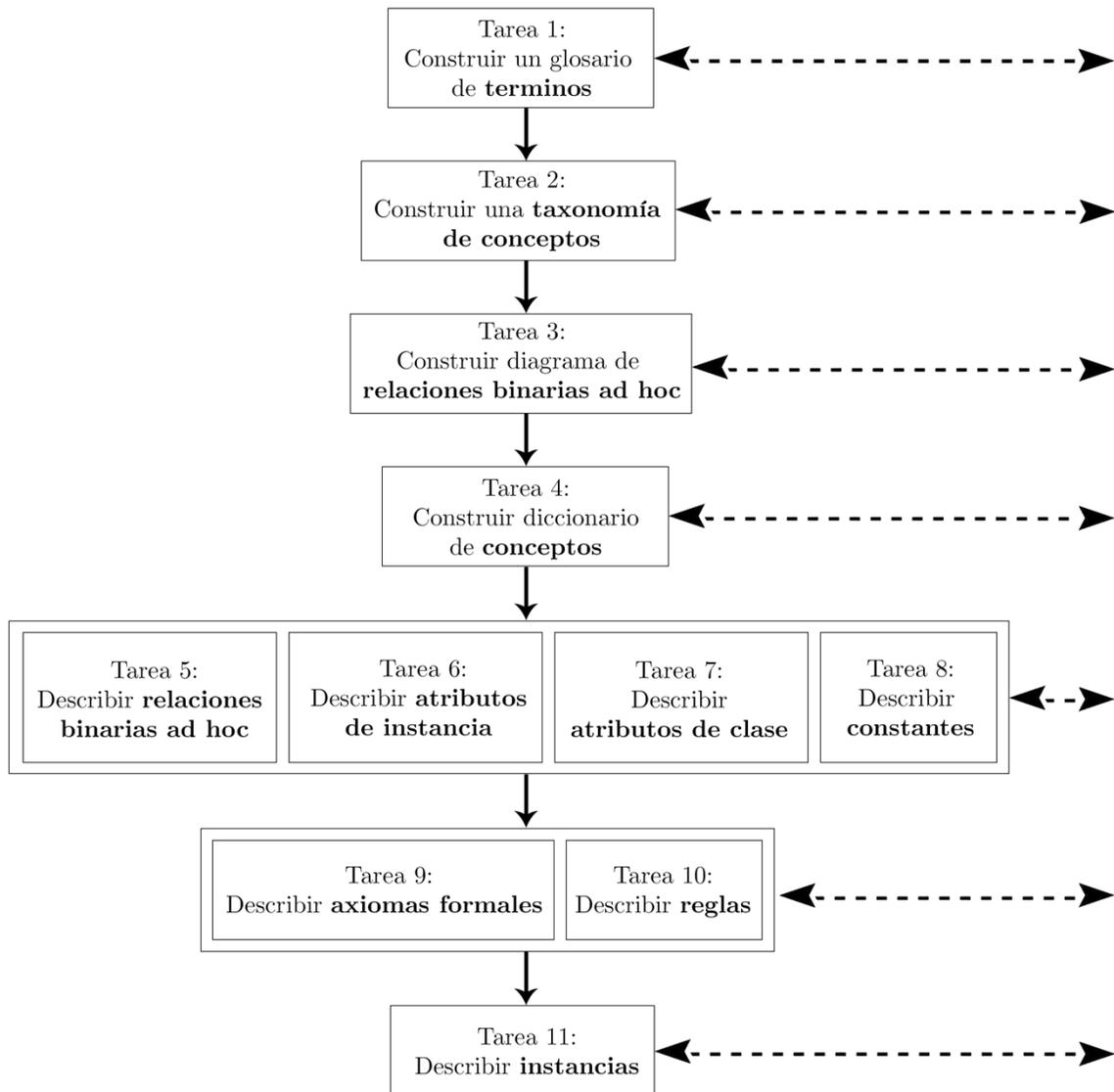


Figura 4.4 Methontology

- Tarea 1: Construir glosario de términos** En primer lugar, se construye un glosario de términos que incluye todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, y sus sinónimos y acrónimos. Es importante mencionar que en las fases iniciales de conceptualización el glosario de términos puede llegar a

contener varios términos que se refieran al mismo componente. En tal caso, el responsable de la ontología debe detectar que se trata de sinónimos.

- **Tarea 2: Construir una taxonomía de conceptos** Una vez que el glosario de términos contiene suficientes términos, el desarrollador de la ontología construye las taxonomías de conceptos que definen su jerarquía.
Para construir taxonomías de conceptos, se seleccionan del glosario de términos aquellos términos que son conceptos.
Una vez que el desarrollador de la ontología ha estructurado los conceptos en la taxonomía de conceptos, y antes de proseguir con la especificación de nuevos conocimientos, éste debe examinar que las taxonomías no contienen errores. Por ejemplo, debe verificar que un elemento no es simultáneamente instancia de dos clases que forman parte de una descomposición disjunta, que no hay ciclos en la taxonomía de conceptos, que no existen varios términos que se refieren al mismo concepto, etc.
- **Tarea 3: Construir diagramas de relaciones binarias ad hoc.** Una vez construida y evaluada la taxonomía, la actividad de conceptualización propone construir diagramas de relaciones binarias ad hoc. El objetivo de este diagrama es establecer las relaciones ad hoc existentes entre conceptos de esta o de distintas taxonomías de conceptos. Del mismo modo que en la tarea anterior, el desarrollador de ontologías debe verificar que los diagramas construidos no contienen errores. El desarrollador de ontologías debe verificar que el origen y destino de todas las relaciones delimitan con exactitud y precisión los conceptos apropiados. En este punto, los errores suelen aparecer cuando el origen o el destino de una relación es impreciso o está sobre- especificado.
- **Tarea 4: Construir el diccionario de conceptos.** Una vez que las taxonomías de conceptos y los diagramas de relaciones binarias ad hoc han sido generados, el desarrollador de la ontología debe especificar cuáles son las propiedades que describen cada concepto de la taxonomía, así como las relaciones identificadas en el diagrama anterior y las instancias de cada uno de los conceptos. El diccionario de conceptos contiene todos los conceptos del dominio, sus relaciones, sus instancias, y sus atributos de clase y de instancia. Las relaciones especificadas para cada concepto son aquellas en las que el concepto es el origen de esta. Las relaciones y los atributos de instancia y de clase son locales a los conceptos, lo que significa que sus nombres pueden ser repetidos en diversos conceptos.
- **Tarea 5: Describir las relaciones binarias ad hoc.** El objetivo de esta tarea es describir en detalle todas las relaciones binarias ad hoc identificadas en el diagrama de relaciones binarias e incluidas en el diccionario de conceptos. Para

cada relación binaria ad hoc, el desarrollador de la ontología debe especificar su nombre, los nombres de sus conceptos origen y destino, su cardinalidad y su relación inversa, si existe.

- **Tarea 6: Describir los atributos de instancia.** El objetivo de esta tarea es describir en detalle todos los atributos de instancia incluidos en el diccionario de conceptos. Cada fila de la tabla de atributos de instancia contiene la descripción detallada de un atributo de instancia. Como ya se comentó anteriormente, los atributos de instancia describen a las instancias del concepto y sus valores pueden ser distintos para cada una de dichas instancias. Por cada atributo de instancia, el desarrollador de la ontología debe especificar su nombre, el concepto al que el atributo pertenece (los atributos son locales a los conceptos), su tipo de valor, su rango de valores (en el caso de atributos numéricos), y sus cardinalidades mínima y máxima.
- **Tarea 7: Describir los atributos de clase.** El objetivo de esta tarea es describir en detalle todos los atributos de clase incluidos en el diccionario de conceptos. Para cada atributo de clase, el desarrollador de la ontología debe rellenar la siguiente información: nombre del atributo, nombre del concepto donde el atributo se define, tipo de valor, cardinalidad y valor(es).
- **Tarea 8: Describir las constantes.** El objetivo de esta tarea es describir en detalle cada una de las constantes identificadas en el glosario de términos. Para cada constante, el desarrollador de la ontología debe especificar su nombre, tipo de valor, valor y unidad de medida en el caso de constantes numéricas
- **Tarea 9: Definir axiomas formales.** Para realizar esta tarea, el desarrollador de ontologías debe identificar los axiomas formales que son necesarios en la ontología y describirlos de manera precisa. Para cada definición de axioma formal, METHONTOLOGY propone especificar la siguiente información: nombre, descripción en lenguaje natural, expresión lógica que define de manera formal el axioma usando lógica de primer orden, y los conceptos, atributos y relaciones ad hoc utilizadas en el axioma, así como las variables utilizadas.
- **Tarea 10: Definir reglas.** De manera similar a la tarea previa, en esta tarea el desarrollador de la ontología debe identificar en primer lugar qué reglas se necesitan en la ontología, y entonces describirlas en la tabla de reglas. Para cada regla, METHONTOLOGY propone incluir la siguiente información: nombre, descripción en lenguaje natural, expresión que describe formalmente la regla, y conceptos, atributos y relaciones ad hoc utilizados en la regla, así como las variables usadas.

METHONTOLOGY propone especificar las expresiones de las reglas utilizando el formato si <condiciones> entonces <consecuente>. La parte izquierda de la regla es una conjunción de condiciones simples, mientras que la parte derecha es una simple expresión de un valor de la ontología.

- **Tarea 11: Describir instancias.** Una vez que el modelo conceptual de la ontología ha sido creado, se pueden definir las instancias que aparecen en el diccionario de conceptos. Para cada instancia se define: su nombre, el nombre del concepto al que pertenece y los valores de sus atributos de instancia, si se conocen.

4.1.3 Mapeo de instancias

Una vez que el modelo ideal del dominio de la aplicación y el conjunto de atributos que lo representan se han definido explícitamente a partir de la ontología, el siguiente paso en la metodología es hacer explícito también el reconocimiento de los atributos del modelo conceptual de cada base de datos en los atributos del modelo ideal. El establecimiento explícito de estas correspondencias permitirá, como se verá más adelante, crear un puente entre los datos de las fuentes originales de información y las clases de la ontología por el cual los datos de todas las fuentes (los valores de los atributos de sus entidades y las entidades mismas) serán reconocidos como instancias de las clases en la ontología.

Esta etapa conocida como mapeo de instancias es una etapa basada por completo en la plataforma SPARQL y permite el acceso unificado a las fuentes de datos originales por parte del usuario a través de consultas o solicitudes de información independientes de los modelos conceptuales específicos de cada base de datos sino solo en atención al modelo ideal reflejado en la ontología.

El proceso se lleva a cabo de forma manual, si bien la dificultad de este proceso radica cuando se considera que el diseño de las fuentes de información incluye el manejo de llaves primarias, relaciones entre entidades y atributos que se organizan bajo cierta estructura para mantener la integración de estos.

A partir del archivo de mapeo, la plataforma D2RQ consigue transformar una base de datos relacional en un grafo RDF virtual de solo lectura al cual se puede acceder vía consultas SPARQL las cuales a su vez son traducidas en consultas SQL hacia las bases de datos originales.

Una ontología es mapeada al esquema de una base de datos con el uso de dos estructuras básicas del lenguaje SPAQRQL: `d2rq:ClassMap` y

d2rq:PropertyBridge. **d2rq:ClassMap** representa una clase o grupo de clases similares de la ontología, especifica cómo las instancias de la clase son identificadas y tiene un conjunto de **d2rq:PropertyBridges** que especifican como se crean las propiedades de una instancia.

La estructura de D2RQ que se utiliza en el archivo de mapeo que define las correspondencias ente la ontología creada y los esquemas de las fuentes de información se detalla a continuación:

d2rq:Database

La estructura **d2rq:Database** define una conexión JDBC u ODBC a una base de datos relacional local y especifica el tipo de las columnas de las bases de datos utilizadas por D2RQ. Un archivo de mapeo D2RQ puede tener varias **dr2q:Databases** para acceder diferentes bases de datos locales [28]

```
map: BDHospital a d2rq:Database;
d2rq:jdbcDSN "jdbc:mysql://localhost/BDHospital ";
d2rq:jdbcDriver "com.mysql.jdbc.Driver";
d2rq:username "root";
d2rq:password "n0m3l0s3";
jdbc:autoReconnect "true";
jdbc:zeroDateTimeBehavior "convertToNull";
```

La propiedad **d2rq:jdbcDSN** indica el URL JDBC de la base de datos. La propiedad **d2rq:jdbcDriver** especifica el nombre de la clase de driver JDBC para la base de datos. Las propiedades **dr2q:username** y **d2rq:password** son el nombre de usuario y password respectivamente requeridos para el acceso a la base de datos. Dos propiedades: **jdbc:autoReconnect** y **jdbc:zeroDateTimeBehavior** son propias del driver de conexión JDBC para MySQL [28].

d2rq:ClassMap

La estructura **d2rq:ClassMap** representa una clase o grupo de clases similares de una ontología OWL o un esquema RDFS. Un **d2rq:ClassMap** define como son identificadas las instancias de la clase. Se conecta a un **d2rq:Database** y tiene un conjunto de **d2rq:PropertyBridges** que asignan propiedades a las instancias.

```
map: Mapeo1 a d2rq:ClassMap;
d2rq:dataStorage map: BDHospital;
d2rq:class onto:Onto_Medicos;
d2rq:uriPattern " Mapeo1 /@@TablaUno.id@@";
```

d2rq:PropertyBridge

La estructura `d2rq:PropertyBridge` relaciona columnas de las tablas de las bases de datos con propiedades RDF. Se usa para asociar propiedades a los recursos RDF generados por el `d2rq:ClassMap`. Los valores de estas propiedades son a menudo literales, pero pueden ser URIs o nodos vacíos que relacionen el recurso a otros recursos.

```
map:BDHospital__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:BDHospital;
  d2rq:property rdfs:label;
  d2rq:pattern "BDHospital #@hospital.id@";
```

```
map:BDHospital_nombre a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:HospitalA;
  d2rq:property onto:Nombre;
  d2rq:column "Hospital.nombre_Hospital ";
  d2rq:datatype xsd:string;
```

```
map:BDHospital_direccion a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:BDHospital;
  d2rq:property onto:Ubicacion;
  d2rq:column "hospital.direccion";
  d2rq:datatype xsd:string;
```

```
map:BDHospital_especialidad a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:BDHospital;
  d2rq:property onto:Especialidad;
  d2rq:column "hopsital.especialidad";
  d2rq:datatype xsd:string;
```

La propiedad `d2rq:belongsToClassMap` es especificada por cada `d2rq:PropertyBridge`; indica que el `d2rq:PropertyBridge` pertenece a un `d2rq:ClassMap`. La propiedad `d2rq:property` es también especificada por cada `d2rq:PropertyBridge`, indica la propiedad RDF que conecta el `d2rq:ClassMap` con el objeto o literal creada por el enlace `d2rq:PropertyBridge`.

La propiedad `d2rq:column` se especifica para propiedades con valores literales, indica la columna de la base de datos que contiene los valores literales. Los nombres de columna se dan de la forma Tabla.Columna. La propiedad `d2rq:datatype` se especifica para propiedades con valores literales; indica el tipo de datos RDF de las literales.

En la figura 4.5 se muestra un ejemplo de un grafo virtual que se crea a partir del archivo de mapeo con el motor de la plataforma D2RQ. Cada arco de un Grafo RDF

representa una sentencia o estatuto en el Modelo RDF. A su vez cada sentencia en un Modelo RDF establece un hecho acerca de un recurso. Como se mostró en capítulos anteriores de este trabajo, las sentencias RDF constan de sujeto (recurso del que salen los arcos), predicado (propiedad que etiqueta el arco) y objeto (recurso o literal apuntada por el arco). Un Modelo RDF es representado por un conjunto de estatutos.

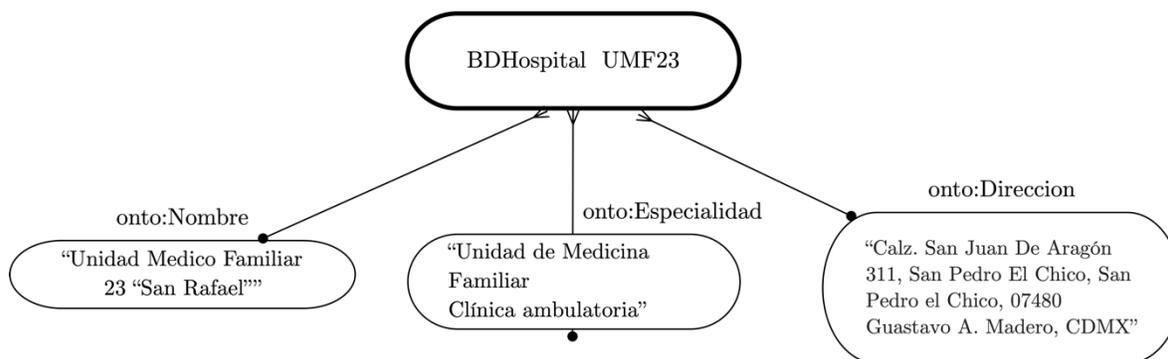


Figura 4.5 Grafo virtual D2RQ

A su vez el grafo anterior (Figura 4.5) se describe como los estatus generados a partir del mapeo, mostrado de la siguiente manera:

```
BDHospital UMF23 onto:Nombre "Unidad Medico Familiar 23 "San
Rafael"
BDHospital # UMF23 onto:Ubicacion "Calz. San Juan De Aragón
311, San Pedro El Chico, San Pedro el Chico, 07480 Guastavo A.
Madero, CDMX"
escuelaA #020 onto:Especialidad "Unidad Medicina Familiar,
Consulta Ambulatoria"
```

4.1.4 Etapa de recuperación y despliegue

En la etapa anterior se obtiene un archivo de mapeo el cual nos permite relacionar las ontologías, se crea la relación entre conceptos para poder recuperar información de manera unificada

De acuerdo con la solución propuesta, la ontología creada actúa como una capa intermedia entre el usuario y las fuentes de datos originales. De esta forma la consulta

original enviada por el usuario se realiza sólo en los términos empleados en las clases de la ontología y sólo obtendrá respuesta de aquellas clases donde existan instancias, y a su vez, las clases en las que existen instancias serán solo aquellas en las que se haya definido la correspondencia con los esquemas de las bases de datos originales por medio de la plataforma D2RQ.

Cuando se mapean los datos a la plataforma D2RQ es que la consultas sobre fuentes de datos cuya estructura de almacenamiento es en forma de grafos RDF. Con la plataforma D2RQ, a través del modelo creado, las consultas SPARQL son reescritas en tiempo de ejecución (en consultas SQL hacia las bases de datos originales, lo que evita la necesidad de replicar los datos en un almacén RDF dedicado.

Las Consultas a través de SPQRL tienen la siguiente estructura a partir del archivo de mapeo definido con anterioridad:

```
SELECT ?nombre, ?ubicacion, ?especialidad
WHERE (
  ?hospital onto:Nombre ?nombre
  ?hospital onto:Direccion ?ubicacion
  ?hospital onto:Especialidad ?especialidad
)
```

A su vez esta consulta se convierte en tiempo de ejecución a una consulta original en la siguiente sentencia SQL la cual se ejecuta sobre el d2rq:Databases y el resultado se transforma en un conjunto de tripletas RDF:

Hasta este punto de la metodología se han trabajado con datos tipo string bajo una estructura relacional en todos los atributos del dominio. Ahora bien, el ultimo punto es representar en un mapa el resultado de la consulta SPARQL anterior.

La visualización de los datos dependerá de las necesidades del usuario, ya que estos pueden ser visualizados de diversas formas, tales como grafos, cuadros, 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.

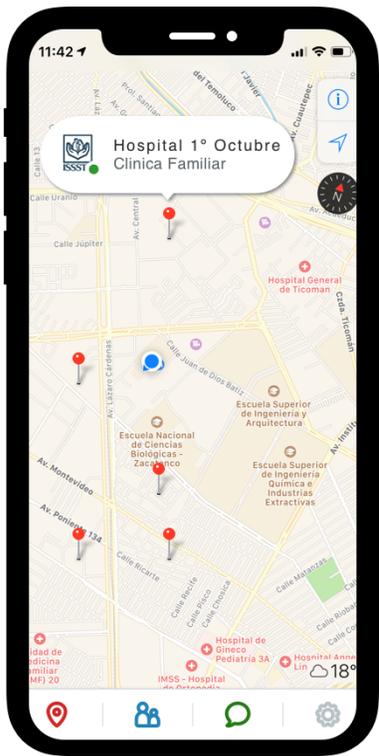
Los datos en esta forma no se pueden llevar a un mapa a menos que de forma directa o indirecta contengan la información espacial que requiere su representación.

En la actualidad existen diversos servidores de mapas para aplicaciones móviles que a través de un conjunto de funciones de una API de programación facilitan la representación de objetos (puntos, polígonos o líneas) superpuestos en un mapa insertado como un objeto dentro de la aplicación móvil

Apple MapKit es la herramienta que se utiliza como proveedor del mapa y, a través de su interfaz de programación es como se consigue representar en forma de puntos o marcadores superpuestos sobre el mapa a los hospitales.

Los datos que debe arrojar el sistema por cada búsqueda son (máximo 5 resultados):

- Nombre
- Ubicación geográfica



```
{  
  "vectorHosp":  
  [  
    ["Hosp 1º Octubre", "19.4931079", "-99.1589498"],  
    ["Hosp Ticomán", "19.5093632", "-99.1463423"],  
    ["Hosp Guadalupe", "19.493188", "-99.1498027"],  
    ["Hosp A.Lindavista", "19.4977614", "-99.1560475"],  
    ["Hosp La Villa ", "19.4923256", "-99.1614389,"]  
  ]  
}
```

Esta información es desplegada en cada punto dentro del mashup de Apple Maps a partir de las coordenadas geográficas recuperadas, como se muestra en la figura 4.6

Figura 4.6 Despliegue

CAPÍTULO V. CASO DE ESTUDIO



5.1 Metodología caso de estudio

El presente trabajo propone el desarrollo de una aplicación para dispositivos móviles que con base en un perfil y haciendo uso de los servicios de ubicación proporcione al usuario recomendaciones acerca de los centros médicos más cercanos a su posición física actual.

Este caso de estudio se fundamenta en la utilización del modelo presentado en capítulos anteriores con base en una ontología que describe los servicios de salud integrados por hospitales y las especialidades que atiende cada uno, así como médicos y personal capacitado que se encuentre cerca de una alerta de emergencia.

El marco de este caso de estudio que sigue la metodología está compuesto de tres etapas:

- Personalización, donde se crea el perfil de usuario y se define el tipo de búsqueda.
- Procesamiento de la información, aquí se realiza la recuperación de información de la ontología tomando en cuenta los parámetros de búsqueda definidos en la etapa anterior
- Visualización de resultados, donde son interpretados los resultados para posteriormente ubicarlos en un mapa.

Para entrar en contexto con el caso de estudio planteado, se propone como ejemplo la siguiente situación: Supongamos que una persona que se encuentra dentro de la Ciudad de México y de manera desafortunada sufre un accidente o bien requiere de algún servicio médico de emergencia, a esta suposición agreguemos el echo de que esta persona desconoce el lugar y por consecuencia los hospitales cercanos a el. Como causa de esta falta de conocimiento del lugar, la emergencia medica puede agudizarse debido a la retrasada atención en los primeros minutos.

Debido a esto, en este trabajo se propone un sistema de recomendación con base en los servicios médicos cercanos al usuario. Esta aplicación móvil permite el uso de un método semántico para sugerir sitios de interés en un contexto exclusivamente medico. Esta aplicación considera aspectos semánticos (como el tipo de emergencia), aspectos geográficos, así como un perfil que permite la personalización de los sitios mostrados, de tal modo que se puedan obtener resultados mucho mas precisos.

En contraste con los sistemas actuales los cuales operan bajo un esquema global en donde las búsquedas son a nivel de coincidencia exacta, bajo un esquema totalmente léxico, lo que impide procesar semánticamente y por consecuencia no se mide conceptualmente la consulta realizada por parte del usuario.

Para tal cometido, se realiza un procesamiento semántico sobre datos definidos previamente en una ontología de aplicación en el dominio medico, con el propósito de obtener la información de manera semántica.

A continuación, la figura 5.1 muestra el diagrama de la metodología propuesta:

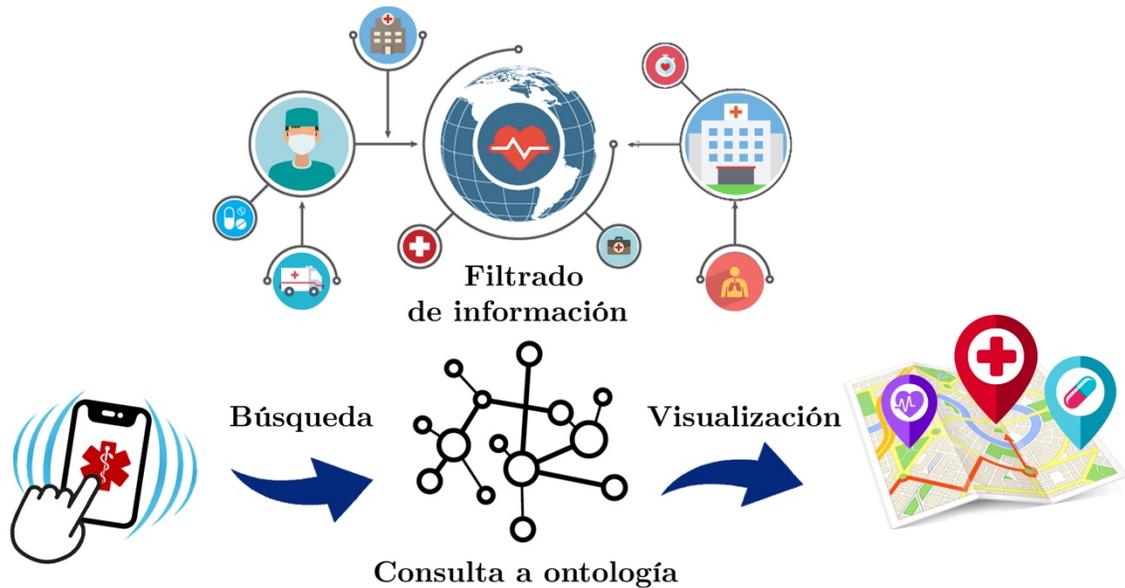


Figura. 5.1 Metodología caso de estudio

Como ya se mencionó al principio de este capítulo esta metodología se divide en 3 bloques detallados a continuación en la siguiente sección.

5.1.1 Caracterización del perfil de usuario

El primer bloque a este sistema tiene como entrada la petición por parte del usuario cuando este requiere un servicio medico cercano a este, esta petición se caracteriza por un perfil propio de cada usuario. Los datos recopilados en la primera etapa son agrupados para generar un vector de características. En la figura 8 se detalla el primer bloque de la metodología propuesta.

Los datos requeridos para la personalización son los siguientes:

- **Coordenadas geográficas:** Posición actual del usuario.
- **Tipo de seguro (Si es que cuenta):** IMSS, ISSTE, Programa oportunidades, PEMEX.
- **Tipo de emergencia:** Dependiendo de la emergencia se selecciona la especialidad correspondiente de cada servicio médico.



Figura 5.2 Primer bloque

Con las características obtenidas a partir de la petición solicitada, se procede a la generación de un vector:

$$\mathbf{V} = (\text{Posición}(\text{latitud, longitud}), (\text{con seguro/sin seguro}), (\text{emergencia}))$$

Este vector en todos los casos va a tener la misma estructura, sin embargo no en todas ocasiones se requerirán las mismas variables por lo que el valor de las mismas cambiara dependiendo de la condición y los datos proporcionados por el usuario.

El siguiente bloque recibe al vector de caracterización, el cual es un servicio que se implementa en Java, el cual se encarga de procesar la información recabada.

5.1.2 Análisis y procesamiento de la información

En este segundo bloque se tiene como entrada al vector de caracterización, el cual representa la salida del bloque anterior. Se realiza un análisis dentro de este bloque para así realizar una consulta a la ontología definida previamente, esta consulta se realiza a través del lenguaje SPARQL. En la figura 5.3 se muestra el proceso descrito.

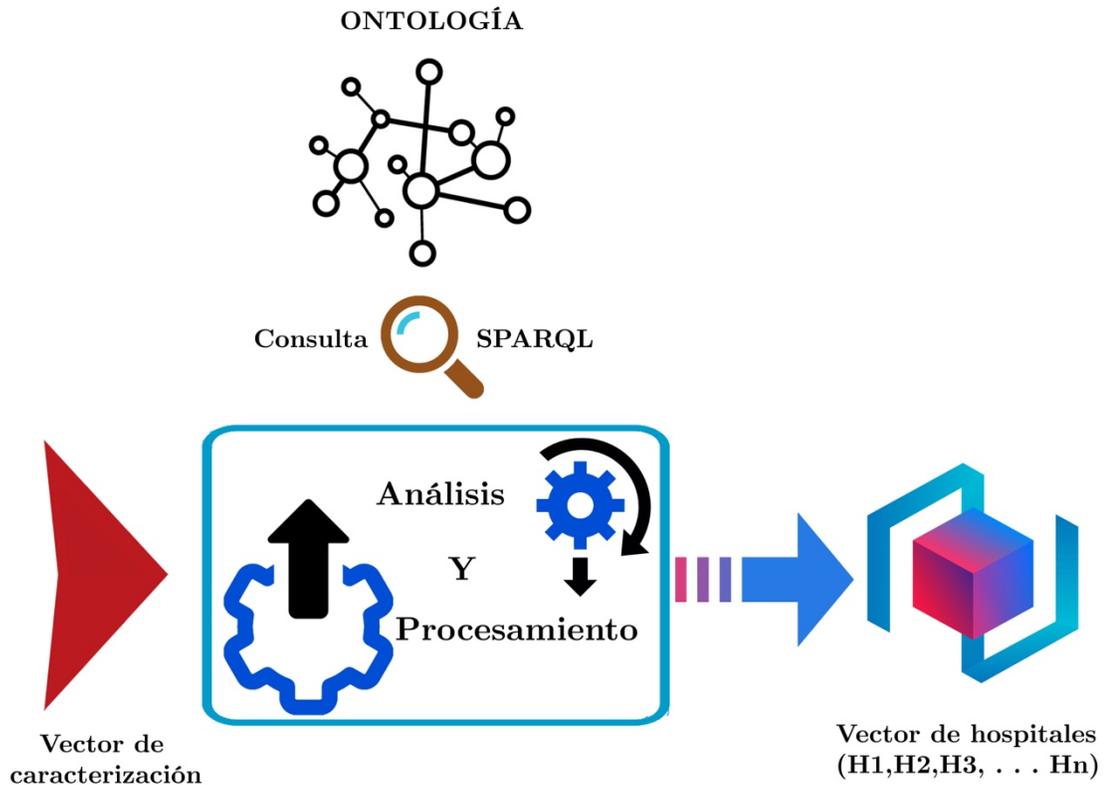


Figura 5.3 Segundo bloque

Para realizar el análisis de la información perteneciente al usuario se lleva a cabo los siguientes procesos:

- **Localización:** Se implementa un buffer para determinar de manera general los hospitales mas cercanos a la ubicación del usuario.
- **Categorización:** Esta categorización depende de la petición inicial, la cual se divide en Hospital general y Urgencias.
- **Recuperación de hospitales disponibles:** Dentro de este proceso se recuperan todas las instancias existentes en la ontología determinada.

- **Búsqueda específica:** Para este apartado se realiza una búsqueda en la ontología, la cual devuelve la información específica, los detalles y la posición geográfica de estos, tomando como punto de partida el tipo de seguro con que cuenta el usuario.
- **Resultado:** Como respuesta al proceso anterior, los resultados se muestran como un conjunto de hospitales que se recomiendan al usuario.

Una vez realizado el análisis de los sitios recomendados para el usuario, se devuelve como respuesta un vector el cual contiene las coordenadas de los hospitales, las cuales se utilizan para agregar marcadores al mapa dentro del dispositivo móvil.

5.1.3 Visualización de resultados

El ultimo bloque para este caso de estudio se encarga de recibir al vector con los hospitales correspondientes, a su vez si el usuario selecciona un hospital en especifico muestra la ruta optima para llegar a este. Para ello se utiliza el framework de desarrollo que proporciona Apple MapKit.

Los resultados son mostrados y desplegados de acuerdo con la distancia que se haya realizado la petición.

La visualización dentro del mapa se observa como distintos marcadores en el, en el que cada uno de ellos representa a un hospital. El usuario es representado por un circulo azul y los marcadores que aparecen en el mapa, se localizan en un radio de 12 km.

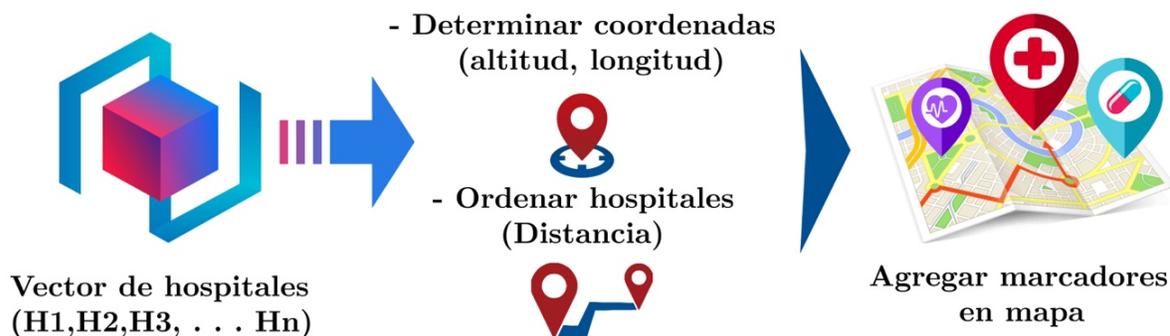


Figura 5.4 Bloque 3

5.2 Desarrollo caso de estudio

Para la creación de esta red de ontologías se utiliza la metodología METHONTOLOGY, descrita en el capítulo IV.

5.2.1 Tarea 1: Glosario de términos

Como se mostró en el capítulo IV se debe construir un glosario de términos de vital relevancia para la ontología. Debido a que el caso de estudio tiene como campo semántico los servicios médicos, con lo cual se presenta a continuación la tarea 1:

Tabla 5.1 Glosario de términos para el caso de estudio

NOMBRE	SINÓNIMOS	ACRÓNIMOS	DESCRIPCIÓN	TIPO
Institución de salud	Servicio medico	---	Lugar destinado a la atención sanitaria de la población, aquellas prestaciones que brindan asistencia sanitaria.	Concepto
Secretaria de salud	---	SSA	Institución encargada de diseñar, ejecutar y coordinar las políticas públicas en materia de servicios sanitarios. Lo anterior incluye elaborar los programas, planes y proyectos de asistencia social, servicios médicos y salubridad general a través del Sistema Nacional de Salud y en conformidad con lo establecido por la Ley General de Salud.	Concepto
Institución de salud pública	---	---	Comprende a todas las instituciones de salud pertenecientes al sector público en sus diferentes niveles: federal, estatal o municipal, que ofrecen servicios de salud a la población derechohabiente y no derechohabiente.	Concepto
Institución de salud privada	---	---	Esta clase comprende a todas las instituciones de salud privadas que ofrecen servicios de salud a la población no derechohabiente o abierta	Concepto

Consulta externa	---	---	Comprende a todas las descripciones genéricas de unidades médicas pertenecientes a los gobiernos estatales, que no es posible ubicarlas en los tipos de unidades médicas anteriores.	Concepto
Instituto Mexicano Del Seguro Social	---	IMSS	Institución de salud y/o seguridad social perteneciente al sector público en sus diferentes niveles: federal, estatal o municipal, que ofrece servicio de salud a la población derechohabiente, como resultado de una prestación laboral al trabajador, por ser pensionado o jubilado, o ser un familiar beneficiario.	Concepto
Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado	---	ISSSTE	Comprende a todas las unidades de medicina familiar pertenecientes al Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE) a través de los cuales se brindan servicios de salud en el primer nivel de atención.	Concepto
Institución de Salud de Petróleos Mexicanos	---	PEMEX	Esta subclase comprende a todas las clínicas externa, hospitales y centros médicos pertenecientes a Petróleos Mexicanos (PEMEX).	Concepto
Institución de Salud de la Secretaría de la Defensa Nacional	---	SEDENA	Comprende a todas las clínicas, hospitales y centros médicos pertenecientes a la SEDENA, a los militares y sus derechohabientes.	Concepto
Institución de Salud de la Secretaría Marina	---	SEMAR	Comprende a todas las clínicas pertenecientes a la Secretaría de Marina a través de los cuales se brindan servicios de salud en el primer nivel de atención, así como algunas jerarquías dentro del ámbito de la Salud Naval.	Concepto
Hospital o Sanatorio	---	---	Comprende a todos los hospitales y sanatorios	Concepto

			pertenecientes a otras instituciones de salud y seguridad social, a través de los cuales se brindan servicios de salud en el segundo nivel.	
Centro de salud y rehabilitación	---	---	Son todos los centros de salud y de rehabilitación pertenecientes al Sistema Nacional para el Desarrollo Integral de la Familia (SNDIF) a través de los cuales se brindan servicios de salud en el primer nivel de atención.	Concepto
Asistencia sanitaria privada	Medicina privada	---	Es la asistencia sanitaria proporcionada por entidades distintas al gobierno, empresas privadas a las que el ciudadano contribuye (generalmente vía la suscripción de seguros de salud).	Concepto
Asistencia sanitaria pública	---	---	Son organismos adscritos a la entidad gestora pública de la asistencia sanitaria, su función es la de gestionar y administrar los centros, servicios y establecimientos sanitarios.	Concepto
Instituciones de salud privadas que brindan servicios médicos subrogados	---	---	Comprende a todas las instituciones de salud pertenecientes al sector privado que ofrecen servicios a los trabajadores de instituciones o dependencias públicas y a sus beneficiarios, previo establecimiento de un contrato o convenio.	Concepto
Institución privada de beneficencia	---	---	Comprende a todas las unidades médicas pertenecientes a instituciones de beneficencia que ofrecen servicios médicos a la población abierta, tales como la Cruz Roja, dispensarios médicos, etc.	Concepto
Otro tipo de institución de salud privada	Consultorio particular	---	Todas las unidades médicas pertenecientes a instituciones de salud privadas que no fue	Concepto

			posible clasificarlas en las subclases anteriores	
Aseguradora, banco y otra institución de prepago de servicio médico	---	---	Esta clase comprende a todas las aseguradoras, bancos y otras instituciones a través de las cuales, las personas que han realizado un pago de manera anticipada o un contrato, tienen garantizado el acceso a servicios médicos, generalmente en instituciones privadas.	Concepto
Aseguradora	---	---	Comprende a todas las instituciones de seguros que ofrecen esquemas de protección médica en diferentes modalidades, desde coberturas básicas hasta las de gastos mayores, sin importar si el contrato o póliza fue adquirido de forma personal, o a través de una empresa, institución o patrón	Atributo de instancia
Tipo_seguro	---	---	Propiedad que establece si un usuario que utiliza los servicios de salud posee o no algún seguro.	Propiedad
No derechohabencia	---	---	Comprende todas descripciones de las que se puede deducir que la persona no tiene acceso garantizado a servicios medico	Propiedad
Derechohabencia no especificada	---	---	Comprende a todas las descripciones generales que no permiten saber si la persona tiene o no derecho a servicios médicos y en caso de tenerlo, la descripción impide ubicar con claridad la institución de derechohabencia	Propiedad
Seguro facultativo	---	---	Es el régimen de afiliación voluntario que ofrece la institución y bajo el cual pueden cotizar los trabajadores independientes, estudiantes.	Restricción

Seguro colectivo	---	---	Da cobertura a todas aquellas personas que formen parte de un mismo grupo homogéneo o tengan algún vínculo común.	Restricción
Especialidad	---	---	Características que se atienden de acuerdo con cada institución sanitaria (traumatología, oftalmología, cardiología, etc.).	Propiedad
Tipo_servicio	---	---	Se define si la institución sanitaria es de carácter estatal o federal.	Propiedad
Seguro_privado	---	---	Define el tipo de seguro contratado (gastos mayores, gastos menores)	Propiedad
No_Poliza	---	---	Clave de identificación proporcionada por la aseguradora al usuario.	Atributo de instancia
Monto de cobertura	---	---	Obligación principal del asegurador que consiste en hacerse cargo, hasta el límite de la suma asegurada, de las consecuencias económicas que se deriven del servicio medico.	Atributo de instancia
Nombre de asegurador	---	---	Designación o denominación verbal que se da a la aseguradora.	Atributo de instancia
Tipo_usuario	---	---	Define las características que conforman a la entidad que usa las instituciones sanitarias (estudiante, trabajador, empleado informal, jubilado, asegurado privado, etc.).	Propiedad
Seguro de gastos mayores	---	---	Contrato que brinda seguridad financiera y cubre todos los gastos médicos (quirúrgicos, de análisis, de medicamentos, etc.) requeridos por el asegurado ante un accidente o una enfermedad, en función de la suma asegurada contratada.	Restricción
Seguro de gastos menores	---	---	Contrato que brinda seguridad financiera es ésta quien proporciona alternativas dentro de su red, en este caso,	Restricción

			se tiene un grupo más cerrado de alternativas Este seguro es preventivo, más enfocado a la conservación de la salud.	
Sin seguro de gastos	---	---	No se cuenta con ninguna cobertura, sustento económico para ser atendido en alguna institución sanitaria.	Restricción
Estudiante	Alumno	---	Es el término que permite denominar al individuo que se encuentra realizando estudios una institución académica.	Restricción
Empleado sector formal	---	---	Se denomina al que se encuentra formalizado mediante la celebración de un contrato de trabajo entre el trabajador y el empleador, y se ajusta a los requerimientos de la ley.	Restricción
Familiar directo de empleado formal	---	---	Individuo perteneciente a la familia que tiene un empleo de manera formal y como consecuencia tiene derecho a recibir atención sanitaria.	Concepto
Jubilado	---	---	Es una prestación económica que recibe un trabajador por parte de su patrón al cumplir con ciertos requisitos.	Restricción
Auto-empleado	---	---	Es la actividad de una persona que trabaje para ella misma de forma directa en unidades económicas de su propiedad, que las dirige, gestiona y que obtiene ingresos de las mismas.	Restricción
Patrono	---	---	Provee un puesto de trabajo a una persona física para que preste un servicio personal bajo su dependencia, a cambio del pago de una remuneración o salario	Restricción
Empleado informal	---	---	Es aquel que carece de un contrato celebrado legalmente entre el patrón y el trabajador, y donde el empleado se encuentra al margen del	Restricción

			control tributario y de la protección que, en materia laboral, le brinda la ley.	
Sin trabajo	Desempleado	---	Desocupación o cesantía, en el mercado de trabajo, hace referencia a la situación del ciudadano que carece de empleo y, por lo tanto, de salario.	Restricción
Dirección	---	---	Es el domicilio en el cual se encuentra ubicado la Institución sanitaria.	Propiedad
Latitud	---	---	Distancia angular que hay desde un punto de la superficie de la Tierra hasta el paralelo del ecuador; se mide en grados, minutos y segundos sobre los meridianos	Atributo de instancia
Longitud	---	---	Distancia angular de un punto de la superficie terrestre al meridiano de Greenwich, determinada por el arco del ecuador comprendido entre dicho meridiano y el punto terrestre considerado; se mide en grados, minutos y segundos hasta los 180°	Atributo de instancia
es_un	---	---	Institución sanitaria (Publica, privada).	Relación
atiende	---	---	Se indica quien es el responsable de administrar dicha institución sanitaria.	Relación
Conforma	---	---	Institución sanitaria, servicio medico, si bien puede ser publica o privada.	Relación

5.2.2 Tarea 2: Taxonomía de conceptos

Esta tarea se describe la jerarquía que presentan las instituciones y servicios médicos a partir del glosario de términos (Tarea 1).

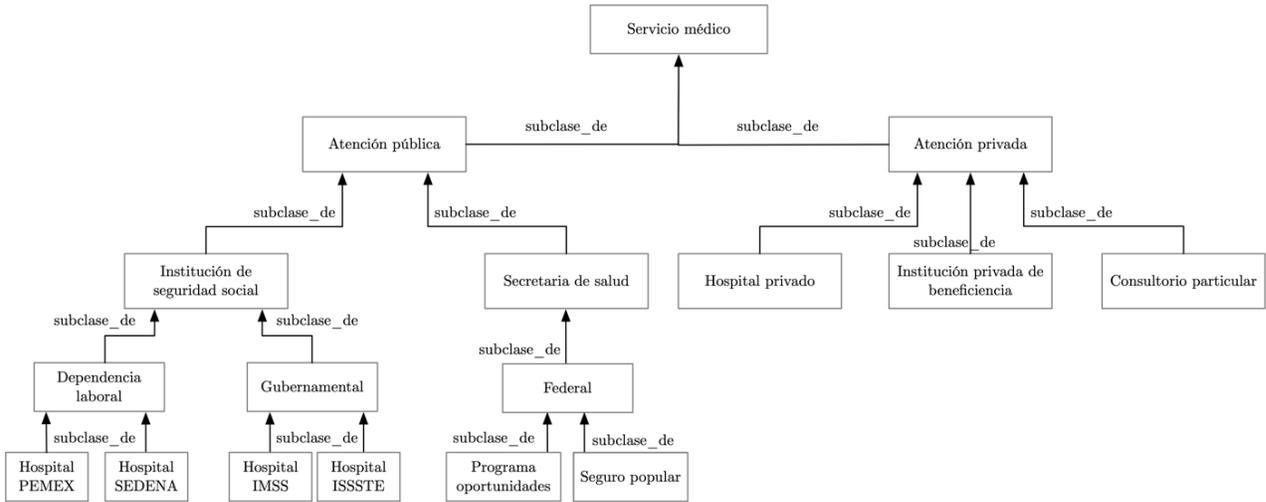


Figura 5.5 Taxonomía de conceptos

5.2.3 Tarea 3: Diagrama de relaciones binarias ad hoc

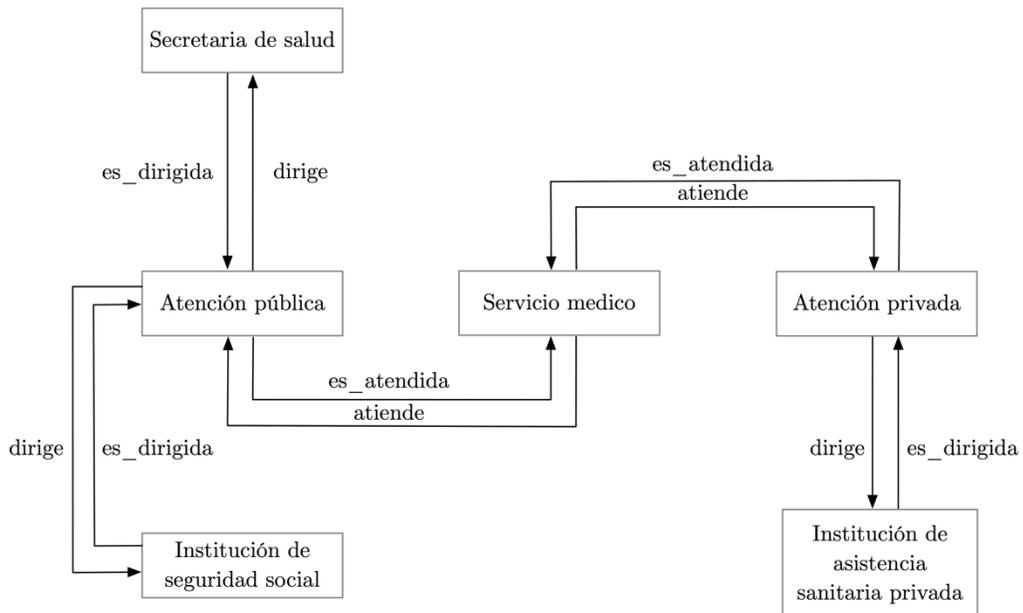


Figura 5.6 Relaciones binarias ad hoc

La tarea 3 en estas relaciones conectan los conceptos raíz de las taxonomías de conceptos de servicio medico y atención pública y privada. Desde un punto de vista de integración de ontologías, estas relaciones ad hoc expresan que la ontología de servicios médicos incluye a la institución de seguridad social y a la secretaria de salud.

Del mismo modo que la tarea anterior, se debe verificar que el diagrama no contenga errores, para ello se verifica el origen y el destino de todas las relaciones delimiten con exactitud y precisión los conceptos apropiados.

5.2.4 Tarea 4: Diccionario de conceptos

Para esta tarea se crea una lista con las relaciones y atributos de instancia de los conceptos que conforman la ontología, así como sus atributos de clase los cuales son locales a los conceptos, lo que significa que pueden ser repetidos en diversos conceptos. Por ejemplo, el concepto “Servicio medico” es el origen de las relaciones ad hoc “Atención pública” y “Atención privada”.

Tabla 5.2 Diccionario de conceptos

Concepto	Instancias	Atributos de clase	Atributos de instancia	Relaciones
Servicio medico	---	---	---	atiende
Atención pública	---	---	---	es_dirigida dirige
Atención Privada	---	Tipo_usuario Tipo_seguro	Aseguradora No_poliza	es_dirigida dirige
Secretaria de salud	---	---	sector	dirige
Institución de seguridad social	---	---	---	dirige
Hospital ISSSTE	-- Hospital regional 1 ^o de octubre.	Especialidad Direccion	---	---

-- Clínica de Especialidades "Indianilla".
 -- Clínica de Especialidades "Neuropsiquiatría".
 -- Clínica de Medicina Familiar "Gustavo A. Madero".
 -- Clinica medicina familiar "Guerrero".
 --Consultorio Auxiliar "Clinica Aragón".
 -- Clinica de especialidades Centro de Cirujia Ambulatoria "1º de octubre".
 --Clinica Giustavo A. Madero C.I y E.A del IPN.

Hospital IMSS	-- Unidad Medico Familiar 2 La Joya -- Unidad Medico Familiar 11 Peralvillo. -- Unidad Medico Familiar 20 Vallejo . -- Unidad Medico Familiar 23 U. Morelos. -- Unidad Medico Familiar 36 C. Atzacocalco. -- Unidad Medico Familiar 41 Fortuna. -- Unidad Medico Familiar 44 La Escalera. -- Unidad Medico Familiar 94 San Juan de Aragón. -- Hospital General Regional 24 Insurgentes.	Especialidad Direccion	---	---
---------------	---	---------------------------	-----	-----

	<ul style="list-style-type: none"> -- Hospital General Regional MF 29 Juan Aragón. -- Hospital U.M. Fisica Magdalena Salinas. -- Hospital Regional Traumatología Magdalena de las Salinas. -- Hospital Ortopedia Magdalena de las Salinas. -- Hospital Regional Psiquiatría Morelos. -- Hospital Unidad de Medicina Física Región Norte. 			
Seguro Popular	<ul style="list-style-type: none"> -- Hospital General Ticomán. -- Hospital Pediátrico La Villa. -- Hospital Materno Infantil Cuauhtepac. 	Especialidad Dirección	---	---
Hospital PEMEX	<ul style="list-style-type: none"> -- Hospital central Norte 	Especialidad Dirección	---	---
Hospital SEDENA	<ul style="list-style-type: none"> -- Hospital Central Militar -- Hospital de Especialidades de la Mujer y Neonatología. 	Especialidad Dirección	---	---
Hospital Privado	<ul style="list-style-type: none"> -- Hospital Ángeles Lindavista. -- Hospital de Especialidades MIG. -- Hospital Boutique Riobamba -- Hospital Santiago Apostol. -- Clínica Guadalupe Tepeyac. 	Especialidad Dirección	---	---

5.2.5 Tarea 5: Relaciones binarias ad hoc

Para cada relación binaria ad hoc, el desarrollo de la ontología debe especificar su nombre, los nombres de sus conceptos origen y destino, su cardinalidad y su relación inversa, si existe. La tabla 5.3 muestra una sección de la tabla de relaciones binarias ad hoc de la ontología de servicios médicos, que contiene la definición de las relaciones y de sus inversas en caso de existir.

Tabla 5.3 Descripción de relaciones binarias ad hoc

Nombre de la relación	Concepto origen	Cardinalidad máxima	Concepto destino	Relación inversa
atiende	Servicio medico	1	Atención publica	es_atendida
atiende	Servicio medico	1	Atención privada	es_atendida
dirige	Atención publica	N	Secretaria de salud	es_dirigida
dirige	Secretaría de salud	N	Institución de seguridad social	es_dirigida
dirige	Atención Privada	N	Institución de asistencia sanitaria privada	es_dirigida

5.2.6 Tarea 6: Atributos de instancia

Los atributos de instancia describen a las instancias del concepto y sus valores pueden ser distintos para cada una de dichas instancias.

Por cada atributo de instancia se debe especificar su nombre, el concepto al que el atributo pertenece (los atributos son locales a los conceptos), su tipo de valor, su rango de valores (en el caso de atributos numéricos), y sus cardinalidades mínima y máxima.

La tabla 5.4 muestra los atributos de instancia de la ontología servicios médicos. Esta tabla contiene los siguientes atributos de instancia del concepto. Atención privada: Aseguradora, No de Póliza, Tipo de cobertura.

Tabla 5.4 Descripción atributos de instancia

Nombre del atributo de instancia	Concepto	Tipo de valor	Rango de valores	Cardinalidad
Aseguradora	Atención Privada	String	---	(1, N)
No. Póliza	Atención Privada	Int	0...	(1, 1)
Tipo de cobertura	Atención Privada	String	---	(1, 1)

5.2.7 Tarea 7: Describir los atributos de clase

Para cada atributo de clase de la ontología se debe agregar la siguiente información: nombre del atributo, nombre del concepto donde el atributo se define, tipo de valor, cardinalidad y valor(es). Tal y como se muestra en la tabla 5.5:

Tabla 5.5 Atributos de clase

Nombre del atributo de clase	Concepto	Tipo de valor	Cardinalidad	Valores
Tipo_usuario	Gubernamental	String	(1, N)	-- Sector formal -- Autoempleado -- Jubilado -- Oportunidades -- Familiar

Tipo usuario	Dependencia laboral	String	(1, N)	-- Sector PEMEX -- Sector SEDENA -- Sector Marina
Tipo_seguro	Gubernamental	String	(1,1)	-- Facultativo -- Colectivo
Tipo de cobertura	Atención Privada	String	(1, 1)	-- Gastos menores -- Gastos mayores -- Sin cobertura
Especialidad	Hospital IMSS	String	(1, N)	---
Especialidad	Hospital ISSSTE	String	(1, N)	---
Especialidad	Hospital Privado	String	(1, N)	---
Dirección	Atención Privada	String	(1, 1)	---
Dirección	Hospital IMSS	String	(1, 1)	---
Dirección	Hospital ISSSTE	String	(1, 1)	---

5.2.8 Tarea 8: Describir las constantes

El objetivo de esta tarea es describir en detalle cada una de las constantes identificadas en el glosario de términos. Para cada constante se debe especificar su nombre, tipo de valor, valor y unidad de medida en el caso de constantes numéricas. Sin embargo, para la construcción de esta ontología no se tiene ningún valor constante, por lo cual esta tarea es omitida.

5.2.9 Tarea 9: Definición de axiomas formales

Para el diseño de esta ontología después de realizar un análisis no se encontraron motivos para los cuales deban de agregarse axiomas formales, estrictamente en la ontología de servicios médicos.

5.2.10 Tarea 10: Definir reglas

Para esta tarea METHONTOLOGY propone especificar las expresiones de las reglas utilizando el formato si <condiciones> entonces <consecuente>. La parte izquierda de la regla es una conjunción de condiciones simples, mientras que la parte derecha es una simple expresión de un valor de la ontología.

Tabla 5.6 Reglas en ontología

Regla	Descripción	Expresión	Conceptos	Atributos	Relaciones	Variables
Atención medica a beneficiarios	Solo se puede brindar atención medica a personas que se encuentren como derechohabiente	Si [Persona](?X) y [derechohabiente(?Y)] entonces [es_atendido(?X,?Y)]	-- Hospital IMSS -- Hospital ISSTE -- Hospital PEMEX -- Hospital SEDENA -- Hospital Marina	Tipo_usuario	Es_atendido	?X ?Y
Atención medica privada con aseguradora	Solo se puede brindar atención medica privada a través de una aseguradora a personas que se encuentren con un No. De poliza.	Si [Persona](?X) y [No_Poliza(?Y)] entonces [es_atendido(?X,?Y)]	-- Hospital Privado	-- No poliza -- Cobertura	Es_atendido	?X ?Y

5.2.11 Tarea 11: Describir instancias

Para cada instancia se define: su nombre, el nombre del concepto al que pertenece y los valores de sus atributos de instancia, si se conocen. La tabla 5.7 presenta algunas instancias de la tabla de instancias de la ontología de servicios médicos: Hospitales ISSSTE, Hospitales IMSS, Hospitales Privados. tal y como se definió en el diccionario de conceptos, y tienen valores definidos para algunos de los atributos Nombre, dirección, especialidad, latitud y longitud.

Tabla 5.7 Atributos de instancia

Nombre de la instancia	Concepto	Atributo	Valores
Hospital regional 1º de octubre	Hospital ISSSTE	Nombre	Hospital regional 1º de octubre
		Dirección	Av. Instituto Politécnico Nacional 1669, Revolución IMSS, 07300
		Especialidad	Centro de Cirugía Ambulatoria
		Latitud	19.486594
		Longitud	-99.134041
Clínica de especialidades “Indianilla”	Hospital ISSSTE	Nombre	Clínica de especialidades “Indianilla”
		Dirección	Audio Bernard 85, Doctores, 06720
		Especialidad	Clínica familiar
		Latitud	19.3146529
		Longitud	-99.2513025
Clínica de Especialidades Neuropsiquiatría	Hospital ISSSTE	Nombre	Clínica de Especialidades Neuropsiquiatría
		Dirección	Guerrero 346, Tlatelolco, 06900
		Especialidad	Neurología
		Latitud	19.4542761
		Longitud	-99.1465098
Clínica de Medicina Familiar “Gustavo A. Madero”	Hospital ISSSTE	Nombre	Clínica de Medicina Familiar Gustavo A. Madero
		Dirección	Tepeyac Insurgentes, 07020
		Especialidad	Clínica ambulatoria, Urgencias
		Latitud	19.4028671
		Longitud	-99.2030296

Clínica de Medicina Familiar “Guerrero”	Hospital ISSSTE	Nombre	Clínica de Medicina Familiar “Guerrero”
		Dirección	Guerrero 65, Guerrero, 06300
		Especialidad	Clínica ambulatoria
		Latitud	19.4410569
		Longitud	-99.1504252
Clínica de Medicina Familiar “Aragón”	Hospital ISSSTE	Nombre	Clínica de Medicina Familiar “Aragón”
		Dirección	Moctezuma 168, Gustavo A. Madero
		Especialidad	Medicina general
		Latitud	19.4782764
		Longitud	-99.1102096
Centro De Cirugía Ambulatoria “1º de octubre”	Hospital ISSSTE	Nombre	Centro De Cirugía Ambulatoria “1º de octubre”
		Dirección	Av. Instituto Politécnico Nacional 1669, Revolución IMSS
		Especialidad	Ortopedia, Cirugía General, Otorrinolaringología, Oftalmología, Urología, Ginecología, Cirugía Plástica y Reconstructiva, Cirugía Maxilofacial
		Latitud	19.3146529
		Longitud	- 99.2513025
Unidad Medico Familiar 2 La Joya.	Hospital IMSS	Nombre	UMF 2
		Dirección	Ote 91, La Joya, 07890
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.4564535
		Longitud	-99.1074837
Unidad Medico Familiar 11 Peralvillo.	Hospital IMSS	Nombre	UMF 11
		Dirección	Caruso S/N Esq. León Cavallo, Vallejo, 07870
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.46632
		Longitud	-99.13321
Unidad Medico Familiar 20 Vallejo.	Hospital IMSS	Nombre	UMF 20
		Dirección	Calzada Vallejo 675, Magdalena de las Salinas, 07760
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.49016
		Longitud	-99.10978

Unidad Medico Familiar 23 U. Morelos.	Hospital IMSS	Nombre	UMF 23
		Dirección	San Juan De Aragón 311, San Pedro El Chico, San Pedro el Chico, 07480
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.62083
		Longitud	- 99.90972
Unidad Medico Familiar 36 C. Atzacualco.	Hospital IMSS	Nombre	UMF 36
		Dirección	Segunda cerrada 5 Mayo entre Ejido y 5 de Mayo S/N col. Atzacualco Cp. 07040
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.4965482
		Longitud	-99.0973707
Unidad Medico Familiar 41 Fortuna.	Hospital IMSS	Nombre	UMF 41
		Dirección	Av. Fortuna, Magdalena de las Salinas, 07760
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.4833
		Longitud	-99.15
Unidad Medico Familiar 44 La Escalera.	Hospital IMSS	Nombre	UMF 44
		Dirección	Plan de San Luis 20, La Purísima Ticoman, 07320.
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.5158684
		Longitud	-99.1394597
Unidad Medico Familiar 94 San Juan de Aragón.	Hospital IMSS	Nombre	UMF 94
		Dirección	Calle Camino Antiguo a San Juan de Aragón No.235, Casas Alemán, 07580
		Especialidad	Unidad de Medicina Familiar
		Latitud	19.4782539
		Longitud	- 99.0851176
Hospital General Zona 24 Insurgentes	Hospital IMSS	Nombre	HGZ 24
		Dirección	Av. Insurgentes Norte. 1322, Magdalena de las Salinas, 07760
		Especialidad	Unidad de Hospitalización, Gral.
		Latitud	19.4822365
		Longitud	-99.1459817
	Hospital IMSS	Nombre	UM.FIS

Hospital U.M. Física Magdalena Salinas.		Dirección	Av. Instituto Politécnico Nacional 1603, Magdalena de las Salinas, 07760
		Especialidad	Unidad De Medicina Física Y Rehabilitación
		Latitud	19.4822365
		Longitud	-99.1459817
Hospital Regional Traumatología Magdalena de las Salinas.	Hospital IMSS	Nombre	UMAЕ
		Dirección	Av. Fortuna S/N, Magdalena de las Salinas, 07760.
		Especialidad	Traumatología
		Latitud	19.4822365
		Longitud	-99.1459817
Hospital Ortopedia Magdalena de las Salinas.	Hospital IMSS	Nombre	HORT
		Dirección	Av. Fortuna 101, Magdalena de las Salinas, 07760.
		Especialidad	Ortopedia.
		Latitud	19.4822365
		Longitud	-99.1459817
Hospital Regional Psiquiatría Morelos.	Hospital IMSS	Nombre	HRP 2N
		Dirección	DM Nacional, San Pedro el Chico, 07480.
		Especialidad	Psiquiatría
		Latitud	19.4836
		Longitud	-99.0956
Hospital Unidad de Medicina Física Región Norte.	Hospital IMSS	Nombre	UMAЕ
		Dirección	Avenida Instituto Politécnico Nacional 5421, Magdalena de las Salinas, 07760
		Especialidad	Rehabilitación Patologías Musculo
		Latitud	19.4833
		Longitud	-99.15
Hospital General Ticomán	Seguro Popular	Nombre	
		Dirección	Plan de San Luis s/n entre Miguel, Ticomán. 07330
		Especialidad	Ginecología, Consulta general, Medicina Familiar
		Latitud	19.5144421
		Longitud	-99.1377583
	Seguro Popular	Nombre	Hospital pediátrico la villa

Hospital pediátrico la villa		Dirección	Hidalgo 200, Villa Gustavo A. Madero, 07050
		Especialidad	Pediatría
		Latitud	19.4872209
		Longitud	-99.1164849
Hospital Materno Infantil Cuautepec	Seguro Popular	Nombre	Hospital Materno Infantil Cuautepec
		Dirección	Av. Emiliano Zapata 700, Cuautepec de Madero, 85800
		Especialidad	Pediatría, Cirugía general, Gineco obstetricia, Urgencias pediátricas, Colposcopia
		Latitud	19.5397889
		Longitud	-99.1409758
Hospital Central Norte	Hospital PEMEX	Nombre	Hospital Central Norte
		Dirección	Campo Matillas 52, San Antonio, 02720
		Especialidad	Consulta Externa, Hospitalización, Urgencias, Centro de Transfusión, Densitometrias Óseas, Diálisis Peritoneal y Hemodiálisis, Ecocardiografía, Electrocardiografía, Electroencefalografía, Quimioterapia
		Latitud	19.4800252
Hospital Central Militar	Hospital SEDENA	Longitud	-99.2009679
		Nombre	Hospital Central Militar
		Dirección	Perif. Blvd. Manuel Ávila Camacho s/n, Militar, 11200
		Especialidad	Neumología Cardiología, Cirugía General. Gastroenterología, Ginecología, Neuropsiquiatría, Endocrinología, Obstetricia, Oftalmología, Ortopedia, Otorrinolaringología, Pediatría, Procesados, Traumatología y Cirugía de Urgencia, Urología, Venereosifilografía.
		Latitud	19.4714207

		Longitud	-99.2498481
Hospital Militar De Especialidades De La Mujer Y Neonatología	Hospital SEDENA	Nombre	Hospital Militar De Especialidades De La Mujer Y Neonatología
		Dirección	Av. Industria Militar 1057, Lomas de Sotelo, Hipódromo de las Américas, 11200
		Especialidad	Neonatología, Ginecoobstetricia, Perinatología, Endocrinología, Pediatría, Consulta externa
		Latitud Longitud	19.4394909 -99.2187241
Hospital Ángeles Lindavista	Hospital Privado	Nombre	Hospital Ángeles Lindavista
		Dirección	Calle Riobamba 639, Lindavista Sur, 07760
		Especialidad	Alergología, Angiología y Cirugía Vasculatología, Audiología, Cardiología, Cirugía General Cirugía Maxilofacial N. M. Cirugía Neurológica, Cirugía Oncológica, Dermatología Endocrinología, Gastroenterología Geriátrica Ginecología Oncológica Ginecología y Obstetricia Hematología, Neumología Odontología, Oftalmología Traumatología.
		Latitud Longitud	19.4867588 -99.1317199
Hospital De Especialidades Mig.	Hospital privado	Nombre	Hospital De Especialidades Mig.
		Dirección	Calle Riobamba 800, Lindavista, 07300
		Especialidad	Gineco-obstetricia, urgencias, pediatría, consulta externa
		Latitud Longitud	19.491584 -99.1288291
Hospital Boutique Riobamba	Hospital privado	Nombre	Hospital Boutique Riobamba
		Dirección	Riobamba 927, 07300
		Especialidad	Cirugía Estética, Cardiología, Hematología, Neumología, Urología, Angiología

		Latitud	19.4948575
		Longitud	-99.1233166
Hospital Santiago Apóstol	Hospital privado	Nombre	Hospital Santiago Apóstol
		Dirección	Av. Cuauhtemoc 32, Jorge Negrete, 07280
		Especialidad	Angiología, Cardiología, Cirugía General, Cirugía Plástica y Reconstructiva, Gastroenterología Ginecología, Medicina General, Neonatología, Neurología, Oftalmología, Oncología, Otorrinolaringología, Pediatría Traumatología y Ortopedia
		Latitud	19.5265958
		Longitud	-99.144363
Clínica Guadalupe Tepeyac	Hospital privado	Nombre	Clínica Guadalupe Tepeyac
		Dirección	Calz. Ticomán 128, Lindavista Nte., 07300
		Especialidad	Urgencias, Ginecología, Ortopedia, Pediatría
		Latitud	19.4928315
		Longitud	-99.1233215

5.3 Transformación y recuperación de información original

De acuerdo con la metodología planteada es necesario realizar un tratamiento de los datos originales. En ese sentido, se procede a transformar a RDF los datos originales de los servicios médicos. Para conseguir lo anterior, es preciso mapear los datos contenidos en las fuentes originales como instancias de la ontología. La forma de hacerlo es a través de la creación de un modelo RDF a partir del establecimiento explícito de correspondencias un archivo de texto entre los elementos de los esquemas de las bases de datos a integrar y las clases definidas en la ontología.

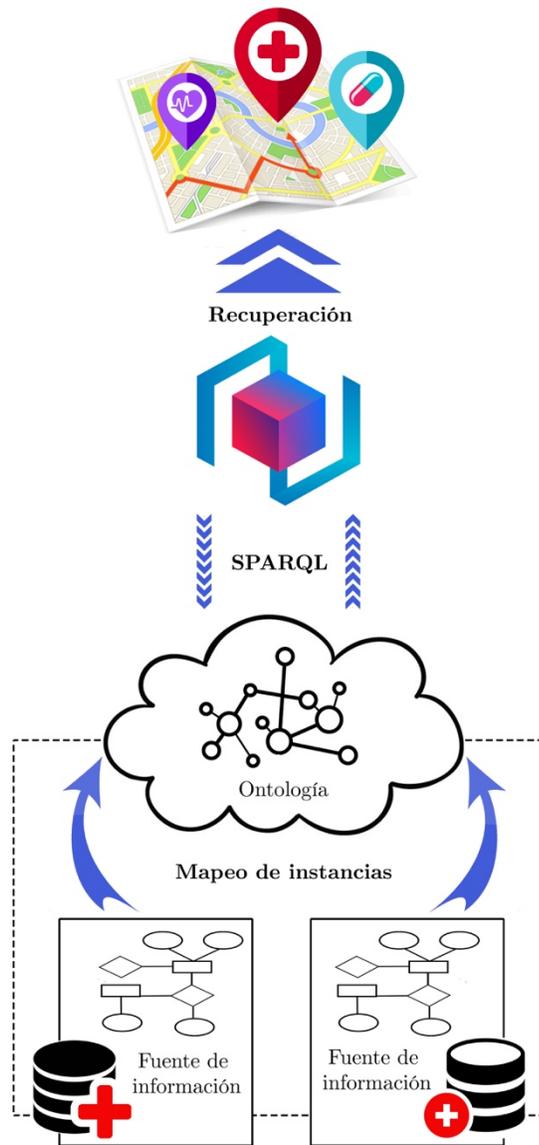


Figura 5.1 Recuperación de fuentes originales

La herramienta utilizada para la creación dicho modelo, DR2Q, tiene el soporte de la API de programación de Jena, la cual también provee las funciones necesarias para el procesamiento de las consultas SPARQL a través del modelo creado. El siguiente esquema Figura 5.7 muestra la metodología que se emplea para utilizar las fuentes originales de datos

La búsqueda de información por parte del usuario se hace a través de ciertos criterios ingresados dentro de la aplicación, que le permiten establecer parámetros para la recuperación de loa misma. Con la definición de estos criterios se envía una consulta SPARQL única cuya respuesta siempre arroja un conjunto de datos que cumplen el criterio definido. Dentro de este conjunto de datos recuperado se tienen además datos en relación con la ubicación geográfica de los objetos del dominio que se desea representar; en este caso de estudio: servicios médicos, los cuales, previa transformación a coordenada geográfica (latitud, longitud), son aprovechados para representar gráficamente en un mapa a los servicios de atención medica que cumplen el criterio de la consulta.

5.3.1 Mapeo de instancias (Hospitales)

Una vez que se ha determinado que fuentes de información, así como que atributos integran la ontología, el siguiente punto es hacer explicito también el reconocimiento de los atributos del modelo conceptual de cada una de las fuentes en atributos ideales para integrar la ontología. El establecimiento de estas correspondencias permite crear un puente entre los datos de las fuentes originales de datos y las clases de la ontología por el cual los datos de todas las fuentes (los valores de los atributos de sus entidades y las entidades mismas) son reconocidos como instancias de las clases en la ontología.

El archivo de mapeo comienza con la declaración de los prefijos que referencian los distintos espacios de nombrado (namespaces) utilizados en el resto del archivo para el establecimiento de correspondencias o mapeos entre los esquemas de bases de datos y la ontología. En general, las líneas que comienzan con # representan comentarios.

```
# Namespace of the ontology
@prefix onto:<http://localhost/hospitales/site/ssmedicos.owl#>
# Namespace of the mapping file
@prefix map: <file:/Users/ios/Documents/iOS/mapeo.n3#> .
# Other namespaces
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
@prefix jdbc: <http://d2rq.org/terms/jdbc/>

map:BDHospital a d2rq:Database;
    d2rq:jdbcDSN "jdbc:mysql://localhost/hospital ";
```

```

    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:username "root";
    d2rq:password "n0m3l0s3";
    jdbc:autoReconnect "true";
    jdbc:zeroDateTimeBehavior "convertToNull";

map:BDHospital a d2rq:ClassMap;
    d2rq:dataStorage map:BDHospital;
    d2rq:class onto:ssmedicos;
    d2rq:uriPattern " BDHospital/@@hospital.id@";

map:BDHospital__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property rdfs:label;
    d2rq:pattern "BDHospital #@@hospital.id@"; .
map:BDHospital_nombre a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Nombre;
    d2rq:column "hospital.nombre_hospital";
    d2rq:datatype xsd:string;

map:BDHospital_direccion a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Ubicacion;
    d2rq:column "hospital.direccion";
    d2rq:datatype xsd:string;

map:BDHospital_altitud a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Altitud;
    d2rq:column "hospital.altitud";
    d2rq:datatype xsd:double;

map:BDHospital_altitud a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Altitud;
    d2rq:column "hospital.altitud";
    d2rq:datatype xsd:double;

map:BDHospital_longitud a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Longitud;
    d2rq:column "hospital.longitud ";
    d2rq:datatype xsd:double;

map:BDHospital_especialidad a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:BDHospital;
    d2rq:property onto:Especialidad;
    d2rq:column "hospital.especialidad";
    d2rq:datatype xsd:string;

```

Finalmente, una vez que se tiene el archivo de mapeo se puede visualizar la construcción del grafo que describe dicho archivo el cual se muestra en la figura 5.9, esta figura muestra la correspondencia de las fuentes de información.

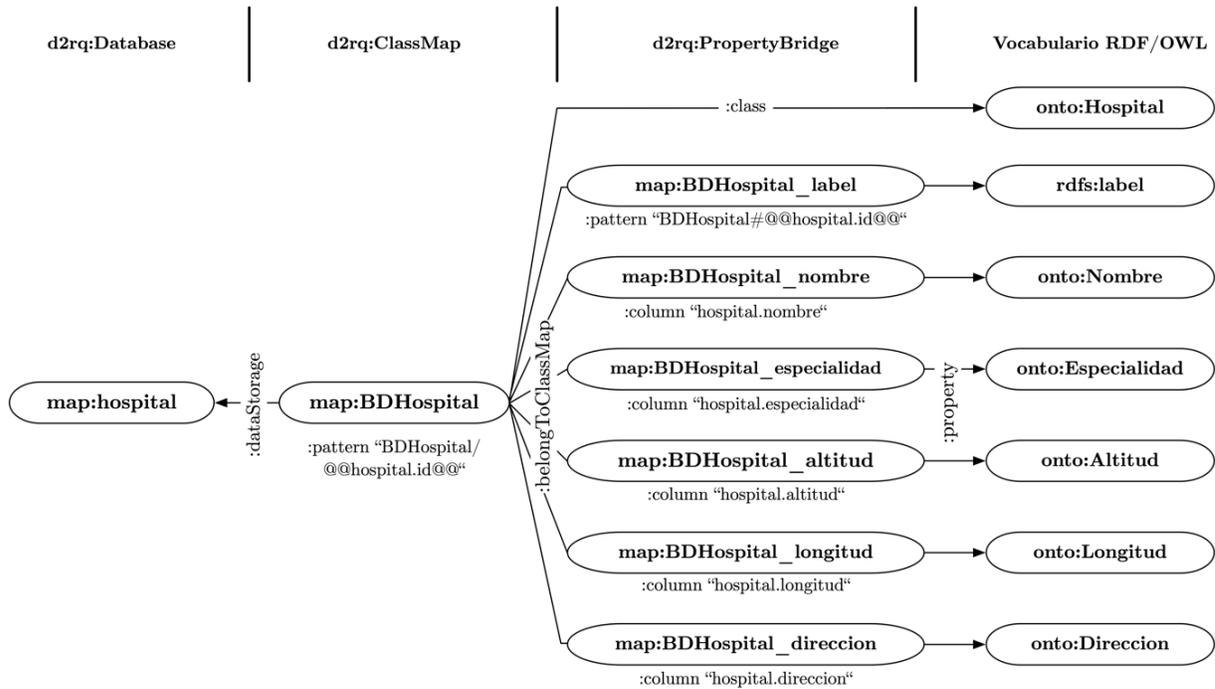


Figura 5.8 Mapeo D2RQ

5.3.2 Consulta mediante SPARQL

Las consultas en SPARQL contienen un conjunto de patrones de tripletas llamado patrón de grafo básico. Los patrones de tripletas son como los patrones RDF, excepto que tanto sujeto, como predicado y objeto pueden ser variables. Un patrón de grafo básico coincide con un subgrafo de los datos RDF cuando los términos RDF del subgrafo pueden ser sustituidos por las variables y el resultado es un grafo RDF equivalente al subgrafo

Retomando el capítulo anterior la plataforma D2RQ permite construir un Modelo RDF de Jena a partir del archivo de mapeo y a través de la implementación de la clase `ModelD2RQ` se consigue convertir los contenidos de las bases de datos en un grafo virtual. Con la plataforma D2RQ, a través del modelo creado, las consultas SPARQL son reescritas en tiempo de ejecución en consultas SQL hacia las bases de datos originales, lo que evita la necesidad de replicar los datos en un almacén RDF dedicado.

Los resultados de las consultas SQL son transformadas en tripletas RDF o en un conjunto de resultados SPARQL y entregados a las capas superiores del framework de Jena para su posterior procesamiento.

Las consultas se llevan a cabo sobre el repositorio de instancias ontológicas `ssmedicos.owl` las cuales tienen la siguiente estructura:

```
String querySPARQL =
    "PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# +
    "PREFIX mapeo.n3:
    <http://www.serviciosmedicos.com/ontologia/ssmedicos.owl#>" +
    "PREFIX owl:<http://www.w3.org/2002/07/owl#>" +
    "PREFIX owl2xml:<http://www.w3.org/2006/12/owl2-xml#>" +
    "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>" +
    "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>" +

    "SELECT DISTINCT * WHERE"+
    "{ "+
    "    ?Hospital result:nombre ?Nombre. "+
    "    ?Hospital result:especialidad ?Especialidad. "+
    "    ?Hospital result:direccion ?Direccion. "+
    "    ?Hospital result:altitud ?Altitud. "+
    "    ?Hospital result:longitud ?Longitud. "+
    "    FILTER ( ?especialidad = "+Especialidad+)" "+
    "};
```

Para poder ejecutar la anterior consulta se crea un modelo en el cual se envía el String **querySPARQL**:

```
OntModel modelo = null;

modelo ModelFactory.createOntologyModel(PelletReasonerFactory.THE_SPEC);

String ruta= getServletContext().getRealPath("/owl")+"/ssmedicos.owl";

java.io.InputStream in = FileManager.get().open(ruta);

    modelo.read(in, "");
    .
    .
    .

Query consulta = QueryFactory.create(querySPARQL);

QueryExecution ex = QueryExecutionFactory.create(consulta, modelo);
```

En el fragmento de código anterior la clase de `QueryExecutionFactory` proporciona un conjunto de métodos para crear objetos del tipo `QueryExecution` a partir de objetos tipo `Query` (`querySPARQL`).

Para poder mandar a llamar la ejecución de este query se emplea el método **QueryExecution**, en donde los objetos empleados tipo Query son la representación de los datos para consultarlos en su forma externa, el cual se obtiene a partir del método Query create de la clase **QueryFactory**, al cual se le da como argumento la cadena de consulta en lenguaje SPARQL.

La cadena que es enviada incluye los prefijos de los espacios de nombrado (namespaces) utilizados para acceder a los recursos especificados. La clase Model es la interface del framework de Jena a través de la cual se implementa el Modelo RDF sobre el que se hacen las consultas. El Modelo RDF es el que se creó en el apartado anterior del motor D2RQ. Posteriormente, el método **execSelect** de la interface **QueryExecution** ejecuta la consulta y devuelve un objeto que implementa la interface **ResultSet** del framework de Jena y que consiste en el conjunto de resultados obtenidos de la consulta en forma de tabla, donde cada fila corresponde al conjunto de elementos coincidentes que cumplen las condiciones de la consulta.

La consulta anterior devolvería como resultados las siguientes tuplas (Tabla 5.8):

Tabla 5.8 Resultados consulta SPARQL

Nombre	Dirección	Altitud	Longitud
Hospital regional 1º de octubre	Av. Instituto Politécnico Nacional 1669, Revolución IMSS, 07300	19.486594	-99.134041
Clínica de Medicina Familiar “Gustavo A. Madero”	Tepeyac Insurgentes, 07020	19.4028671	-99.2030296

La respuesta a esta consulta devuelve al usuario datos que inicialmente se tenían de forma aislada o distribuida en dos conjuntos de datos diferentes, heterogéneos y sin relación previa alguna, ahora unificados a través de la ejecución de una consulta única.

Hasta este punto ya se tiene una manera de representar explícitamente el modelo conceptual ideal para la localización de servicios médicos en este dominio de aplicación. Sin embargo, aun resta asignar una correcta visualización al momento de interpretar dichos datos, por lo cual el siguiente paso será hacer el reconocimiento y tratamiento de la consulta anterior.

Después de ejecutar y obtener la información de la ontología, esta se encapsula y se ordena en un vector, el procesamiento del vector se trata en el siguiente apartado.

5.3.3 Visualización de marcadores en mapa

La interfaz de visualización como ya se menciona en apartados previos es una aplicación móvil, la cual proporciona la funcionalidad para que el usuario pueda realizar consultas a la ontología.

Las opciones de búsqueda disponibles son:

- Búsqueda General
- Emergencia

La información que viaja hacia la aplicación móvil, específicamente al mapeo de instancias, el cual se utiliza para realizar la consulta en SPARQL sobre el modelo persistente de datos.

Con los resultados de la consulta SPARQL, acompañados cada uno de ellos de la coordenada geográfica que le corresponde, se puede crear un archivo vector cuyos elementos contienen la información de los hospitales que cumplen con el criterio establecido en la consulta y las coordenadas geográficas que corresponden de su ubicación.

El framework de desarrollo utilizado para la visualización de los marcadores es proporcionada por AppleMap Kit.

El primer paso es importar el framework en el **UIViewController**. Para ello, se coloca el siguiente código al principio del controlador.

```
import MapKit
```

A continuación, se conecta el mapa del Storyboard al código del controlador que lo contiene.

```
@IBOutlet weak var map: MKMapView!
```

Se cambia la declaración de la clase ViewController para que implemente el protocolo **MPMapViewDelegate**.

```
class ViewController: UIViewController, MKMapViewDelegate
```

Además se define el controlador como delegado del mapa para que responda a los eventos relacionados con éste, dentro de la función `viewDidLoad()`.

```
map.delegate = self
```

Seguidamente, se declara la latitud y longitud del mapa (estas coordenada son obtenidas a través del vector de resultados)

```
let altitud:CLLocationDegrees = 19.4028671
let longitud:CLLocationDegrees = -99.2030296
```

Posteriormente se crea el campo de vision del mapa, es decir, la extensión de area visible cuando éste cargue.

```
let latDelta:CLLocationDegrees = 0.01
let longDelta:CLLocationDegrees = 0.01
```

Con la latitud, la longitud y el campo de visión se crea el span y la location.

```
let span:MKCoordinateSpan = MKCoordinateSpanMake(latDelta, longDelta)
let location:CLLocationCoordinate2D = CLLocationCoordinate2DMake(latitude,
longitude)
```

Se crea la región final.

```
let region:MKCoordinateRegion = MKCoordinateRegionMake(location, span)
```

Por último, se añade la region al mapa.

```
map.setRegion(region, animated: true)
```

En este punto, ya se tiene en la aplicación un mapa con el sitio fijado por la latitud y la longitud, este proceso se realiza con un ciclo para agregar cada uno de los marcadores correspondientes.

Ahora bien se añaden las anotaciones en el mapa. Para eso se crea una `MKPointAnnotation` en la location creada anteriormente.

```
let annotation = MKPointAnnotation()
annotation.coordinate = location
annotation.title = "Hospital 1° de Octubre"
annotation.subtitle = "Av. Instituto Politécnico Nacional 1669,
Revolución IMSS, 07300"
```

Y luego se añade en el mapa.

```
func mapView(mapView: MKMapView, viewForAnnotation annotation:
MKAnnotation) -> MKAnnotationView? {
    if !(annotation is CustomAnnotation) {
        return nil
    }
    let reuseId = "ISSSTE"
    var anView = mapView.dequeueReusableAnnotationViewWithIdentifier(reuseId)

    if anView == nil {
        anView = MKAnnotationView(annotation: annotation,
reuseIdentifier: reuseId)
        anView!.canShowCallout = true
        mapView.addAnnotation(annotation)
    }
    else {
        anView!.annotation = annotation
    }

    return anView
}
```

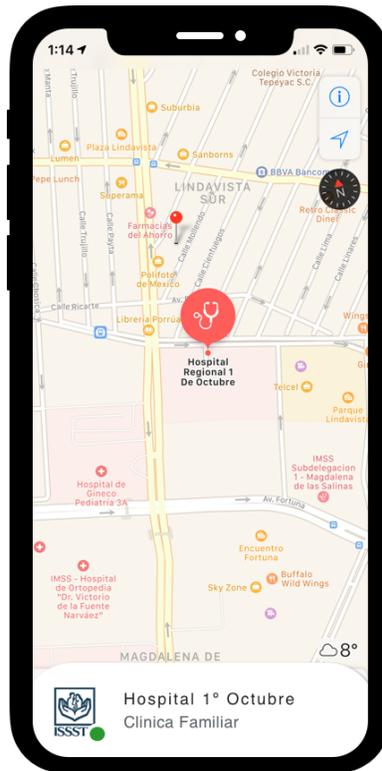


Figura 5.9 Agregar anotaciones

CAPÍTULO VI. PRUEBAS Y
RESULTADOS



En esta sección se presentan algunas pruebas realizadas y sus implicaciones, igualmente, se muestran algunos resultados y comentarios acerca de su interpretación que estos tienen.

El capítulo se encuentra dividido de la misma manera que la metodología; esto con la finalidad de presentar los resultados por cada una de las etapas definidas. La sección de recuperación se refiere a la aplicación móvil y los experimentos realizados en un dispositivo móvil.

6.1 Creación capa de integración de información

Una vez que se generó el modelo, el cual se definió en el capítulo 5, se realiza la codificación de la ontología utilizando un lenguaje formal; para esto se seleccionó la herramienta de edición Protégé, a través de la cual, al definir la ontología, se genera la codificación en el lenguaje OWL.

La fase de integración con la metodología Methontology consiste propiamente en la recopilación de las definiciones contenidas en distintos repositorios de datos como conceptos básicos para la ontología de servicios médicos a elaborar en el presente trabajo. La construcción de la ontología se realizó en Protégé [52], generando un archivo OWL con la estructura de esta, sus propiedades y especificaciones. De la misma forma, se exportaron en este mismo archivo las instancias de Hospitales que se dieron de alta para utilizarse en el caso de estudio.

Al iniciar Protégé, se habilitan las siguientes opciones (en Window -> Tabs):

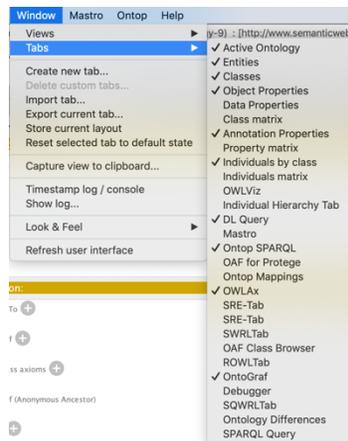


Figura 6.1 Activar opciones Protégé

La siguiente pestaña que nos interesa es Classes, ahí es donde se colocan los conceptos que definen a la ontología. Tomando como modelo la tarea 1 de Methontology.



Figura 6.2 Subclases principales

Para esta ontología las dos subclases principales son:

- Atención Pública
- Atención Privada

A cada clase se le asigna un nombre y de manera automática se da una URL, esto es debido a que cada concepto se está almacenando en un lenguaje estándar para ser manipulado, por ejemplo, con Web Semántica, una regla importante es que las clases deben ser únicas.

La forma gráfica de esta ontología en Protégé queda de la siguiente manera:

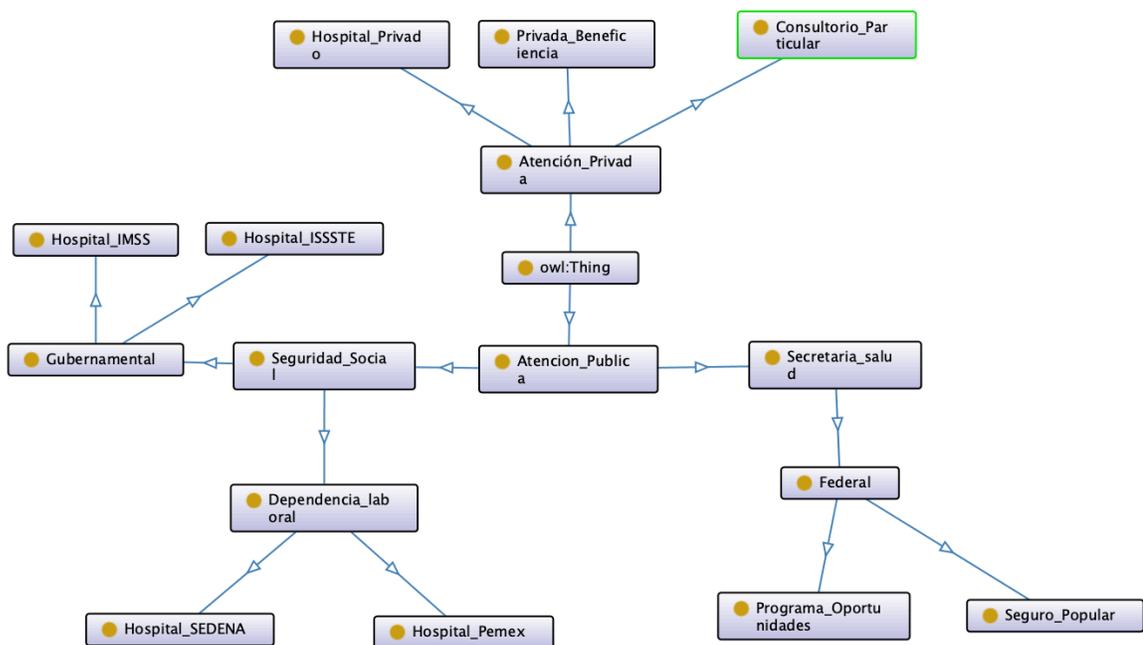


Figura 6.3 Red ontológica Servicios Médicos

Con OWL y en particular con Protégé 5.5, la forma de establecer una relación entre dos clases distinta a is_a (relación superclase-subclase) es a través de la especificación explícita del dominio y rango de una propiedad o a través de la creación de restricciones.

Una restricción es un tipo de clase descrita a partir de las relaciones en las que participan sus individuos. La clase de los individuos que tiene al menos una relación a través de una propiedad específica con individuos que son miembros de otra clase, se conoce como restricción existencial.

Posteriormente se procede a agregar los atributos de clase correspondientes de cada entidad para ello se accede a la pestaña Data Properties.

Una vez definidos los atributos de la ontología lo siguiente es agregar instancias, es decir objetos que prueben la validez de la ontología, para ello nos desplazamos a la pestaña de Individuals, para agregar un individuo o instancia usamos el botón de agregar individuos, marcado en la imagen.



Figura 6.4 Agregar instancias

A su vez se tiene que ligar estos atributos de clase con las entidades correspondientes por lo que se procede a definir de manera individual cada instancia.

Una vez definidas las instancias se deben de vincular con los atributos definidos previamente, seleccionando en Data property assertions:

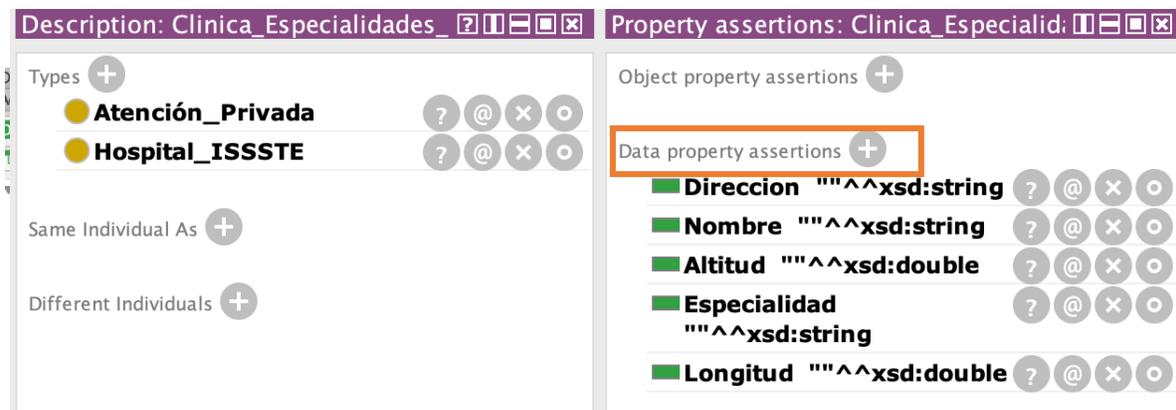


Figura 6.5 Vincular atributos

Las instancias que se agregan no están asociadas a ninguna clase en particular, para asignarlas hay que regresar a la pestaña de Classes, seleccionar la clase a la que le queremos agregar individuos y seleccionar la opción Members para integrararlos. Estas instancias se obtienen de la tarea 11 de Methontology.

Una vez que se abren las instancias se agregan a la clase que se seleccionó de acuerdo con el dominio establecido.

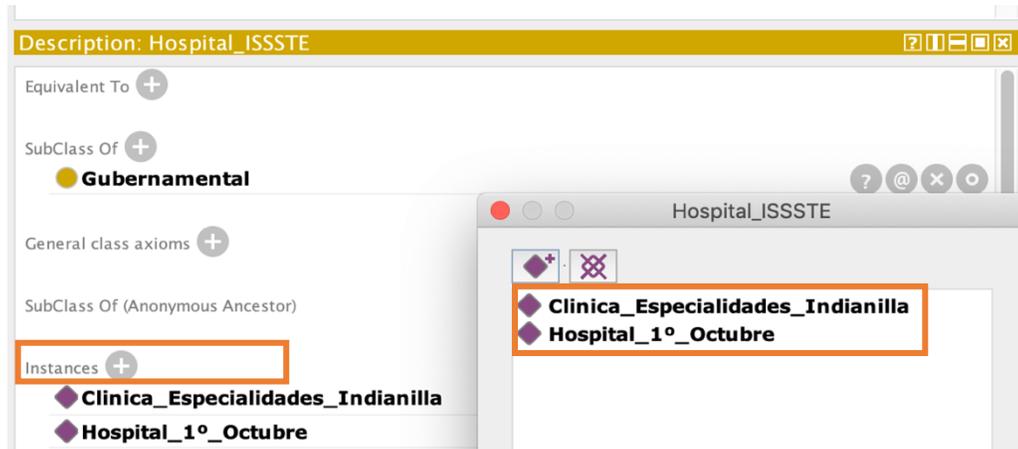


Figura 6.6 Vincular instancias en clases

Para poder relacionar semánticamente la información de cada clase se establecen las relaciones así como sus inversos, definidos en la tarea 3 relaciones binarias

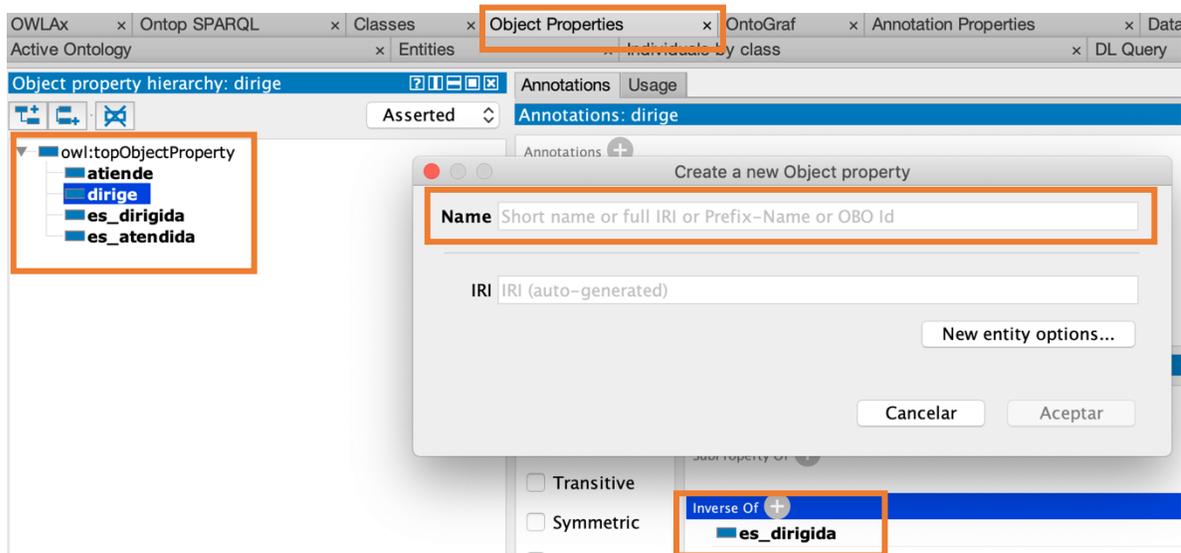


Figura 6.7 Relaciones de clases

6.2 Creación de ontología en el estándar OWL

De manera general se tiene la base de esta ontología. Esta ontología se almacena en formato OWL/XML o que estos dos formatos principales nos quieren decir es que las ontologías van más allá del contenido de información: se toma también en cuenta la estructura y contexto del modelo generado, el cuál se puede compartir e implementar en diferentes tecnologías. A continuación, se muestra un fragmento de esta ontología:

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"

xml:base="http://www.semanticweb.org/macbook/ontologies/2018/10/ssmedicos.o
wl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

ontologyIRI="http://www.semanticweb.org/macbook/ontologies/2018/10/ssmedico
s.owl">
  <Prefix name=""
IRI="http://www.semanticweb.org/macbook/ontologies/2018/10/ssmedicos.owl#" /
>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace" />
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Annotation>
    <AnnotationProperty abbreviatedIRI="rdfs:comment" />
    <Literal>Ontología de servicios medicos</Literal>
  </Annotation>
  <DataProperty IRI="#Nombre" />
</Declaration>
<Declaration>
</SubClassOf>
<ClassAssertion>
  <Class IRI="#Atención_Privada" />
  <NamedIndividual IRI="#Clinica_Especialidades_Indianilla" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Hospital_ISSSTE" />
  <NamedIndividual IRI="#Clinica_Especialidades_Indianilla" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Atención_Privada" />
  <NamedIndividual IRI="#Hospital_1º_Octubre" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Hospital_ISSSTE" />
  <NamedIndividual IRI="#Hospital_1º_Octubre" />
</ClassAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#Altitud" />
  <NamedIndividual IRI="#Clinica_Especialidades_Indianilla" />
```

```

    <Literal
datatypeIRI="http://www.w3.org/2001/XMLSchema#double"></Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#Direccion"/>
    <NamedIndividual IRI="#Clinica_Especialidades_Indianilla"/>
    <Literal></Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#Especialidad"/>

```

6.3 Diseño caso de estudio

6.3.1 Requerimientos funcionales

Los requerimientos funcionales se han abordado respecto a lo que se espera que el sistema cubra en funcionalidad para localizar servicios médicos. A continuación, se muestra cada uno de los requerimientos identificados en la aplicación móvil, especificando para cada uno de ellos un identificador único con el cual se hará referencia a los mismos en el documento, el nombre del requerimiento y su descripción.

La nomenclatura utilizada es la siguiente:

- **Requerimiento Funcional:** RF#
Ejemplo: RF1, RF2, ..., RFN
- **Requerimiento no Funcional:** RNF#
Ejemplo: RNF1, RNF2, ..., RNFN

En donde el número de requerimientos puede ir desde 1 a N, siendo N un número finito natural.

RF01-AM Agregar datos personales

Descripción: Se crea una cuenta única con los datos personales del usuario

RF02-AM Registro de Contactos

Descripción: Se registran contactos dentro de la aplicación, denominados “Contactos Favoritos”, en los cuales incluyen amigos y familiares.

Estos contactos recibirán un mensaje de texto (SMS) con la ubicación y el tipo de emergencia.

RF03-AM Realizar búsqueda

Descripción: El usuario realiza una petición con distintos parámetros para la búsqueda dentro de la ontología

RF04-AM Visualizar Resultado

Descripción: Podrá visualizar en la pantalla del dispositivo móvil el resultado obtenido de la consulta a la ontología.

6.3.2 Requerimientos no funcionales

A continuación, se muestran los requerimientos no funcionales los cuales están separados en Restricciones y en Propiedades, especificando para cada uno de ellos un identificador único con el cual se hace referencia a los mismos en el documento, el nombre del requerimiento y su descripción.

6.3.2.1 Restricciones

Las restricciones son características que no pueden ser negociadas y que son como guía o definición para el sistema

RNF01-R Lenguaje de programación

Descripción: El lenguaje de programación es Swift utilizando el SDK oficial (Xcode) para desarrollo de aplicaciones para el sistema operativo iOS.

RNF02-R Sistema operativo

Descripción: El sistema operativo para ejecutar la aplicación, es iOS 12, de esta manera se busca que la aplicación sea compatible con versiones superiores.

RNF03-R Características del hardware donde se ejecuta la aplicación

Descripción: El equipo donde se ejecute la aplicación móvil deberá contar como mínimo con las siguientes características del hardware:

- Procesador dual-core de 1.2 Ghz.
- 1 GB de memoria RAM.
- Pantalla touch capacitiva.

RNF04-R Características de la aplicación móvil

Descripción: La aplicación móvil que realiza las consultas a la ontología debe contar como mínimo con las siguientes características:

- La aplicación debe ser compatible con sistema operativo iOS 11 y versiones superiores.
- Los mapas para la localización deberán estar actualizados.
- Inicio rápido de la aplicación y bajo consumo de recursos.

6.3.2.2 Propiedades

Las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las propiedades del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema.

RNF04-R Interfaz gráfica de la aplicación

Descripción: La interacción de la aplicación móvil con el usuario será mediante una interfaz gráfica de usuario sencilla.

6.3.3 Reglas de negocio

Las Reglas del Negocio describe las políticas, normas, operaciones, definiciones y restricciones presentes en este trabajo y que son de vital importancia.

6.3.3.1 Definición

RND01 Contactos favoritos

Tipo: Definición

Descripción: Los contactos de confianza son aquellos a contactos previamente seleccionados por el usuario, a los cuales, les llegará el mensaje de texto con el nivel de la emergencia medica así como su ubicación.

RND02 Mensaje de texto

Tipo: Definición

Descripción: Los mensajes de texto son el medio por el cual el usuario envía información a un contacto de confianza previamente registrado en la aplicación. En donde el contenido del mensaje de texto es la ubicación actual del usuario y su tipo de emergencia medica.

RND03 Localización

Tipo: Definición

Descripción: La aplicación cuenta con un sistema de localización GPS implementado, esto es para que el usuario pueda dar a conocer a sus contactos de confianza su ubicación y así mismo poder ver las ubicaciones de los servicios médicos.

6.3.3.1 Restricción

RNR01 Apartado de emergencias

Tipo: Restricción

Descripción: La aplicación móvil contiene un espacio exclusivamente para reportar una emergencia medica.

RNR02 Información incorrecta

Tipo: Restricción

Descripción: Los datos proporcionados al sistema que son marcados como “requeridos” no se deben omitir. Todos los datos proporcionados al sistema deben respetar el formato y pertenecer al tipo de dato especificado; así como estar dentro de la longitud máxima o mínima definida.

RNR03 Información de registro del usuario

Tipo: Restricción

Descripción: Los datos del usuario requeridos para el uso de la aplicación son:

- Nombre
- Apellido
- Edad

- Tipo de seguro medico

6.3.2 Modelo de arquitectura 4 + 1

El modelo de 4+1 [55] vistas fue desarrollado por Philippe Kruchten, el cual encaja con el estándar IEEE 1471-2000 Recommended Practice for Architecture Description of Software Intensive Systems que se utiliza para describir la arquitectura de un sistema de software intensivo basado en el uso de múltiples puntos de vista. El modelo 4+1 describe la arquitectura del software usando cinco vistas concurrentes. Tal como se muestra en la Figura 6.8 Modelo 4+1 de Kruchten, donde cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema.

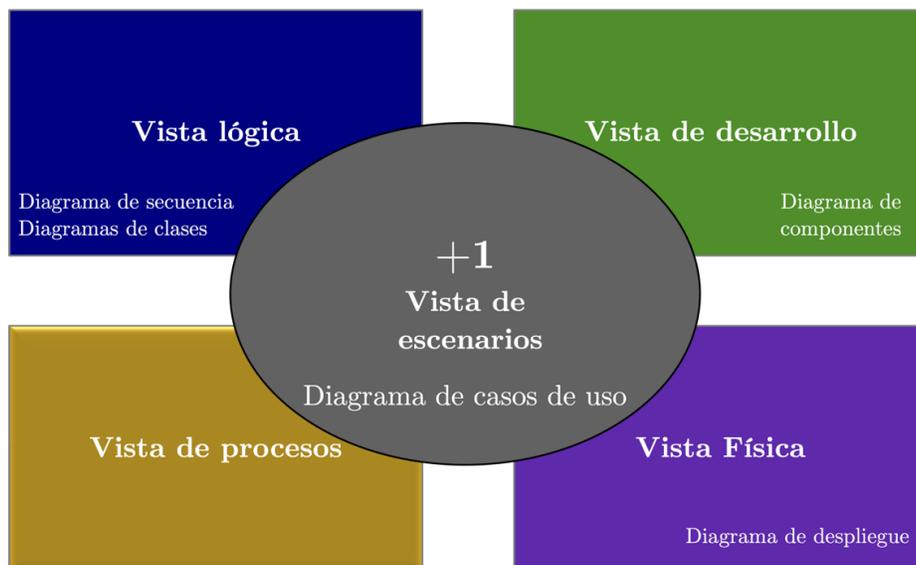


Figura 6.8 Modelo de Kruchten

- La vista lógica describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos.
- La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.
- La vista de procesos describe los aspectos de concurrencia y sincronización del diseño.
- La vista física describe el mapeo del software en el hardware y refleja los aspectos de distribución.

La descripción de las decisiones arquitecturales se puede organizar en estas cuatro vistas y luego ilustrarlas con un conjunto reducido de casos de uso o escenarios, los cuales constituyen la quinta vista. La arquitectura evoluciona parcialmente a partir de esta vista de escenarios.

En el presente trabajo se expone una vista adicional al modelo 4+1, la vista de datos, en la cual se describe el esquema de la ontología

6.3.2 Vista de escenarios

6.3.3.1 Casos de uso aplicación móvil

La siguiente imagen (Figura 6.9) muestra el diagrama de los casos de uso del sistema, contemplando a un solo actor final: Usuario.

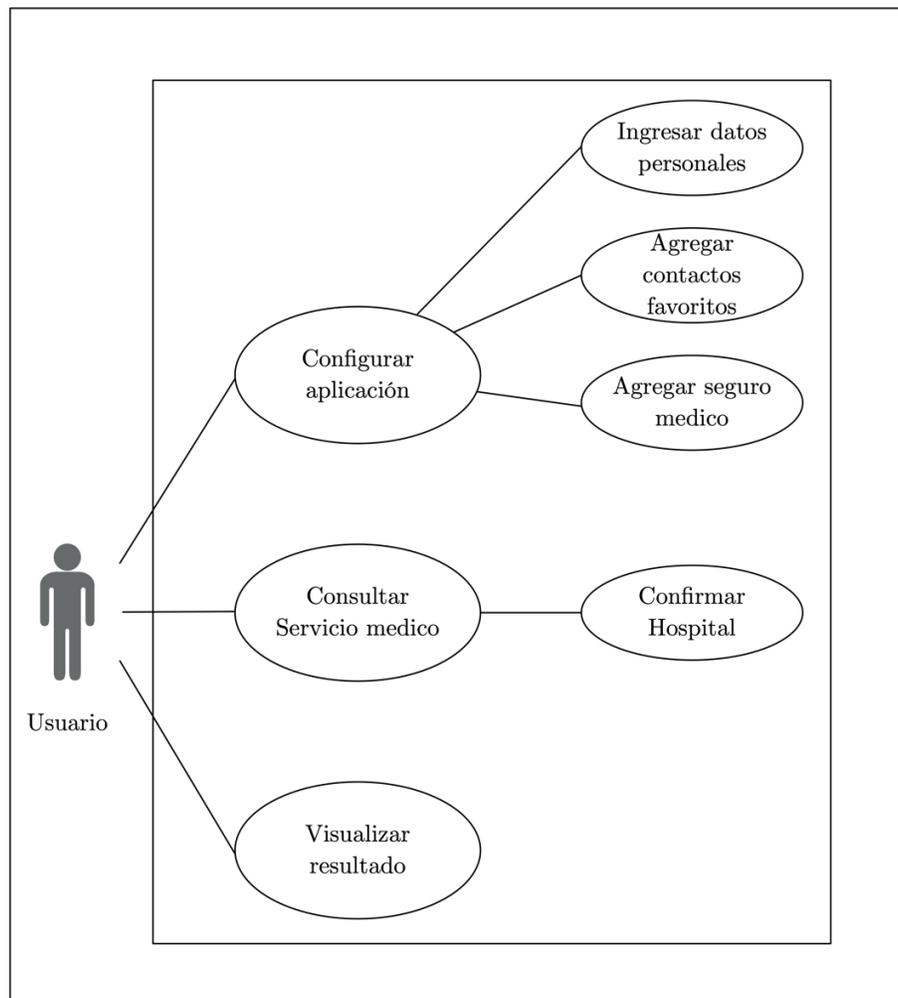


Figura 6.9 Diagrama de casos de uso

6.3.3.1.1 CUA01 Configurar aplicación

Resumen

Este caso de uso brinda al usuario la posibilidad de personalizar la aplicación ingresando su nombre, y dando altas y bajas de nuevos contactos para la agenda de contactos favoritos de la aplicación, así como también el registro del seguro médico.

Descripción

Tabla 6.1 CUA01 Configurar aplicación

Caso de uso:	CUA01 Configurar aplicación
Actor	Usuario
Objetivo	Permitir al usuario hacer los ajustes necesarios a la aplicación para su correcto funcionamiento.
Entradas	Para llevar a cabo la configuración de la aplicación, el usuario tiene que entrar a la pantalla de inicio y buscar la opción “Ajustes”.
Salidas	Se muestra los campos que se pueden modificar.
Precondiciones	Ninguna.
Pos condiciones	Se agrega a la base de datos local los datos del usuario.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR02 Información correcta▪ RNR03 Información de registro de usuario
Errores	Se muestra en la Pantalla el mensaje de error. Falta un dato para efectuar la operación solicitada cuando el usuario no haya ingresado datos en campos obligatorios.
Tipo	Primario

6.3.3.1.2 CUA02 Ingresar datos personales

Resumen

Este caso de uso brinda al usuario la posibilidad de ingresar los datos personales, para posteriormente tener un resultado mas personalizado en la localización de servicios médicos.

Descripción

Tabla 6.2 CUA02 Ingresar datos personales

Caso de uso:	CUA02 Ingresar datos personales
Actor	Usuario
Objetivo	Permitir al usuario ingresar sus datos personales.
Entradas	Para poder ingresar los datos personales el usuario debe acceder a la pantalla IU02 Ingresar datos personales en donde se ingresa: <ul style="list-style-type: none">▪ Nombre▪ Edad▪ Sexo▪ Correo electrónico
Salidas	Se muestra el mensaje de notificación Operación realizada exitosamente
Precondiciones	Ninguna.
Pos condiciones	Se agrega a la base de datos local los datos del usuario.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR02 Información correcta▪ RNR03 Información de registro de usuario
Errores	Se muestra en la Pantalla el mensaje de error. Falta un dato para efectuar la operación solicitada cuando el usuario no haya ingresado datos en campos obligatorios.
Tipo	Secundario, viene del CUA01 Configurar aplicación.

6.3.3.1.3 CUA03 Agregar contacto favorito

Resumen

Este caso de uso brinda al usuario la posibilidad de agregar un contacto llamado favorito, al cual llegará una notificación por medio de un mensaje de texto, información al destinatario el estado actual de la persona, así como su ubicación.

Descripción

Tabla 6.3 CUA03 Agregar contacto favorito

Caso de uso:	CUA03 Agregar contacto favorito
Actor	Usuario
Objetivo	Permitir al usuario agregar un contacto favorito desde su agenda de contactos a la aplicación móvil.
Entradas	Para llevar a cabo el agregar un contacto como favorito el usuario tiene que acceder a la pantalla de Agregar Favorito dentro de Ajustes
Salidas	Se muestra el mensaje de notificación Operación realizada exitosamente
Precondiciones	Ninguna.
Pos condiciones	Se agrega el contacto desde la agenda a la base de datos.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR02 Información correcta▪ RNR03 Información de registro de usuario
Errores	Se muestra en la Pantalla el mensaje de error. Falta un dato para efectuar la operación solicitada cuando el usuario no haya ingresado datos en campos obligatorios.
Tipo	Secundario, viene del CUA01 Configurar aplicación

6.3.3.1.4 CUA04 Consultar servicio médico

Resumen

Este caso de uso brinda al usuario la posibilidad realizar la búsqueda de acuerdo con los datos ingresados.

Descripción

Tabla 6.4 CUA04 Consultar servicio médico

Caso de uso:	CUA04 Consultar servicio médico
Actor	Usuario
Objetivo	Permitir al usuario realizar una búsqueda personalizada de acuerdo a sus necesidades.
Entradas	Para llevar a cabo una consulta el usuario debe presionar el botón realizar búsqueda de la pantalla IU01 Pantalla de inicio .
Salidas	Se muestran los resultados en la pantalla IU01 Pantalla de inicio .
Precondiciones	Ingresar los datos personales y seguro medico.
Pos condiciones	Se muestra una lista con servicios médicos a elegir.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR01 Apartado de emergencias
Errores	Se muestra en la Pantalla el mensaje de error. Falta un dato para efectuar la operación solicitada cuando el usuario no haya ingresado datos en campos obligatorios.
Tipo	Primario

6.3.3.1.5 CUA05 Confirmar hospital

Resumen

Este caso de uso brinda al usuario la posibilidad de confirmar el servicio médico de una lista mostrada.

Descripción

Tabla 6.5 CUA05 Confirmar hospital

Caso de uso:	CUA05 Confirmar hospital
Actor	Usuario
Objetivo	Permitir al usuario confirmar a que servicio medico desea acudir.
Entradas	Seleccionar de una lista desplegable las opciones disponibles.
Salidas	Se muestran la ruta y la dirección al hospital deseado.
Precondiciones	Haber realizado una petición de consultar servicio.
Pos condiciones	Se muestra una lista con servicios médicos a elegir.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR01 Apartado de emergencias
Errores	Se muestra en la Pantalla si ocurrió un error al realizar el trazo o visualización de la ubicación geográfica.
Tipo	Secundario viene del CUA04 Consultar servicio médico.

6.3.3.1.6 CUA06 Visualizar resultado

Resumen

Este caso de uso brinda al usuario la posibilidad de visualizar los servicios médicos una vez realizada la petición e ingresado los datos.

Descripción

Tabla 6.6 CUA06 Visualizar resultado

Caso de uso:	CUA05 Visualizar resultado
Actor	Usuario
Objetivo	Permitir al usuario visualizar las distintas opciones de servicios médicos de acuerdo a los parámetros de este.
Entradas	Para visualizar los sitios de interés el usuario tiene que haber ingresado previamente sus datos.
Salidas	Se muestran la ruta y la dirección al hospital deseado.
Precondiciones	Haber realizado una petición de consultar servicio.
Pos condiciones	Se muestra la ruta del servicio medico seleccionado.
Reglas del negocio	<ul style="list-style-type: none">▪ RNR01 Apartado de emergencias
Errores	Se muestra en la Pantalla si ocurrió un error al realizar el trazo o visualización de la ubicación geográfica.
Tipo	Primario

6.4 Pruebas

Se realizaron una serie de pruebas para corroborar la eficacia de la ontología y determinar los posibles errores cometidos; el resultado de las pruebas son los siguientes:

Primera prueba

Se realizó esta prueba para comprobar el tiempo de búsqueda en la ontología, los parámetros de búsqueda son los siguientes:

- Seguro medico: IMSS
- Especialidad: Cardiología

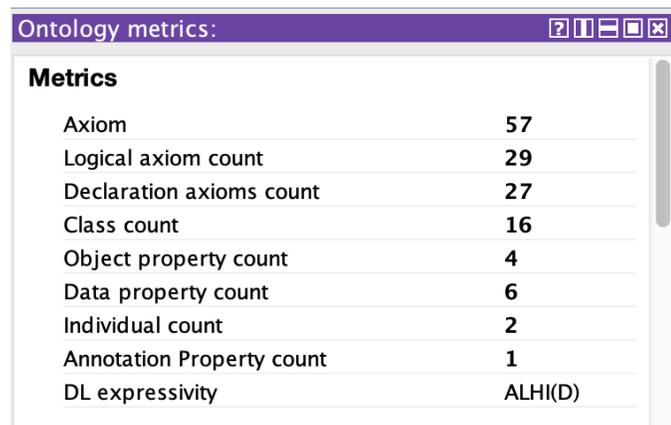
Correctos	Incorrectos
19	1

Los resultados de dicha prueba arrojan que al realizar 20 consultas 1 de ellas presentaba inconsistencias en el vector de los hospitales mostrados, es decir el resultado de dicha consulta no era completamente acorde a los parámetros iniciales.

Segunda Prueba

Con esta prueba se procede a medir el tiempo de búsqueda en la ontología, para ello se emplean los mismos parámetros de búsqueda que la prueba anterior. El tiempo se determina desde Protégé.

Dentro de la herramienta Protégé se proporciona la característica de medir el tiempo de búsqueda en una ontología dentro del menú de herramientas.



Ontology metrics:	
Metrics	
Axiom	57
Logical axiom count	29
Declaration axioms count	27
Class count	16
Object property count	4
Data property count	6
Individual count	2
Annotation Property count	1
DL expressivity	ALHI(D)

Figura 6.10 Métricas en la ontología

Este tiempo de respuesta esta directamente relacionado con la cantidad de axiomas (57) y reglas que presenta la ontología (29), las clases que están involucradas (16) y el numero de declaraciones que permiten el correcto funcionamiento (27). Una vez mencionado lo anterior se obtuvo un promedio de 25 consultas realizadas, el cual arrojó un tiempo de respuesta de 10ms.

Tercera Prueba

Esta prueba se enfoca en verificar la correcta visualización de los servicios médicos en el mapa. Esta prueba depende directamente de los resultados en la prueba 1, debido a que en primera instancia se obtiene el vector de hospitales y posteriormente se procede a ubicarlos en la aplicación móvil. Una vez aclarado lo anterior se realizaron distintas pruebas con distintos parámetros (seguro medico y especialidad).

- Seguro medico: IMSS
- Especialidad: Cardiología

Correctos	Incorrectos
15	0

- Seguro medico: IMSS
- Especialidad: Traumatología

Correctos	Incorrectos
14	1

- Seguro medico: ISSSTE
- Especialidad: Medicina Familiar

Correctos	Incorrectos
15	0

- Seguro medico: ISSSTE
- Especialidad: Neurología

Correctos	Incorrectos
15	0

Se puede observar que dentro de los distintos experimentos en esta prueba solo se presentó un error al momento de visualizar correctamente el marcador en el mapa, esto pudo haber ocurrido debido a que las coordenadas eran incorrectas.

CAPÍTULO VII. ANÁLISIS Y
DISCUSIÓN DE RESULTADOS



Los retos que se persiguen en la actualidad van encaminados hacia la búsqueda de métodos que permitan integrar datos de diversas fuentes de información de una manera coherente y que esta resulte útil en un propósito específico. En ese sentido existen trabajos significativos sobre integradores de información cuyas características van acotadas a trabajar con las fuentes originales de datos, evitando así la duplicidad.

Es por ello que el modelo aquí expuesto propone el uso de una arquitectura ontológica como capa intermedia transparente para el usuario para el manejo y consulta de información en sus fuentes originales. El modelo propuesto es el resultado de un análisis y comparación con diversas arquitecturas existentes una de ella es el siguiente modelo presentado por C. Allocca en el año 2009 [56].

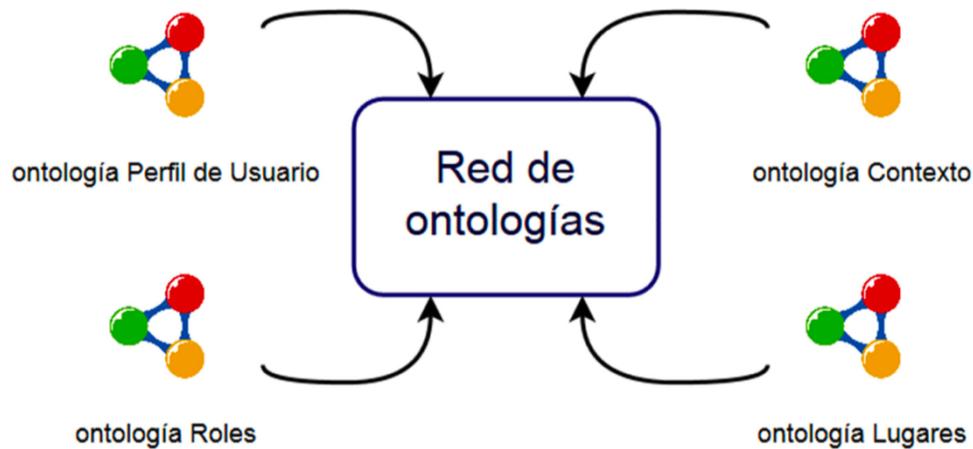


Figura 7.1 Red ontológica

La figura 7.1 contempla la creación de una red de ontologías (colección de ontologías) las cuales se relacionan a través de distintos mecanismos tales como un mapeo, una modularización y un control de versiones. De manera particular esta colección se encuentra en un formato RDF con el fin de que las consultas puedan realizarse bajo un estándar.

Estas ontologías poseen los conceptos, relaciones, así como los tipos de datos, pero estos no poseen ninguna instancia, lo cual obliga a esta colección a realizar una consulta externa para obtener dichas instancias.

Como consecuencia el mantener almacenadas las instancias en un apartado diferente conlleva un retraso en las consultas realizadas, según este autor el tiempo de respuesta se determina según el lugar en donde se encuentre la instancia original, teniendo como promedio un tiempo de respuesta de 5 segundos. En comparativa con el modelo propuesto en este trabajo las pruebas realizadas arrojan que el tiempo de

respuesta en promedio es de 10 ms, este tiempo varía dependiendo del tipo de consulta realizada y la cantidad de tuplas que esta devuelva.

Por tanto, se puede concluir que tener almacenadas las instancias dentro de la misma ontología reduce considerablemente el tiempo de consulta. No obstante, el planteamiento de esta red de ontologías aporta al modelo propuesto ver a las ontologías como un conjunto y no como una entidad única.

A continuación, se presenta el análisis de la información después de haber llevado a cabo la metodología descrita en el capítulo anterior.

Determinar las fuentes de información

La etapa inicial de la metodología consiste en la selección de las diversas fuentes a integrar, sin embargo, es importante mencionar que dicha selección se realizó de acuerdo con un análisis previo, en donde para dicha selección se realizó de manera manual.

Si bien algunos trabajos ya realizan esta tarea de manera automática, este modelo no lo contempla, a pesar de ello existen ciertas ventajas al realizar dicho proceso de esta forma:

- Se discriminó fuentes de información que no pertenecían al dominio semántico.
- Se redujo el margen de error en los datos obtenidos.

El punto anterior se obtuvo a través de la comparación directa con un trabajo enfocado directamente en la selección automática de las fuentes de datos [17], dicho trabajo se enfoca únicamente en determinar que fuentes pertenecen a una misma semántica, si bien este esquema representa una reducción en el tiempo que se requiere para realizar la tarea de la selección, presenta el inconveniente que si se desea obtener resultados mas precisos se debe de realizar una validación manual del resultado final que arroja el sistema.

Haciendo una comparación con el trabajo presente se obtiene que las fuentes seleccionadas no requieren pasar por una segunda validación por parte del usuario, con lo cual la siguiente etapa que implica la parte medular del proceso de integración arrojó resultados con mayor precisión.

Proceso de integración

En esta sección como se plantea en el capítulo 4 se procede a analizar las fuentes que se seleccionaron en el punto anterior lo cual implicó la definición formal de las ontologías que representan dicha información.

A partir de aquí se encontraron diversos trabajos los cuales contemplaron una alineación automática para la creación de las ontologías, dentro de los cuales destaca el siguiente [22] por la similitud con los resultados obtenidos, tomando como consideración que el desarrollo del modelo propuesto requiere de la creación manual de las ontologías.

MatchBoost [22] contempla la consulta a través de matrices de similitud los resultados obtenidos por cada fuente en este sentido, MatchBoost recorre la jerarquía de clases de cada ontología, y asigna un número identificador a cada par de clases posible. Cada matriz contiene la similitud de doscientas parejas posibles de clases. Las columnas representan los nombres de todas las clases de la ontología A, mientras que las filas contienen los nombres de la clase de la ontología B. Sin embargo, la debilidad que presenta trabajar con esquemas que construyan de manera automática es el margen de error que constituye la integración semántica y léxica dentro de los esquemas de bases de datos.

A pesar de existir valores de similitud léxica bajos, no significa que no existen correspondencias entre las clases que representan, siempre y cuando se realicen otros procesos para medir su similitud semántica como los siguientes. Tomando en cuenta los aspectos anteriores y precisamente para reducir el margen de error que presentaban otras investigaciones, se optó por la creación manual de las ontologías que constituyen a la arquitectura propuesta, debido a esto es que se escogió la herramienta Protégé.

Un factor importante para la construcción de la ontología es la herramienta con la que estas se construyeron, si bien en capítulos anteriores se menciona que la herramienta empleada es Protégé se puede tener un análisis de las ventajas que esta herramienta nos aporta:

- Es una herramienta multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Amplia documentación.
- Software libre.
- Filosofía orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia.

Como resultado de la utilización de esta herramienta se obtuvo como resultado la siguiente ontología:

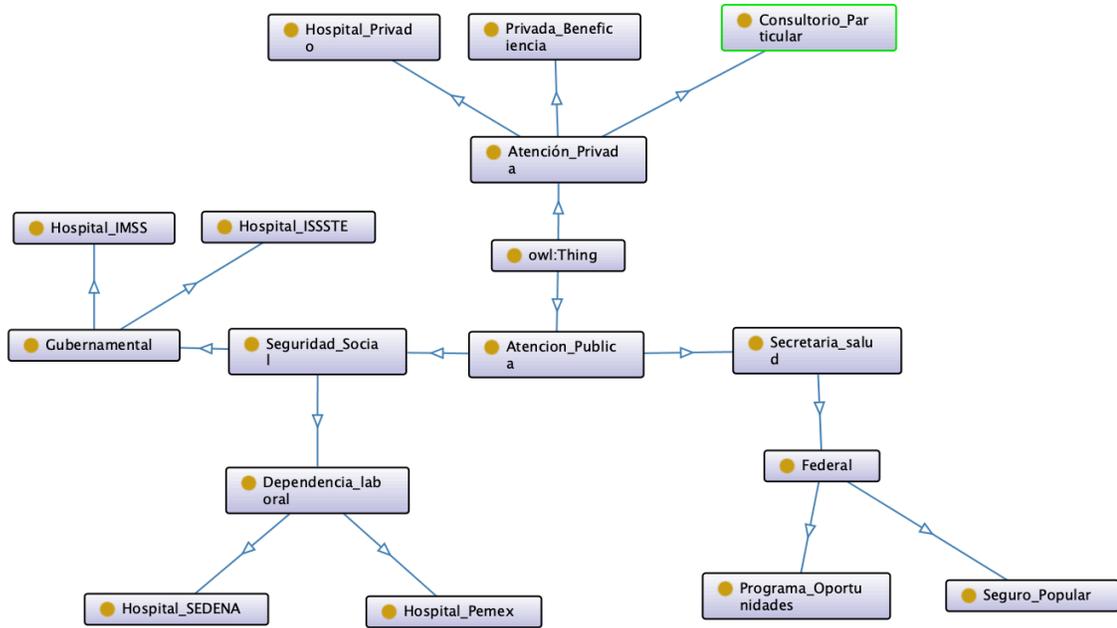


Figura 7.2 Ontología servicios médicos

De la antología anterior se puede destacar los siguiente:

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 servicios médicos. En otras palabras, las fuentes de datos contienen información acerca de ubicaciones geográficas y la especialidad que es atendida en cada institución.

Posteriormente con la aplicación Protégé se obtiene un diagrama que determina las características fundamentales de cada institución figura 6.5. Dicho grafo se empleó como base para cada una de las instancias dentro de la ontología.

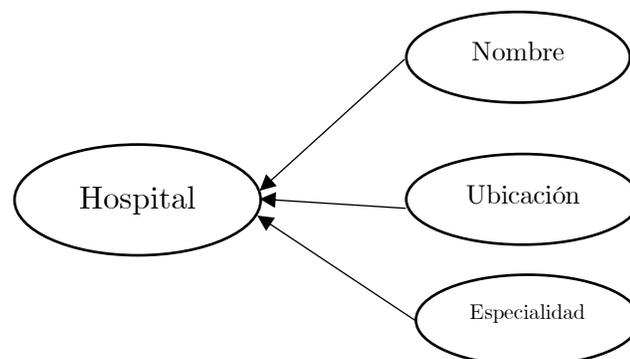


Figura 7.3 Grafo hospital

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 servicios médicos.

Con el uso de la plataforma D2RQ [32] se generan diferentes mapeos los cuales son descritos a continuación. Además con esta framework se puede obtener el archivo de alineación en diferentes formatos.

El proceso para generar la alineación con este framework es el siguiente:

- Se leen 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 [26], en particular nos interesan la distancia de Levenshtein y SMOA para la parte léxica y la similitud de propiedades para la parte semántica.

En la figura 6.6 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.

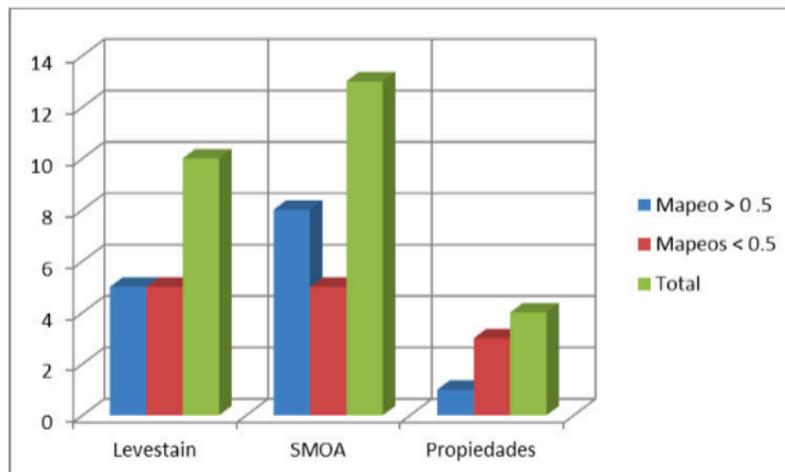


Figura 7.4 Comparación de mapeos

Realizar búsquedas en la ontología

Una vez creadas nuestras ontologías es necesario acceder al contenido de esta, 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 construyeron 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 interfaz 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 interfaz a través de la cual se implementa el Modelo OWL sobre el cual se hicieron 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 podrá dar la información que se requiera específicamente. Posteriormente si se requiere, estas consultas podrían darnos información que no este explícitamente descrita en la ontología (inferencia).

El siguiente fragmento de código es la consulta SPARQL empleada:

```
String querySPARQL =
    "PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# +
    "PREFIX mapeo.n3:
    <http://www.serviciosmedicos.com/ontologia/ssmedicos.owl#>" +
    "PREFIX owl:<http://www.w3.org/2002/07/owl#>" +
    "PREFIX owl2xml:<http://www.w3.org/2006/12/owl2-xml#>" +
    "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>" +
    "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>" +
    "SELECT DISTINCT * WHERE"+
    "{ "+
    "    ?Hospital result:nombre ?Nombre. "+
    "    ?Hospital result:especialidad ?Especialidad. "+
    "    ?Hospital result:direccion ?Direccion. "+
    "    ?Hospital result:altitud ?Altitud. "+
    "    ?Hospital result:longitud ?Longitud. "+
    "    FILTER ( ?especialidad = "+Especialidad+")" + "};
```

Se puede notar que comparte cierta similitud con una consulta en el formato tradicional SQL, sin embargo, existen limitantes con SQL al momento de que se intenta acceder en una misma consulta a distintas fuentes, por lo cual se optó por descartar este formato y en su lugar emplear a SPARQL.

Del estudio realizado se puede concluir que los lenguajes de consultas sobre plataformas RDF presentan fortalezas en el manejo y manipulación de los datos recuperados con base en los operadores aritméticos y las expresiones condicionales, así mismo en el trabajo con cadena de caracteres, el ordenamiento y secuencia de los mismos.

De este lenguaje SQL el cual realiza consultas sobre documentos RDF, su ventaja es que se complementan efectivamente con los modelos RDF, sin embargo, no permiten hacer consultas de tipo semántico, en el sentido de que no tienen porqué basarse necesariamente en elementos (conceptos, atributos y relaciones) de una ontología, sino exclusivamente en el modelo RDF.

Esta comparativa se aprecia en la siguiente tabla, en donde se consideraron otros aspectos y otros lenguajes de consulta.

Tabla 6.3 Comparativa lenguaje de consulta

Lenguaje/ Criterios	EC	CE	OM	CO	RD	RA	PR	DR	RF
XQUERY	+	+	+	-	-	-	-	-	-
XQL	+	+	+	-	-	-	-	-	-
SPARQL	+	-	+	+	+	+	+	+	+
RQL	+	-	+	-	+	-	+	-	-
seRQL	+	-	+	+	+	-	-	+	-
OWL - QL	-	-	-	+	+	+	+	+	+

EC = Expresiones Condicionales.

CE = Cuantificadores Existenciales.

OM = Operadores Matemáticos.

CO = Clases y Objetos.

RD = Rango y Dominio.

RA = Recursos Adyacentes.

PR = Predicados sobre recursos.

DR = Distancia entre recursos.

RF = Resource Framework.

Con dicha tabla se aprecia las ventajas aportadas por el lenguaje que se empleó SPARQL, no se descarta la posibilidad de que el sistema propuesto pueda trabajar con alguno diferente.

Ahora bien, una vez determinado los aspectos importantes en cuanto a la consulta y el modelado de la ontología, la aplicación móvil desarrollada arroja importantes resultados en comparación con otras aplicaciones similares. La principal comparativa fue con Google Maps detallando lo siguiente:

En la Figura 6.7 se muestra un ejemplo de una búsqueda de hospitales realizada en Google Maps. Los campos a modificar por un usuario son la barra de búsqueda para ingresar el sitio a localizar, el punto origen y el punto destino de una ruta a calcular ya sea en algún tipo de transporte o caminando.

Haciendo una comparación entre Google maps y la aplicación desarrollada se aprecia que la primera muestra de manera general los diversos hospitales, hospital de celulares, hospital de mascotas, produciendo así un tiempo de retraso al momento de que el usuario seleccione el hospital que realmente esta buscando de acuerdo con su tipo de seguro y diversas características.

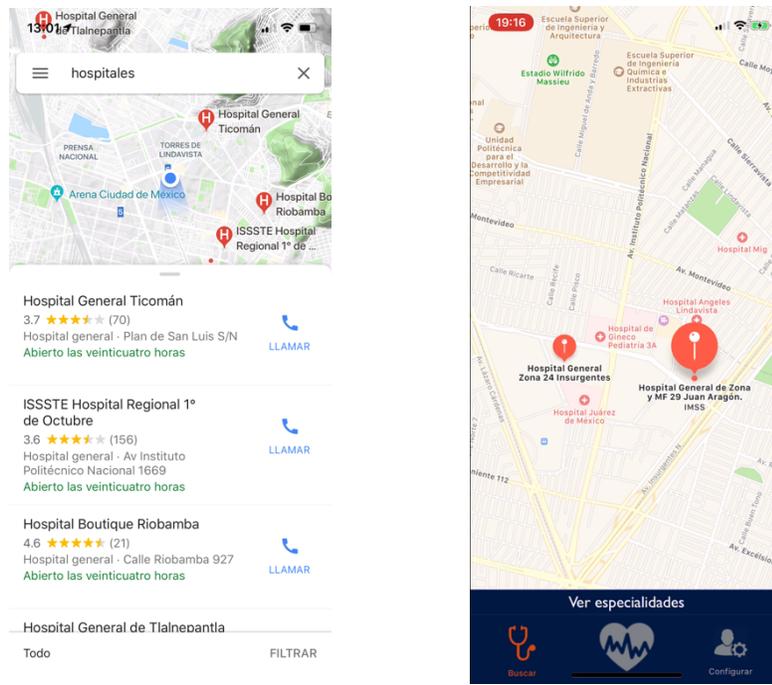


Figura 6.7 Comparación con MapGO

Como se observa la información de manera personalizada de acuerdo con el tipo de usuario y las características que lo definen trae consigo una optimización y reducir el tiempo empleado en situación críticas.

Por otro lado, aplicaciones basadas en servicios de localización como Google Maps resultan muy útiles a los usuarios al permitirles ubicar distintos puntos de interés. Dentro de las ventajas proporcionadas por estas aplicaciones destaca la de Google Maps, gracias a que en sus últimas versiones muestra la calificación en cuanto a la calidad de servicio que tiene algún punto de interés buscado sin embargo para un dominio en específico es mejor contar con aplicaciones echas a la medida y resultados personalizada de acuerdo con la situación, estos puntos serán abordados mas adelante dentro de las conclusiones del presente trabajo.

7.2 Comparación con otros buscadores

En este apartado se realizó una comparación con otras aplicaciones que se enfocan precisamente en la localización de distintos sitios de interés, es por ello Durante el desarrollo del presente trabajo de investigación se revisaron diversos trabajos que involucran aspectos relacionados con el tema de tesis propuesto. De entre ellos, destacan la creación y uso de ontologías, el empleo de aplicaciones móviles, servicios de ubicación y el manejo de operaciones espaciales.

Soto, J. [57] presenta el desarrollo de una aplicación móvil que hace uso de la plataforma de geolocalización junto con el protocolo de comunicación de información clínica HL7 (Health Level Seven) y Sistemas de Posicionamiento Global (GPS), la cual realiza el cálculo de la distancia y la localización de centros médicos tomando la ubicación geográfica exacta del dispositivo móvil del usuario.

Dicha aplicación cobra gran importancia al hacer uso del Sistema de Posicionamiento Global y realizar el cálculo de la ubicación exacta del dispositivo que está haciendo uso de la aplicación a través del método de trilateración. Esta tecnología, al estar incluida en la mayoría de los dispositivos móviles del mercado, la hace aún más factible para ser usada.

Por otro lado, el uso de bases de datos espaciales facilita el desarrollo de operaciones espaciales para el cálculo de distancias entre la ubicación del origen y la del destino. Es importante resaltar que al no hacer uso de ontologías, la información reside en una base de datos, específicamente sobre el gestor PostgreSQL, favoreciendo la integración con el módulo de PostGIS para el manejo de datos geográficos. El modelado de consultas planteado por radio de cobertura, categoría y servicios garantiza que el usuario recibirá

los resultados más aproximados y confiables respecto a su ubicación y especialidad deseada.

Otro trabajo que emplea el estándar HL7 para el intercambio de información médica, es el expuesto por Orgun B. y Vu J. [44] que desarrolla un Sistema Electrónico de Agentes Médicos eMAGS, el cual trabaja con una ontología basada en el estándar de mensajes de salud HL7 y múltiples agentes, para facilitar la interacción de los Sistemas Médicos Distribuidos evitando las limitaciones que ofrece una arquitectura cliente-servidor.

En cuanto a aplicaciones que dan seguimiento y control médico a condiciones crónicas en los pacientes, Paganelli, F. y Giuli D. [25] presentan una tecnología basada en una plataforma de servicios sensibles al contexto haciendo uso del Modelo de Cuidado Crónico (CCM, por sus siglas en inglés).

En este caso, el uso de cuatro ontologías, tres de dominio (paciente, hogar, administración de alarmas) y una de contexto (social) facilita la recuperación de información consistente al permitir una mayor especificación de la consulta sobre la situación presentada en el paciente. Por otro lado, el prototipo de emisión y recepción de alarmas a través de niveles controla de manera eficiente el tipo de atención y velocidad de respuesta proporcionada.

Sin embargo, la tecnología móvil solo hace uso del envío de mensajes SMS o mails a los doctores por medio de un servidor, por lo que no se explotan completamente todas las características ofrecidas por los dispositivos móviles.

Otra aplicación para dispositivos móviles que hace uso de información médica es ICE [51], la cual ofrece el servicio de llamadas a los contactos de emergencia (incluido el médico personal) y proporciona información médica referente al usuario. El uso de esta aplicación resulta útil gracias al manejo de la información imprescindible de un usuario, así como también la facilidad de uso de la aplicación por parte de otra persona en caso de que el propietario del dispositivo este incapacitado para hacerlo. No obstante, esta no proporciona la orientación necesaria para saber a qué hospital o centro médico dirigirse para recibir atención.

CAPÍTULO VIII. CONCLUSIONES



El trabajo desarrollado ofrece una integración de distintas fuentes de información, fueron tomados en cuenta aquellas fuentes que presentan heterogeneidad entre si, pero que sin embargo comparten un mismo dominio.

Es por ello que este trabajo se apoya en el uso de ontologías lo cual ha cambiado y simplificado en gran medida la forma de cómo plantearse el problema de la integración de datos al punto que puede incluso decirse se ha convertido en el nuevo paradigma de la integración de fuentes de datos heterogéneas.

A su vez la información no ha sido aprovechada de una manera importante, ya que las aplicaciones que nos informan la ubicación de hospitales no son de acuerdo con las características de cada usuario.

8.1 Definir un modelo a capas empleando una metodología con base en ontologías

En el presente trabajo se ha descrito el desarrollo de un modelo a capas que permita la integración de diferentes fuentes de datos, concretamente en la propuesta del modelo se abordó el desarrollo de la capa integradora de información. Para tal efecto esta capa se constituyó por una Arquitectura ontológica, en donde justo esta arquitectura es la parte medular y la aportación mas importante de este trabajo. No obstante, la conceptualización y abstracción de esta arquitectura se ve reflejada en una metodología propuesta, en la cual el sistema de consultas recae en un modelo ontológico.

8.2 Definir las capas que conforman al modelo propuesto

En este escenario, el verdadero reto en la actualidad en la integración de diversas fuentes de datos va mas allá de la implicación de la distribución de estas, de el protocolo de comunicación e incluso de el formato con el que se almacenan los datos, esto va enfocado a la conciliación de los protocolos, en la conceptualización que cada fuente tiene acerca de un mismo dominio es decir la semántica e interpretación que esta otorga a los datos.

El trabajo presente aborda una metodología propuesta la cual integra información de diferentes fuentes de datos heterogéneas entre si mediante el mapeo de estas, a su vez se apoya de consultas SPARQL hacia una ontología de dominio creada previamente, en donde las ontologías resultan una herramienta clave para el funcionamiento de esta metodología propuesta.

8.3 Desarrollar la capa de integración de información

De esta manera, a través del desarrollo de una metodología quedan cubiertos dos de los principales objetivos planteados al inicio, los cuales incluyen la definición de las capas del modelo propuesto y el desarrollo de la capa de integración de este, No

obstante, el tercer objetivo el cual es la creación de la ontología tiene estrecha relación con este segundo y con el ultimo objetivo, debido a esto al culminar dicha ontología se llegó a una importante conclusión en donde se destaca lo siguiente

- Se debe tener en claro que es lo que se quiere integrar y cual es el fin de esta integración, para posteriormente determinar con mayor facilidad que fuentes son las que van a conformar a la ontología.

8.4 Crear las ontologías correspondientes con base en el estándar OWL

La aplicación de ontologías ha cambiando y simplificado en gran medida la forma de cómo plantearse el problema de la integración de datos, sin embargo se debe tener claro el objetivo que se persigue cuando se pretende realizar una integración que este basada únicamente con ontologías y ser consiente de que aspectos o características del dominio deben de ser accesibles y a su vez saber si estos se encuentran representados o no en el modelo conceptual de las fuentes de datos originales a integrar.

8.5 Diseñar un caso de estudio que implemente el modelo propuesto

Otro de los alcances realizados es el planteamiento de un caso de estudio en el cual se abstrae la conceptualización y aplicación del modelo propuesto. Es aquí donde toma vital importancia haber respondido la pregunta que se planteó en el punto anterior ya que con ello se puede eliminar la ambigüedad de solamente decir que se quiere integrar información de diversas fuentes.

Para el desarrollo del caso de estudio se trabajó a través de una metodología (Methontology) para la construcción de una ontología de aplicación en el dominio de servicios médicos a partir de la delimitación de estos en la delegación Gustavo A. Madero. Dicha ontología se elaboró bajo el lenguaje OWL (Ontology Web Language) el cual es un estándar en la Web semántica, así mismo los datos que conforman dicha ontología fueron obtenidos de los portales de la Secretaria de Salud, teniendo como resultado información real y confiable. En consecuencia, se espera que futuros trabajos incluso en diferentes líneas de investigación puedan hacer uso de esta ontología de servicios médicos, favoreciendo así el trabajo cooperativo e interdisciplinario entre distintas instituciones y estudios de investigación.

Adicionalmente trabajar con el framework nativo de Apple (Map Kit) para la visualización de los puntos de interés geográficos facilitó la tarea de agregar marcadores y calcular la ruta más optima en comparación con la API de Google Maps definiendo así de una manera mas precisa la ubicación del usuario y el proceso de visualización del conjunto de hospitales en su ubicación geográfica.

No obstante haber cubierto a buen termino la conclusión de cada uno de los objetivos, restan algunos elementos de la aplicación susceptibles a ser optimizados, los cuales se presentan más adelante como trabajo futuro.

Finalmente, el diseño del modelo propuesto en conjunto con las operaciones de búsqueda implementadas hacia la ontología permitió la obtención de resultados mucho mas precisos y lo mas importante personalizados de acuerdo con cada situación y tipo de usuario.

Limitaciones

Pese a que se cumplieron todos los objetivos planteados, existen ciertas limitaciones listadas a continuación:

- No existe un banco único de datos en México el cual contenga la descripción, las especialidades, así como la ubicación geográfica de los servicios médicos existentes, por lo cual el proceso de agregar dichas entidades a la ontología se realiza de manera manual.
- La rapidez con la que se realiza el procesado de búsqueda de servicios médicos esta acotada a la capacidad de procesamiento del dispositivo móvil, por lo que si la ontología crece en numero de entidades la velocidad de respuesta será inferior.
- La ontología desarrollada únicamente contempla servicios médicos ubicados en la delegación Gustavo A. Madero, de tal forma que si el usuario sale de esta delimitación geográfica no podrá obtener resultados.

Trabajo Futuro

Las líneas de investigación para dar una continuidad a este trabajo parten de la pretensión de lograr una integración a mayor escala que la alcanzada hasta este momento, por ello se propone abordar los siguientes rubros:

- La creación de nuevas ontologías se realice de manera automática o bien semiautomática, generando así los archivos correspondientes en formato OWL. A partir de ese punto, se puede argumentar que el modelo es fácilmente escalable desde el punto de vista de la interacción con el usuario.
- El número de instancias contenidas en la ontología de servicios médicos está acotado a aquellas existentes para el caso de estudio; sin embargo, la aplicación

móvil queda abierta a la actualización y crecimiento con la introducción de más instancias a través de la ontología original.

- Agregar un mayor número de características y reglas de inferencia a la ontología, generando con ello un razonador capaz de incrementar la precisión de búsquedas más complicadas.
- En el apartado de la aplicación móvil ofrecer al usuario el poder tener su historial médico y un recordatorio con consultas y recomendaciones por parte del médico.

Bibliografía

- [1] I. Barona, “La descentralización de la información en Finanzas”, *El Universal*, pp. 78-79, 2016.
- [2] H. O. Unver, “An ISA-95-based manufacturing intelligence system in support of lean initiatives”, *International Journal of Advanced Manufacturing Technology*, vol. 65, pp. 853-866, 2013.
- [3] B. Amann, C. Beeri, I. Fundulaki y M. Scholl. “Ontology based integration of xml web resources”, International Semantic Web Conference, Monterey, CA, EUA, 2002, pp. 117–131.
- [4] M. Barcellos, "Revisiting Ontologies: a necessary clarification", *Journal of the American Society for Information Science and Technology*, vol. 64, no. 8, pp. 1682-1693, Julio 2013.
- [5] W. Mark, “The origins of ubiquitous computing research at PARC in the late 1980s”, *IBM Systems Journal*, vol. 38, no 4, pp. 693-696, 1999.
- [6] A. Cechich, A. Nieves, “Ontology based data integration methods: a framework for comparison”, *Revista Colombiana de Computación*, vol. 6, no. 1, pp. 234-275, 2005.
- [7] C. Britton, *IT Architectures and Middleware: Strategies for Building Large, Integrated Systems*, 2a edición, Pearson Education, 2004.
- [8] A.S. Dexter, “Electronic data interchange and small organizations: adoption and impact of technology”, *Management Information Systems Quarterly*, vol. 19, no. 4, pp. 465-485, Diciembre 1995.
- [9] R.J. Miller, Y.E. Ioannidis, y R. Ramakrishnan, “Schema equivalence in heterogeneous systems: bridging theory and practice”, *Information Systems*, vol. 19, no. 1, pp. 3-31, Enero 1994.
- [10] M.F. Worboys y M. Duckham. *GIS: A Computing Perspective*, 2a edición, Taylor & Francis Group, 2004.
- [11] G. Smyth, “From Data Mining to Knowledge Discovery in databases”. *AI Magazine*, vol. 23, no. 5, 1996.

- [12] A. Botello, “Aplicando técnicas de correspondencia semántica para la integración de bases de datos dispersas”, Tesis de maestría, IPN, CIC, CDMX, Oct 2011.
- [13] M. Kokla. y M. Kavouras, “A Formal Method for the Semantic Integration of Classification Schemata”, 3ra Conference on Geographic Information Science, Finlandia, Mayo 2000.
- [14] C.J. Date, *Introducción a los Sistemas de Bases de Datos*, editorial Pearson Educación, 2001.
- [15] I. Cruz y H. Xiao, “The role of ontologies in data integration”, *Science*, vol. 21, no. 3, 2005.
- [16] H.L Roger, C. Veda, “Framework and knowledge for database integration. Managing information technology resources and applications in thse world economy”, *Storey*, pp. 218, 1997.
- [17] D. Rey, “Un modelo de integración y preprocesamiento de información distribuida”, Tesis doctoral, Universidad Politécnica de Madrid, Nov 2007.
- [18] J. Henriques y P. Tomé, “Ubiquitous Integration Architectural Issues”, Tesis doctoral, Instituto Politécnico de Viseu, Portugal, 2012.
- [19] C. Britton, *IT Architectures and Middleware: Strategies for Building Large, Integrated Systems*, 2a edición. Pearson Education, 2004.
- [20] A. Monge, y C. Elkan, “The field matching problem: Algorithms and applications”, Berkeley, 2a Int. Conf. Knowledge Discovery and Data Mining. 1996.
- [21] C. Batini, M. Lenzerini, “A comparative analysis of methodologies for database schema integration”, *ACM Computing Surveys* vol. 18, no. 4, Diciembre 2016.
- [22] M. I. Martínez, “Alineación de ontologías para la integración de fuentes de datos”, Tesis de maestría, IPN, CIC, CDMX, 2012.
- [23] A. Doan y Y. Halevy. “Semantic integration research in the database community: A brief survey”. *Database Science*, Toronto, 2005.

- [24] T. Berners-Lee, J. Hendler, “The semantic web”. *Scientific american*, Berkeley CA, 2001, pp. 284- 288.
- [25] J. Jones, (2011, Mayo), Resource Description Framework (RDF): concepts and abstract syntax. World Wide Web Consortium Std. En línea <http://www.w3.org/TR/rdf-concepts>.
- [26] Web Ontology Language. World Wide Web Consortium Std. En línea <http://www.w3.org/2004/OWL/>.
- [27] SPARQL Query Language for RDF. World Wide Web Consortium Std. En línea <http://www.w3.org/TR/rdf-sparql-query/>.
- [28] J. Pacheco, H. Najera, K. Estrada, “MDE Framework for semi-automatic Development of Web Applications”. In proceedings of the 1st International Conference on Model-Driven Engineering and Software Development, 2013.
- [29] J. Pacheco, H. Najera, K. Estrada, “Semantic Web Builder: A Framework for Semantic Web Applications Development”, Workshop on Semantic Web and Linked Open Data, 2016.
- [30] Web semántica: facilitar la búsqueda e integración de información. En línea <https://s3uswest2.amazonaws.com/centrosconacyt/wpcontent/uploads/2015/04/Infografia-Web-Sem--ntica.jpg>.
- [31] H. Stuckenschmidt, “Enabling Technologies for Interoperability”, Tesis doctoral Center for Computing Technologies University of Bremen, 2000.
- [32] 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”, *IJCAI-01 workshop: ontologies and information sharing, volume*, pp. 108–117. Citeseer, 2001.
- [33] R. Bayardo, J. Bohrer, W. Brice, R. Cichochi, A. Fowler, J. Helal, A. Kashyap, V. Ksiezk, T. Martin, G. Nodine, M. Rashid, M. Rusinkiewicz, M. Shea, R. Unnikrishnan, C. Unruh, A. “Infosleuth: Agent- based semantic integration of information in open and dynamic”, *Science Environments*, 2017.
- [34] P. Cannata, P. Huhns, M. Jacobs, N. Ksiezzyk, T. Ong, K. Sheth, P., Tomlinson, C. Woelk, “The Carnot heterogeneous database project: Implemented applications”, *Distributed and Paralleled Data- bases*, 2015.

- [35] P. Visser, M. Beer, M. Bench, T. Diaz, B. Shave, “Resolving ontological heterogeneity in the KRAFT Project”, In 10th International Conference and Workshop on Database and Expert Systems Applications DEXA’99, Universidad de Florencia, Italia, 2009.
- [36] D. Jones, “Developing shared ontologies in multi-agent systems”. In ECAI 2008 Workshop on Intelligent Information Integration, Brighton, U.K, 2008.
- [37] C. Goh, H. Bressan, S. Madnick, “Context interchange: New features and formalisms for the intelligent integration of information”, ACM Transaction on Information Systems, Universidad de Valencia, 2011.
- [38] E. Mena, V. Kashyap, *Domain specific ontologies for semantic information brokering on the global information infrastructure*. In Guarino, no. 5, editor, Formal Ontology in Information Systems, 2016.
- [39] A. Aransay, “Diseño de interacción centrada en el usuario”, Tesis doctoral Universidad de Vigo, 2009.
- [40] F. Hakimpour, “Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata”, Zürich, Universidad de Irán, 2013.
- [41] A. Halevy, N. Ashish, D. Bitton, J. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka, “Enterprise information integration: successes, challenges and controversies” , SIGMOD Conference, pp. 778–787, 2005.
- [42] “Fundamentos De Bases De Datos”, Laboratorios, S. Sudarshan, Instituto Indio de Tecnología, Bombay, 2015.
- [43] P. Ziegler y R. Dittrich, “Database Technology Research Group”, Department of Informatics, Universidad de Zurich, Record, pp. 53–58, 2005.
- [44] M. Ehrig, *Ontology alignment: bridging the semantic gap*, Editorial Semantic web and beyond. Springer, 2007.
- [45] T.R. Gruber et al. “A translation approach to portable ontology specifications. Knowledge acquisition”, Tesis doctoral, Universidad de Stanford, pp. 199–220, 1993.
- [46] R. Neches, “Enabling technology for knowledge sharing”, AI Magazine 12, 1991.

- [47] N. Guarino. “Formal ontology in information systems”, The first international conference (FOIS’98), June 6-8, Trento, Italy, vol. 46. Ios Press Inc, 1998.
- [48] W. Borst, N. Akkermans, J. Revista Avances en Sistemas e Informática, Vol.7 No.1, marzo de 2010.
- [49] P. Haase, J. Broekstra, A. Eberhart, y R. Volz. “A comparison of rdf query languages”, The Semantic Web ISWC 2004, vol. 3298 of Lecture Notes in Computer Science, pp. 502–517. Springer Berlin Heidelberg, 2004.
- [50] G. Pérez, A. Corcho, O. Fernández, “Ontological Engineering: With Examples from the Areas of Knowledge Management”, E-Commerce and Semantic Web, Advanced Information and Knowledge Processing, Springer Verlag, 2003.
- [51] H. Mohamed, Y. Jincai, y J. “Qian, Towards integration rules of mapping from relational databases to semantic web ontology”. In Web Information Systems and Mining (WISM), International Conference on, vol. 1, pages 335 –339, oct. 2010.
- [52] A free, open-source ontology editor En línea en <http://protege.stanford.edu/>.
- [53] F. Hernández, “Metodología de la Investigación”. Colombia, McGraw Hill, 1994.
- [54] L. Munch, “Métodos y Técnicas de Investigación”, 4ta edición, México, Trillas, 2012.
- [55] D. Garlan y M. Shaw, “An Introduction to Software Architecture,” Advances in Software Engineering and Knowledge Engineering, Vol. 1, World Scientific Publishing Co, 1993.
- [56] C. Allocca, C. Bekiari, Y. Marketakis, *Ontology-based Integration of Heterogeneous and Distributed Information of the Marine Domain*, ERCIM News 2014, Special theme: Linked Open Data, Enero 2014.