

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

**Ontología genérica para la integración de servicios web
semánticos
Reporte Final**

José Angel Sánchez Revelo
208335914

Trimestre 2014 Primavera

Maricela Claudia Bravo Contreras
Profesor Asociado
Departamento de Sistemas

Agosto de 2014

1 Resumen

Durante el desarrollo de este informe final, se detallará cada uno de los pasos que se siguieron para desarrollar el proyecto terminal titulado “**Ontología genérica para la integración de servicios web semánticos**”, el cual estuvo asesorado y supervisado por la Dra. Maricela Claudia Bravo Contreras y se desarrolló en un periodo total de un año.

Este proyecto terminal se dividió en 2 fases esenciales para su desarrollo, la primera fase comprendió todo lo relacionado a la documentación necesaria para poder obtener las reglas de mapeo, así como analizar cómo están constituidos cada uno de los servicios web semánticos que fueron seleccionados (WSDL 1.X, WSDL 2.0 y OWL-S); dichas reglas de mapeo son las encargadas de realizar el objetivo principal planteado para este proyecto, esto al poder obtener un sistema homogéneo el cual sea compatible con los 3 servicios web mencionados anteriormente.

La segunda fase de este proyecto se enfocó principalmente en el desarrollo de la ontología que cumple con lo establecido en el objetivo principal y en dos de los tres objetivos específicos. Finalmente se desarrolló una interfaz la cual tiene como funcionalidad la comunicación eficiente entre la ontología y el usuario final.

Índice

1	Resumen	1
2	Introducción	3
3	Antecedentes	5
4	Justificación	7
5	Objetivos	10
5.1	Objetivo General	10
5.2	Objetivos Específicos	10
6	Marco Teórico	10
7	Desarrollo del proyecto	11
7.1	Diseño de la Ontología Genérica	11
7.1.1	Arquitectura General del Sistema	11
7.1.2	Generación de las Reglas de Mapeo	12
7.2	Desarrollo de la Ontología Genérica	16
7.2.1	Reglas para nombrar a los individuos	28
7.3	Desarrollo de la Interfaz Gráfica	30
8	Resultados	39
9	Conclusiones	41
10	Bibliografía	44
11	Apéndices	46
11.1	Propuestas para Proyectos Terminales futuros	46

2 Introducción

Durante décadas el uso del internet dentro de la rama de la computación ha sido fundamental, esto debido en gran parte a la evolución tan acelerada que han tenido las redes de comunicaciones, así como los recursos necesarios que se disponen para poder ser explotada adecuadamente. Estos recursos han permitido poder desarrollar herramientas que facilitan en gran medida su uso, desde exploradores de internet, redes de comunicación más veloces, etc. y esto ha dado pie a que las empresas tengan una mayor difusión y puedan dar a conocer más ampliamente los productos o servicios que ofrecen.

Dentro de todas estas herramientas disponibles, existen los denominados **servicios web**¹, los cuales han cobrado un interés especial desde la publicación de un artículo en la revista "*Scientific American Magazine*", el cual fue dado a conocer el 17 de Mayo de 2001. Esta herramienta fue impulsada por un renombrado científico de la computación norteamericano de nombre Tim Berners-Lee, apuntando a un entorno en línea idílico donde las máquinas son capaces de interpretar páginas web de la misma forma que lo pueden hacer los seres humanos. [1]

Los servicios web (Web Services en inglés) son definidos dentro del argot computacional como un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Entre los estándares manejados por este tipo de tecnologías se encuentran:

¹ También conocida como web 3.0

Web Services Protocol Stack², XML³, SOAP⁴, XML-RPC⁵ y WSDL⁶, esto por citar algunos ejemplos.

Además de eso, un concepto que está tomando mucha relevancia en las ciencias de la computación son las denominadas ontologías, las cuales han adoptado diversas connotaciones muy peculiares y un tanto alejadas de su significado filosófico original. El término y las definiciones han proliferado con gran magnitud desde principios de la década de los 90's, las cuales están basadas en la conceptualización, conjuntando una serie de conceptos y sus relaciones dentro de un mismo dominio o una misma aplicación.

Las ontologías de este tipo deben tener un lenguaje formal que pueda ser entendido por máquinas y personas. La relación que tienen las ontologías, los lenguajes formales y las aplicaciones, son descritas por el matemático e informático de origen estadounidense John F. Sowa en el año de 1999, indicando que las ontologías pueden ser vistas como herramientas útiles para algún propósito, un dominio o una aplicación en concreto. [2]

Los elementos que deben de contener las ontologías son variados, y todos tienen que estar relacionados entre sí. Estos elementos parten desde un nivel básico que corresponde a la enumeración de todos los individuos relevantes; a estos individuos citados se les puede asignar propiedades las cuales harán sus características muy específicas.

² Así se denomina al conjunto de servicios y protocolos de los servicios web.

³ XML: Acrónimo en inglés para "Extensible Markup Language" que es el formato estándar para los datos que se vayan a intercambiar

⁴ SOAP: Acrónimo en inglés para "Simple Object Access Protocol"

⁵ XML-RPC: Acrónimo en inglés para "XML Remote Procedure Call", que hace referencia a los protocolos sobre los cuales se establece el intercambio de datos.

⁶ WSDL: Acrónimo en inglés para "Web Services Description Language", el cual se refiere al lenguaje que usa la interfaz pública para los servicios web, además de ser una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web.

Posteriormente tenemos a las denominadas clases, las cuales son un conjunto de individuos donde tienen la función de ser los componentes principales de cualquier ontología; cabe mencionar que estas clases también pueden dar la posibilidad de crear subclases especializadas, las cuales heredarán la mayor parte de las características que posee cada uno de los individuos que poblarán dicho recurso (ontología).

Este tipo de herramientas inteligentes (ontologías) deben cumplir con ciertas reglas denominadas reglas de inferencia, las cuales trabajan con una serie de formalismos lógicos que deben ser especificados para que puedan laborar de manera correcta. Estas reglas son diseñadas para que puedan cumplir con funciones esenciales entre las que se encuentran:

- Tener una serie de restricciones y reglas de validación las cuales tienen a su cargo asegurarse que la ontología tenga una coherencia lógica y sobretodo que sea funcional.
- Todas deben tener una definición intencional de las clases, esto con el objeto de que puedan agruparse dos o más clases en otra clase distinta.

En términos generales este proyecto terminal se encarga de conjuntar todos los conceptos arriba mencionados, con el objetivo de que el sistema planteado pueda ayudar a minimizar el tiempo de búsqueda de un servicio web, esto gracias a las reglas mencionadas con anterioridad.

3 Antecedentes

Los Servicios Web surgieron ante una necesidad de estandarizar la comunicación entre distintas plataformas (PC,

Mainframe, Mac, etc.) y lenguajes de programación (PHP, C#, Java, etc.).

Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM⁷ y CORBA⁸, por ser dependientes de la implementación del vendedor DCOM - Microsoft, y CORBA - ORB (a pesar de que CORBA tiene múltiples vendedores pueden operar entre sí, hay ciertas limitaciones para aplicaciones de niveles más altos en los cuales se necesite seguridad o administración de transacciones).

Otro gran problema es que se hacía uso de RPC (Remote Procedure Call) para realizar la comunicación entre diferentes nodos. Esto, además de presentar ciertos problemas de seguridad, tiene la desventaja de que su implementación en un ambiente como es Internet, es casi imposible (muchos firewalls bloquean este tipo de mensajes, lo que hace prácticamente imposible a dos computadoras conectadas por Internet comunicarse).

Los Web Services surgieron para finalmente poder lograr la tan esperada comunicación entre diferentes plataformas. En la actualidad muchos sistemas legacy están pasando a ser web services. Es por

⁷ Siglas en inglés para **Distributed Component Object Model** la cual es una tecnología perteneciente a Microsoft para desarrollar componentes de software distribuidos sobre varios ordenadores y que se pueden comunicar entre sí.

⁸ Siglas en inglés para **Common Object Request Broker Architecture** el cual es un estándar definido por Object Management Group (OMG) que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diferentes computadoras, puedan trabajar juntos; es decir, facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos

esto que en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP⁹, WSDL, y UDDI¹⁰.

4 Justificación

En la actualidad muchos sistemas se desarrollan con servicios web, esto hace que los sistemas de información sean más eficientes a la hora de ser ejecutados, y esto se puede ver reflejado principalmente en los tiempos que tarda en ejecutar alguna de las acciones para los cuales son invocados, y consecuentemente con esto ayudan a usuarios y a programadores por igual a realizar ciertas operaciones para las cuales son requeridos. Estos servicios pueden realizar diferentes tipos de acciones, las cuales van desde unas simples operaciones aritméticas hasta funciones con un alto grado de complejidad y enfocados a objetivos específicos como lo es el poder ver la disponibilidad en tiempo real de vuelos dentro de una aerolínea.

Cabe destacar que el estudio de este tipo de servicios puede llegar a ser muy tedioso para los programadores, esto además de tener un fuerte impacto en cuestiones de desarrollo de software, donde el tiempo que se le puede invertir en la búsqueda de servicios que resuelvan problemas como los citados anteriormente ocasionan altos costos y los cuales representan problemas muy grandes para las empresas que llegan a contratar a personas especializadas en el desarrollo de este tipo de sistemas inteligentes.

Otro problema importante a resaltar dentro del desarrollo de este tipo de sistemas es la falta de una buena documentación de todos los servicios disponibles, esto debido a que se hace más complicado la

⁹ Siglas en inglés para **Simple Object Access Protocol**, el cual es un protocolo que sirve para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP y basado en XML.

¹⁰ Siglas en inglés para **Universal Description, Discovery and Integration**, los cuales son catálogos independientes, basados en XML, que lista los negocios de internet de todo el mundo.

invocación de los mismos, así como el estudio para averiguar la manera en que realizan las cosas y con ese fin poder adecuarlo a las necesidades que se tengan en ese preciso momento.

Dentro de este proyecto terminal, lo que se buscó fue la integración de estos dos conceptos introducidos en los párrafos de arriba (Ontología y Web Service), y por medio de una serie de reglas de mapeo específicas, poder buscar las equivalencias existentes entre los siguientes tipos de servicios web WSDL 1.0¹¹, WSDL2.0¹² y OWL-S¹³, esto con la finalidad de que los programadores de sistemas no tengan tanta complicación en cuestiones tiempo para poder realizar las operaciones de sustitución (operación que consiste en reemplazar un servicio web dañado por otro que sea funcional) que se puedan requerir en algún momento dado.

Los trabajos relacionados con este proyecto terminal están descritos en la Tabla 1, así como la descripción de su funcionamiento.

Titulo	Descripción de su funcionamiento
Sistema de recuperación de Información semántica. [3]	Este proyecto lo que hizo fue explorar y aplicar técnicas del Procesamiento del Lenguaje Natural a la recuperación de la información para diseñar un sistema de recuperación de información semántico, y es similar en la forma de utilizar la información semántica, la diferencia se encuentra en que el proyecto terminal ya concluido usó esos recursos para hacer una aplicación, que como ya se mencionó, fue usada para diseñar el sistema antes dicho, mientras que en este proyecto se usará la semántica de las ontologías para apoyar la búsqueda de soluciones.

¹¹ WSDL 1.1: Siglas en inglés para Web Services Description Language en su versión 1.1

¹² WSDL 2.0: Siglas en inglés para Web Services Description Language en su versión 2.0

¹³ OWL: Acrónimo en inglés para “Ontology Web Language”

Extracción automatizada y representación de servicios web median- te ontologías. [4]	Este proyecto terminal es similar en la cuestión de trabajar con ontologías, ya que, en este proyecto terminal extraerá los elementos más relevantes de múltiples descripciones de servicios web y estarán representados por medio de un modelo ontológico, mientras que en este proyecto se diseñara e implementará un modelo ontológico que represente cuatro diferentes tipos de servicios web.
Sistema clasificador de ontologías mediante métodos de máquinas de soporte vectorial. [5]	Este proyecto terminal tiene similitudes en la cuestión de realizar un buscador con una interfaz web, la diferencia radica en que este proyecto en desarrollo realizará búsquedas de ontologías, mientras que en este proyecto el buscador también hará consultas a servicios web que sean representados en la ontología general.
Solución de problemas de mediana dificultad utilizando composición de servicios web. [6]	Este proyecto tiene como objetivo el hacer una aplicación que resuelva una tarea solicitada por un usuario usando la técnica de composición de servicios web. Mientras que en este proyecto, se realizó una sistema de integración y mapeo de ontologías de diferentes servicios web.
Sistemas de almacenamiento semántico de textos de investigación mediante ontologías. [7]	Este proyecto es similar solo en la cuestión de trabajar con ontologías, ya que en lo demás difieren totalmente, porque mientras en uno se almacenan los textos usando estas herramientas, en este proyecto lo que se busca es que la ontología general represente todos los servicios web dependiendo del tipo que se representan.

Tabla 1: Trabajos Relacionados

5 Objetivos

5.1 Objetivo General

- Diseñar e implementar un modelo ontológico general para la integración y mapeo de ontologías de servicios web.

5.2 Objetivos Específicos

- Diseñar y construir un modelo ontológico general para la representación de las características comunes de los lenguajes de descripción de servicios web WSDL 1.1, WSDL 2.0, OWL-S y SAWSDL¹⁴.
- Diseñar e implementar un sistema de integración de ontologías de servicios web en sus versiones WSDL 1.1, WSDL 2.0, OWL-S y SAWSDL en el modelo general mediante un conjunto de reglas de mapeo.
- Diseñar e implementar un sistema de consulta general al modelo ontológico para evaluar los resultados de la integración y mapeo de ontologías de servicios web.

6 Marco Teórico

Una vez desarrollados todos los conceptos necesarios, se pueden sacar opiniones muy diversas acerca de los servicios web. Dentro de estas opiniones las más relevantes son las de computólogos de renombre como lo es Berners-Lee y Sowa; ellos defienden la postura de que los servicios web serán los componentes más importantes de cualquier sistema que se programe a futuro.

Los componentes de los servicios web son muy variados y cada uno de ellos tiene una función en específico la cual deberá realizar, y en base a este fundamento se llegó a la teoría de que si los servicios

¹⁴ SAWSDL: Acrónimo en inglés para “Semantic Annotation for WSDL”

web en general hacen operaciones muy similares, porque no tratar de hacer unas equivalencias que sean capaces de poder reconocer todas las operaciones similares.

El fundamento anterior tomó más fuerza al momento de revisar la forma en como trabajaban ciertos sistemas en los cuales están implementados los servicios web, la teoría que se plantea se cumplió en todos y cada uno de los casos que fueron examinados. Con base a los resultados obtenidos por parte de los sistemas se dedujo que podría existir una equivalencia con todas esas operaciones y esta no afectaría a otros servicios.

Tomando en consideración la forma en que trabajan las ontologías y observando que en proyectos pasados se hicieron una ontología para cada uno de los servicios web que fueron ocupados, en este proyecto se propuso construir un modelo general el cual sea capaz de reconocer los miembros de todos los servicios web descritos en la propuesta; esto con fundamentos teóricos que previamente me proporcionó mi asesora acerca de los componentes que construyen este tipo de recursos.

7 Desarrollo del proyecto

7.1 Diseño de la Ontología Genérica

7.1.1 Arquitectura General del Sistema

Se realizó un estudio minucioso acerca de la forma en cómo están conformados los servicios web los cuales fueron objeto de análisis, esto con la finalidad de saber cómo se comportan cada uno de ellos. Los resultados arrojados por este análisis son mostrados dentro de la tabla 2, esto en base a lo propuesto por Sowa en 1999 [2].

Usando estos datos como punto de entrada se empezaron a deducir las reglas que se deberían seguir para poder encontrar una equivalencia total o parcial en cada uno de estos tipos de servicios web, y los cuales quedaron representados en el modelo ontológico

genérico. El lenguaje que se usó como punto de partida para este análisis fue el OWL, esto por recomendaciones que hace el consorcio encargado de las principales tecnologías sobre la que se basa la Web, es decir, El World Wide Web Consortium (abreviado W3C por sus siglas en inglés) [8] la cual es dirigida por Tim Berners-Lee, personaje que fue mencionado con anterioridad en este reporte; otra de las razones por las que se optó por usar como punto de partida los OWL es porque actualmente es el lenguaje más utilizado para la creación de las ontologías; finalmente de ese análisis se obtuvieron los siguientes resultados los cuales quedaron plasmados en la Tabla 2.

		WSDL	OWL
Descripción Formal.		Operación	Perfil: Nombre del Servicio
Transformación de la información	Entrada	Entrada	Perfil: Tiene Entradas
	Salida	Salida	Perfil: Tiene Salidas
Cambio de Estado	Estado Anterior		Perfil: Tiene Precondiciones
	Estado Posterior		Perfil: Tiene Resultados

Tabla 2: Resultados del análisis de los servicios web

El sistema es capaz de recibir colecciones de archivos los cuales describen las funcionalidades que tiene cada uno de los servicios web que los representa. Los servicios web que son soportados en esta ontología son los que están representados por los siguientes lenguajes de descripción:

- WSDL 1.X
- WSDL 2.0
- OWL

La arquitectura de este sistema está representada en la Ilustración 1.

7.1.2 Generación de las Reglas de Mapeo

Una vez obtenido los datos anteriores, se procedió a realizar las reglas de mapeo, las cuales se establecieron como el objetivo general a desarrollar dentro de este proyecto terminal. Estas reglas se hicieron en base a las operaciones principales que realiza cada uno

de los servicios web descritos; dichas reglas dieron como resultado una ontología capaz de separar en cada una de las clases los servicios web correspondientes, así como también cada uno de los individuos que constituyen los servicios ya descritos con anterioridad.

Cabe resaltar que la Tabla 2: Resultados del análisis de los servicios web fue de mucha utilidad al momento de obtener dichas reglas, ya que en ella se encuentran bien definidas y descritas las operaciones que realiza cada uno de los servicios web; en base a la función que realiza cada una estas operaciones se obtuvieron unas equivalencias, las cuales no son otra cosa que las mismas acciones, pero especificadas de manera distinta en cada uno de los servicios web que se usaron para la realización de este proyecto final.

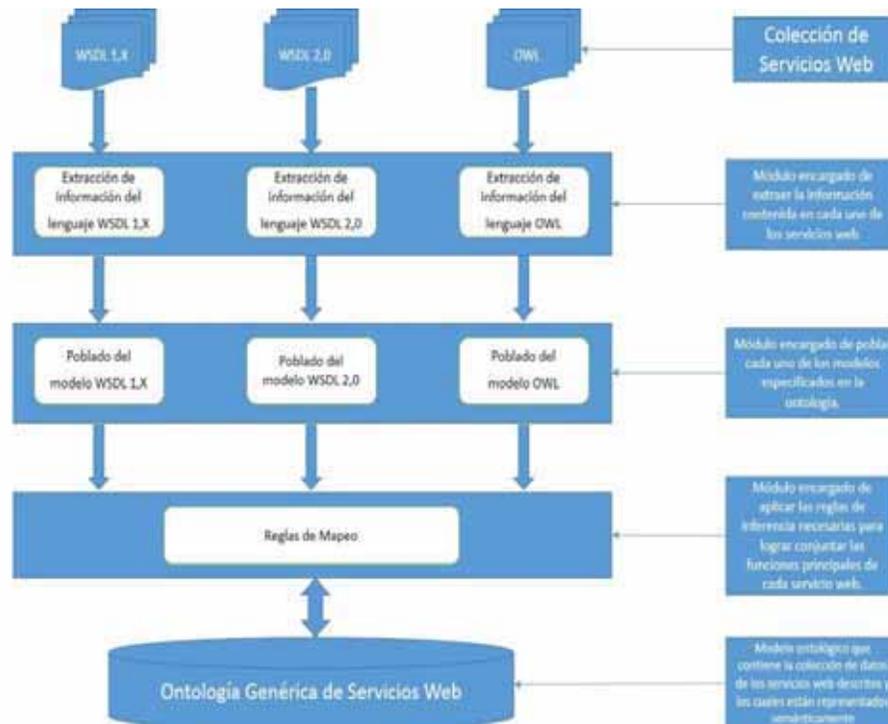


Ilustración 1: Arquitectura General del Sistema

Dentro del análisis que se realizó y el cual ya se mencionó con anterioridad, se llegó a una conclusión acerca de cómo están constituidos los servicios web. Además de ello se identificó con precisión que en los servicios del tipo WSDL tanto en su versión 1.X como en su versión 2.0 la clase principal se denota como una Operación, mientras que en los servicios OWL-S se les denomina

como Profile. Ese fue el punto de partida y el cual derivo a encontrar cada una de las equivalencias dentro de las operaciones señaladas; Dichas funciones se detallan en los siguientes diagramas, los cuales se muestran en forma de estructura la manera en cómo están conformados, y de manera similar es la forma en cómo están constituidas cada una de las ontologías.

Servicio WSDL 1.X

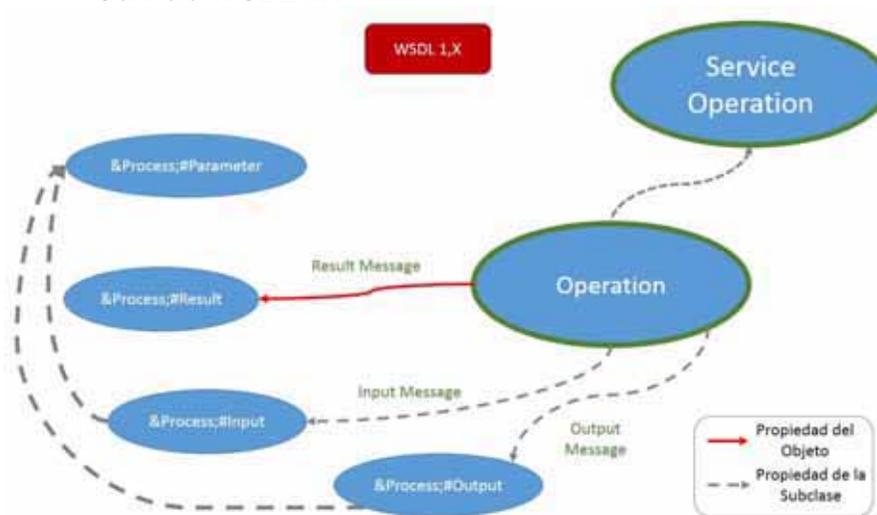


Ilustración 2: Diagrama del Servicio Web WSDL 1.X

Servicio WSDL 2.0

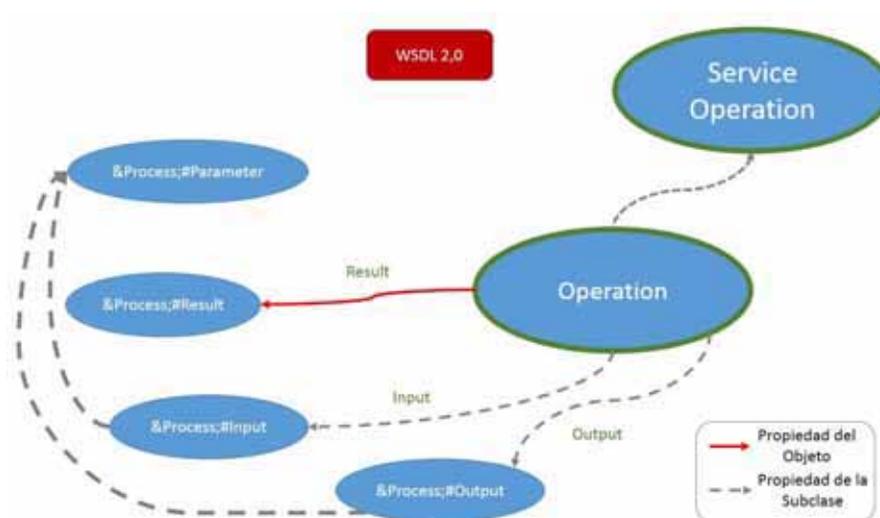


Ilustración 3: Diagrama del Servicio Web WSDL 2.0

En ambos servicios web se manejan parámetros muy similares uno del otro, estos parámetros son los responsables de que el servicio web funcione de manera eficiente y los cuales se detallan a continuación.

Operación (Operation): Es la clase principal del servicio web, en él se contienen todas las funciones que puede realizar el servicio web.

Parámetro (Parameter): Son los valores que acepta los servicios web. Estos se rigen por el tipo de operación que realiza, y pueden ser desde cadenas enteras hasta valores complejos.

Entrada (Input): Son los valores que entran dentro de las funciones de los servicios web, estos tienen la función de realizar las operaciones correspondientes a dichas funciones.

Salida (Output): Son los valores que devuelve la función después de haber realizado las operaciones correspondientes.

Resultado (Result): Es el resultado final de la operación que realizó el servicio web.

Servicio Web OWL-S

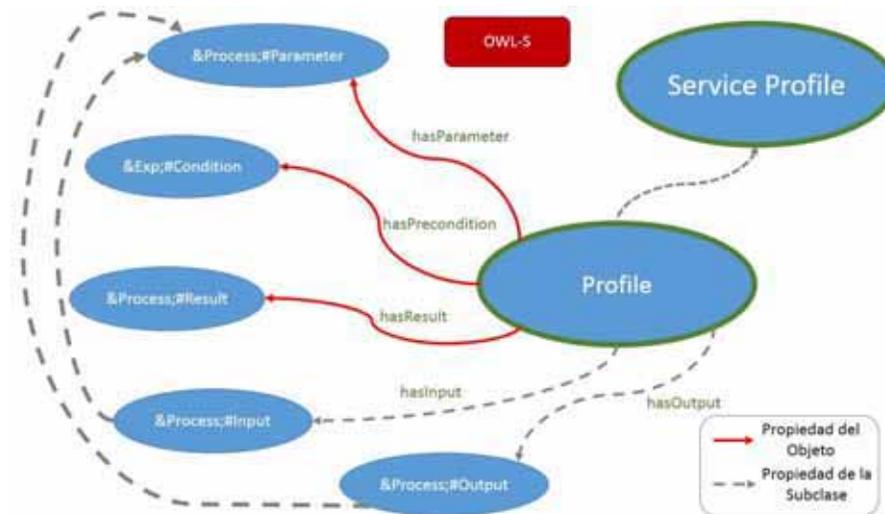


Ilustración 4: Diagrama del Servicio Web OWL-S

Los servicios web en formato OWL-S también tienen una estructura bastante similar a los mostrados en los WSDL, en lo único que varían es en el nombre de la clase principal el cual se le denomina

Profile, además de que se incluye una operación extra la cual es un parámetro de condición.

Con base a los diagramas anteriores, se procedió a investigar la manera en cómo se podrían relacionar cada una de las operaciones de estos servicios web y con ello poder ser equivalentes, esta investigación tuvo como base fundamental la documentación que existen dentro del portal de la W3C, se obtuvo un resultado el cual fue satisfactorio, y se detalla en la Ilustración 5 ;

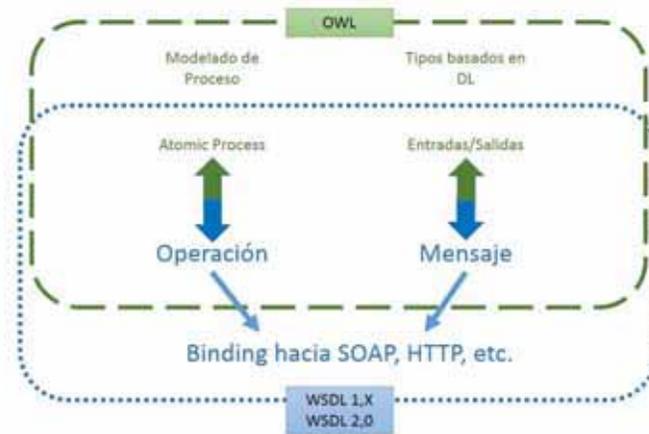


Ilustración 5: Estructura final de las equivalencias entre servicios web

Finalmente y con todos los datos obtenidos, se procedió a estructurar el diagrama final que servirá como guía para la realización de la ontología general representativa de servicios web y con ello cumplir el objetivo principal de este proyecto. Ver Ilustración 6.

7.2 Desarrollo de la Ontología Genérica

En base con toda la información recabada y anteriormente descrita, se procedió a construir la ontología genérica. Para comenzar con la realización de dicha tarea se apoyó en el software libre Protegé, el cual se desarrolla y se da soporte en la Universidad de Oxford. Este

programa realizado totalmente en JAVA fue el responsable de cargar con el enorme compromiso de generar la ontología final con los parámetros que como usuario le asigné a cada una de las clases.

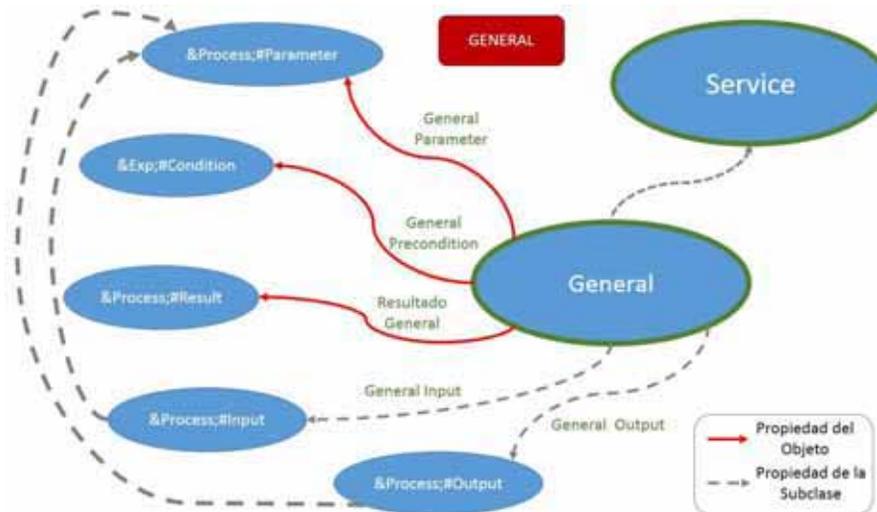


Ilustración 6: Diagrama Final

El desarrollo de la ontología comenzó detallando cada uno de los servicios web que fueron seleccionados; estos servicios se representaron en formas de clases principales nombrándolas con cada uno de los tipos de servicios web que representan y las cuales parten de una clase maestra denominada “**THING**” (Análogamente es la forma de llamar a la ontología dentro de Protegé) y la cual se detalla en la Ilustración 7

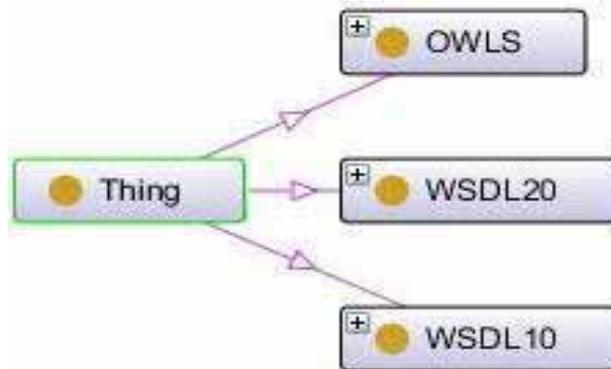


Ilustración 7: Gráfica que representa la ontología con sus tres clases principales.

Una vez realizado esto, se estudió con detalle los XML que conforman cada uno de los tipos servicios web usados y con base a ese estudio se realizó una tabla, la cual identifica cada una de las propiedades que constituyen los objetos de dichos servicios, así como se destaca el análisis que se realizó para poder obtener cada uno de los rangos y dominios los cuales conforman dichos elementos. Dichos resultados se encuentran plasmados dentro de la Tabla 3

ObjectProperties	Dominio	Rango
hasBinding	Endpoint	Binding
hasDataTypeComplexType	Part	ComplexType
hasDataTypeSimpleElement	Part	SimpleElement
hasDataTypeSimpleType	Part	SimpleType
hasEndpoint	Service	Endpoint
hasInterface	Binding	Interface
hasOperation	Interface	Operation
hasParameterInput	Operation	ParameterInput
hasParameterOutput	Operation	ParameterOutput
hasPart	Parameter	Part

Tabla 3: Propiedades de los servicios web, dominios y rangos.

Posteriormente también se obtuvieron cada uno de las propiedades que contienen los datos, así como sus respectivos dominios y sus rangos, los cuales se detallan en la Tabla 4.

DataProperties	Dominio	Rango
hasAttributeBinding	Binding	String
hasAttributeDataType	DataType	String
hasAttributeComplexType	ComplexType	String
hasRestrictionSimpleType	SimpleType	String
hasAttributeEndpoint	Endpoint	String
hasInterfaceName	Interface	String
hasAttributeOperation	Operation	String
hasAttributeParameter	Parameter	String
hasAttributeService	Service	String
hasAttributePart	Part	String

Tabla 4: Propiedades de los datos, Rangos y Dominios

Una vez habiendo obtenido los resultados anteriores se dispuso a realizar cada una de las clases, las cuales se les tuvo que añadir el nombre del servicio web como terminación de palabra; esto porque el software no permite que haya individuos repetidos dentro de la ontología. Dichos resultados se observan en la Tabla 5 y en la Tabla 6 para los servicios web del tipo WSDL 1.X, en la Tabla 7 y en la Tabla 8 se observan las propiedades contenidas para el servicio web WSDL 2.0, y finalmente en la Tabla 9 y en la Tabla 10 se observan las descripciones que contienen los servicios web OWL-S.

Con la información descrita en las tablas mencionadas como punto de referencia, se inicializó la construcción de cada una de las clases principales, las cuales se conformaron en base a los dominios y rangos que manejan cada uno de los servicios web. En la Ilustración 13, Ilustración 14, y en la Ilustración 15 se describe de manera gráfica como quedaron finalmente conformadas las clases principales, ya con esto construido se arrancó el razonador, en este caso se usó **Pellet**, el cual se encargó de encontrar alguna inconsistencia dentro de las clases, y si ese fuera el caso, el mismo razonador se ocupa de notificarlo, así como de eliminar las clases que no están lógicamente bien definidas.

Una vez teniendo los tres modelos ontológicos hechos, se procedió a aplicar las reglas de mapeo que previamente se identificaron, y las cuales se explicó oportunamente la forma en cómo se obtuvieron. Estas reglas fueron posible por medio de equivalencias que se aplicaron a las clases más importantes de cada uno de los modelos, en ellas se especifica que individuos son capaces de realizar acciones similares. Las clases identificadas con los individuos que la conforman fueron los siguientes:

- Datatype (Ilustración 8).
 - SimpleTypeWSDL1.X
 - SimpleTypeWSDL2.0
 - ComplexTypeWSDL1.X
 - ComplexTypeWSDL2.0

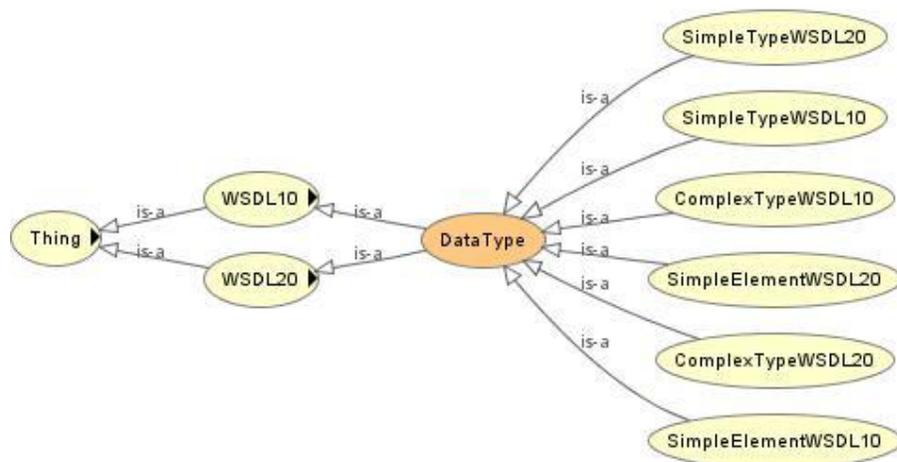


Ilustración 8: Gráfica donde se especifica el Datatype

- Result. (Ilustración 9)
 - ResultWSDL1.X
 - ResultWSDL2.0
 - ResultOWL-S

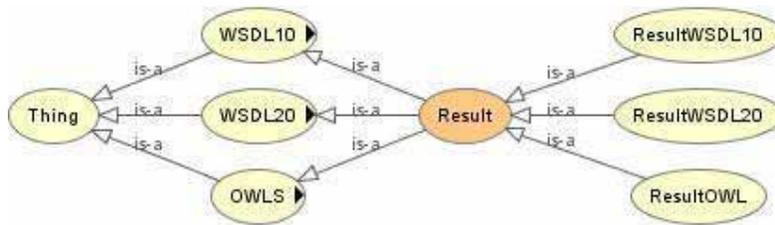


Ilustración 9: Gráfica donde se especifica el Result

- General. (Ilustración 10)
 - OperationWSDL1.X
 - OperationWSDL2.0
 - Profile (Corresponde al servicio web OWL-S)

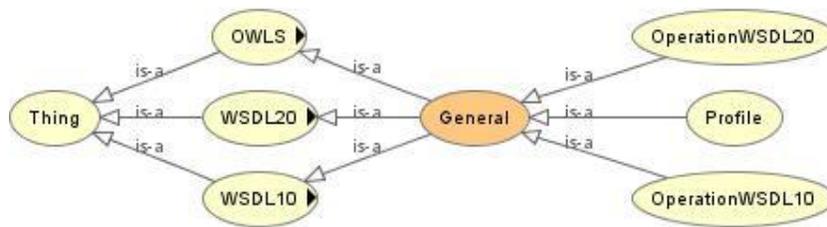


Ilustración 10: Gráfica donde se representa General

- Parameter. (Ilustración 11)
 - ParameterInputWSDL1.X
 - ParameterInputWSDL2.0
 - ParameterInputOWL-S
 - ParameterOutputWSDL1.X
 - ParameterOutputWSDL2.0
 - ParameterOutputOWL-S

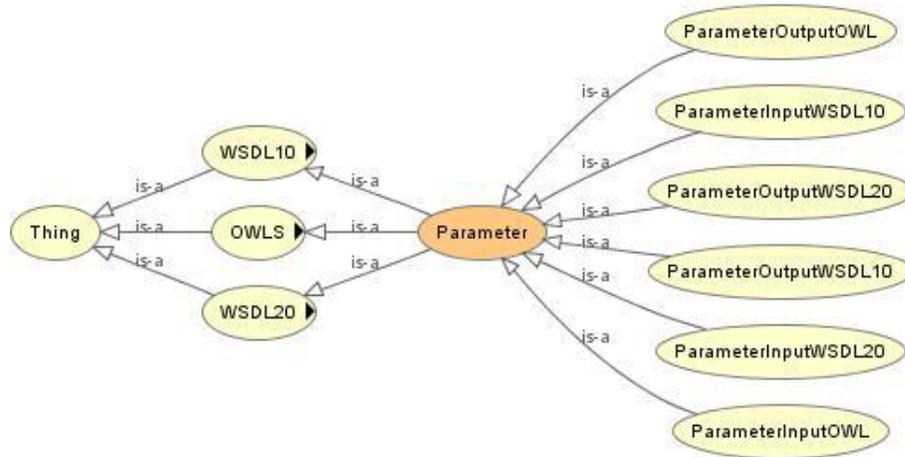


Ilustración 11: Grafica que especifica Parameter

Una vez terminadas las equivalencias, se procedió a realizar un “**Axioma de Cobertura**” a cada una de las equivalencias, esto con la finalidad de que no quede excluido ningún elemento, es decir, que si por algún motivo un dato pudiera estar repetido en servicios web diferentes, éste tiene que ser tomado en cuenta, ya que a pesar de que estos elementos tengan nombres iguales o muy similares, los servicios web en donde se incluyen son totalmente distintos, esto dentro de los conjuntos que contienen cada uno de los servicios mencionados y los cuales están obligados a realizar disyunciones que son lo que hacen posible este tipo de acciones.

Habiendo terminado todos los axiomas necesarios, se procedió a iniciar de nuevo con el razonador, esto con la finalidad de que encuentre algún tipo de incoherencias al momento de realizar la lógica necesaria para poder hacer las disyunciones que previamente se explicaron y con ello no se quede afuera ningún tipo de miembro. Una vez que el razonador no encontró problema alguno dentro de la ontología propuesta, se llegó a la conclusión, avalado previamente por la asesora, de que finalmente el modelo ontológico estaba completamente terminado, se le introdujeron algunos datos de servicios web para comprobar que estaba separando correctamente

todos los tipos de individuos que se introducían y dando como resultado la ontología mostrada en la Ilustración 12.

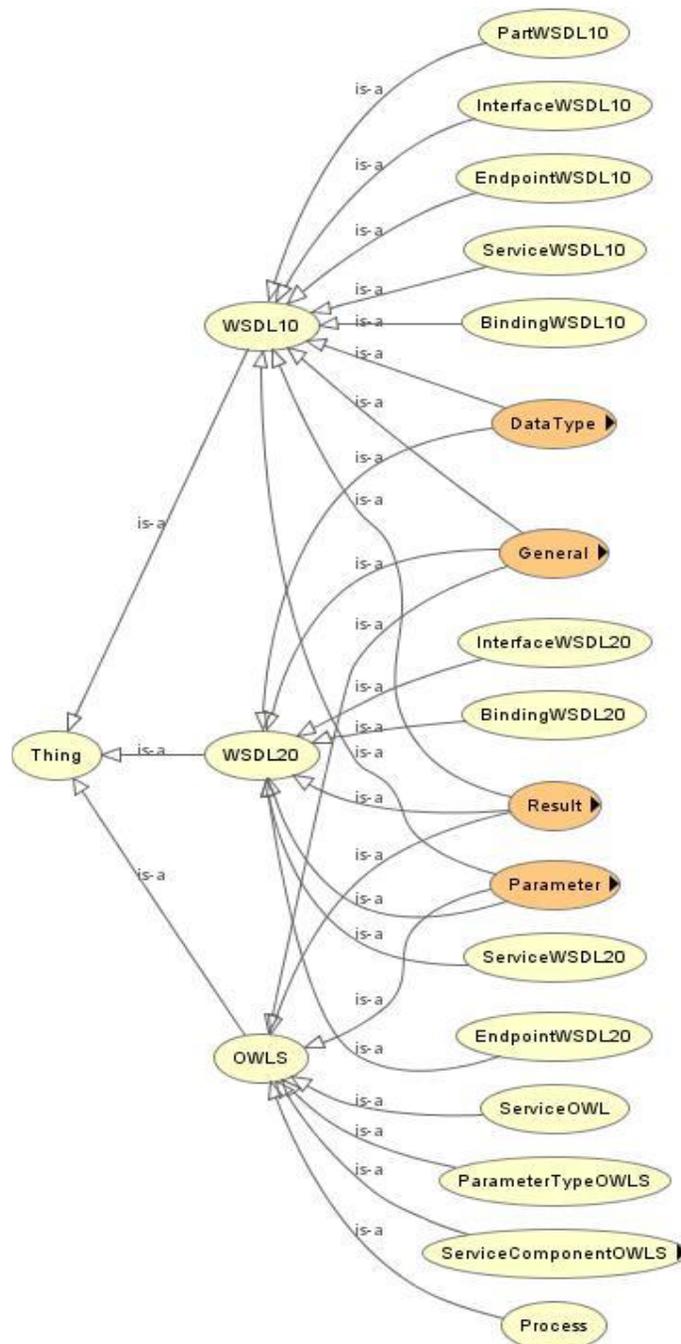


Ilustración 12: Modelo ontológico general

ObjectProperties	Dominio	Rango
hasBinding	Endpoint	Binding
hasDataTypeComplexType	Part	ComplexType
hasDataTypeSimpleElement	Part	SimpleElement
hasDataTypeSimpleType	Part	SimpleType
hasEndpoint	Service	Endpoint
hasInterface	Binding	Interface
hasOperation	Interface	Operation
hasParameterInput	Operation	ParameterInput
hasParameterOutput	Operation	ParameterOutput
hasPart	Parameter	Part

Tabla 5: Propiedades de los servicios web WSDL1.X

DataProperties	Dominio	Rango
hasAttributeBinding	Binding	String
hasAttributeDataType	DataType	String
hasAttributeComplexType	ComplexType	String
hasRestrictionSimpleType	SimpleType	String
hasAttributeEndpoint	Endpoint	String
hasInterfaceName	Interface	String
hasAttributeOperation	Operation	String
hasAttributeParameter	Parameter	String
hasAttributeService	Service	String
hasAttributePart	Part	String

Tabla 6: Datos de los Servicios Web WSDL1.X

ObjectProperties	Dominio	Rango
hasBinding	Endpoint	Binding
hasDataTypeComplexType	Part	ComplexType
hasDataTypeSimpleElement	Part	SimpleElement
hasDataTypeSimpleType	Part	SimpleType
hasEndpoint	Service	Endpoint
hasInterface	Binding	Interface
hasOperation	Interface	Operation
hasParameterInput	Operation	ParameterInput
hasParameterOutput	Operation	ParameterOutput
hasPart	Parameter	Part

Tabla 7: Propiedades de los servicios web WSDL 2.0

DataProperties	Dominio	Rango
hasAttributeBinding	Binding	String
hasAttributeDataType	DataType	String
hasAttributeComplexType	ComplexType	String
hasRestrictionSimpleType	SimpleType	String
hasAttributeEndpoint	Endpoint	String
hasInterfaceName	Interface	String
hasAttributeOperation	Operation	String
hasAttributeParameter	Parameter	String
hasAttributeService	Service	String
hasAttributePart	Part	String

Tabla 8: Datos de los servicios web WSDL 2.0

ObjectProperties	Dominio	Rango
hasBinding	Endpoint	Binding
hasDataTypeComplexType	Part	ComplexType
hasDataTypeSimpleElement	Part	SimpleElement
hasDataTypeSimpleType	Part	SimpleType
hasEndpoint	Service	Endpoint
hasInterface	Binding	Interface
hasOperation	Interface	Operation
hasParameterInput	Operation	ParameterInput
hasParameterOutput	Operation	ParameterOutput
hasPart	Parameter	Part

Tabla 9: Propiedades de los servicios web OWL-S

DataProperties	Dominio	Rango
hasAttributeBinding	Binding	String
hasAttributeDataType	DataType	String
hasAttributeComplexType	ComplexType	String
hasRestrictionSimpleType	SimpleType	String
hasAttributeEndpoint	Endpoint	String
hasInterfaceName	Interface	String
hasAttributeOperation	Operation	String
hasAttributeParameter	Parameter	String
hasAttributeService	Service	String
hasAttributePart	Part	String

Tabla 10: Datos de los Servicios Web OWL-S

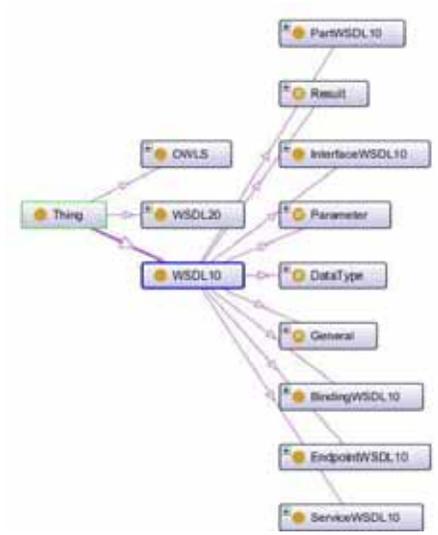


Ilustración 13: Gráfica que representa el servicio web WSDL 1.X

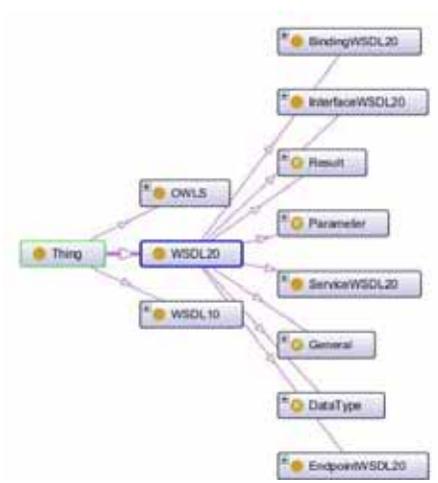


Ilustración 14: Gráfica que representa el servicio web WSDL 2.0

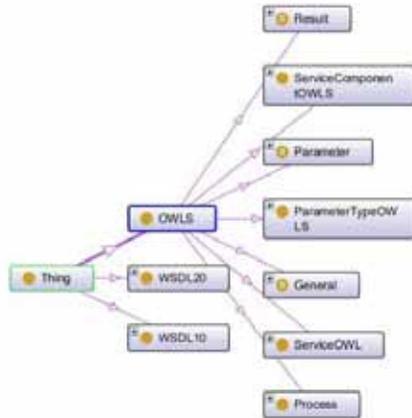


Ilustración 15: Gráfica que representa el servicio web OWL-S

7.2.1 Reglas para nombrar a los individuos.

Estas reglas se acordaron y definieron en común acuerdo con la asesora para que en un futuro, si se requiriera, esta ontología pudiera acoplarse a otro tipo de sistemas, o en su defecto acoplarse a un sistema más grande. Se usó un estándar el cual se manejó en otros proyectos terminales relacionados con las ontologías, estos estándares están compuestos de tal manera que no se produzcan conflictos entre los individuos por llamarse de la misma manera y con ello también evitamos conflictos al momento de poblar una ontología. Primeramente se tienen los Identificadores de Recursos Internacionales (Abreviado como IRI) los cuales ayudan como identificadores de la ontología, o en este caso el servicio web que representa.

El formato usado para este fin fue el siguiente:

Para el nombrado del servicio

IRI + nombre del servicio + \$numeroconsecutivo

Nombrado de los individuos

IRI + nombre del individuo + \$ prefijo \$ + nombredelservicio
--

En las siguientes tablas, se detallarán cada uno de los términos usados para el nombrado de cada uno de los individuos.

Clase	Prefijo	Clase	Prefijo
Binding	bin	Operation	opi
Interface	intf	Services	
Endpoint	endp	Part	part
ParameterInput	pin	ParameterOutput	pout
SimpleElement	se	ComplexType	ct
SimpleType	st		

Tabla 11: Clases y prefijos para WSDL1.X

Clase	Prefijo	Clase	Prefijo
Binding	bin	Operation	opi
Interface	intf	Services	
Endpoint	endp	ParameterOutput	pout
ParameterInput	pin	ComplexType	ct
SimpleElement	se	SimpleType	st

Tabla 12: Clases y prefijos para WSDL2.0

Clase	Prefijo	Clase	Prefijo
Service		ParameterType	
Process	process	ParameterInput	pin
ParameterOutput	pout	ServiceProfile	sp
ServiceModel	sm	ServiceGrounding	sg

Tabla 13: Clases y prefijos para OWL-S

7.3 Desarrollo de la Interfaz Gráfica

Para el desarrollo de la interfaz gráfica se tomó como punto de partida el código generado por Protegé. Este código representa al modelo ontológico que se obtuvo de los razonamientos que oportunamente se explicaron y detallaron en este reporte. La función principal de la interfaz es la de comunicar de manera eficiente al usuario con la ontología por medio de un ambiente gráfico capaz de correr en todos los navegadores web (Firefox, Google Chrome, Internet Explorer, Safari) además de que sea intuitivo para el usuario y con ello no tenga problema alguno para navegar dentro de la aplicación web que se realizó. Cabe resaltar que la aplicación web de la interfaz gráfica fue realizada usando el IDE de desarrollo **Netbeans** en su versión **8.0**, se usó el framework “**Primefaces**” y está montado para que corra bajo un servidor “**Apache Tomcat 7.0.42**”, ya que al momento de correrlo con la versión 8.0 de este servidor, el proyecto sufría de muchos inconvenientes debido a que en el momento de realizar este proyecto, el servidor antes mencionado (8.0) todavía se encontraba en la etapa de pruebas trayendo como consecuencia que fuera muy inconsistente y poco fiable.

Un punto importante dentro de esta carpeta proyecto son las librerías las cuales tienen una ruta relativa para que esta aplicación pueda ser desplegada (Deploy) dentro de cualquier computadora que cuente con las especificaciones arriba mencionadas. Las librerías que se usaron para el desarrollo fueron las siguientes:

- JSF 2.2 – javax.faces
- JAVA EE 6 API Library

Estas librerías se encuentran de manera predeterminada dentro del IDE de desarrollo. Así mismo los JAR's ocupados fueron los siguientes:

- bootstrap-1.0.10
- commons-fileupload-1.2.2
- commons-io-2.4
- javax.faces-2.1.9
- jaxen-1.1.3
- jdom-2.0.2
- jstl
- org.protege.editor.owl.codegeneration
- owlapi-bin
- prettyfaces-jsf2-3.3.3
- primefaces-3.4
- standard
- xercesImpl
- xml-apis

Todos estos .JAR se encuentran en la siguiente dirección

PTOGWS -> JAR -> librerías

Como se mencionó anteriormente la dirección descrita arriba esta referenciada de manera relativa, ya que si se escribía de manera absoluta habría el error de que no se vincularan de manera automática los componentes necesarios para que la aplicación pueda funcionar de manera óptima.

La aplicación desarrollada toma como referencia para iniciar un índice realizado en XHTML, el cual le da la bienvenida al usuario e indica los datos del autor de este proyecto terminal y de la asesora que tomo parte como responsable de que se cumpliera cada uno de los puntos establecido en la propuesta. Debo destacar que la interfaz web fue tomada de un proyecto terminal pasado [4], esto con la finalidad de que se siga un estándar que tiene como finalidad la integración de este sistema a un proyecto más grande, tal y como se mencionó con anterioridad.



Ilustración 16: Página de Inicio del Sistema PTOGWS

En esta primera ventana de navegación se puede observar que se cuenta con dos vínculos en el encabezado, los cuales están hechos de manera independientes con la finalidad de poder reutilizarlos cuantas veces sean requeridos. El primero de estos vínculos se encuentra en la palabra **Inicio**, el cual tiene como función llevar al usuario al índice principal del sistema, y el segundo de ellos se encuentra en la palabra “**Sistema**”. Este enlace lleva al usuario a la interfaz que corresponde a la parte funcional del proyecto.

Esta página, como se mencionó anteriormente, cuenta con los datos principales del alumno y la asesora, así como los links que llevan a los servidores donde se encuentran las herramientas usadas para la realización de este proyecto.

Al momento de dar clic en la palabra “**Sistema**”, la aplicación llevará al usuario a una segunda pantalla que como se mencionó en

la parte de arriba, es la encargada de realizar todo el trabajo de la aplicación, y la cual se detallará a continuación

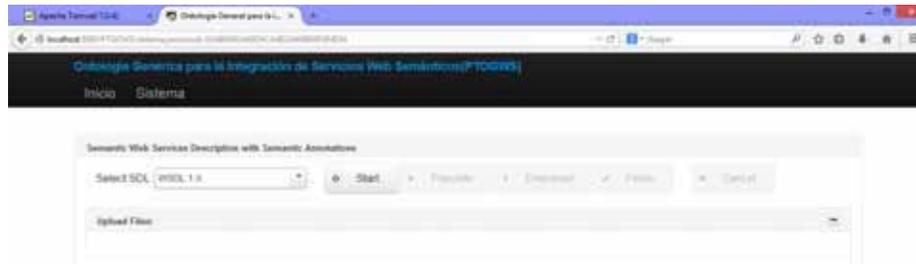


Ilustración 17: Pantalla del Sistema

Esta pantalla cuenta con diferentes elementos que son interactivos con la persona que llegue a utilizar este sistema. Primeramente tenemos el “Header” el cual está compuesto por los dos elementos que hacen posible la navegación dentro del sistema (“Inicio” y “Sistema”), posteriormente tenemos un menú desplegable el cual ofrece las opciones de los servicios web que se pueden seleccionar (véase la Ilustración 18) .



Ilustración 18: Menú de Servicios Web

Una vez elegido el servicio web con el que trabajará el sistema, procede a iniciarlo, esto presionando el botón de “**Start**” (Ilustración 19).



Ilustración 19: Menú con el botón para iniciar el Sistema

Una vez iniciado el sistema, aparecerán 3 botones indicando cada una de las acciones que se pueden realizar (Ilustración 20).

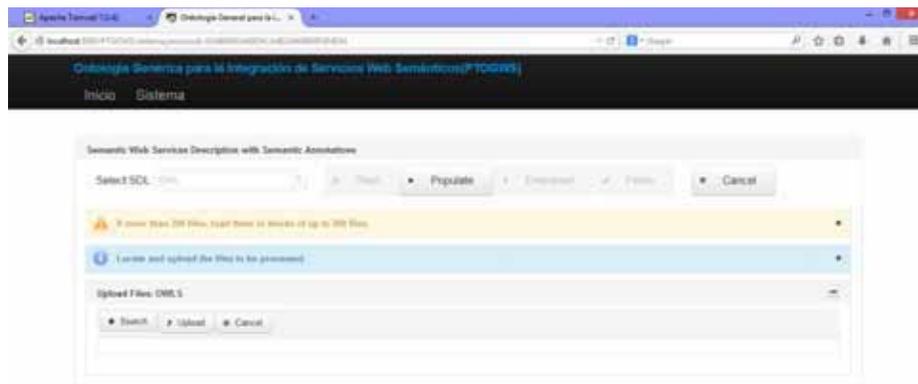


Ilustración 20: Sistema en funcionamiento

El primer botón **“Search”** abre una pequeña ventana (véase la Ilustración 21) para que el usuario pueda seleccionar los servicios web que desee, siempre y cuando cumplan con el requisito de ser del tipo que se seleccionó previamente; una vez seleccionados se procede a dar clic al botón de **“Abrir”**. Cabe resaltar el hecho que solo se pueden seleccionar y cargar grupos de 200 servicios web, esto quiere decir que si un usuario quisiera hacer una extracción de 500 servicios, el procedimiento que tendría que seguir será el siguiente.

- 1.- Seleccionar los primeros 200 servicios web.
- 2.- Cargar por medio del botón **“Upload”** los servicios al sistema.
- 3.- Seleccionar los segundos 200 servicios web.
- 4.- Cargar por medio del botón **“Upload”** los servicios al sistema.

5.- Seleccionar los últimos 100 servicios web.

6.- Cargar por medio del botón “**Upload**” los servicios al sistema.

Esto se recomienda hacer debido a las limitantes que maneja el framework de “Primefaces”, el cual no es capaz de soportar toda la gran cantidad de trabajo que conlleva cargar y parsear cada uno de los servicios cargados, ya que si se hiciera lo contrario, lo que provocaría sería que el framework deje de funcionar y con ello detener el script de ejecución dentro del sistema teniendo como consecuencia que éste se pase y deje de funcionar de manera temporal, hasta que el usuario decida detener el script o aguardar a que pueda ejecutarse.

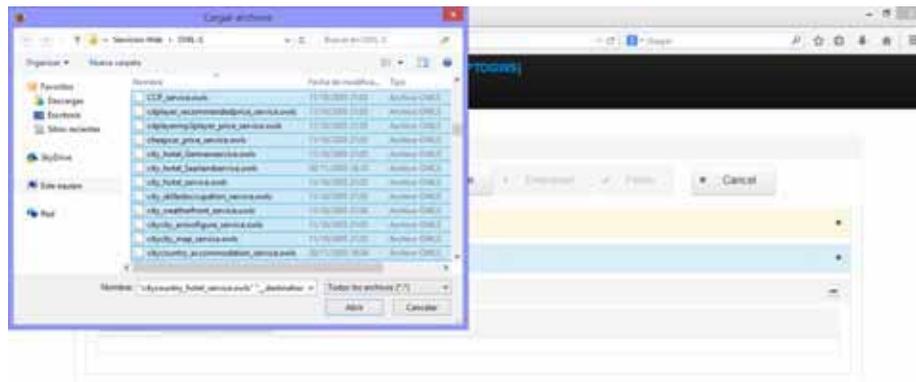


Ilustración 21: Ventana para seleccionar servicios

Como se mencionó al momento de abrir los servicios web seleccionados, estos se cargarán en la página del sistema para esperar a ser procesados (Ilustración 22).



Ilustración 22: Pantalla donde se muestran los servicios cargados

Una vez cargados los servicios web (Ilustración 23) se procede a poblar la ontología con el botón “**Populate**”, este botón lo que hace es la interacción que debe tener el sistema web con el sistema en JAVA, ya que es el encargado de llamar a los parsers correspondientes por medio de SistemaBean.java para que estos identifiquen cada uno de los miembros que componen el servicio y así poder colocarlos en las clases que corresponden a cada uno de los elementos identificados dentro del modelo genérico que se obtuvo en la ontología (véase la Ilustración 24).



Ilustración 23: Pantalla con los servicios web cargados.

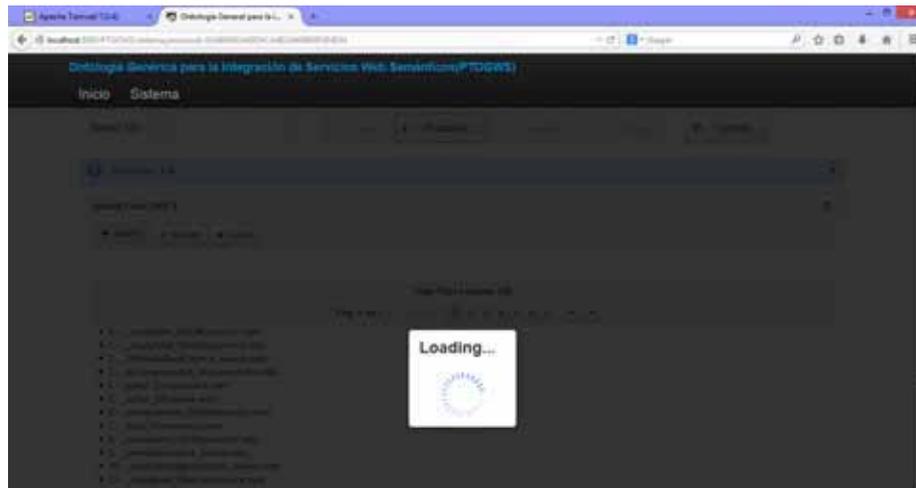


Ilustración 24: Momento en que es poblada la ontología genérica

Una vez poblada la ontología, se activará el botón “Download” (Ilustración 25), el cual le permitirá al usuario poder descargar el archivo OWL generado (Predeterminadamente se descargará en la carpeta que está por defecto en el sistema operativo) y así poder visualizarlo con **Protegé** (Ilustración 26).



Ilustración 25: Pantalla con el botón Download activado

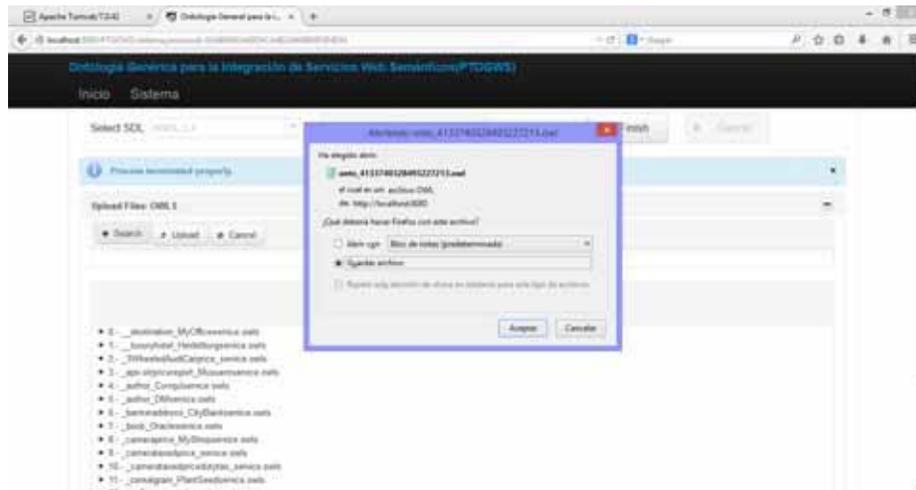


Ilustración 26: Pantalla con el archivo a descargar

Finalmente y una vez descargada la ontología, se procede a dar clic en el botón de “Finish”, para que el sistema termine con su trabajo hasta una nueva extracción de miembros (véase la Ilustración 27).

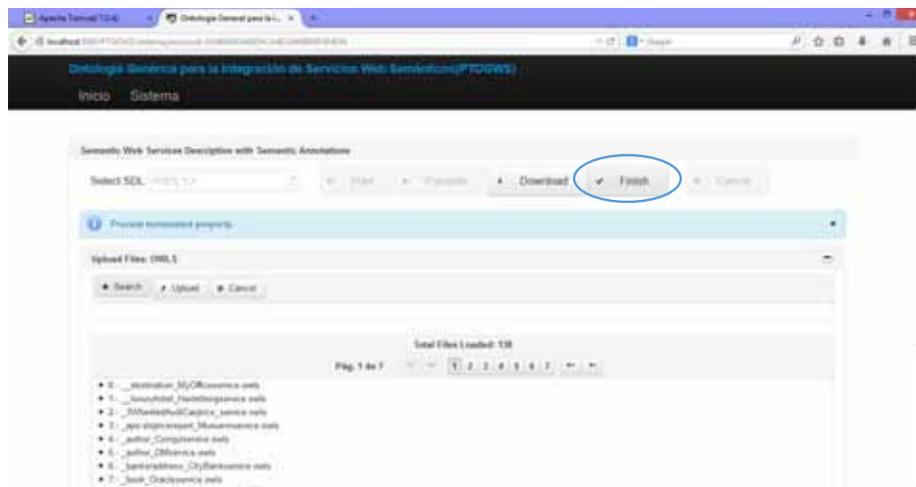


Ilustración 27: Pantalla con el botón “Finish” activado

8 Resultados

Después de haber realizado varias pruebas al sistema, el resultado que se obtuvo fue satisfactorio; el sistema es capaz de poder parsear cada uno de los elementos introducidos y con ello poder identificar a cada uno de los miembros que lo componen y colocarlos dentro de las clases que le corresponden. En un principio los resultados no fueron los deseados debido a pequeñas inconsistencias que había dentro de la ontología, dichas inconsistencias fueron arregladas y se pudo obtener la meta esperada de este proyecto de integración.

Los resultados de este sistema se pueden observar en las imágenes Ilustración 28, Ilustración 29 y en la Ilustración 30, en ellas se realizó una prueba con un servicio web OWL-S y se puede observar lo que se afirmó en el párrafo anterior, finalmente gracias a este resultado y en común acuerdo con la asesora se dio por entregado y finalizado este proyecto terminal.

Cabe destacarse que este proyecto que ahora se entrega es la base para proyectos terminales futuros, ya que este modelo ontológico será de mucha utilidad al momento de desarrollar un sistema más complejo; por ello la importancia de estandarizar todos los parámetros y los miembros que se manejaron durante la realización de este trabajo recepcional. Uno de dichos proyectos a futuro podría ser la realización de un parser inteligente que sea capaz de identificar dentro de la estructura del XML el tipo de servicio web que se maneja y con ello poder llamar a las funciones que se encargan de poblar y realizar todas las acciones correspondientes dentro del proyecto que entrega este servidor; esto con la finalidad de que el sistema sea capaz de recibir cualquier tipo de servicio y las pueda integrar dentro de un mismo archivo OWL logrando así el sistema homogéneo requerido para los fines que tenga la asesora.

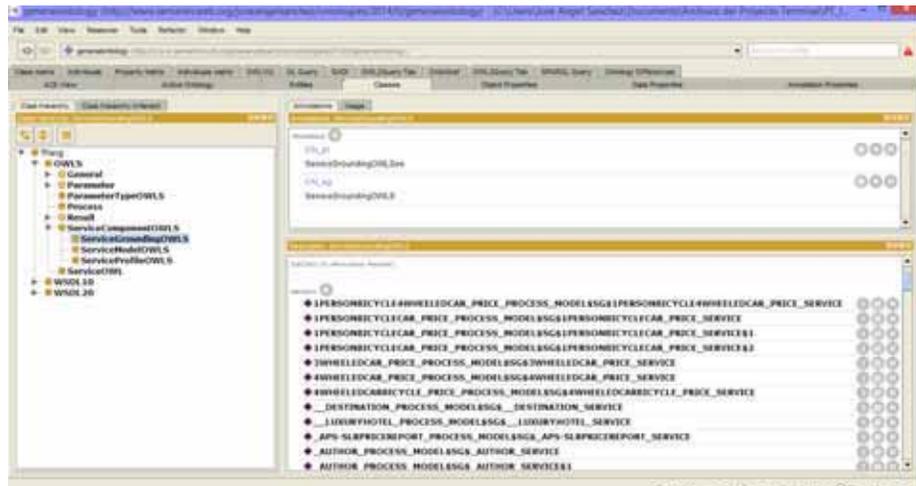


Ilustración 30: Pantalla donde se muestran los miembros del Service Ground en OWL-S

9 Conclusiones

Las conclusiones obtenidas del desarrollo de este proyecto terminal fue el haber adquirido un mayor conocimiento en el manejo de las ontologías, así como haber aprendido a usar la lógica computacional a un nivel más avanzado, esto gracias a las reglas de inferencia y a los diferentes axiomas que se tuvieron que ocupar e hicieron que repitiera el modelo ontológico general más de una vez.

Otra conclusión positiva de este proyecto, fue que me ayudo en gran medida a comprender cómo funcionan los servicios web más en concreto, así como resaltar el papel tan importante que juegan en la actualidad; me enseñaron mucho acerca de su funcionamiento, el cómo es que pasan sus parámetros y con ello poder llamar a las funciones descritas que permiten la realización de las operaciones correspondientes y para las cuales son requeridas. Así como los puntos anteriores, también quisiera recalcar la utilidad de haber aprendido a usar el software Protegé, que no había usado en mi vida académica, descubrí lo poderosa que puede llegar a ser ésta herramienta en cuestión de diseño y modelado de las ontologías.

Por supuesto no podría olvidar que este trabajo recepcional me enseñó mucho acerca de las ontologías, estoy consciente de que es un curso que se imparte o impartía dentro de la carrera, pero al haber nuevos planes, este servidor no pudo tomar dicho curso, y por eso me llena de satisfacción saber que logré aprender el funcionamiento de estos sistemas de información inteligentes, y eso me llena de mucho orgullo y dignidad.

Además de todo lo mencionado anteriormente, este proyecto terminal me enseñó lo más importante de todo; me enseñó el valor de todos los conocimientos que adquirí a lo largo de estos años de carrera y los cuales fueron muy valiosos para poder resolver las complicaciones que se me presentaron a lo largo del desarrollo de este proyecto y los cuales me permitieron salir adelante para poder continuar con su desarrollo. Y así finalmente yo me quedaría con la siguiente cita textual, la cual refleja con exactitud mi sentir acerca de todo este trabajo terminal:

“El sueño detrás de la Web es un espacio común en el que nos comunicamos compartiendo información. Su universalidad es esencial: el hecho de que un enlace de hipertexto pueda apuntar a cualquier cosa, sea personal, local o global, sea un borrador o algo definitivo. Hay una segunda parte del sueño, que consiste en nuestras interacciones en línea, trabajar, vivir en sociedad y jugar. Podemos usar los ordenadores para analizar esas interacciones, encontrarle el sentido a lo que estamos haciendo, donde encajamos y cómo podemos trabajar mejor juntos. Con el flujo dramático de material de todo tipo que sufre la Web a partir de los 90's, la primera parte del sueño está cumplida. La segunda parte todavía no se cumple, pero hay signos y planes que nos hacen confiar. La gran necesidad de información acerca de la información, para ayudarnos a categorizar, ordenar, pagar, y poseer información que está derivando en lenguajes diseñados para que la Web sea procesada por máquinas, más que por la gente. La Web de documentos legibles por humanos se está

mezclando con la Web de datos entendibles por las máquinas. El potencial de la mezcla de humanos y máquinas trabajando juntos puede ser inmenso.” [9]

10 Bibliografía

- [1] T. Berners-Lee, «Semantic Web,» *Scientific American Magazine*, 2001.
- [2] J. Sowa, Knowledge representation: logical, philosophical and computational foundations, United States: Brooks/Cole Publishing, 1999.
- [3] S. M. Ugalde Chavez, «Sistema de recuperación de información semántico», proyecto terminal,» División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, México, 2012.
- [4] J. Pascual Martínez, «Extracción automatizada y representación de servicios web mediante ontologías», proyecto terminal,» División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, México, 2012.
- [5] M. García Sierra, «Sistema Clasificador de ontologías mediante métodos de máquinas de soporte vectorial», proyecto terminal,» División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, México, 2012.
- [6] D. A. Malgón Mercado, «Solución de Problemas de mediana dificultad utilizando composición de servicios web», proyecto terminal,» División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, México, 2012.
- [7] F. Tébar Martínez, «Sistemas de almacenamiento semántico de textos de investigación mediante ontologías», proyecto terminal,» División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, México, 2012.
- [8] T. Berners-Lee, «www.w3.org,» Octubre 1994. [En línea]. Available: www.w3.org. [Último acceso: 01 Julio 2014].
- [9] T. Berners-Lee, «The World Wide Web: A Very Short Personal History,» W3 Organization, 07 Mayo 1998. [En

línea]. Available: <http://www.w3.org/People/Berners-Lee/ShortHistory.html>. [Último acceso: 22 Agosto 2014].

11 Apéndices

11.1 Propuestas para Proyectos Terminales futuros

Al momento de entregar este proyecto, de común acuerdo con mi asesora se acordó que el parser inteligente de servicios web se propondría para otro proyecto terminal, por mi parte anexo la ubicación específica del código dentro de SismaBean.java, el cual hay que modificar de tal manera que no tenga restricción alguna al momento de poner servicios web de diferentes tipos dentro de una misma selección de archivos.

```
if (opcionSel == 1) {
    templateOntology = ((ServletContext) FacesContext.getCurrentInstance().getExternalContext().getContext())
        .getResourceAsStream("/resources/ontologiasTemplate/Ontologia.owl");
    Ontologia = File.createTempFile("onto_", ".owl", tempDirColeccion);
    Ontologia.deleteOnExit();
    IOUtils.copy(templateOntology, new FileOutputStream(Ontologia));
    PopulationWSDL1X poblarOnto=new PopulationWSDL1X(Ontologia,archivos);
    poblarOnto.poblarOntologia(errores);
} else if (opcionSel == 2) {
    templateOntology = ((ServletContext) FacesContext.getCurrentInstance().getExternalContext().getContext())
        .getResourceAsStream("/resources/ontologiasTemplate/Ontologia.owl");
    Ontologia = File.createTempFile("onto_", ".owl", tempDirColeccion);
    Ontologia.deleteOnExit();
    IOUtils.copy(templateOntology, new FileOutputStream(Ontologia));
    PopulationWSDL20 poblarOnto=new PopulationWSDL20(Ontologia,archivos);
    poblarOnto.poblarOntologia(errores);
}
else if (opcionSel==3){
    templateOntology=((ServletContext)FacesContext.getCurrentInstance().getExternalContext().getContext())
        .getResourceAsStream("/resources/ontologiasTemplate/Ontologia.owl");
    Ontologia = File.createTempFile("onto_", ".owl",tempDirColeccion);
    Ontologia.deleteOnExit();
    IOUtils.copy(templateOntology, new FileOutputStream(Ontologia));
    OntologyPopulationOWLS poblarOnto=new OntologyPopulationOWLS(Ontologia, archivos);
    poblarOnto.poblarOntologia(errores);
}
```