

MOTOR DE BÚSQUEDA SEMÁNTICA EN SERVIDORES DE INTERNET DE  
OBJETOS (INTERNET OF THINGS – IOT)

JESUS ALBERTO NIÑO ZAMBRANO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICO – MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS  
BUCARAMANGA

2014

MOTOR DE BÚSQUEDA SEMÁNTICA EN SERVIDORES DE INTERNET DE  
OBJETOS (INTERNET OF THINGS – IOT)

JESUS ALBERTO NIÑO ZAMBRANO

Trabajo de Grado para optar al título de  
Ingeniero de Sistemas

DIRECTOR

MSC. LUIS CARLOS GÓMEZ FLÓREZ

CODIRECTOR

MSC. MIGUEL ANGEL NIÑO ZAMBRANO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERAS FISICO – MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS  
BUCARAMANGA

2014

A Sandy

## **AGRADECIMIENTOS**

En primera medida doy agradecimientos a mis tutores el Profesor Luis Carlos Gómez Flórez y el Profesor Miguel Angen Niño Zambrano que con paciencia y esfuerzo me condujeron por este difícil sendero para alcanzar la exelencia profesional y académica. Agradezco mi madre que siempre estuvo pendiente de mi proceso de formación y aporto lo más importante que fue su incondicional, afecto, amor y dedicación, agradezco al alma mater la Universidad Industrial de Santander por recibirme en su seno, y hacer de mi una persona útil a la sociedad y a mi familia, agradezco a mis hermanos que siempre estuvieron atentos a brindarme una mano en los momentos mas necesarios.

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	20
1.1 Contexto General .....	20
1.2 Declaración del Problema.....	21
1.3 Hipótesis de Investigación .....	21
1.4 Escenario de Motivación.....	21
1.5 Contribuciones.....	22
1.6 Alcance.....	24
1.7 Objetivos.....	24
1.7.1 Objetivo General.....	24
1.7.2 Objetivos Específicos .....	25
1.8 Contenido de la Monografía .....	25
MARCO REFERENCIAL .....	27
2. Introducción .....	27
2.1 Internet de Objetos .....	27
2.1.1 Modelo de Conceptos de la IoT .....	27
2.1.2 Relación entre la IoT y las redes existentes .....	28
2.1.3 Propiedades de la Red de Objetos .....	29
2.2 Convergencia de Visiones en la IoT .....	30
2.3 Características de los objetos.....	30
2.3.1 Taxonomía de Objetos de la IoT.....	31
2.3.2 Tecnologías del Internet de Objetos .....	32
2.3.3 Aplicaciones del IoT.....	33
2.3.4 Arquitectura de la IoT .....	33
2.4 Middleware de la IoT .....	35
2.4.1 Clases de Taxonomía de Middleware.....	35
2.5 Web Semántica .....	35
2.5.1 Tecnologías de la Web Semántica .....	36

2.5.2 Ontologías .....	36
2.5.3 Indexación Semántica .....	37
2.5.3.1 Anotaciones Semánticas .....	37
2.5.3.2 Anotaciones del Natural Language Processing - NLP .....	38
2.5.3.3 Anotaciones sobre Información Semántica Contextual .....	41
2.6 Semántica en la IoT .....	43
2.6.1 Estándares Semánticos en el IoT .....	43
2.6.2 Estructuras usadas para indexar en la IoT .....	44
2.6.3 Aplicaciones de Búsqueda en la IoT.....	44
2.6.4 Motores de búsqueda en la IoT .....	44
2.6.5 Ontologías para la IoT .....	44
2.7 Publicación de datos en la Web.....	45
2.8 Recuperación de la Información .....	46
2.8.1 Modelos de Recuperación de la Información.....	47
2.8.2 Consulta, Búsqueda y Clasificación.....	47
2.8.3 Estructuras de indexación en la Web .....	49
2.8.4 Semántica aplicada a la búsqueda de datos de Sensores en la IoT .....	49
2.9 Servicios Web - WS.....	50
2.10 Servicios Web Semánticos - SWS.....	50
2.11 Conclusiones del Marco de Referencia .....	51
METODO DE INDEXACIÓN SEMANTICA EN LA IOT - SIMIOT .....	52
3.1 Método Propuesto .....	52
3.1.2 Preguntas que debe responder el método.....	52
3.1.3 Arquitectura del Índice Semántico en IoT .....	52
3.2 Método para crear Índices Semánticos en WoT .....	54
3.2.1 Fase 1: Conceptualización del Proyecto de Indexación Semántica (FCPIS)...	55
3.2.2 Fase 2: Creación de la Base de Conocimiento y Servicios (FCBCS) .....	58
3.2.3 Fase 3 Construcción del Índice Semántico (FCIS) .....	59
3.2.4 Fase 4 Despliegue del Índice Semántico (FDIS) .....	67

3.3 Especificación del Método para la Creación de Índices Semánticos en la IoT en SPEM 2.0 .....	68
3.4 Conceptos SPEMv2.0 y Diagramas de SIMIOT .....	69
3.5 Conclusiones Método SIMIOT .....	69
BUSCADOR SEMANTICO EN LA WOT ACERCA DE LA CONTAMINACION AMBIENTAL.....	71
4.1 Fase I Actividad 1 Tarea 1 Visión del Sistema .....	71
4.1.1 Introducción .....	71
4.1.2 Definición del problema .....	71
4.1.3 Ubicación de los Productos .....	73
4.2 Fase I Actividad 1 Tarea 2 Especificación de Requisitos Buscador Semántico BSIoT.....	73
4.2.1 Introducción .....	73
4.2.2 Requerimientos Funcionales .....	73
4.2.1 Interfaces del Sistema .....	74
4.2.2 Interfaces a Sistemas Externos .....	74
4.2.3 Reglas de Negocio (Business Rules) .....	74
4.2.4 Documentación.....	75
4.2.5 Ubicación de los Productos .....	75
4.3 Fase I Actividad 1 Tarea 3 Arquitectura del Sistema BSIoT .....	75
4.3.1 Introducción .....	75
4.3.2 Objetivos de la Arquitectura y Filosofía .....	75
4.3.3 Dependencias y Asunciones.....	76
4.3.4 Vistas Arquitectónicas .....	77
4.3.4.1 Ubicación de los Productos .....	80
4.4 Fase I Actividad 1 Tarea 4 Modelo Semántico BSIoT .....	80
4.4.1 Introducción .....	80
4.4.2 Preguntas de Referencia del Modelo.....	80
4.4.3 Modelos del Índice Semántico a Construir.....	82
4.4.4 Modelo de Consultas Personalizadas.....	83

4.4.5	Modelo de Servicios .....	84
4.4.6	Ubicación de los Productos .....	87
4.5	Fase I Actividad 2 Tarea 1: Evaluar el middleware / servidor IoT a utilizar (AEMS) .....	87
4.5.1	Introducción .....	87
4.5.2	Marco de Referencia de Servidores IoT .....	87
4.5.3	Ubicación de los Productos .....	90
4.6	Fase I Actividad 2 Tarea 2 Alineación de Conceptos del Servidor IoT y Fuentes Externas .....	90
4.6.1	Ubicación de los Productos.....	90
4.7	Fase II Actividad 1 Tarea 1 Seleccionar una metodología de desarrollo adecuada.....	90
4.7.1	Introducción .....	90
4.7.2	Implicaciones y Relaciones con el Modelo Semántico Propuesto .....	91
4.7.3	Principios de diseño de la ontología .....	92
4.7.4	Metodología de Diseño de Ontologías.....	94
4.7.5	Ubicación de los Productos .....	96
4.8	Fase II Actividad 1 Tarea 2 Crear la estructura de conocimiento .....	96
4.8.1	Ontología de Contaminación Ambiental .....	96
4.8.2	Ubicación de los Productos .....	97
4.9	Fase II Actividad 2 Tarea 1 Crear los servicios del API de la Fuente de datos . .....	97
4.9.1	Ubicación de los Productos .....	97
4.10	Fase II Actividad 2 Tarea 2 Realimentación de la funcionalidad.....	97
4.11	Fase III Actividad 1 Tarea 1 Extraer Información de Objetos .....	98
4.11.1	Ubicación de los Productos .....	98
4.12	Fase III Actividad 1 Tarea 2, 3 y 4 Análisis Léxico, Eliminación de Palabras vacías y Aplicación de Lematización .....	98
4.12.1	Ubicación de los Productos .....	99
4.13	Fase III Actividad 2 Tareas 1, 2 y 3 Selección de Términos, Obtención de los Conceptos, Desambiguar Conceptos.....	99

4.13.1	Introducción .....	99
4.13.1	Extracción de Conceptos .....	100
4.14	Fase III Actividad 3 Tareas 1, 2 y 3 Análisis de Coocurrencia y Análisis de Representatividad.....	100
4.14.1	Análisis de coocurrencia .....	100
4.15	Fase III Actividad 4 Tarea 1 Implementación del Índice Semántico .....	101
4.15.1	Ubicación de los Productos.....	101
4.16	Fase III Actividad 4 Tarea 2 Implementación de Servicios Web Semántico de Objetos .....	101
4.17	Fase III Actividad 4 Tarea 3 Implementación de Servicios de Índice .....	102
4.18	Fase III Actividad 4 Tarea 4 Evaluación del índice construido .....	102
4.19	Fase IV Actividad 1 Tarea 1, 2 Elaboración del Plan de Restitución y Ejecutar el despliegue .....	102
4.20	Conclusiones del Capitulo.....	102
	VALIDACIÓN DEL PROTOTIPO .....	104
5.1	Modelo de Evaluación de SIMIOT .....	104
5.2	Desarrollo de Dispositivos Virtuales para las Pruebas .....	104
5.3	Curva de Precisión-Recuerdo (Precision-Recall): .....	105
5.4	Pruebas Alfa .....	106
5.5	Métricas Kappa Cohen .....	108
5.6	Métricas Kappa Fleiss .....	109
5.7	Prueba de usabilidad.....	110
5.7.1	Preguntas de usabilidad .....	110
5.7.2	Tabla de resultados de la prueba de usabilidad .....	111
5.8	Conclusiones de las Pruebas .....	112
	CREACIÓN DE SERVICIOS WEB SEMÁNTICOS DEL BUSCADOR SEMANTICO PARA LA IOT.....	114
6.1	Introducción .....	114
6.1.1	Coreografía de servicios .....	114
6.1.2	Orquestación de servicios.....	114
6.1.3	Composición de servicios .....	115

6.1.4 Enfoques de Coreografía, Orquestación y Composición de servicios .....	115
6.1.5 Mediación de servicios .....	115
6.2 Etapas para el diseño de SWS .....	115
6.2.1 Descubrimiento del Servicio .....	115
6.2.2 Definición del Servicio .....	116
6.2.3 Entrega del Servicio.....	116
6.3 Herramientas de Desarrollo de un SWS.....	116
6.4 Prototipo de Servicios Web Semánticos para SIMIOT .....	117
6.5 Diseño general del Servicio Web Semántico SIMIOT.....	117
6.6 Ubicación del diseño.....	117
6.6.1 Ontologías .....	118
6.6.2 Servicios .....	118
6.6.3 Metas.....	119
6.6.4 Mediadores entre Ontologías.....	119
6.6.5 Mediadores entre Servicios .....	119
7 CONCLUSIONES .....	120
BIBLIOGRAFIA.....	122
ANEXOS.....	128

## LISTA DE TABLAS

Tabla 1 Características de los Objetos .....	31
Tabla 2 Dimensiones de clasificación de los objetos del IoT .....	31
Tabla 3 Clasificación de los objetos del (IoT). .....	32
Tabla 4 Tecnologías del IoT .....	33
Tabla 5 Sistema de Estrellas para Publicar en la web .....	45
Tabla 6 Enfoques de Publicación en la web .....	46
Tabla 7 Modelos de RI.....	47
Tabla 8 Técnicas semánticas para la Búsqueda de datos de sensores .....	49
Tabla 9 Visión de BSIOT .....	73
Tabla 10 Convenciones de Servidores Mediadores de la IoT.....	88
Tabla 11 Comparación de Características de Servidores Mediadores de la IoT ...	88
Tabla 12 Tipos de Servidores Mediadores de la IoT.....	89
Tabla 13 Explicación de medidas de Precisión - Recuerdo .....	105
Tabla 14 Preguntas del dominio Contaminación.....	107
Tabla 15 Resumen de Precisión Promedio para cada consulta .....	107
Tabla 16 Tabulación de resultados Kappa.....	108
Tabla 17 Resumen de Kappa Cohen para cada consulta.....	108
Tabla 18 Resumen de Kappa Fleiss para cada consulta.....	109
Tabla 19 Aspectos y Preguntas de Usabilidad.....	110
Tabla 20 Diseño de Ontologías SIMIOT .....	118
Tabla 21 Diseño de Servicios SIMIOT .....	118
Tabla 22 Diseño de Metas SIMIOT .....	119
Tabla 23 Diseño de Mediador de Ontologías SIMIOT.....	119
Tabla 24 Diseño de Mediador de Conceptos SIMIOT.....	119
Tabla 25 Protocolo para dominio Hogar .....	133
Tabla 26 Middleware por Componentes Funcionales .....	136
Tabla 27 Descripción de las Capas del IoT .....	137
Tabla 28 Descripción de los Enfoques del IoT.....	138
Tabla 29 Middleware por Enfoques .....	138
Tabla 30 Cloud of Things.....	142
Tabla 31 Middleware por Capas VO .....	143
Tabla 32 Aspectos de concordancia entre Agentes y SO.....	144
Tabla 33 Middleware por Plataformas Multi-Agente .....	145
Tabla 34 Descripción de Tecnologías de la Web Semántica.....	146
Tabla 35 Tipos de datos del IoT.....	148

Tabla 36 Dimensiones de Diseño para aplicaciones en la IoT .....	149
Tabla 37 Técnicas para el Descubrimiento de entidades .....	150
Tabla 38 Servicio para construir aplicaciones del IoT .....	152
Tabla 39 Retos vs. Características de los Índices Semánticos en la IoT .....	157
Tabla 40 Tabla de conceptos del Modelo Medio Ambiental del Buscador SIMIOT .....	160
Tabla 41 Alineación de Ontologías SSN-XG y AWS.....	166
Tabla 42 Tabla de indicadores para selección de Middleware .....	171
Tabla 43 Evaluación del Servidor IoT a utilizar en el caso de estudio .....	172
Tabla 44 Evaluación del PS para cada servidor IoT seleccionado .....	173
Tabla 45 Contaminantes mas referenciados .....	199
Tabla 46 Limite maximo en tiempo de exposición .....	199
Tabla 47 Unidades utilizadas .....	200
Tabla 48 Niveles por contaminante.....	200
Tabla 49 Regiones Dpto Cauca con Sensores virtuales.....	200
Tabla 50 Regiones Dpto Santander con Sensores virtuales.....	202
Tabla 51 Simulación de Canales Xively .....	204
Tabla 52 Datos prueba de relevancia .....	207
Tabla 53 Excel Presición at K .....	210
Tabla 54 Formulas Excel Presición at K .....	210
Tabla 55 Recopilación de datos en la hoja Muestra en Excel.....	211
Tabla 56 Calculo de Presición at K .....	211
Tabla 57 Modelo Excel para Kappa Cohen .....	213
Tabla 58 Primera forma de calcular K. Cohen en Excel .....	214
Tabla 59 Segunda forma de calcular K. Cohen en Excel.....	214
Tabla 60 Kappa Cohen consulta 1 .....	214
Tabla 61 Kappa Cohen consulta 2.....	215
Tabla 62 Kappa Cohen consulta 3.....	215
Tabla 63 Kappa Cohen consulta 4.....	215
Tabla 64 Modelo Excel para Fleiss .....	216
Tabla 65 Formulas Excel para calcular K. Fleiss para 2 clases .....	217
Tabla 66 Formulas Excel para calcular K. Fleiss para 3 clases.....	217
Tabla 67 Kappa Fleiss consulta 1 y 2 .....	217
Tabla 68 Kappa Fleiss consulta 3 y 4 .....	218

## LISTA DE FIGURAS

Figura 1 Modelo de Conceptos del IoT .....	28
Figura 2 Relación IoT y otras redes .....	29
Figura 3 Propiedades de la Red de Objetos .....	30
Figura 4 Arquitectura del IoT.....	34
Figura 5 Índice de Anotaciones del NLP .....	39
Figura 6 Anotación en base a NLP .....	40
Figura 7 Anotaciones de Información Semántica Contextual .....	41
Figura 8 Anotación en base a la Información Semántica Contextual.....	42
Figura 9 Adaptación del clásico modelo de Espacio Vectorial .....	48
Figura 10 Arquitectura de Índices Semánticos en la IoT.....	53
Figura 11 Diagrama de Actividad SIMIOT de EPF Composer .....	69
Figura 12 Arquitectura Lógica del Buscador Semántico IoT .....	78
Figura 13 Modelo General de Casos de Uso .....	79
Figura 14 Diagrama Metodológico y de Componentes de la Ontología de Interacción Semántica WoT .....	91
Figura 15 Modelo de Dependencia de Ontologías.....	95
Figura 16 Resultado encuesta de Usabilidad.....	111
Figura 17 Internet de Objetos convergencia de visiones .....	130
Figura 18 Componentes Funcionales de Middlewares .....	135
Figura 19 Enfoques Middleware .....	137
Figura 20 Arquitecturas Middleware .....	139
Figura 21 Orientado a SOA .....	140
Figura 22 Orientado a Grid/SOA.....	142
Figura 23 Organizaciones Virtuales .....	143
Figura 24 Arquitectura de la Web Semántica.....	146
Figura 25 Triple RDF .....	148
Figura 26 Diagrama de actividad con desglose de tareas .....	158
Figura 27 Diagrama detallado de actividad con roles, tareas y productos de trabajo.....	158
Figura 28 Diagrama de dependencias de productos de trabajo. ....	159
Figura 29 Publicación web de SIMIOT.....	159
Figura 30 Mapa Mental del Modelo Medio Ambiental del proyecto.....	162
Figura 31 Vista Rápida de Alineación a la Ontología ENVO.....	165

Figura 32 Modelo Conceptual del Objeto Semántico Tomado Tesis doctoral Miguel Angel Niño Zambrano .....	166
Figura 33 Diagrama de Componentes para SIMIOT.....	188
Figura 34 Diagrama de Secuencia de Interfaces para SIMIOT.....	189
Figura 35 Interfaz de Inicio para SIMIOT .....	190
Figura 36 Interfaz de Búsqueda para SIMIOT .....	191
Figura 37 Interfaz de Biotipo para SIMIOT .....	192
Figura 38 Búsqueda de Sensores por Variables Ambientales y Biotipo .....	193
Figura 39 Búsqueda de Sensores por Contaminación del Aire y por Biotipo .....	194
Figura 40 Modelo de secuencia de servicios .....	222
Figura 41 Modelo semántico en WSMO .....	223
Figura 42 SIMIOT en la herramienta WSMT.....	223
Figura 43 Modelo de Ontologías en WSMO .....	224
Figura 44 Crear nueva Ontologías en WSMT .....	225
Figura 45 Modelo de Conceptos en WSMO .....	226
Figura 46 Representación grafica de Concepto en WSMT.....	226
Figura 47 Representación grafica de Atributos en WSMT .....	228
Figura 48 Representación grafica de Relaciones en WSMT .....	229
Figura 49 Modelo de Parametros en WSMO .....	230
Figura 50 Representación grafica de Parametros en WSMT.....	230
Figura 51 Modelo de Axiomas en WSMO.....	231
Figura 52 Representación grafica de Axiomas en WSMT .....	232
Figura 53 Modelo de Servicios en WSMO .....	234
Figura 54 Modelo de Capacidades en WSMO.....	234
Figura 55 Modelo de Capacidades de servicio solicitante .....	235
Figura 56 Modelo de Capacidades de servicio búsqueda .....	237
Figura 57 Modelo de Capacidades de servicio concepto.....	238
Figura 58 Modelo de Metas para WSMO.....	239
Figura 59 Modelo de Mediador para WSMO .....	240
Figura 60 Representación grafica para mediador de ontologias.....	241
Figura 61 Representación grafica para mediador de servicios .....	241

## LISTA DE ANEXOS

ANEXO A MARCO REFERENCIA.....	130
ANEXO B METODO DE INDEXACIÓN SEMANTICA EN LA IOT - SIMIOT .....	157
ANEXO C IMPLEMENTACIÓN DEL BUSCADOR SEMÁNTICO EN LA IOT .....	160
ANEXO D VALIDACIÓN DEL PROTOTIPO .....	195
ANEXO E CREACIÓN DE SERVICIOS WEB SEMÁNTICOS .....	220

## RESUMEN

**TITULO:** MÉTODO DE INDEXACIÓN SEMÁNTICA EN INTERNET DE LAS COSAS

**AUTOR:** JESUS ALBERTO NIÑO ZAMBRANO

**PALABRAS CLAVE:** Internet de las Cosas, Indexación Semántica, Método, Ontologías.

**DESCRIPCION:** El futuro de la Internet de las cosas depende en gran medida de la capacidad de gestionar adecuadamente los datos y la información producida por los sensores. En este trabajo se presenta un método para la indexación semántica en la IOT y la construcción de un motor de búsqueda inteligente a partir de estructuras de conocimiento mediado por ontologías y adaptado a las necesidades de información de los usuarios en la web de las cosas. Se implementaron servicios web para presentar todas las funciones del buscador semántico para que sean invocados por cualquier aplicación, se realizó un diseño de servicios web semánticos para automatizar el proceso de búsqueda, selección e invocación de los servicios web que presentan las funciones del buscador semántico, de esta manera se presentan servicios no solamente para usuarios a través la aplicación web sino que también para aplicaciones clientes en la web. La implementación de tecnologías semánticas en la IOT apoya la escalabilidad y la interoperabilidad de los dispositivos superando la heterogeneidad y haciendo posible la gestión de grandes cantidades de datos y crear servicios útiles para los usuarios. Los servicios desarrollados para la IOT corresponden a búsqueda y procesamiento semántico en el dominio de la contaminación medioambiental, se analizan los dispositivos almacenados en portales web que hacen parte de la infraestructura de la IOT, esto posibilita la reutilización de estos recursos por la usuarios de la web los cuales pueden ser otras aplicaciones.

---

\* **Trabajo de Grado**

\*\* **Facultad:** Ingenierías Físico – Mecánicas. Escuela de Ingeniería de Sistemas.

**Director:** Msc. Luis Carlos Gómez Flórez.

**Codirector:** Msc. Miguel Ángel Niño Zambrano.

## ABSTRACT

**TITLE:** SEMANTIC INDEXING METHOD IN INTERNET OF THINGS

**AUTHOR:** JESUS ALBERTO NIÑO ZAMBRANO

**KEYWORDS:** Internet of Things, Semantic Indexing, method ontologies.

**DESCRIPTION:** The future of the Internet of things depends largely on the ability to properly manage data and information produced by the sensors. In this work a method for semantic indexing in IOT and building a smart search engine from knowledge structures mediated by ontologies and tailored to the information needs of users in the web of things arises. Web services to present all the features of the semantic search engine to be invoked by any application implemented a semantic web services design was conducted to automate the process of search, selection and invocation of web services that have semantic search functions, thus services are presented not only to users through the Web application but also for web client applications. The implementation of semantic technologies in IOT supports scalability and interoperability of devices exceeding the heterogeneity and enabling the management of large amounts of data and create useful services for users. Services developed to correspond to the IOT search and semantic processing in the domain of environmental pollution, the devices stored in web sites that are part of the infrastructure of the IOT are discussed, this enables the reuse of these resources by users web which may be other applications.

---

\* **Bachelor Thesis**

\*\* **Facultad:** Ingenierías Físico – Mecánicas. Escuela de Ingeniería de Sistemas.

**Director:** Msc. Luis Carlos Gómez Flórez.

**Codirector:** Msc. Miguel Ángel Niño Zambrano.

# INTRODUCCIÓN

## 1.1 Contexto General

La Web es una red distribuida en permanente evolución, la información semi-estructurada de la web tradicional es de difícil reutilización, por ello, con la vinculación e integración de información que logró la web semántica, se hizo posible automatizar tareas de manera inteligente.

Actualmente, la red Internet ha evolucionado en varias nuevas formas, una de ellas es la conexión de las cosas (objetos) a través de sensores y dispositivos de comunicación, este es llamado el internet de las cosas (Internet of Things – IoT)(Atzori, Iera et al. 2010). El IoT es un escenario complejo que integra varias tecnologías y soluciones de comunicación, sin embargo la falta de estándares(Handong and Lin 2011) en dispositivos de entrada y salida, interfaces hardware y software, y en observaciones y flujos de datos, plantea la necesidad de proveer abstracciones para superar la heterogeneidad semántica(Chaqfeh and Mohamed 2012).

Los datos y servicios provistos por la IoT se les denomina la Web de las Cosas (Web of Things – WoT), éstos datos se generan en el orden de exabytes, constituyendo como prioridad la necesidad de la adquisición de la información relevante a través del uso de técnicas semánticas de recuperación de información y minería de datos. Una estrategia para afrontar las dificultades técnicas en la IoT se ha realizado con la implementación de componentes software denominados middleware(Bandyopadhyay, Sengupta et al. 2011) que por medio de anotaciones semánticas reconcilian las diferencias entre formatos e interfaces, además los datos se comunican en la WoT a través de arquitecturas: REST, SOA y estándares como: XML, JSON entre otros.

El presente trabajo se enfocó en la necesidad de encontrar información útil en la WoT con el fin de que los usuarios puedan reutilizarla, se construyó un motor de búsqueda que permite incorporar la indexación semántica basada en los servicios(Gigli and S. 2011) existentes en la WoT, y con el uso de portales web como fuente de información de datos. Dado que la indexación semántica es eficiente en la búsqueda en grandes cantidades de datos, se diseñó un método para desarrollar un buscador con indexación semántica en el campo medioambiental cumpliendo con el estándar de desarrollo SPEM v2.0, que permite

que el método sea reutilizable; adicionalmente, se desplegaron servicios web para dar apoyo a las aplicaciones que requieran servicios de datos del índice. Finalmente, se realizó un diseño de Servicios Web Semánticos – SWS con el objetivo futuro de automatizar tareas de búsqueda e invocación de servicios web. El buscador está enfocado a la consulta de información medio ambiental concretamente a la calidad del aire, alineando ontologías estándar de la WoT. La evaluación del motor de búsqueda se llevó a cabo con medidas e indicadores para la recuperación de la información para validar motores de búsqueda web.

## **1.2 Declaración del Problema**

La pregunta de investigación del presente proyecto es: ¿Cómo aprovechar adecuadamente la información provista por los objetos del Internet de Objetos para satisfacer los requerimientos de información de los usuarios?

## **1.3 Hipótesis de Investigación**

Una posible solución es añadir a los objetos información semántica y la posibilidad de crear nuevos servicios de información dentro del WoT. En el campo de la recuperación de la información se han utilizado técnicas que integran ontologías para incorporar conocimiento a los recursos web con el uso de servicios web, y así poner a disposición de los usuarios la información de los objetos físicos. El presente proyecto pretende aplicar éstos conceptos mediante la construcción de un motor de búsqueda semántica para el WoT, el enfoque del proyecto se basa en la indexación y búsqueda de datos de objetos físicos localizando las entidades de sensores de acuerdo a ciertos requerimientos de búsqueda, un motor de búsqueda para el WoT tiene que apoyar la búsqueda de contenido estructurado lo cual se puede solucionar utilizando bases de conocimiento como ontologías, por ultimo para apoyar la interoperabilidad el motor implementa un sistema de búsqueda basado en servicios web.

## **1.4 Escenario de Motivación**

Los avances en el desarrollo de dispositivos móviles y sistemas distribuidos provocó la aparición del concepto de computación ubicua<sup>1</sup>(Friedewald and Raabe 2011), la cual pretende integrar a las tecnologías de la Información y la Comunicación - TIC<sup>2</sup> y el entorno de la vida diaria de las personas. El desarrollo

---

<sup>1</sup> Tecnología Ubicua es un concepto creado por Mark Weiser en The Computer for the Twenty-First Century, propuso que la tecnología ubicua es introducir a los ordenadores en el mundo real de las personas como dispositivos que interactúan físicamente con las personas.

<sup>2</sup> TIC tecnologías de la información y la comunicación: conjunto de elementos y técnicas para el tratamiento y transmisión de la información.

de dispositivos de Radio Frequency Identification – RFID, hardware de sensores, y el avance de la nanotecnología, han establecido nuevas formas de comunicación: entre personas, personas y dispositivos o también dispositivos que se conectan con otros dispositivos.

La interconexión de dispositivos, es decir objetos físicos, ha trascendido a la web, y se le conoce como Web de las Cosas. Los objetos físicos reciben una identidad digital o etiqueta para ser identificados como recursos digitales, por ejemplo el teléfono móvil despliega la interfaz para interactuar con sensores de temperatura y humedad en las diferentes ciudades y así obtener un reporte del clima en cualquier momento, a esto se le denomina Physical Mobile Interaction - PMI(Vetter, Hamard et al. 2007).

La interacción de estos dispositivos con las personas y con su entorno lleva al concepto de productos inteligentes(Kiritsis 2011), los cuales deben cumplir requerimientos de diseño y desarrollo tales como: contener sensores, manejo adecuado de memoria, procesamiento inteligente de datos, capacidad de razonamiento, capacidad de comunicación y varios niveles de inteligencia.

Para poder aprovechar convenientemente la gran cantidad de sensores y dispositivos conectados a Internet, es necesario construir mecanismos de búsqueda y recuperación de información, esta información está relacionada a las características y servicios que pueden ofrecer los objetos y que puedan cumplir con las necesidades de sus potenciales usuarios que pueden ser personas y productos inteligentes.

## 1.5 Contribuciones

- **Revisión, análisis y aporte a la formalización estado del arte sobre la IoT:** El estudio de la IoT está en pleno auge y se siguen haciendo esfuerzos para documentar y normalizar temas de vital interés en el contexto de la IoT, aunque existe abundante documentación de aplicaciones prácticas, los conceptos básicos y revisiones de tema son escasas y generalizadas lo que acarrea un gran esfuerzo para comprender claramente las líneas de investigación y los beneficios contenidos en la IoT.
- **Formalización de un método de indexación semántica reutilizable que sigue el estándar SPEMv2.0:** La indexación semántica es una herramienta utilizada en la recuperación de información de recursos web de difícil

clasificación e identificación en la web sintáctica<sup>3</sup> como información multimedia, imágenes, videos y servicios web. La WoT contienen una gran cantidad de datos que por sus características (flujos en tiempo real de información bruta y no estructurada) son candidatos para ser indexados semánticamente, sin embargo este proceso no es sencillo de realizar, por ello se construyó un método que permite a un grupo de usuarios y desarrolladores reutilizarlo y aplicarlo para indexar la información de la WoT de acuerdo a sus propias necesidades con el objetivo de facilitar este proceso.

- **Desarrollo de un prototipo software que crea un índice semántico para la búsqueda de información en la WoT:** Utilizando el método de indexación semántica propuesto, se implementó un índice semántico que expone la información a través de servicios web. Para el caso de estudio del proyecto, se creó una ontología de contaminación ambiental y se alinearon con ontologías estándar de la WoT y en el campo medioambiental. Un elemento importante, es que el índice puede cambiar su dominio de aplicación cambiando la ontología de dominio correspondiente. Finalmente, se construyó un buscador web que consulta el índice y realiza un despliegue de la información de sensores que podrían medir la contaminación ambiental en un lugar geográfico, soportada en el servidor IoT llamado Xively. Para la construcción del índice se crearon de servicios web para exponer su funcionalidad de manera estándar a otras aplicaciones. Posteriormente se creó un Buscador Web que permite encontrar información de los sensores relacionados a una consulta de usuario realizada en lenguaje natural en el campo de contaminación medioambiental. Para lo anterior se utilizaron técnicas de ontologías, uso de middleware de objetos e interfaces de aplicación (Application Programming Interface – API) de desarrollo del IoT.
- **Desarrollo de sensores virtuales que simulan información en el ámbito de la contaminación ambiental en dos regiones de Colombia:** Se Implementó 60 canales que generan información de contaminación siguiendo las normas<sup>4</sup> Colombianas para niveles permitidos de contaminación del aire y tipos principales de contaminantes del aire.

---

<sup>3</sup> Web sintáctica: Se refiere a que los recursos están enlazados entre sí, sin tener en cuenta lo que significa o representa cada recurso.

<sup>4</sup> \*Normas ambientales mínimas y dictar regulaciones de carácter general para controlar y reducir la contaminación atmosférica en el territorio nacional (Numerales 10, 11 y 14 Artículo 5º de la Ley 99 de 1993).

- **Diseño de Servicios Web Semánticos SWS para el Índice Semántico Creado:** Utiliza ontologías para las coreografías de servicios que sirven como anotaciones semánticas a los servicios web creados en el prototipo de índice semántico en la IoT, alineada con la ontología de contaminación ambiental creada. Se utilizaron herramientas CASE para el desarrollo de SWS. Su utilidad es con el fin de apoyar futuros despliegues en servidores de SWS para búsqueda e invocación de servicios de manera automatizada.
- **Ponencia Internacional:** Método de Indexación Semántica en Internet de las Cosas en el Workshop Semantic Web and Linked Open Data: En la Mexican International Conference on Computer Science ENC 2013 en Morelia, Michoacán, México Octubre 30, 31 y Noviembre 1st, del 2013.
- **Pasantía de investigación:** Diseño desarrollo e implementación de 7 prácticas que pasaran a hacer parte de un libro guía para la materia de Tecnologías Ubicuas de la Universidad del Cauca.

## 1.6 Alcance

Uno de los retos que se afrontaron en el presente proyecto fue aplicar técnicas de indexación semántica y recuperación de la información en la WoT con el fin de aportar a la solución del manejo y reutilización de información producida por los sensores. Así el presente proyecto presento un método de indexación semántica en la IoT que puede ser reutilizado para crear índices semánticos en cualquier dominio, adicionalmente provee un buscador web semántico en el área de la contaminación medioambiental y soportado sobre el servidor de objetos Xively, permitiendo indexar y recuperar su información a través de búsquedas formuladas en lenguaje natural, finalmente de implementó un diseño de SWS que pueden desplegarse en un trabajo posterior con el fin de automatizar la utilización de sus servicios web en el futuro.

## 1.7 Objetivos

### 1.7.1 Objetivo General

Crear un motor de búsqueda semántica para el Internet de Objetos - IoT aplicando los conceptos de Servicios Web Semánticos y Ontologías.

---

\*\*Norma nacional de calidad del aire, o nivel de inmisión, para todo el territorio nacional (Artículos 6º, 10 y 12 del Decreto 948 de 1995).

### **1.7.2 Objetivos Específicos**

- Definir un método de almacenamiento y recuperación de información sobre Servidores Mediadores existentes en el IoT.
- Implementar un buscador semántico que utilice ontologías con el fin de encontrar y reutilizar la información producida por objetos del IoT.
- Validar el motor de búsqueda semántica de IoT con medidas de precisión y recuerdo de recuperación de la información, se toma como caso de estudio los objetos de servidores mediadores del IoT como Microsoft SenseWeb y Cosm.

## **1.8 Contenido de la Monografía**

- **CAPITULO II. MARCO REFERENCIAL**

Se establecen los paradigmas conceptuales del tema central, los núcleos temáticos son la recuperación de la información, internet de objetos y web semántica. Identificando los referentes teóricos para la constitución del estado del arte, así como las investigaciones relacionadas.

- **CAPITULO III. METODO DE INDEXACIÓN SEMANTICA EN LA IOT - SIMIOT**

Se presenta el diseño y desarrollo de un método que sirve como guía para construir aplicaciones en la WoT, orientado al manejo del conocimiento a través de ontologías con un modelo semántico de recuperación de la información. Finalmente, realiza su especificación utilizando el estándar SPEM 2.0 con el fin de hacerlo fácilmente reutilizable y ampliable.

- **CAPITULO IV. BUSCADOR SEMANTICO EN LA WOT ACERCA DE LA CONTAMINACION AMBIENTAL**

Se presenta el diseño y desarrollo de un motor de búsqueda semántica para la IoT, basado en el método definido en el capítulo anterior, para esto se desarrolló un índice semántico que expone servicios de búsqueda de información sobre la infraestructura del IoT existente, específicamente el servidor Xively (antes Cosm). Presenta una interfaz en lenguaje natural en dos idiomas (español – inglés) para realizar consultas y fue creado utilizando un modelo arquitectónico en capas donde se instancian por módulos.

- **CAPITULO V. VALIDACIÓN DEL BUSCADOR: SEMANTIC SEARCH ON INTERNET OF THINGS**

El capítulo realiza la evaluación del buscador desarrollado. Para evaluar los detalles cuantitativos del proceso de recuperación de la información se utilizó un conjunto de medidas de relevancia con usuarios finales.

- **CAPITULO VI. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO**

Se toman los resultados obtenidos y se plantean conclusiones haciendo reseñas y recomendaciones sobre escenarios en los que se desarrollaron los productos y se presentan propuesta de trabajos futuros los cuales aprovechen los beneficios de los productos desarrollados.

# MARCO REFERENCIAL

## 2. Introducción

En este capítulo se pretende hacer referencia a los principales conceptos que encierra la IoT, haciendo énfasis en sus retos y desafíos, además, se describen los fundamentos teóricos y conceptuales que permitieron reunir los elementos esenciales para desarrollar la aplicación de indexación semántica y el motor de búsqueda semántica en la IoT del presente proyecto.

### 2.1. Internet de Objetos

Los avances en las TIC, han permitido dejar de lado las barreras geográficas y temporales para transmitir información, creando consigo una nueva revolución de pensamiento y maneras de hacer las cosas. La Internet ha evolucionado a tal punto que en nuestro estilo de vida actual se ha convertido en una necesidad básica para poder interactuar con la sociedad. Los avances en la telefonía móvil y los servicios Web en los últimos años han marcado la vida de las personas cambiando la manera de comunicarnos, recrearnos, trabajar y hacer vida social entre otros aspectos.

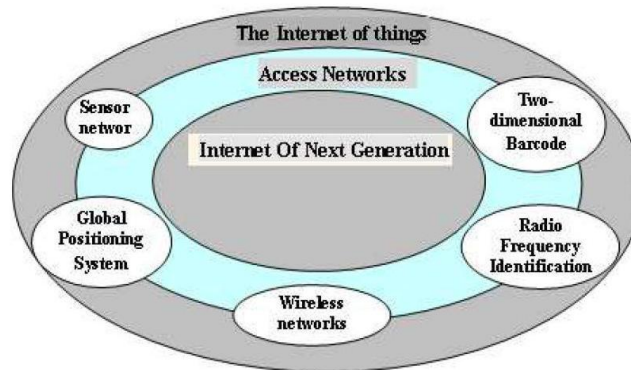
Normalmente las personas vivimos en una realidad mediada por nuestra relación con los (objetos físicos y conceptuales) y como a través de ellos intervenimos en dicha realidad, parece lógico pensar, que la siguiente evolución natural de la Internet es que éstos objetos se conecten a la misma, para que se comuniquen con nosotros y entre ellos mismos, con el fin de crear mejores servicios y bienestar; es precisamente lo que se busca en el Internet de Objetos (Gubbi, Buyya et al. 2013).

#### 2.1.1 Modelo de Conceptos de la IoT

El IoT se le ha denominado la “*Tercera Ola de la Información*” en el mundo de la industria después de la computadora y el Internet. El “*Internet de las Cosas*” se refiere a la información proveniente de dispositivos de detección tales como redes de sensores inalámbricos (Wireless Sensor Network – WSN), tecnologías de Identificación por radiofrecuencia (*Radio Frequency Identification – RFID*), código de barras bidimensionales, códigos de respuesta rápida (Quick Response Codes – QR códigos), sistemas de posicionamiento global (Global Positional Systems - GPS) y redes de corto alcance ad hoc, que se basan en el concepto de comunicación de máquina a máquina ( *Machine to Machine - M2M*) combinando

una variedad de redes de acceso y el internet para conformar una gran red inteligente, según la Figura 1 según (Yuxi and Guohui 2012) ver (Figura 1).

Figura 1: Modelo de Conceptos del IoT



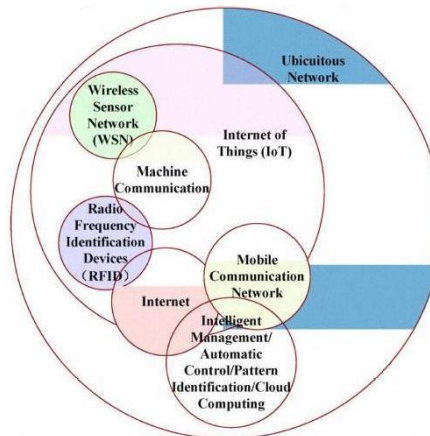
Tomado de: Key Technologies and Applications of Internet of Things, (Yuxi and Guohui 2012)

Según la Unión Internacional de Telecomunicaciones (International Telecommunication Union - ITU) en 2005 existen otros modelos de conceptos (Miao and Bu 2010) del IoT como la Web de Servicios, Sensores Infrarrojos y el Escáner Laser adicionales a los ya expuestos por Yuxi como RFID y GPS.

### 2.1.2 Relación entre la IoT y las redes existentes

La IoT ha aportado en la optimización del recurso humano, los recursos medio ambientales, gestión de la seguridad urbana, la gestión de la producción y en general la mejora de la calidad de vida; en la Figura 2 según (Miao and Bu 2010) establece la forma en que se relacionan la IoT y las demás redes.

Figura 2: Relación IoT y otras redes



Tomado de: Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid, (Miao and Bu 2010)

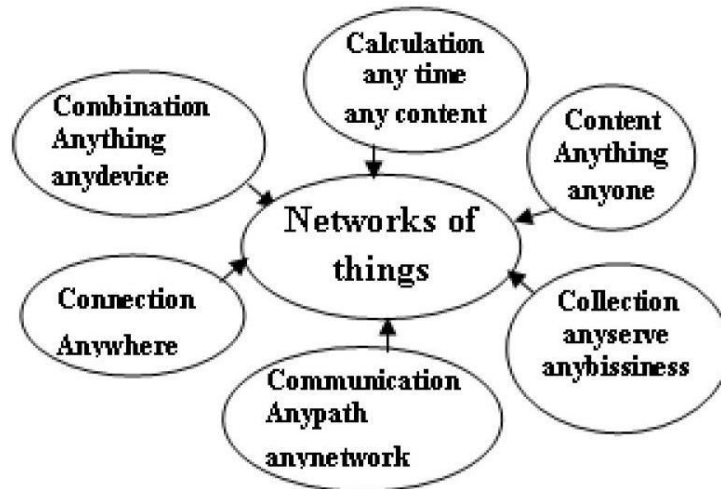
La figura es interpretada siguiendo tres aspectos principales:

- **Sentido Integral:** Uso de tecnología RFID, sensores, códigos de dos dimensiones para recoger información en cualquier momento y en cualquier lugar.
- **Transmisión Fiable:** Obtención de información de los objetos referente al concepto de tiempo real a través de redes de telecomunicaciones e internet.
- **Procesamiento Inteligente:** Usa del cómputo inteligente, difuso y computación en la nube para identificar, analizar y procesar grandes cantidades de datos e información, todo esto con el propósito de control inteligente de los objetos.

### 2.1.3 Propiedades de la Red de Objetos

El IoT extiende el mundo físico hacia ubicuidad de (tiempo – espacio) según (Yuxi and Guohui 2012) En la Figura 3 se incluyen propiedades de: contenidos, recopilación, comunicación, conectividad, combinación y cálculos, con el fin de permitir la construcción de entornos complejos de objetos y personas los cuales se pueden comunicar e interactuar entre sí para colaborar.

Figura 3: Propiedades de la Red de Objetos



Tomado de: Key Technologies and Applications of Internet of Things, (Yuxi and Guohui 2012)

Según (Tao and Dongxin 2012) formalmente podemos decir que el IoT busca lograr conectividad entre cualquier objeto o cosas y además entre las personas y los objetos, todo el tiempo y en cualquier momento y lugar. Estas capacidades requieren dispositivos que puedan interactuar con el medio que los rodea como los sensores, e introducir capacidades de inteligencia y conectividad la cual ha creado servicios y soluciones tecnológicas para el hogar y la industria.

## 2.2 Convergencia de Visiones en la IoT

La IoT está compuesta por diversas visiones las cuales están orientadas a las cosas, a la semántica y al internet, en el la (Tabla 1) se presenta una descripción de la intersección entre estas visiones las cuales nos ayuda a hacernos una idea más detallada de como todas las tecnologías desarrolladas para cada ámbito se relaciona en la IoT para crear conceptos nuevos sobre las relaciones e interacciones que se plantean entre el hombre y las tecnologías de la información.

## 2.3 Características de los objetos

Los objetos del IoT se les puede atribuir el desempeño de una función o rol (Yuxi and Guohui 2012), dependiendo de sus capacidades informáticas tales como: conectividad de red, energía disponible, escenarios de tiempo y espacio entre otras, dependiendo de sus características individuales, efectos producidos y relaciones que se plantean. En la Tabla 1 se puede apreciar con más detalle dichas características.

Tabla 1: Características de los Objetos

Características	Descripción
<b>Básicas</b>	Un objeto puede ser un objeto del mundo real o del mundo virtual puede haber comunicación entre el mundo físico real y el mundo virtual generando una información de su estado.
<b>Generales (Adicionales a las características funcionales básicas)</b>	Un objeto puede utilizar un servicio como interfaz para comunicarse con otro objeto, un objeto contiene sensores y actuadores para interactuar con el medio ambiente.
<b>Sociales</b>	Un objeto puede Interactuar con otro objeto y con personas generando una información semántica de dicha relación.
<b>Autonomía de funciones</b>	Un objeto realiza tareas autónomas y capacidad de razonamiento, capacidad de interactuar y aprender de otros objetos y el medio ambiente.
<b>Auto replicación y control de funciones</b>	Red de información con capacidades de detección para establecer relaciones entre los humanos y sistemas físicos.

### 2.3.1 Taxonomía de Objetos de la IoT

El Objeto es un concepto básico que considera a cada nodo como un objeto lógico siempre conectado y con información asociada o generada. Las dimensiones de clasificación de los objetos en el (IoT) depende de las propiedades de los objetos de la esperanza de vida de sus estados dinámicos de los atributos intrínsecos y servicios según (Mathew, Atif et al. 2011) realiza su clasificación de acuerdo a las capacidades intrínsecas y se presentan en la Tabla 2.

Tabla 2: Dimensiones de clasificación de los objetos del IoT

Capacidades Intrínsecas	Descripción	Ejemplos
<b>Identidad</b>	Identificación de los objetos.	Código de barras, códigos RFID.
<b>Procesamiento</b>	Control y administración de los objetos.	Uso de chip, uso de sistemas operativos embebidos.
<b>Comunicación</b>	Interfaces de entrada y salida, medios de comunicación y protocolos.	Puertos USB, Bluetooth, APIs de programación web.
<b>Almacenamiento</b>	Capacidad de almacenamiento para los estados de los objetos y los valores.	Almacenamiento del identificador del objeto por ejemplo un código RFID, almacenamiento para interfaces de comunicación de puertos y direccionamiento, memoria RAM y discos duros HDD.

Los objetos de la (IoT) se clasifican teniendo en cuenta aspectos como la Longevidad (Lifespan), Propietario (Ownership), Compartibilidad (Shareability), Amigos (Friends), Buscabilidad y Accesibilidad (Searhability and accessibility),

también se pueden clasificar según las características núcleo (core), Primitivo (Primitive), Complejo (Complex) inteligente (Smart).

Según las capacidades de los objetos se pueden clasificar de la siguiente manera en la Tabla 3.

Tabla 3: Clasificación de los objetos del (IoT).

Capacidades		Capacidades Intrínsecas				Ejemplos
		Identidad	Procesamiento	Comunicación	Almacenamiento	
Core		√				RFID o códigos de barras.
Primitive	Fuzzy	√	√			Lavadora, hornos microondas.
	Plug	√		√		Altavoces, auriculares.
	Fat	√			√	CD, DVD.
Complex	Social	√	√	√		Control remoto, teléfonos fijos.
	Sticky	√		√	√	USB stick, RFID Tags.
	Gizmo	√	√		√	Calculadora, juegos de mano.
Smart Things		√	√	√	√	PDA's, PCs.

Teniendo en cuenta las características de los objetos, las propiedades del IoT y la intercepción de las tres visiones que intervienen en el IoT, podemos identificar los temas principales que se deben abordar al momento de crear una red de objetos en el IoT, además de permitir reconocer los elementos básicos que se pueden utilizar en el IoT. Se puede resumir éstos elementos en cinco preguntas:

1. ¿Qué tipo de Objetos se desea crear de acuerdo a sus características?
2. ¿Qué propiedades de la red de Objetos se van a implementar?
3. ¿Qué tecnologías para representar e identificar los objetos se debe utilizar?
4. ¿Qué servicios se desea desarrollar en la Web de Objetos?
5. ¿Qué capacidades semánticas tiene los objetos?

### 2.3.2 Tecnologías del Internet de Objetos

El concepto de Planeta Inteligente (Smart Planet)(Tao and Dongxin 2012) es propuesto en el año 2008 por IBM, el cual se centra en el concepto de sabiduría y

su implantación en el IoT. Las tecnologías clave son el RFID (Radio Frequency Identification), el (Wireless Sensor Network – WSN), los (Global Positioning Systems – GPS), redes móviles, servicios web (Web Service – SW), aplicaciones industriales y el desarrollo de chips son componentes del concepto Smart Planet. Según (Miao and Bu 2010) las tecnologías de la IoT y la relación de las distintas clases de redes se resumen en la Tabla 4.

**Tabla 4: Tecnologías del IoT**

<b>Tecnologías</b>	<b>Especificaciones</b>
<b>RFID technology / Tecnología RFID</b>	La inducción y la propagación electromagnética evita el contacto físico, ha desarrollado seguridad anti replica, cifrado y certificados.
<b>Sensor Network / Red de sensores</b>	Nueva plataforma para censar el mundo con un gran número e nodos que interactúan, posee aplicaciones en la defensa, industria, agricultura y funcionamiento de las ciudades.
<b>Smart technology / Tecnología inteligente</b>	Los objetos reciben inteligencia para comunicarse activa y pasivamente, esta tecnología entra en el campo de la (Inteligencia Artificial – IA), interacción (Maquina humano – M2H) y tecnologías de control inteligente y procesamiento inteligente de señales.
<b>Nanotechnology</b>	Estudio de partículas diminutas y objetos que interactúan y se conectan se han aplicado en campos como la medicina, energía y transporte.

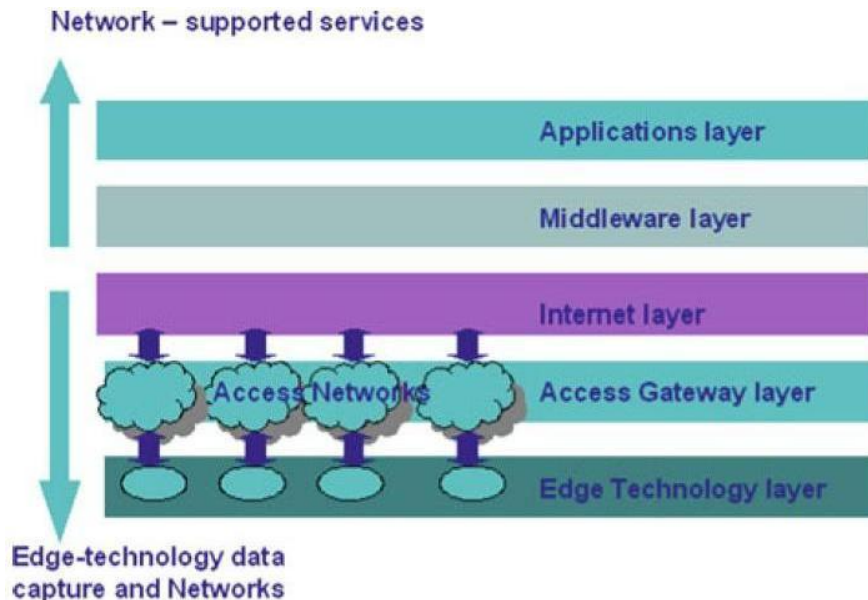
### 2.3.3 Aplicaciones del IoT

La IoT está desarrollando una infraestructura tecnológica de grandes proporciones, la aplicabilidad de estas tecnologías son múltiples e innumerables, en el (Anexo ) se presenta una colección de los campos en donde las aplicaciones de la IoT han sido más significativas.

### 2.3.4 Arquitectura de la IoT

Se han realizado varias propuestas alrededor de la arquitectura de la IoT, según (Bandyopadhyay and Sen 2011) en la Figura 4 muestra que sin embargo existe una estrategia de capas bien definidas, en la cual, las capas inferiores adquieren los datos y las capas superiores utilizan los datos en las aplicaciones y la capa del medio se convierte en un mediador común de comunicación.

Figura 4: Arquitectura del IoT



Tomado de: Internet of Things: Applications and Challenges in Technology and Standardization, (Bandyopadhyay and Sen 2011)

La descripción general de cada una de las capas es como sigue:

- **Edge Technology Layer / Tecnología capa de borde:** En la primera capa de frontera, están los objetos físicos dotados de las tecnologías de identificación como: etiquetas RFID, sensores para recoger información del entorno y lectores para almacenamiento de datos, además de sistemas embebidos para procesamiento y comunicación.
- **Access Gateway Layer / Acceso a capa de entrada:** La capa de acceso o puerta de enlace se realiza el enrutamiento de mensajes, publicación y suscripción. Esta capa permite tener acceso a la información recopilada por los objetos específicos hacia la red (capa Internet), ocultando los detalles técnicos o físicos de transporte y conversión de datos.
- **Internet Layer / Capa de Internet:** En la (capa Internet) los datos de los objetos se transportan a través de protocolos IP, almacenándose en servidores de información físicos y accesibles por un software intermediario (Middleware), que pondrá a disposición de las aplicaciones los mismos.
- **Middleware Layer / Capa de middleware:** La capa intermedia es una interfaz entre la capa de componentes físicos y la capa de aplicación, se lleva a cabo la gestión de dispositivos, datos, filtrado, agregación y análisis semántico. Un ejemplo es el descubrimiento de información a través de formas de servicios como de código de producto electrónico (*Electronic*

*Product Código – EPC*) y sus servicios web de información de nombres objetos (*Object Naming Service – ONS*).

- **Application Layer / Capa de aplicación:** Finalmente en la capa de aplicación se encuentran aplicaciones en producción, logística, medio ambiente, seguridad pública y salud entre otras, además, con el creciente desarrollo de la tecnologías RFID se están integrando nuevas aplicaciones.

## **2.4 Middleware de la IoT**

Según (Zhou 2012) un middleware “*es el desarrollo de sistemas en tiempo de ejecución que permite interacciones a nivel de aplicación entre programas en un entorno de computación distribuida*”, estas interacciones pueden ser síncronas o asíncronas, los middlewares logran que las aplicaciones se ejecuten en múltiples plataformas logrando interoperabilidad entre aplicaciones, escalabilidad, portabilidad, independencia de la ubicación y oculta la heterogeneidad de hardware en ambientes distribuidos.

Los desafíos (Handong and Lin 2011) del IoT radican en que los recursos de sensores son profundamente heterogéneos y se consumen en tiempo real, están en redes distribuidas cambiantes, además presentan necesidades de limpieza de datos debido a condiciones ambientales y también las limitaciones de los objetos de la baja capacidad de proceso y consumo de energía, la cantidad de datos y falta de integración de datos heterogéneos y de aplicaciones de minería de datos, búsqueda en indexación.

### **2.4.1 Clases de Taxonomía de Middleware**

En el presente estudio en el **ANEXO A-3** se pretende describir y clasificar los middlewares de acuerdo a los diferentes aspectos funcionales y técnicos, un mismo middleware puede estar en varias taxonomías según corresponda, estas clasificaciones pretenden hacer una descripción de cada uno los middlewares e identificar las estrategias, técnicas y arquitecturas desplegadas en la IoT las cuales son soluciones ya probadas que puedan ser incorporadas al desarrollo de los motores de búsqueda semántica.

## **2.5 Web Semántica**

La web sintáctica requiere de metadatos adicionales para que una aplicación pueda procesar los datos en la web, los metadatos son datos acerca de los datos, los cuales entregan información de los recursos, el objetivo de la web semántica (Daconta and Obrst 2003) es que los datos sean más inteligentes para mejorar el procesamiento por una máquina, inteligencia se entiende por el contenido sea interpretado por una máquina y sus relaciones obedezcan a un

orden o categorías conceptuales. La web semántica pretende resolver varios problemas inherentes a la arquitectura de la web como por ejemplo la sobrecarga de la información o la especialización de los sistemas, estos sistemas se caracterizan por ser monopolísticos, no extensibles y no interoperables; por último un problema de gran relevancia es la pobre agregación de contenido entre fuentes dispares de información, como por ejemplo: fuentes financieras, comparación de precios entre portales de compras y minería de datos.

Una tecnología empleada para identificar el contenido de un recurso es el XML, y tecnología encargada de establecer las relaciones entre recursos es el RDF – XML, esta es la tecnología base para la interoperabilidad sintáctica, es decir, una máquina puede interpretar la información cuando ambas partes que comparten la información conocen los nombres o detalles sintácticos; para la interoperabilidad semántica se requieren las tecnologías las cuales se apoyan en conocimiento, como OWL para las ontologías.

### **2.5.1 Tecnologías de la Web Semántica**

Una descripción de las tecnologías de la web semántica se encuentra en el este apartado pretende establecer con exactitud los principales enfoques tecnológicos de la web semántica, y así poder establecer bases sólidas en el entendimiento de la IoT, ya que casi todas las tecnologías de web semántica son utilizadas en la IoT.

### **2.5.2 Ontologías**

La definición más común con el ánimo de no ser tan específicos según (Sánchez, Cavero et al. 2007) que define la ontología como una: "*especificación explícita de una conceptualización*" Esta definición se basa en la idea de "conceptualización" como una visión simplificada del mundo que queremos representar. La conceptualización es el proceso por el cual la mente humana forma su idea acerca de una parte de la realidad. Esta idea es una representación mental libre de propiedades accidentales y basadas en las características esenciales de los elementos; por lo tanto en ciencias de la computación establece que el concepto de ontología está unido a un dominio o un mini-mundo y la especificación representada en la ontología se ocupa de ese dominio, si el dominio (o parte de ella) cambia, la conceptualización también debe cambiar y por lo tanto la ontología que representa este mini-mundo.

Actualmente se han desarrollado muchas Ontologías de Dominio general como WordNET (Miller, Beckwith et al. 1990) y CyC (Guha and Lenat 1990), las cuales están a disposición de los usuarios de Internet para integrarlas o interoperarlas con

las aplicaciones como los Buscadores Semánticos<sup>5</sup>, los cuales pretenden mejorar la precisión de las consultas, ya sea refinando los resultados de los buscadores tradicionales o creando un buscador propio.

### 2.5.3 Indexación Semántica

La Recuperación de la Información Semántica se relaciona con las Knowledge Base - KB que es un tipo de base de datos para la gestión de conocimiento que se toma como fuente de información, por ejemplo son las ontologías, tesauros, vocabularios etc. Se puede extraer las referencias en un texto para anotar documentos puede ser llevado a cabo con LUCENE(Kara, Alan et al. 2012) que es un API de código abierto para la recuperación de la información; comparando la semejanza entre los conceptos semánticos, por medio de métricas fuzzy(Thomas and Sheth 2006) utilizando la distancia Levenshtein, calcula los pesos de las palabras clave con una técnica llamada Frecuencia de termino - Frecuencia inversa de documento (Term frequency - Inverse document frequency - Tf-idf), el peso es la relevancia de la entidad o el cálculo de la frecuencia de la entidad en el documento asociando la entidad con las palabras claves presentes en el documento(Fernández, Cantador et al. 2011) como se ve en la Ecuación 1.

**Ecuación 1. Frecuencia de Termino**

$$d_{x=\frac{Frec_{x,d}}{\max_y Frec_{y,d}}} * \log \frac{|D|}{n_x}$$

- $d_x =$  Pesos de la instancia de x para el documento d
- $Frec_{x,d} =$  Numero de apariciones en d de las palabras clave asociadas
- $\max_y Frec_{y,d} =$  Frecuencia de la instancia que es más repetida y para el documento d.
- $n_x =$  Numero de documentos anotados con x instancia.
- $|D| =$  Número total de documentos en el espacio.

#### 2.5.3.1 Anotaciones Semánticas

En la IR(Ceri and Bozzon 2013) la anotación es el proceso por el cual se logra relacionar los contenidos de un recurso a los conceptos, una forma de anotar es mapear una página web utilizando un identificador URI, lograr el procesamiento automático no se alcanzaba con este tipo de anotaciones ya que se necesitaba un

---

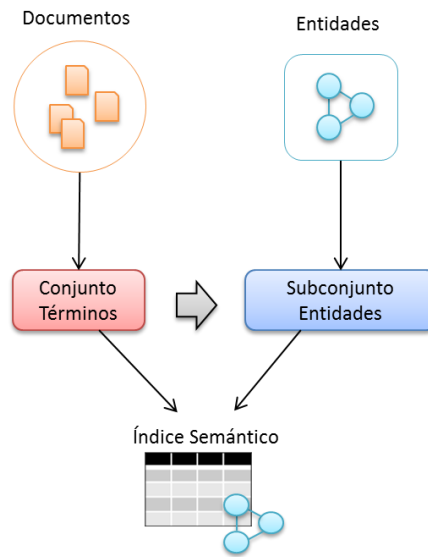
<sup>5</sup> En esta Web se encuentra una Lista de los principales Buscadores Semánticos existentes <http://www.javi.it/semantic.html>.

modelo de dominio y un vocabulario de conocimientos relevantes por ejemplo una ontología, con las ontologías se pueden obtener conclusiones y tomar decisiones beneficiando el procesamiento automático de los recursos. La anotación explícita no tiene la necesidad de incrustar las anotaciones dentro de los documentos sino que se crean instancias de la ontología las cuales se relacionan con los documentos analizados, este proceso se realiza dinámicamente y permite tener desconectados los recursos, este proceso es beneficioso para la escalabilidad ya que se pueden relacionar documentos a múltiples bases de conocimiento ontológico de esta manera el buscador semántico puede devolver solo los documentos relacionados con la información de interés. Existen diversos métodos que pretenden mejorar la exactitud de las anotaciones, encontrar nuevas anotaciones en los documentos y mejorar la precisión de los pesos de las anotaciones en nuestro estudio analizaremos las anotaciones de NPL y las anotaciones que se basan en información semántica contextual.

#### **2.5.3.2 Anotaciones del Natural Language Processing - NLP**

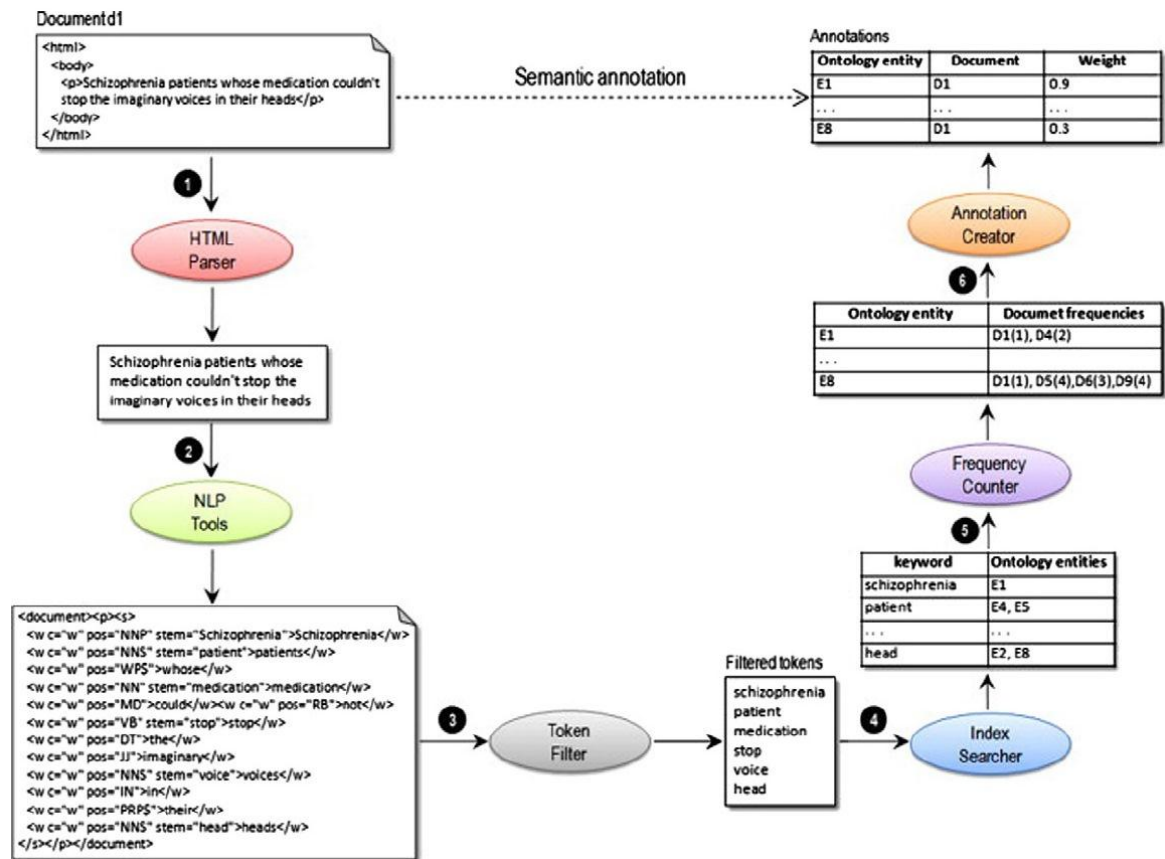
Este enfoque extrae las palabras o grupos de palabras que potencialmente pueden representar entidades semánticas como clases, propiedades e instancias, esto se hace al verificar la semejanza entre los conceptos o sentido de la palabra, a partir de un umbral determinado se añaden estas instancias de las entidades como anotaciones del documento en el índice. En este proceso primero se hace un análisis lingüístico al texto de los documentos para luego si buscar semejanzas con los conceptos de las entidades. Este tipo de anotaciones ayuda encontrar nuevas anotaciones en los documentos y mejorar la precisión de los pesos de las anotaciones. Ver Figura 5.

**Figura 5. Índice de Anotaciones del NLP**



Este proceso de anotación de procesamiento de lenguaje natural es planteado por (Fernández, Cantador et al. 2011) en la Figura 6.

Figura 6. Anotación en base a NLP



Tomado de: Semantically enhanced Information Retrieval: An ontology-based approach, (Fernández, Cantador et al. 2011)

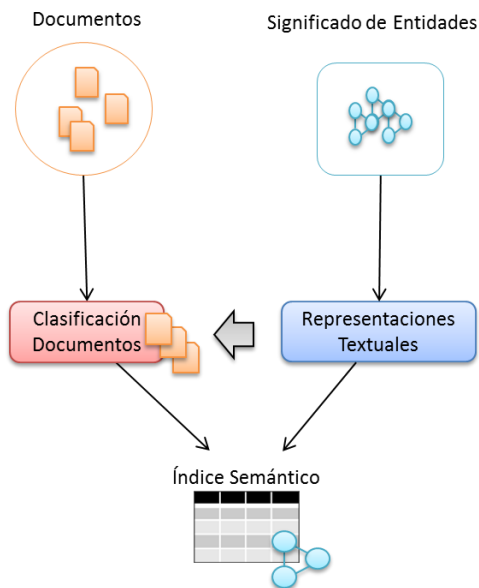
Los pasos que sigue son:

- Paso 1: Del texto del documento borra etiquetas HTML, palabras sin sentido dejando solo el contenido esencial.
- Paso 2: Para el texto se usa (herramientas Wraetic) con etiquetas (PoS) como las palabras tienen plurales, conjugaciones, verbos etc. Asocia las palabras con todas sus posibles formas base. Por ejemplo "borracho" de lema tiza como "beber" si es verbo y como borracho si es sustantivo.
- Paso 3: Análisis lingüístico para filtrar términos e identificar el conjunto de términos como unidades de información individuales.
- Paso 4: Buscar en el índice términos filtrados para obtener un subconjunto de entidades semánticas a anotar.
- Paso 5: Ponderación de las anotaciones hallando la frecuencia de las entidad semántica en los documentos individuales.
- Paso 6: Las anotaciones se añaden a la base de datos relacional.

### 2.5.3.3 Anotaciones sobre Información Semántica Contextual

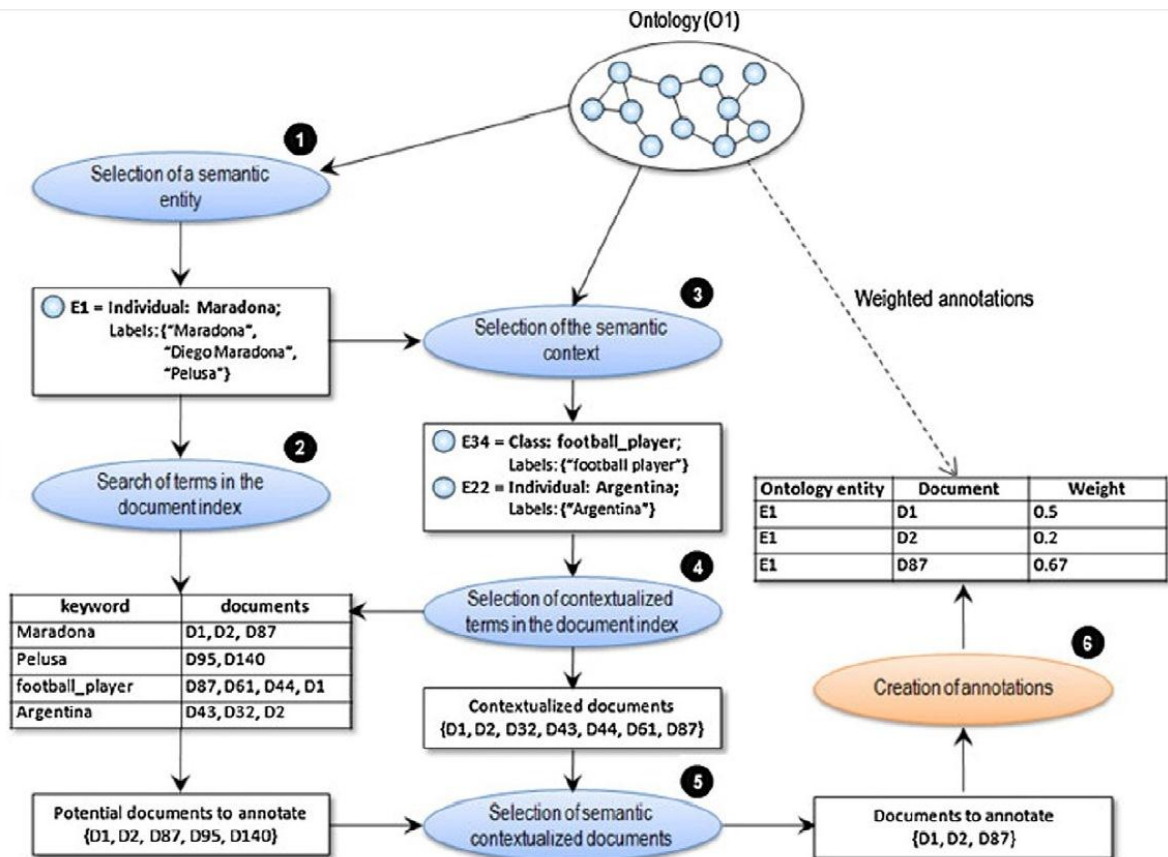
Se realiza un pre procesamiento de los documentos antes de extraer los conceptos de las entidades, en esta anotación adicionalmente se extraen los significados de las entidades siguiendo los enlaces explícitos de las entidades lo que se denomina contexto semántico, esto lo hace obteniendo las representaciones textuales para ser anotadas. En este proceso primero se hace una búsqueda en las entidades semánticas para después comparar con un Standard Keyword Based Index o índice de documentos por palabra clave. Este proceso es como una inversión del proceso de anotación pero tiene beneficios como por ejemplo se baja el coste computacional por que los documentos se indexan por adelantado y se puede mejorar la exactitud de las anotaciones ya que el análisis preliminar de las entidades establece un contexto de conceptos, puede usar cualquier índice de documentos. Este tipo de anotación ayuda a mejorar la exactitud de las anotaciones reduciendo la ambigüedad de las mismas con el conocimiento de fondo contextual en este aspecto aparece un equilibrio entre cantidad y calidad de las anotaciones ya que algunas ontologías carecen de conocimiento contextual. Ver Figura 7.

Figura 7: Anotaciones de Información Semántica Contextual



Este proceso de anotación sobre información semántica contextual es explicado por (Fernández, Cantador et al. 2011) en la Figura 8.

Figura 8: Anotación en base a la Información Semántica Contextual



Tomado de: Semantically enhanced Information Retrieval: An ontology-based approach, (Fernández, Cantador et al. 2011)

Los pasos propuestos son:

- Paso 1: Extraer la representación textual de las entidades por ejemplo la entidad Maradona puede tener varios nombres como Diego, Pelusa.
- Paso 2: Encontrar documentos a anotar con las representaciones textuales por medio de técnicas de búsqueda de calidad y técnicas de clasificación. Algo importante es que el documento que tenga la representación no implica que tenga su significado.
- Paso 3: Extrae el contexto semántico de entidad explorando las relaciones ontológicas. En nuestro ejemplo en Futbol, Jugador, Argentina.
- Paso 4: Encontrar el conjunto contextualizado de documentos por ejemplo las representaciones textuales que se utilizaron para extraer fueron Argentina y Jugador de Futbol.

- Paso 5: Se inserta con su correspondiente Entidad del Paso 2 y significado contextual de entidad del paso 4. Las representaciones textuales de Concepto y Contexto semántico por ejemplo Jugador de futbol, "", "", Argentina.

## 2.6 Semántica en la IoT

### 2.6.1 Estándares Semánticos en el IoT

Las aplicaciones iniciales debían ser diseñadas conociendo perfectamente el sistema de sensores (Le-Phuoc, Nguyen-Mau et al. 2012) y su entorno, para solucionar esta dificultad se desarrollaron estándares y normas para soportar la integración y reutilización de los datos de sensores heterogéneos, por ejemplo las normas IEEE 1451 para transductores, las normas ANSI N42 para la radiación, adicionalmente métodos para acceder a los datos del sensor con paginas HTTP el cual se usó para el sistema GSN, y estándares sintácticos XDS para integración de datos sensor usado en el sistema Xively y Sensorbase, también se desarrollaron estándares semánticos como (Open Geospatial Consortium - OGC) y metadatos basados en RDF con ontologías OWL que representan el conocimiento de dominio, también tenemos protocolos para el intercambio de datos de sensores como él (Extended Environments Markup Lenguaje – EEML), usados para implementar conexiones entre sensores por medio de servicios web, también se propuso RDFa para anotar conceptos ontológicos y propiedades de (Sensor Web Enablement - SWE) utilizando Xlink para gestión y apoyo de funcionalidades semánticas.

Un aspecto importante propuesto, es añadir semántica al flujo de datos sensor para hacer consultas, se hicieron descripciones usando Prolog y representaciones de flujo con OWL, por ultimo llegamos a la descripción de consultas usando metadatos de flujo que utiliza aplicaciones para publicar las descripciones de datos de sensores como (Linked Stream Middleware - LSM) otro ejemplo es Sersormasher que posee un motor mashups visual para la publicación de datos de sensores, parecida a linked data en la web semántica, otro ejemplo es Senseweb que se basa en metadatos sensor para seleccionar las fuentes más apropiadas de sensores, y un último ejemplo es el Sensorgrid4env que utiliza anotaciones semánticas para apoyar las interacciones entre sensores heterogéneos, cabe mencionar SPARQLstream el cual hace énfasis en la integración y no en la consulta ni en la escalabilidad de los datos sensor.

### **2.6.2 Estructuras usadas para indexar en la IoT**

En el método de "Indexación etiqueta" propuesto por (Mingxia Gao 2008) se creó para dar una mejor solución a los problemas de almacenamiento e indexación, la estructura básica de almacenamiento cuenta con tres capas base y algunos mecanismos para acelerar el proceso de indexación, en la primer capa de almacenamiento encontramos la Tabla Principal de usuarios que contiene la información del usuario, la segunda capa almacena todos los Objetos que pertenecen a un usuario en particular y almacena todas las propiedades de los objetos. Hay tres maneras de indexación que se benefician de la estructura de almacenamiento; Primera: la indexación de Interfaz Visual se utiliza para construir un diagrama 3D de objetos que es un mapa de objetos; Segunda: la indexación Etiqueta Orientada a Objetos que encuentra los objetos por medio de etiquetas Orientada a Objetos o Etiquetas Categoría; Tercera: indexación de Palabras Clave.

### **2.6.3 Aplicaciones de Búsqueda en la IoT**

Los fundamentos y requisitos que se deben tener en cuenta cuando nos disponemos a construir aplicaciones que apoyen las funcionalidades características del IoT, se hace un especial énfasis en la búsqueda y descubrimiento de entidades del IoT, los aspectos que se relacionan se presentan en el **ANEXO A-3.5**.

### **2.6.4 Motores de búsqueda en la IoT**

El motor Dyser (Ostermaier, Ro et al. 2010) en el cual los tipos de sensores y sus estados son sólo etiquetas de texto, la búsqueda no se basa en la última medición conocida del sensor sino en un modelo de predicción de mediciones que calcula una probabilidad para la medición buscada. El indizador rastrea la web en forma periódica con el fin de construir un índice de las páginas pertinentes. Ordena los resultados en orden descendente de probabilidades generales. La implementación del prototipo de Dyser se basa en Java y PHP, el motor de búsqueda fue implementado en Java y cuenta con un servicio web Simple Object Access Protocol - SOAP sencillo para plantear las solicitudes de búsqueda, el motor proporciona un mecanismo de auto sugieren que ayuda al usuario para completar el término de búsqueda al sugerirle posibles coincidencias. El motor implementa una base de datos relacional que se accede mediante la interfaz JDBC, usa la base de datos MySQL.

### **2.6.5 Ontologías para la IoT**

(Wei, De et al. 2012), presentan una ontología en la cual modelan los aspectos de la IoT con el fin de almacenar conocimiento en ella. Definen unos *principios de*

*diseño* de la ontología y posteriormente definen varios módulos que pretenden abarcar todo el conocimiento del área. Realizan alineaciones con la ontología *DUL*, *SSN-XG* y *los servicios con el estándar OWL-S*. Los principios de diseño presentados se tuvieron en cuenta para la construcción de las ontologías de la presente propuesta, adicionalmente algunos de los módulos presentan conceptos interesantes que nos permiten abstraer algunas características de la IoT, por ejemplo la *definición de los servicios* y tipos de servicios utilizados. Otro elemento interesante es que reutilizan las ontologías más aceptadas en este momento sobre la IoT, lo cual garantiza un buen aporte a las mismas.

## 2.7 Publicación de datos en la Web

La web se convirtió en la plataforma favorita para publicación de datos, en sus inicios el comercio electrónico utilizaban formularios para hacer búsquedas en bases de datos a esto se le denominó web profunda “Deep Web” por qué los buscadores cubrían datos superficiales de la web, en este momento tecnologías nuevas permiten consultar cualquier base de datos en la web, además cada página se podía identificar por un Identificador Internacional de Recursos (International Resource Identification - IRI) para poder ser buscada, posteriormente se pretende pasar los recursos de formato fuente a formatos abiertos lo cual es costoso. Un editor puede utilizar una licencia de datos abiertos (PDDL, ODC, CC06) y también del sistema de estrella de datos abiertos para determinar que se puede hacer con los datos, en la Tabla 5 se muestra la clasificación según (Thomas and Sheth 2006):

**Tabla 5: Sistema de Estrellas para Publicar en la web**

Estrellas	Descripción
*	Datos en cualquier formato web e imágenes con licencia abierta, se puede realizar trabajo manual sobre los datos y pueden una interface de para extraer los datos.
**	Datos con una estructura legible por una maquina por ejemplo el formato Excel en lugar de una imagen de una tabla.
***	Datos en formatos abiertos para que pueda ser leído por cualquier aplicación por ejemplo formato Comma Separate Values - CVS en lugar del formato Excel.
****	Datos con tres estrellas y estar identificados por IRIs para poder ser marcados y vinculados.
*****	Datos con cuatro estrellas y además se vinculan a otros datos publicados en la web o a algún esquema de elementos que representa alguna entidad del mundo real o relacionado con algún concepto.

En la Tabla 6, según (Thomas and Sheth 2006) se listan los enfoques utilizados para la publicación de datos en la web y su relación con el sistema de estrellas para datos abiertos en la web como se ve en la Tabla 6.

**Tabla 6: Enfoques de Publicación en la web**

<b>Enfoque</b>	<b>Estrellas</b>	<b>Descripción</b>
<b>Deep Web / Web profunda</b>	De una a tres	Publicación de tablas HTML, formatos Excel y CVS.
<b>Web APIs / APIs Web</b>	De tres a cuatro	Datos en formato no propietario por ejemplo XML, JSON. Rara vez se identifican por una IRI
<b>Microformat / Microformatos</b>	De tres a cinco	Datos en formato no propietarios utilizando formatos abiertos. Los microformatos reutilizan las etiquetas HTML para transmitir metadatos de la información a los usuarios finales es decir microformatos que utiliza IRIs.
<b>RDFa y Linked Data</b>	Cinco	Enfoques centrados en bases de datos back-end enriqueciendo paginas para usuarios finales con formato de datos RDFa. Linked Data es el conjunto de buenas prácticas para publicar datos en la web, linked data se basa en IRIs para identificar elementos de datos sobre HTML y RDF, RDFS y OWL como lenguajes para crear vocabularios.

## 2.8 Recuperación de la Información

En la Recuperación de la Información – IR (Ceri and Bozzon 2013) se evalúa la relevancia de la necesidad de información, no se debe confundir con la consulta propiamente dicha, un ejemplo de requisito de información puede ser: “Comer es beneficioso para reducir la presión arterial” y la consulta podría ser “El efecto del chocolate en la presión”. Para formalizar los mismos se denotara lo siguiente Ecuación 2:

**Ecuación 2. Función de Recuperación**

$$R(q_i, d_j)$$

$$d_j \in D = \text{Documento}$$

$$q_i \in Q = \text{Consulta}$$

### 2.8.1 Modelos de Recuperación de la Información

Los modelos de representación de documentos y consulta tienen gran importancia en el rendimiento y la recuperación, se centra el trabajo en los modelos clásicos que comprenden los modelos Booleano, Vectorial y probabilístico, aunque existen modelos basados en lógica formal, Lógica fuzzy(Thomas and Sheth 2006) e inteligencia artificial ver Tabla 7.

Tabla 7: Modelos de RI

Modelos de la (IR)	Descripción
<b>Booleano</b>	Un conjunto de términos y las consultas representan a los documentos son expresiones booleanas, se requiere de una coincidencia exacta para la recuperación.
<b>Espacio Vector (VSM)</b>	Vectores de términos ponderados representan los documentos y las consultas, la recuperación se realiza según el grado de similitud con vector consulta y tiene la capacidad de recuperar los documentos parcialmente coincidentes.
<b>Probabilístico</b>	La recuperación de los documentos se realiza teniendo en cuenta las probabilidades de pertenencia al conjunto.

### 2.8.2 Consulta, Búsqueda y Clasificación

Cuando se genera una consulta, se recupera un conjunto de tuplas, para recuperar los documentos se accede al índice, posteriormente se calcula el valor de similitud entre la consulta y cada documento recuperado, donde cada documento es representado por un vector que corresponde a cada entidad. A continuación se presenta un ejemplo del modelo vectorial en la Ecuación 3:

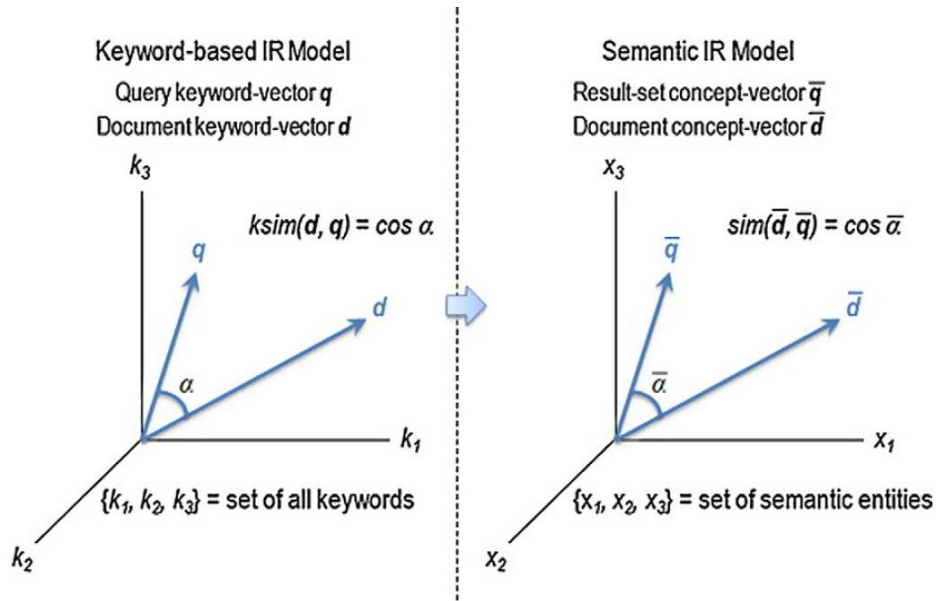
Ecuación 3. Similitud entre Consulta y Documento

$$sim(d, q) = \frac{d \times q}{|d| * |q|}$$

$$\begin{aligned} sim(d, q) &= \text{Medida de Similitud} \\ d &= \text{Documento} \\ q &= \text{Consulta} \end{aligned}$$

La adaptación del modelo vectorial establece el peso de la anotación entre el documento y la entidad según (Fernández, Cantador et al. 2011) en la ver Figura 9.

Figura 9: Adaptación del clásico modelo de Espacio Vectorial



Tomado de: Semantically enhanced Information Retrieval: An ontology-based approach, (Fernández, Cantador et al. 2011)

Como se puede observar el modelo de búsqueda sintáctica se basa en palabra encontradas y el modelo de búsqueda semántica recupera conceptos constituidos por grupos de palabras esto es representado por la barra superior de las variables (d, q).

La recuperación semántica de la información (Semantic Information Retrieval - SIR) busca extender el campo de la Recuperación de la Información al no poder recuperar el significado de la información que se transporta, un problema que se presenta es el nivel polisémico de las palabras, el significado que lleva una palabra en diferentes oraciones y sus relaciones con otras palabras de la frase, el proceso de encontrar el significado correcto de las palabras en la frase se conoce como desambiguación (*Word Sense Disambiguation - WSD*), existen técnicas estadísticas o modelado de dominio, el modelo de dominio que describa los conceptos relevantes y un modelo estocástico que aprovecha la información lingüística eliminando la ambigüedad de las palabras.

### 2.8.3 Estructuras de indexación en la Web

Las estructuras (Delbru, Campinas et al. 2012) de indexación registran los datos con cierto orden creando un índice, los motores de búsqueda semántica indexan recursos automáticamente usando bases de conocimiento ontológicas que recuperan la información utilizando conceptos o significados. Existen tres técnicas de búsqueda de documentos: semi – estructurados el esquema de Índice Nodo que utiliza el etiquetado de nodos, el esquema Índice Grafico que utiliza índices secundarios o resúmenes y el esquema Índice de estructura secuencial que requiere de técnicas de cadenas coincidentes para responder a las consultas. Se plantean un modelo de búsqueda de alto rendimiento que sea aplicable a grandes colecciones de datos en la web, las unidades básicas que se desean recuperar están compuestas por un conjunto de atributos, valores y relaciones entre entidades, el esquema está compuesto por un índice invertido con forma de árbol que realiza consultas sobre grafos RDF con empleo de palabras clave.

### 2.8.4 Semántica aplicada a la búsqueda de datos de Sensores en la IoT

Los componentes tales como las redes de sensores, los servicios web y las bases de datos se deben combinar para construir aplicaciones en la IoT. La introducción de la semántica para combinar estos ingredientes constituye una potente solución para el manejo de las dificultades que se tienen con los flujos de datos del sensor.

Las soluciones de los desafíos en el desarrollo de aplicaciones en el IoT más influyentes ha sido el consorcio Sensor Web Enablement - SWE los cuales han aportado en observaciones, mediciones, descripciones de red de sensores, transductores y transmisión de datos , y también en el campo de los diferentes tipos de servicios generados por las fuentes de datos se han apoyado la planificación, alerta, observación, recogida de medición y gestión Las tendencias en las soluciones (Corcho, Ra et al. 2010) que se han presentado para la integración y fusión de datos heterogéneos utilizando técnicas semánticas en la Tabla 8.

Tabla 8: Técnicas semánticas para la Búsqueda de datos de sensores

Técnica	Descripción
<b>Transformación de base de datos sensor a en RDF</b>	Se transforma el flujo de datos de sensores a RDF y la publicación con HTTP mediante identificación del sensor con la URI. Ampliamente utilizado en el Linked Open Data.
<b>Consulta semántica adaptadas a datos de</b>	Con los lenguajes de consultas SPARQL genera vistas lógicas para consultar los flujos de datos de sensores.

## 2.9 Servicios Web - WS

Son aplicaciones de software las cuales se pueden “descubrir y describir” en XML y utilizan protocolos de web estándar a través de las redes. La parte de “describir” tiene que ver con la sintaxis del mensajes XML a través del protocolo HTTP y la parte de “descubrir” tiene que ver con el UDDI y ebXML registros que permite a las aplicaciones descubrimiento automático en el **ANEXO A-3.4** se hace una descripción detallada de esta tecnología.

## 2.10 Servicios Web Semánticos - SWS

Esta descripción de las tecnologías existentes para SWS no pretende dar definiciones exhaustivas de su funcionamiento pero si plantear los propósitos de desarrollo y la evolución de las estrategias adoptadas para solucionar las dificultades para automatizar los servicios web.

Son servicios software identificados por un URI que se describen, descubren y acceden a través de protocolos web. Los servicios web consumen XML en esta característica encajan con la Web Semántica, los servicios web proliferan igual que las páginas web pero existe el problema de que son difíciles de descubrir y las aplicaciones avanzadas de servicios web implican: comparación, composición y orquestación las cuales requieren que sean automatizadas, en este punto la web semántica a dispuesto soluciones como lo son los SWS, la web semántica y los servicios web van de la mano pero WSDL el lenguaje que describe los interfaces SOAP no es suficiente ya que se requiere dar apoyo automatizado, uno de los primeros esfuerzos fue el lenguaje DAML-S de ontologías para describir las propiedades y capacidades de los servicios, posteriormente se desarrolló (Semantic Web Enable Service – SWES) que es una descripción completa de descubrimiento de servicios y proporciona un servicio de mediación web escalable, al construir ontologías de modelos de cómo funcionan la cosas se tiene un lenguaje común para describir servicios web con mucho más detalle para lograr la automatización. Los lenguajes ontológicos que pretenden describir: “que requiere y que proporciona el servicio, como funciona y como puede ser utilizado”, los enfoques principales(Kashyap and Bussler 2008) basados en ontologías para la representación de servicios web semánticos se pueden consultar en el **ANEXO A-3.7**

## 2.11 Conclusiones del Marco de Referencia

Finalmente, después de la revisión del estado del arte se pueden obtener conclusiones y decisiones importantes de trabajo para el presente proyecto:

1. La arquitectura utilizada en el desarrollo se fundamentó en capas, centrados en el ámbito de aplicación, es decir más en la WoT que la IoT, ya que el proyecto no pretendía trabajar a nivel de protocolos e integración de dispositivos, sino a nivel del tratamiento y uso de la información de los mismos. Para solventar el problema de interfaz con los protocolos se seleccionó un middleware o servidor IoT existente, el cual provee un API REST para interactuar con los dispositivos.
2. Dado que la indexación semántica es un proceso que involucra varias herramientas, modelos y tecnologías, se decidió crear un método de indexación semántica en la IoT, con el fin de aplicar las técnicas de la web semántica para procesar y recuperar información en la WoT, con el objetivo de facilitar su adopción para aprovechar los servicios de información de la IoT.
3. Las tecnologías de la web semántica que se utilizaron en el presente proyecto son: las ontologías como elementos de conocimiento para la indexación de la información de los sensores en un dominio particular, el modelo vectorial para facilitar el desarrollo del buscador semántico y además provee buenos resultados y un diseño orientado por un modelo semántico el cual integra las ontologías existentes de sensores y los estándares más usados en la IoT.
4. Con respecto a la Indexación semántica se decidió utilizar un proceso de anotación semántica de los conceptos de la ontología en los documentos recuperados de los sensores, con el fin de incrementar la relevancia en la búsqueda. Adicionalmente se incorporó un proceso de expansión de consulta en lenguaje natural con respecto a la ontología, con el fin de mejorar la consulta que hacen los usuarios.
5. El Buscador no se centrará en modelos de predicción, sino en encontrar los sensores más adecuados para las necesidades de información de los usuarios.
6. Se decidió proveer un conjunto de servicios web SOAP de índice semántico y del dominio particular a desarrollar. Los servicios web semánticos se decide hacer un diseño de cómo sería su implementación, ya que la misma se saldría de los alcances del proyecto, dada su complejidad.

## METODO DE INDEXACIÓN SEMANTICA EN LA IOT - SIMIOT

### 3.1 Método Propuesto

Teniendo en cuenta el estado del arte descrito anteriormente, las características de los objetos, las arquitecturas de la IoT, las herramientas de interoperabilidad semántica y el propósito de la indexación semántica, podemos identificar los temas y procesos principales que se deben abordar al momento de crear un índice semántico en la IoT. Con el fin de formalizar adecuadamente el método propuesto, se hace uso de las especificaciones en el estándar SPEM 2.0 (OMG 2008).

#### 3.1.2 Preguntas que debe responder el método

Inicialmente se realiza una aproximación teórica y después se formaliza el método de desarrollo con el estándar. Las preguntas que debe responder el método propuesto son:

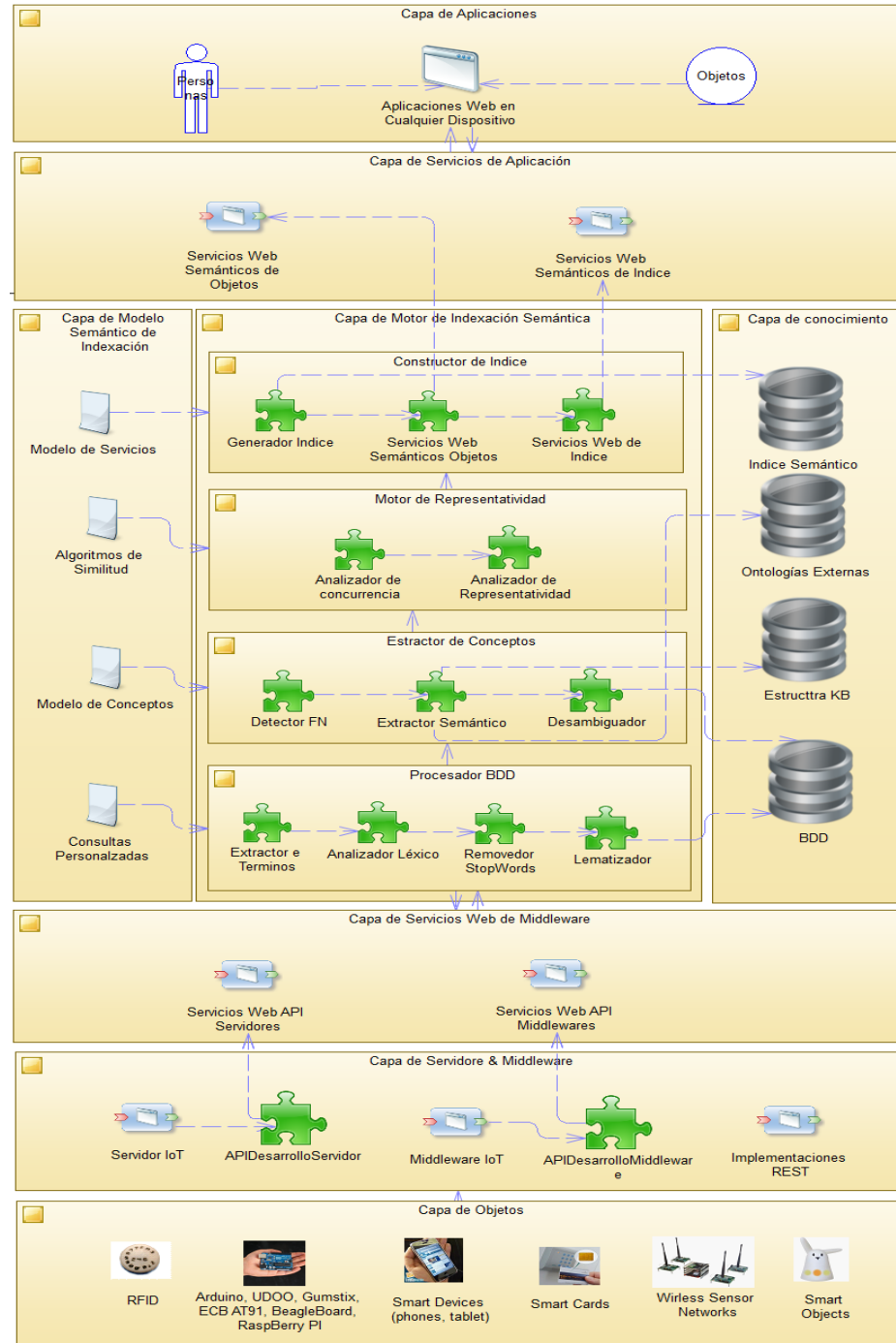
- 1 ¿Cuáles son los objetivos de la construcción de un índice semántico en la IoT?
- 2 ¿Cuáles son los requerimientos principales para un índice semántico en la IoT?
- 3 ¿Qué fases, actividades y tareas se deben seguir para crear un índice Semántico en la IoT?
- 4 ¿Qué roles participan en la construcción y cuáles son sus habilidades?
- 5 ¿Qué propiedades de objetos se deben tener en cuenta para su anotación semántica?
- 6 ¿Qué tecnologías para representar e identificar los objetos se debe utilizar?
- 7 ¿Qué decisiones importantes debe tomar al momento de crear un índice semántico?

#### 3.1.3 Arquitectura del Índice Semántico en IoT

Con el fin de establecer una abstracción global de los índices semánticos en la IoT, se presenta una arquitectura (Figura 10) de funcionamiento del mismo. Se puede distinguir una arquitectura multicapa en la cual las primeras tres capas de abajo hacia arriba crean una abstracción de los objetos de la IoT. La capa de servicios web de middleware es una propuesta con el fin de generar interfaces uniformes de acceso a los datos de los objetos, permitiendo añadir en el futuro cualquier tipo de middleware o Servidor IoT que soporte o desarrolle dichos servicios, adicionalmente permite a las aplicaciones conectarse de manera

transparente a los servicios de esta capa, sin importar de que servidor provengan o como es su lógica de negocio.

Figura 10: Arquitectura de Índices Semánticos en la IoT



Por otro lado, las tres capas intermedias que están paralelas, son las encargadas de crear el índice semántico. La propuesta, es que se parte de un modelo semántico para la creación del índice, este modelo está dirigido por la solución de una serie de requisitos personalizados al desarrollo de cada índice semántico y por unas decisiones de diseño, que guían la construcción del mismo en cada actividad. Los elementos principales son:

- **Modelo de consultas personalizadas:** es un modelo que permite establecer el dominio de objetos que se deben indexar y como se va a caracterizar dicho dominio al momento de seleccionar la información y los objetos a indexar.
- **El modelo de conceptos:** es un documento que establece las estructuras a utilizar (ontologías, tesauros), cómo utilizarlas y las decisiones de modelos conceptuales y algoritmos a utilizar en cada tarea desarrollada para la obtención de los conceptos. En esta capa se toman decisiones de cómo manejar el contexto y demás información semántica de los objetos.
- **Algoritmos de Similitud:** es un documento con un conjunto de decisiones alrededor de los algoritmos que se van a utilizar para el análisis de coocurrencia y representatividad.
- **Modelo de servicios:** Es un documento que establece las interfaces a desarrollar con respecto a los servicios web semánticos de los objetos y los servicios Web del índice mismo.

Finalmente, las dos últimas capas pretenden generar una interfaz de servicios web, de tal forma que cualquier aplicación web corriendo en cualquier dispositivo, pueda utilizar los servicios del índice y de los objetos a los cuales hace referencia. Estas aplicaciones pueden dar servicios a personas o a objetos inteligentes con capacidades de procesamiento.

### **3.2 Método para crear Índices Semánticos en WoT**

A continuación se presenta el método estableciendo fases y pasos que se deben seguir para crear un índice semántico en la WoT. Para esto, se tomó como punto de partida en el trabajo de (Niño, López et al. 2012) en el cual presentan cuatro fases para la construcción de índices semánticos para la web. El trabajo fuerte y diferenciador radica en cómo se adapta este proceso a la indexación semántica en la WoT, modificando y agregado nuevas fases, actividades y tareas específicas para la IoT. Finalmente, se hace una especificación formal del método propuesto utilizando el estándar SPEM 2.0 (OMG 2008, Abril), tomando algunas

características del proceso de desarrollo OpenUP (org. 2013) y buscando acomodarlo más a un proceso ligero, con el fin de que se pueda usar y entender por máquinas y humanos en su posterior reutilización en un proceso de desarrollo de software particular.

### **3.2.1 Fase 1: Conceptualización del Proyecto de Indexación Semántica (FCPIS)**

El Objetivo de esta fase consiste en determinar los objetivos y requerimientos por los cuales se construye el índice semántico. Con esta información se debe decidir el tipo de estructura que se utilizará para almacenar el conocimiento (Tesauros, Ontologías, otras) y clasificar la información de los objetos de la IoT, con fin de definir como obtener los conceptos ocultos en cada dispositivo, adicionalmente establecer la infraestructura a utilizar para mediar con los objetos de la IoT. Las actividades definidas son:

**Actividad 1. Definir Requerimientos del índice (ADRI):** Se debe hacer una lista de requisitos software de acuerdo al modelo de negocio para el cual se necesita el índice. Se priorizan las funcionalidades y se evalúan los riesgos de la misma, similar a cualquier desarrollo de sistema de información.

**Tarea 1. Diseño de la Visión del Proyecto:** el analista lleva a cabo revisiones con las partes interesadas y crea un vocabulario común de términos en un documento glosario, define el alcance y establece las dificultades de la creación de un motor de búsqueda semántica en un documento visión y una lista de objetos de trabajo.

**Tarea 2. Formalización de los Requisitos Software:** El analista crea una lista de requerimientos del índice tales como: Consultas en tiempo real, tiempos de respuesta, cantidad de información semántica a manejar y objetivos del índice. Los requerimientos se deben dialogar con los **clientes** (stakeholders), con el fin de llegar a un acuerdo con los mismos. El producto es un documento de requerimientos y un documento de conceptos. Aquí se aplica una práctica llamada visión compartida.

**Tarea 3. Formalización de la Arquitectura:** El arquitecto crea la arquitectura y realiza el proceso de adaptación de la arquitectura al modelo del negocio particular y hace los refinamientos que crea necesarios. El producto es un documento con la arquitectura que se va a implementar.

**Tarea 4. Crear el modelo conceptual semántico de indexación:** Esta es una de las tareas más relevantes del método, ya que requieren un

trabajo creativo importante para resolver varios problemas en la búsqueda en la IoT. El trabajo de (Römer, Ostermaier et al. 2010) hacen un acercamiento temprano a este tipo de indexación estableciendo los problemas a tener en cuenta en la búsqueda, se debe hacer un diseño cuidadoso de la manera como se va a manejar varios elementos importantes en la consulta de índices semánticos. El **programador**, el **analista**, el **experto en conocimiento** y el **cliente**, se reúnen para generar un modelo conceptual semántico de indexación en la IoT. Para esto deben responder las siguientes preguntas:

- ¿Cuáles son los usuarios del índice semántico?
- ¿Cuáles son los tipos de consultas que va resolver el índice semántico?
- ¿Cómo se van a manejar las consultas que hacen los usuarios para que se pueda entregar resultados relevantes?
- ¿Qué cantidad de información se va a indexar y cual se va a consultar directamente al sensor?
- ¿Cómo se va a manejar la concurrencia de consultas a un mismo sensor?
- ¿Cómo se va a modelar el contexto que necesariamente está asociado a los sensores?
- ¿Cómo se va a manejar la frecuencia de actualización del índice?
- ¿Cuál es la estructura de datos más adecuada para almacenar el índice?
- ¿Cómo se va a compartir el índice para su correcto aprovechamiento?

El producto final de esta tarea, en un conjunto de documentos (Modelo de consultas personalizadas, Modelo de conceptos, Algoritmos de Similitud y Modelo de Servicios) con las respuestas a estas preguntas y detalles de implementación y uso de tecnologías para implementar el modelo. Dependiendo de la complejidad del modelo, se puede hacer una realimentación a la arquitectura del índice y a los módulos y paquetes a crear en este mismo método básico propuesto.

**Actividad 2. *Evaluar el middleware / servidor IoT a utilizar (AEMS):*** Se debe seleccionar o decidir crear una infraestructura que permita conectar los diferentes dispositivos de la IoT a la web, permitiendo la interacción con las aplicaciones web a través de protocolos ampliamente aceptados. Existe

la posibilidad de reutilizar middleware gratuitos o servicios de los diferentes servidores de datos de la IoT, a través del uso de librerías (APIs) de desarrollo. En lo posible se debe buscar la captura metadatos y datos en formato XML, con el fin de crear una base de datos documental (BDD), que será utilizada para la indexación. Se evalúa si el API de desarrollo cumple con las funcionalidades básicas para su incorporación: debe permitir mínimo lectura para los dispositivos, consultar otros dispositivos y si usa un protocolo estándar de comunicación como REST, MQTT, más los requisitos propios del índice. Hay que tener en cuenta que los servidores IoT están más enfocados a poner datos de los objetos en la web con diferentes formatos, pero son pocos los que desarrollan una interoperabilidad en dos vías, es decir, que se pueda enviar información a los objetos, en este caso los middleware se han desarrollado pensando más en esta funcionalidad. Otro elemento importante, es que el hecho que de requerir enviar datos a los objetos, implica una programación con el API de desarrollo en los objetos inteligentes o en los gateway inteligentes, así se debe que tener en cuenta el hardware y entorno de desarrollo utilizados. Sin embargo, los índices semánticos no tienen como principal objetivo enviar mensajes a los recursos que indexa, por ello la decisión aquí en la mayoría de los casos sería usar un servidor IoT o una plataforma que permita cómo mínimo consultar su información.

**Tarea 1. Definir la fuente de datos de la IoT:** Los desarrolladores realizan una comprobación de las características de los middleware o servidores IoT a utilizar como fuente de datos. Entre los pasos tenemos: Descargar el API, hacer test de consulta de datos y metadatos, hacer test de consulta de objetos por un criterio. El producto es un documento con los servidores seleccionados y las características a tener en cuenta al momento de utilizar los métodos de consulta. El estudio realizado en la tabla del **ANEXO B-1**. puede servir como insumo de trabajo en esta tarea.

**Tarea 2. Alineación de Conceptos del Servidor IoT y Fuentes Externas:** Cada fuente de datos define una estructura de metadatos, en este punto el analista estudia las fuentes seleccionadas en la tarea anterior y revisa la estructura de metadatos, las relaciona a la Ontología de Dominio o estructura de conocimiento elegida en la cual se clasifica la información de cada uno de los objetos. Adicionalmente, es necesario analizar si los datos provistos por la fuente de datos y establecer si es necesario obtener datos de otras fuentes que deben ser alineadas a la ontología o

estructura seleccionada de conocimiento, para que responda a las preguntas en el dominio particular seleccionado. Esta relación la hace el **experto en conocimiento** identificando nuevos sinónimos o nuevos nodos que deben ser agregados a la estructura de conocimiento utilizada e incluso instancias. El producto es unas modificaciones al documento que presenta la estructura de conocimiento seleccionada con los nuevos conceptos.

### **3.2.2 Fase 2: Creación de la Base de Conocimiento y Servicios (FCBCS)**

En esta Fase se crea una primera versión de la estructura de conocimiento, además de crear los métodos que permitirán convertir al API de desarrollo en servicios Web para su posterior uso por el índice. Finalmente se hace una realimentación a los requisitos definidos y a la arquitectura propuesta teniendo en cuenta lo que se encontró en este proceso. Posterior a esa realimentación se puede hacer otra iteración de desarrollo de la base de conocimiento con el fin de refinar la misma, utilizando una práctica de iteraciones incrementales en el proceso de desarrollo:

#### ***Actividad 1. Preparar o Crear la Estructura de Conocimiento (APCEC):***

Una vez desarrollada la conceptualización de la base de conocimiento se procede a su construcción. Esta tarea puede presentar varios pasos dependiendo de la metodología de desarrollo de ontologías seleccionada, si se ha decidido construir una. Adicionalmente se debe presentar la estructura final a los clientes para su realimentación, continuando la práctica de la visión compartida.

***Tarea 1. Seleccionar una metodología de desarrollo adecuada:*** El **experto en conocimiento** y el **desarrollador**, tomando como base los requerimientos y decisiones tomadas con la estructura de conocimiento, seleccionan una metodología de desarrollo de la misma y estiman el trabajo a desarrollar para aplicarla. El producto es un plan de desarrollo de la estructura, con recursos, tiempo y productos bien definidos y enmarcados en el proyecto macro que es la creación del índice semántico en la IoT.

***Tarea 2. Crear la estructura de conocimiento:*** El **desarrollador** ejecuta el plan de desarrollo de la estructura, en lo posible utilizando herramientas como PROTEGE (Noy, Sintek et al. 2001) en el caso de una ontología de dominio específico (Fernández-López and Gomez-Perez 2003). Otra opción, es utilizar una ontología estandarizada como SSN-XG y a partir de ella soportar la construcción del índice. La estructura se ubica en la capa de

conocimiento del índice. El producto es un archivo OWL (ontología), RDF o cualquier otro formato de almacenamiento de conocimiento, los cuales se representa con un documento de resultados de pruebas y un documento de archivo devueltos en la ejecución.

**Actividad 2. Adaptar el API al Entorno del Índice (AAEI):** En esta actividad se toman los API de las fuentes de datos y crean los servicios web necesarios para su uso en la capa de servicios de la arquitectura del índice.

**Tarea 1. Crear los servicios del API de la Fuente de datos:** En este punto, el **programador** toma el API de desarrollo, dependiendo del servidor o middleware usado y se programa de acuerdo a la interfaz desarrollada para crear el índice semántico. Los métodos se crean como servicios Web a ser consumidos por el motor del índice. Los API se podrían programar directamente en el índice, pero esto cierra al indexador a que sólo use este API, lo ideal es crear una capa de Servicios y en ellos colocar módulos que desarrollan el API hacia un servidor particular, esto hace uso de una práctica de software en la arquitectura propuesta de encapsulamiento por capas de desarrollo. El producto son los códigos fuente y programas en el repositorio de Servicios Web, los documentos de diseño específico son un documento de datos de objetos particulares y un documento de conjunto de elementos a partir de la consulta.

**Realimentación de la funcionalidad:** Una vez desarrollados los servicios Web se realimenta su funcionamiento con el **equipo de desarrollo**, el **administrador de proyecto** y los **clientes** con el fin de establecer tempranamente otros requerimiento o funcionalidades no detectadas. Las modificaciones a realizar o las apreciaciones obtenidas se registran, las cuales se representa con un documento de resultados de pruebas y un documento de archivo devueltos en la ejecución.

### 3.2.3 Fase 3 Construcción del Índice Semántico (FCIS)

En esta fase se construye el índice semántico. Las actividades desarrolladas se pueden realizar cada una en una iteración, con el fin de tener hitos bien definidos en los productos intermedios y así diferenciar cada una de las etapas de construcción del índice. Las principales actividades son: el procesamiento de la base de datos documental, extracción de conceptos, estudio de representatividad y construcción de la estructura de índice:

**Actividad 1. Preprocesamiento de la base de datos documental (APBDD):** Esta actividad crea la base de datos documental (BDD), cada documento o conjunto de documento se han extraído de las fuentes de

datos y se realizan varias tareas de preprocesamiento de los datos de cada uno de los documentos, reduciendo el número de datos a analizar y obteniendo el conjunto de términos (tokens) a ser indexados. Para esta actividad se puede apoyar en herramientas software que contienen librerías para este procesamiento como LUCENE.(Hernández and Hernández 2008). Las tareas son:

**Tarea 1. *Extraer Información de Objetos:*** En esta tarea el **desarrollador** implementa un módulo (extractor de términos) en un paquete denominado preprocesador de BDD, el cual usando los servicios web de las API hace un escaneo sistemático de toda la base de datos del Servidor o Middleware, extrayendo la información de cada objeto a través de los servicios web que usan las API de desarrollo. Aquí también hay que decidir si se hace un escaneo de datos específicos dependiendo del contexto de uso del índice. El producto es la base de datos documental (BDD), un conjunto de archivos de texto que será la fuente de indexación principal.

**Tarea 2. *Análisis Léxico del Texto:*** El **desarrollador** implementa en un módulo (analizador léxico) en el paquete preprocesador de BDD. Dado que las fuentes de información de los objetos proveen información estructurada (metadatos XML) y no estructurada (Descripciones y Datos sensores), aquí se hace un manejo un poco diferente de la información estructurada, ya que las categorías, etiquetas y relaciones encontradas en los documentos XML de metadatos se convierten en sí mismas conceptos por los cuales se representa el objeto y se indexa, por ello los documentos de este tipo se eliminan los caracteres sobrantes como etiquetas XML y demás, para dejar solo los términos importantes de acuerdo a las categorías identificadas en la fase anterior, así los documentos estructurados se convierten en texto plano. Para los documentos no estructurados se puede utilizar una herramienta de procesamiento de lenguaje natural como tokenizadores, los cuales separan la secuencia de caracteres de los textos para generar una secuencia de palabras que posteriormente pueden ser los términos índice del documento. Remoción de caracteres especiales, como “/ \ - : ? ; ) ( & #”, entre otros. El producto es la misma BDD modificada con archivos llenos de términos.

**Tarea 3. *Eliminación de palabras vacías (stopword removal):*** El **desarrollador** crea un módulo (Removedor StopWord) en el paquete preprocesador de BDD, el cual busca eliminar los términos con poco valor en la recuperación de información como: pronombres, partículas

interrogativas y ciertas preposiciones. Entre los artículos están por ejemplo: “un, la, los, el, ellos”, las partículas interrogativas son: “que, cuál, quién, cómo, dónde”, entre otros. Algunas preposiciones son: “con, desde, entre, hasta, por, según”, etcétera. Dependen los términos del idioma que se escoja. El producto es la misma BDD modificada con archivos llenos de términos reducidos en tamaño, sin las palabras que no aportan semántica.

**Tarea 4. Aplicación de Lematización (stemming):** El desarrollador crea un módulo (Lematizador) en el paquete preprocesador de BDD el cual permiten la reducción de las palabras a su forma básica o raíz, por ejemplo, eliminando las partes no esenciales de los términos como prefijos y sufijos o derivando las palabras en plural a su raíz en singular. A través de Lucene y otros proyectos se pueden usar diversos algoritmos de stemming, como por ejemplo el algoritmo de Porter para inglés. El producto es la misma BDD modificada con archivos llenos de términos reducido a palabras que representan una clase de variaciones del mismo término y términos normales, reduciendo aún más la cantidad de información a indexar.

**Actividad 2. Extracción de Conceptos (AEC):** En esta actividad se hace una selección de términos candidatos, es decir, los términos relevantes en los documentos y que son candidatos para extraer sus conceptos. Estos conceptos aparecerán a partir de un proceso de consulta en las estructuras de información diseñadas para almacenar conocimiento en la primera fase, como los tesauros y las ontologías u otras aproximaciones de la teoría cognitiva. El proceso de obtener los conceptos tiene dos objetivos principales según (Pezo, Perez et al. 2013): El primer objetivo consiste en **ampliar-reducir los conceptos (Anotación Semántica)** encontrados en las estructuras de conocimiento consultadas, principalmente abstrayendo o especializando los conceptos. En este punto, cada proyecto de indexación semántica define el conjunto de conceptos relacionados de acuerdo a sus necesidades. Una de las reglas más fáciles para ampliar la representación es obtenido de las estructuras de conocimiento con respecto a cada concepto el conjunto de *sinónimos*, *hiperónimos*, *hipónimos* y *merónimos*, entre otras posibles relaciones de conceptos que existan en las estructuras seleccionadas. El segundo objetivo consiste en reducir los problemas de *homonimia*, *sinonimia* y *polisemia*: Para este caso se pueden aplicar algoritmos que permitan hacer *desambiguación* de las palabras. Uno de los más utilizados es el Word Sense Disambiguation – WSD (Sanderson 1996; Rada and Moldovan 2000; Sánchez and Moreno 2008) y sus variaciones.

En esta fase se identifican dos tareas importantes:

**Tarea 1. Selección de Términos:** El **desarrollador** crea el módulo (Detector FN) en el paquete extractor de conceptos, el cual toma cada documento pre-procesado en la fase anterior y se extraen sus términos en una estructura de datos que permita representar cada documento como un conjunto de términos, entre ellos las *frases nominales*<sup>6</sup>. El uso de frases nominales es decisión de los desarrolladores y depende de los objetivos del índice, generalmente es aconsejable realizarlo, con el fin de aumentar la posibilidad de encontrar conceptos derivados. Se puede usar el algoritmo propuesto por Baziz (2005), relacionándolos con Tesoros y Ontologías generales (Ej. WordNet) para comparar las relaciones jerárquicas de estas con las frases extraídas. El producto es un vector de términos candidatos y que están directamente relacionados a cada documento de la BDD.

**Tarea 2. Obtención de los Conceptos:** El **desarrollador** implementa un módulo (extractor semántico) en el paquete extractor de conceptos, implementando cualquiera de las siguientes dos opciones, que se pueden implementar por separado o combinados: La primera forma es la **Obtención de conceptos basado en ontologías y/o tesauros**, esta es la más usada, los términos obtenidos en la tarea anterior, son consultados en la estructura de representación de conocimiento elegida construida en la fase 1 (los tesauros o las ontologías), para obtener la *semántica*<sup>7</sup> de los mismos, la cual se expresa a su vez como otro conjunto de términos (conceptos) relacionados que se obtienen de éstas estructuras. La segunda forma es la utilizar de un **espacio conceptual** (Schatz 1997; Chung, He et al. 1999), el cual parte fundamentalmente de la premisa de que los conceptos presentes en un documento tienen interrelaciones que se pueden obtener a partir de técnicas de correlación estadística. En la construcción de espacios conceptuales es necesario extraer frases nominales y normalmente los proyectos agregan anotaciones sobre las mismas en los documentos preprocesados identificando sus características para el siguiente análisis, para ello utilizan etiquetadores semánticos, los cuales son módulos de programa

---

<sup>6</sup> Una frase nominal es un conjunto de términos y constituye en sí misma un nuevo término (concepto) de indexación

<sup>7</sup> Según el diccionario de la Real Academia de la Lengua Española es “Estudio del significado de los signos lingüísticos y de sus combinaciones, desde un punto de vista sincrónico o diacrónico” [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=semántica](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=semántica), Consultado 28/04/2011.

que se encargan de asignar etiquetas a cada concepto. El producto final es un vector de conceptos, relacionados directamente con cada documento correspondiente de la BDD.

**Tarea 3. Desambiguar los Conceptos:** El **desarrollador** implementa el módulo (Desambiguador), en el paquete extractor de conceptos. En este módulo, cualquiera que haya sido la opción de obtención de los conceptos se lleva a un analizador lexicográfico con el fin de hacer el proceso de desambiguación. El producto es un conjunto de conceptos que están relacionados a cada documento de la base de datos documental. Se tienen vectores de conceptos relacionados a los documentos, los conceptos pueden estar presentes en el documento mismo o no, de acuerdo a la expansión de conceptos realizada.

**Actividad 3. Estudio de Representatividad (AER):** Esta actividad toma los vectores de conceptos y documentos obtenidos en la fase anterior con el fin de establecer pesos y/o prioridades de los conceptos con respecto a cada uno de los documentos a los cuales está relacionado. Esto se logra de dos formas: Aumentando la importancia a conceptos muy relacionados y/o eliminando conceptos que alejan el documento, eso depende de lo que se busca. Esto último se logra realizando un estudio o evaluación de representatividad de los conceptos y sus relaciones con los documentos y sus relaciones entre ellos mismos, para esto se utilizan los algoritmos de similitud semántica, los cuales pretenden encontrar, clasificar y ordenar por mayor cercanía o similitud los conceptos y por ende los documentos a los cuales están relacionados. Es aquí donde se encuentra la principal diferencia con los indexadores clásicos. Las tareas de esta fase se resumen a continuación:

**Tarea 1. Análisis de coocurrencia:** El **desarrollador** crea un módulo (Analizador de coocurrencia) en el paquete de Representatividad de Conceptos. En este módulo cualquiera que sea el modelo a seguir normalmente se inicia con un análisis de coocurrencia, con el cual se obtienen las frecuencias de aparición de los conceptos y/o frases nominales en los documentos. Dependiendo del algoritmo o estrategia de obtención de conceptos de la fase anterior, éste análisis puede ser un poco más complejo que simplemente hacer un conteo de las apariciones, por ejemplo si se decidió hacer un espacio conceptual, el algoritmo de coocurrencia debe tener encuenta las anotaciones hechas a las frases nominales encontradas y usar estructuras de datos más complejas como las matrices de coocurrencia (Chang and Schatz 1999). El producto es el mismo conjunto de vectores de la actividad

anterior pero con frecuencias estimadas de sus pesos en cada documento.

**Tarea 2. Análisis de representatividad:** El desarrollador crea el módulo (Analizador de Representatividad), en el paquete Representatividad de conceptos, el cual para el análisis de representatividad, se puede usar cualquier algoritmo de similitud semántica existente (Lin 1998; Chung, He et al. 1999; Resnik 1999; Desmontils and Jacquin 2002; Cord, Lombardi et al. 2005; Mazuel and Sabouret 2008; Lv, Zheng et al. 2009; Samaneh Chagheri 2009; Song, Li et al. 2009); sin embargo, la medida de similitud y ponderación de frecuencia con TF-IDF (Salton and McGill 1986) y las variaciones personalizadas por cada investigación (Avello 2005; Herrera 2006; Samaneh Chagheri 2009), sigue siendo uno de cálculos más simples y usados para representar los conceptos más relevantes en los textos. Por otra parte, uno de los algoritmos que ha tomado mayor fuerza es la Indexación Semántica Latente (Novoa and Ballen 2007) o LSI (Latent Semantic Indexing), es una teoría matemática usada como modelo de RI, que permite determinar el uso y las relaciones de un término con su contexto, esto lo logra con la creación de vectores multidimensionales, los cuales permiten encontrar las relaciones existentes entre palabras, palabras y párrafos y entre párrafos, obteniendo lo que se conoce como un espacio conceptual. El producto final es un conjunto de vectores con conceptos más relevantes que representan a cada documento de la base de datos documental. A este conjunto de vectores algunos autores lo llaman Espacio Conceptual, independientemente de que se hayan usado tesauros y ontologías o por el uso de espacios conceptuales, ya que está basado en los conceptos y no en los términos. Por tal razón en este trabajo se define el producto de esta fase como el espacio conceptual.

**Actividad 4. Construcción del Índice Semántico (ACIS):** Finalmente, en esta actividad se procede al construir el índice semántico, el cual se puede entender como una estructura de datos sobre el texto de los documentos, para acelerar la búsqueda (Yates and Neto 1999). Existen varios tipos de estructuras como los arboles invertidos (Zobel 2006), arreglos de sufijos, archivos firmados y los índices distribuidos (Benavides 2011). Aquí se tiene en cuenta el modelo desarrollado en la primera fase, el cual servirá como principal guía de implementación de la estructura. Los pasos son los siguientes:

**Tarea 1. Implementación del Índice Semántico:** La mínima información que se debe almacenar en la estructura definida son los vectores de conceptos vs. el documento que representan. Así, por cada conjunto de conceptos, se debe poder obtener los documentos con los que están relacionados. Sin embargo, los objetivos y las particularidades establecidas en el modelo semántico de indexación podrían establecer información adicional, como almacenar posiciones en el texto, relaciones con otros conceptos, contextos, datos de sensores, entre otras, junto a los conceptos y a los documentos representados. Esto hace una buena diferencia con respecto de los índices normales, ya que se podría obtener más información de los documentos buscados, consultando sólo el índice. El **desarrollador** implementa el módulo (generador de índice) en el paquete motor de indexación, el cual puede ser tan complejo como el modelo lo requiera y tener a su vez más submódulos que trabajen entre sí. El producto es código fuente que permite construir una estructura de datos compleja con la información extraída de la BDD.

**Tarea 2. Implementación de Servicios Web Semántico de Objetos:** En este punto se crean servicios Web asociados a los servicios de información de los sensores que han sido indexados, con el fin de poner a disposición de las aplicaciones las consultas de los objetos. Para esto se crea un repositorio de Servicios Web, que puede ser consultado y reutilizado por las aplicaciones. El **desarrollador** implementa un módulo (Creador de Servicios Web de Sensores) en el paquete motor de indexación, el cual puede incluir procesos automáticos o semiautomáticos de despliegue de éstos servicios en un repositorio que puede ser consultado por aplicaciones que consuman servicios.

**Tarea 3. Implementación de Servicios de Índice:** En esta tarea el **desarrollador** crea un conjunto de Servicios Web, que capturan la consulta de los clientes para ser procesada con respecto de los recursos indexados y los servicios que éstos proveen. Este módulo puede ser tan complejo como el modelo que se haya creado en la primera fase, ya que se pueden incluir trabajos de preprocesamiento semántico a la consulta (Niño, López et al. 2012) de los usuarios y así aumentar la posibilidad de dar una mejor respuesta en los resultados. Los productos son un módulo (Interfaz de Servicios de Índice) que expone los servicios web del índice creado e interactúa con el paquete de motor de indexación.

**Tarea 4. Evaluación del índice construido:** Esta tarea participa el **ingeniero de pruebas** (tester) el cual es el responsable de implementar

los casos de test, implementarlos y ejecutarlos. Se pueden diseñar pruebas de diversos tipos: utilizando simuladores (Römer, Ostermaier et al. 2010), pruebas con los clientes y usuarios o utilizando testbeds (Gluhak, Krco et al. 2011) que son servicios reales de sensores con los cuales se pueden probar las implementaciones. El equipo de desarrollo decide las pruebas a desarrollar y las implementan de acuerdo a su modalidad. Se sugieren las siguientes actividades de evaluación:

- a. Realización de consultas para verificar la relevancia de resultados obtenidos con el índice semántico en pruebas alfa. Posteriormente, hacer las pruebas beta para tener realimentaciones tempranas con los clientes.
- b. Depurar la Consulta (opcional). Se puede realizar un procesamiento a la consulta de manera que se obtengan conceptos relevantes en ella. Esto se hace por medio de un mapeo de los términos de la misma con los conceptos de la ontología (dominio específico). Si no encuentra los conceptos de la consulta en la ontología, se realiza un mapeo semántico con un tesaurus, un vocabulario controlado o una base de datos léxica de dominio general. Al realizar esta comparación se extraen los conceptos que se asemejen (sinónimos) a los términos de la consulta y así, se comparan con los documentos relevantes de acuerdo al índice construido.
- c. Para observar los resultados con respecto a la relevancia, se puede utilizar los indicadores de evaluación de recuperación de la información, para esto se realizan evaluaciones exhaustivas, las cuales pueden ser realizadas con uno de los siguientes enfoques: Las primeras orientadas a evaluaciones automatizadas sobre colecciones de información existentes (cerradas) y previamente valoradas (documentos, consultas y documentos relevantes para cada consulta), para establecer la *Precisión* (Kent, Berry et al. 1955; Salton and McGill 1986; Frakes 1992), *Recuerdo o Exhaustividad* (Kent, Berry et al. 1955; Swets 1969; Salton and McGill 1986) y el *índice MAP*, entre otras; las segundas están relacionadas con evaluaciones realizadas por usuarios (colecciones abiertas) donde se pueden evaluar medidas derivadas de la precisión y el recuerdo y medidas como el *estadística Kappa* (Manning, Raghavan et al. 2008) para evaluar la confianza de los resultados. Finalmente, es necesario establecer otros índices relevantes para la IoT, dado que allí se manejan consultas diferentes a los buscadores tradicionales de la web.

Los productos son los documentos detallados de las pruebas hechas y los documentos de análisis de las mismas, con detalle en diseño de la solución y la implementación de la misma.

#### **3.2.4 Fase 4 Despliegue del Índice Semántico (FDIS)**

En esta fase se implanta el índice semántico en la infraestructura disponible para su puesta en marcha. Se debe desarrollar un plan de instalación de versiones finales y su correcta comunicación a los usuarios, verificando la existencia de los manuales correspondientes que debe acompañar un software:

**Actividad 1. *Elaboración de los documentos de despliegue (AEDD):***

Esta actividad el ingeniero en despliegue (deployment Engineer) realiza las tareas necesarias para que el índice llegue a sus clientes como un producto finalizado.

**Tarea 1. *Elaboración del Plan de Restitución:*** La tarea consiste en elaborar un documento al equipo de despliegue por parte del **ingeniero en despliegue**, con toda la información que necesita la organización de apoyo a la producción para determinar si se necesita una restauración o ¿qué se autorice? O ¿cómo se va a realizar?, entre otras. El producto es un documento de restitución.

**Tarea 2. *Ejecutar el despliegue:*** Una vez elaborado el plan los **ingenieros de despliegue** siguen el plan y recolectan la información necesaria para establecer que ha quedado bien implementado y puesto en producción en índice semántico. El producto es un informe del despliegue y el índice funcionando correctamente, más una comunicación del mismo a sus usuarios.

Este método es una aproximación a los pasos y elementos importantes que se deben tener en cuenta al momento de crear un índice semántico en la IoT. Debe ser refinado y revisado con cada una de las implementaciones que se desarrollen del mismo.

El método propuesto es muy adaptable a las necesidades de implementación, ya que puede funcionar como complemento software autónomo y que apoya a otros servicios o aplicaciones, o también se pueden eliminar actividades y tomar decisiones de implementación dependiendo del uso que se le desee dar, por ejemplo los middleware y servidores IoT pueden implementarlos tomando sólo la fase de construcción, ya que ellos proveen las capas de acceso y así desarrollar estructuras semánticas que mejorarían sus servicios.

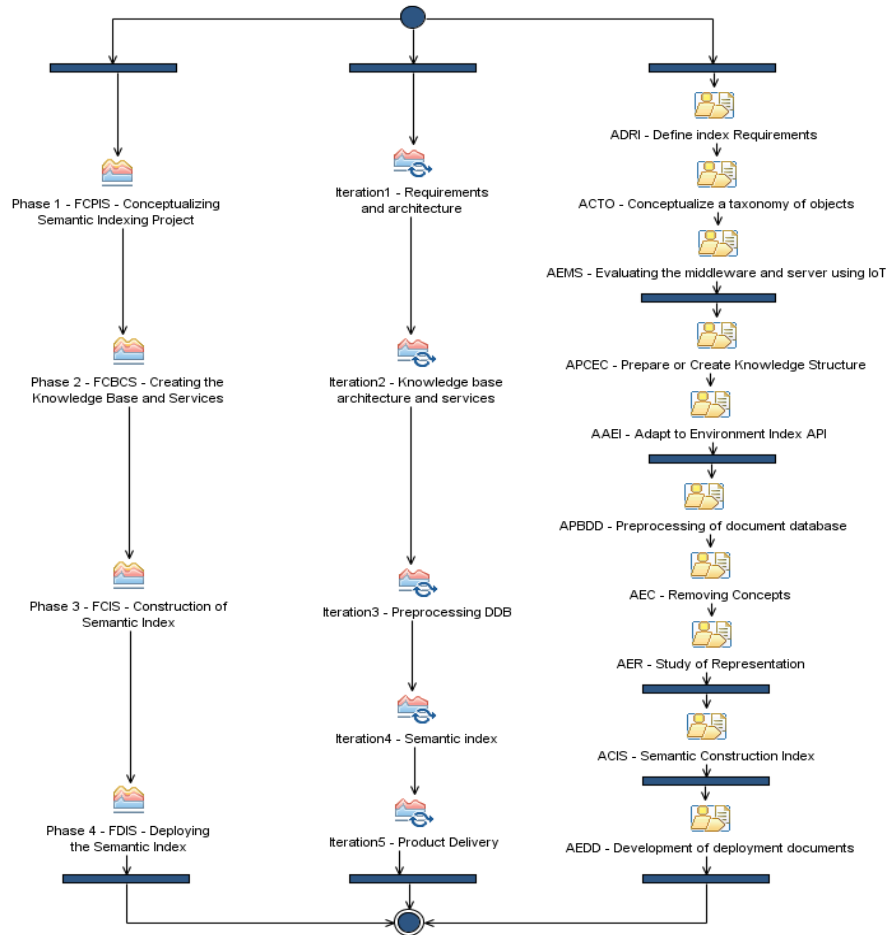
En el siguiente apartado se presenta una especificación formal en SPEM y utilizando la herramienta EPF – Composer.

### **3.3 Especificación del Método para la Creación de Índices Semánticos en la IoT en SPEM 2.0**

EPF Composer administra, busca y desplegar una publicación web el método SIMIOT que sigue la metodología ágil de desarrollo OpenUP. EPF Composer hace una diferenciación entre los contenidos de método (Method Content) y los procesos (Processes), estos contenidos se van incorporando a los procesos teniendo en cuenta el rol que realizara la cierta tarea y los productos de trabajo que son los insumos o resultado de dichas tareas, también se llevan guías que sirven como ayuda para realizar las tareas o crear los productos de trabajo adecuadamente.

En la Figura 11 se muestra el diagrama de actividades para el método en la primer columna se muestra las fases de SIMIOT en la segunda las iteraciones que definen los micro incrementos, y en la columna tres las actividades definidas para el método. SIMIOT.

Figura 11. Diagrama de Actividad SIMIOT de EPF Composer



### 3.4 Conceptos SPEMv2.0 y Diagramas de SIMIOT

En el ANEXO B-2 se pueden observar más detalles de los productos que genera la herramienta EPFC Composer y la especificación desarrollada. En este sitio Web: <http://semanticsearchiot.net/sswt/simiot/publish/index.htm>, se puede ver toda la especificación con el fin de que sea reutilizada por un grupo de desarrollo.

### 3.5 Conclusiones Método SIMIOT

- La Web Semántica de los Objetos encierra en sí misma más retos que los que maneja la web semántica actual, dado que debe incorporar las mismas tecnologías semánticas, pero adicionalmente sortear correctamente las restricciones propias de la IoT, la propuesta de crear índices semánticos en la IoT, guiado por un modelo semántico conceptual y personalizado a las

necesidades de información de los usuarios, podría ser una solución que aporta adecuadamente al aprovechamiento de la IoT.

- La creación de un índice semántico en la IoT, requiere tener en cuenta muchos aspectos de diseño e implementación utilizando tecnologías semánticas y de la IoT que deben armonizarse correctamente entre sí, para poder lograr un equilibrio entre los objetivos del negocio y las capacidades reales de los objetos. Así la propuesta de un método para la creación de índices semánticos en la WoT, se convierte en una buena herramienta que permite guiar a los desarrolladores y los usuarios de la web en su aprovechamiento de la IoT generando nuevos servicios de conocimiento.
- Es necesario desarrollar varias instancias del método propuesto con el fin de afianzar y realimentar el mismo, para que se convierte en una herramienta más estable y que permita ganar confianza en su posibilidades.

## **BUSCADOR SEMANTICO EN LA WOT ACERCA DE LA CONTAMINACION AMBIENTAL**

Este capítulo realiza la implementación de un índice semántico sobre la WoT, para el caso del dominio de contaminación ambiental. Se realiza una instancia del método definido en el capítulo anterior, mostrando en cada fase y actividad las decisiones tomadas para la construcción de dicho índice. Finalmente, se crea el prototipo de buscador semántico para WoT que utiliza el índice.

### **4.1 Fase I – Actividad 1 – Tarea 1: Visión del Sistema**

#### **4.1.1 Introducción**

El presente documento tiene como finalidad mostrar la visión del índice semántico creado y un análisis de las funcionalidades implementadas en el buscador semántico. Los productos principales fueron dos: El índice semántico que accede al servidor de Objetos Xively y una aplicación Web que permite encontrar objetos dada una consulta en el dominio de contaminación medioambiental en ciudades específicas de Colombia.

#### **4.1.2 Definición del problema**

Para el caso particular se desea crear un buscador semántico sobre el servidor IoT llamado Xively, que fue seleccionado de acuerdo a unas características particulares. Este servidor tiene conectados millones de objetos con sus correspondientes datos y metadatos. La idea principal es que cualquier persona que utiliza el buscador pueda encontrar fácilmente objetos que le permitan reutilizar los datos que pertenecen al dominio particular. El dominio seleccionado como el caso de estudio ha sido la información y contaminación medioambiental. La razón de la selección de dicho dominio es que es uno de los campos en los que más sensores se han desplegado en la IoT y de los cuales sería más fácil obtener mayor cantidad de datos. Adicionalmente es muy poco lo que se ha encontrado de aplicaciones de la WoT en el campo de la contaminación medioambiental.

Existen buscadores tradicionales como google, bing, altavista y en el mismo servidor IoT, en los cuales se pueden realizar búsquedas sobre los eventos medioambientales. Sin embargo, es necesario hacer navegación por varios minutos para poder elegir las fuentes de datos y en algunos casos se trata de páginas web que proveen la información y no de los sensores existentes y públicos de la WoT. Para el caso colombiano la entidad oficial que está encargada de implementar el sistema de gestión ambiental es el IDEAM (<http://www.ideam.gov.co/jsp/index.jsf>), esta entidad ha implementado en todo el

país sistemas hardware con sensores para medir variables climatológicas y ambientales, además de construir sistemas de información geográfica y bases de datos para recoger las mediciones. Lamentablemente, los usuarios que deseen acceder a dicha información lo deben hacer a través de las aplicaciones Web de la entidad y los datos para usuarios generales corresponden en su mayoría son históricos y no mediciones en tiempo real. Para éstos últimos es necesario firmar convenios o realizar pagos por dicha información.

Con la llegada de los servidores de objetos IoT, los cuales permiten a cualquier usuario de la Web crear y conectar objetos y sensores a la Web, se ofrece la posibilidad de utilizar esta información como posibles fuentes gratuitas para realizar búsquedas de información relevantes a los usuarios de la misma Web.

La idea principal es mejorar la búsqueda de objetos de la IoT con información semántica de contexto de su: localización, variables que mide, si está activo o no a través de un modelo de conocimiento mediado con una ontología de aplicación de contaminación medioambiental y otras ontologías existentes que optimizan el proceso como la SSN-XG y objeto semántico.

Para el caso particular el usuario desea buscar información de los sensores relacionados a variables de contaminación medioambiental en una región particular de Colombia. La aplicación presenta un Ranking de los sensores que cumplan con los criterios de búsqueda y despliega su información para conocimiento del usuario.

En la Tabla 9 se muestra los cuatro aspectos principales de la visión del sistema.

**Tabla 9: Visión de BSIOT**

El problema de:	¿Cómo realizar consultas a la WoT (Servidor IoT) con el fin de mejorar el descubrimiento de sensores relacionados a variables específicas de la contaminación medioambiental?
Afecta:	Usuarios de la Web que desean obtener información de los dispositivos de la WoT relacionados a la información de contaminación medioambiental en una región.
El impacto es:	La creación de Índice medioambiental y del servicio de búsqueda pueden ayudar a: <ul style="list-style-type: none"><li>• Mejorar el tiempo de Descubrimiento y Búsqueda de los objetos de la IoT que se pueden utilizar en la contaminación medioambiental.</li><li>• Agregar rápidamente los servicios de los objetos a aplicaciones.</li></ul>
Una solución viable sería:	Tener un servicio de búsqueda Web de sensores de la WoT que provea información contextualizada a su localización y capacidades de medición de variables de interés en el campo de contaminación medioambiental.

### **4.1.3 Ubicación de los Productos**

El producto “Visión del Sistema” que contiene un resumen de los requisitos básicos establecidos para sistema. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 1\Actividad 1\Tarea 1\ Visión del Sistema

## **4.2 Fase I – Actividad 1: Tarea 2: Especificación de Requisitos Buscador Semántico – BSIoT**

### **4.2.1 Introducción**

En este apartado se presentan los requerimientos funcionales y no funcionales del sistema, también se establecen el conjunto de productos que se obtendrán en las fases posteriores.

### **4.2.2 Requerimientos Funcionales**

La lista de requerimientos funcionales es la siguiente:

- R1. **Registro:** Esta es una funcionalidad que le permite al usuario darse de alta en el sistema con el fin de poder recaudar la información de sus calificaciones de relevancia.
- R2. **Iniciar Sesión:** Es una funcionalidad que le permite al usuario identificarse en el buscador con el fin de activar las funcionalidades reservadas para los usuarios.
- R3. **Realizar Búsqueda:** Esta funcionalidad permite a los usuarios digitar una consulta en lenguaje natural con el fin de obtener resultados acerca de los objetos que cumplen con la consulta realizada.

#### **4.2.1 Interfaces del Sistema***Interfaces de Usuario*

La interfaz debe proveer una zona para digitar la consulta, una zona para ejecutar la consulta, otra zona en los que se encuentren unos menús textuales de registro, inicio de sesión y finalización de la sesión.

#### **4.2.2 Interfaces a Sistemas Externos**

##### **4.2.2.1 Interfaces de Software**

Las interfaces están desarrolladas como un repositorio de servicios Web que podrían ser semánticos. En la salida de los sensores que coinciden con el criterio de búsqueda se les coloca un enlace a las funcionalidades de los servicios web desarrollados para consumir sus datos. En este punto, sin embargo, es de esperar que no todos los sensores tengan los servicios desarrollados, ya que por ahora se enfoca a un trabajo manual y no automatizado, por salirse de los alcances del presente proyecto. Los servicios web del índice se expondrán mediante un enlace a una página de servicios web disponibles para su uso con otras aplicaciones.

##### **4.2.2.2 Interfaces Hardware**

Sin interfaces de hardware.

##### **4.2.2.3 Interfaces de Comunicaciones**

Las interfaces de comunicación son abstraídas por el servidor IoT.

#### **4.2.3 Reglas de Negocio (Business Rules)**

Cómo el índice que se va a trabajar es en el área de la información medioambiental y la contaminación, se espera que el índice permita alimentar un buscador que soluciones las siguientes reglas de negocio:

- R1. **Consultas de variables medioambientales:** Esta es una funcionalidad que le permite al usuario solicitar información de las variables medioambientales tales como: magnetismo, humedad, radiación solar, presión atmosférica, calidad del aire y viento en un lugar específico.

**R2. Consultas sobre contaminación medioambiental:** Es una funcionalidad que le permite al usuario solicitar información de los agentes contaminantes que son medidos a través de los sensores en un lugar específico.

#### **4.2.4 Documentación**

Se documenta la instanciación del método y del código fuente desarrollado. Finalmente se debe crear un manual de usuario de la herramienta.

#### **4.2.5 Ubicación de los Productos**

El producto “Especificación de Requisitos v2” los requerimientos funcionales y no funcionales del sistema. Se encuentra en la siguiente ruta del CD de entrega:

C: \CD del Proyecto \IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 1\Actividad 1\Tarea 2

### **4.3 Fase I – Actividad 1: Tarea 3: Arquitectura del Sistema BSIoT**

#### **4.3.1 Introducción**

El presente documento resume la arquitectura definida para crear el índice semántico en la IoT. Del estado del arte se han definido servicios específicos que se van a utilizar para la creación del índice. Por ello la arquitectura aquí presentada define los servicios a utilizar. Sin embargo, el servidor de objetos IoT no se había seleccionado en su momento, por lo tanto se asumieron algunas características que debería tener dicho servidor y que se tuvieron en cuenta al momento de hacer el estudio de las características del servidor.

#### **4.3.2 Objetivos de la Arquitectura y Filosofía**

Debido a que esta arquitectura se deriva de la propuesta realizada en el método para crear índices semánticos en la IoT, los referentes de su filosofía son los mismos expuestos en esta parte. Se adaptan al desarrollo específico los objetivos así:

- **Eficacia en la extracción:** Consiste en obtener la mayor cantidad de semántica posible de los objetos como consecuencia de esto se mejora la extracción. Los datos se obtendrán de los metadatos provistos por el Servidor IoT seleccionado. El servidor debe proveer dos tipos de información a indexar:
  - **Información Estructurada:** Esta información viene en formatos XML o JSON y campos de captura de datos a manera de metadatos, los campos se llenan en el momento en que cada usuario crea su sensor en el servidor IoT; la información de los metadatos es relacionada directamente a los conceptos en las ontologías, así se asegura una alineación correcta de dichos datos para poder ser incluidos al momento de ser indexados. Para que sean tenidos en cuenta, se realiza una

anotación semántica de dichos conceptos en el archivo después de eliminar las palabras vacías.

- **Información No estructurada:** Los formatos XML y JSON del servidor, proveen información no estructurada en palabras clave, descripciones y demás campos abiertos en los cuales el usuario incorpora información de acuerdo a sus percepciones. Son convertidos en una bolsa de palabras que se agregan a las anotaciones anteriores y posteriormente es indexada semánticamente.
- **Mecanismo eficiente de captura de datos:** El servicio del API del servidor IoT tiene ya una forma eficiente de captura y reutilización de datos, utilizan metadatos que están en XML u otro formato estándar para cada uno de los sensores que almacena. Estos datos son ofrecidos través del API que ofrece el servidor IoT para usuarios finales y aplicaciones.
- **Correcto uso de estándares:** Se utilizará como principal ontología la SSN-XG para clasificar características de los sensores, con respecto a los conceptos medioambientales se ha alineado la ontología ENVO, la cual ha sido creada por expertos internacionales. Por otro lado, el formato en los cuales se transmite la información es XML o JSON.
- **Selección adecuada del Modelo Semántico:** Se refiere al correcto análisis de los datos obtenidos provenientes de los dispositivos, se estudia la representatividad (inferencia de conceptos) y adecuado almacenamiento en una estructura de información eficiente, se realiza recuperación y actualización con uso de (estructura de datos y algoritmos). Adicionalmente, se deb manejar un modelo adecuado para la información en tiempo real y la información que se consulta posteriormente. Este modelo se presenta en detalle en la solución propuesta más adelante.
- **Exposición de servicios:** Se utilizarán Servicios web que expondrán los datos del índice semántico desarrollado.

#### 4.3.3 Dependencias y Asunciones

La lista de todo lo que se asumió para la arquitectura:

- Se utilizará un servidor IoT que será seleccionado en tareas posteriores, del cual se implementará a partir de sus API, la funcionalidad necesaria para consultar y extraer los archivos de metadatos en JSON.
- Se utilizará la ontología SSN-XG, WordNet y una ontología de Objeto Semántico propuesta por la tesis de doctorado del Magister Miguel Ángel Niño, con el fin de almacenar el conocimiento necesario para indexar los objetos relacionados en la consulta, así como la alineación a la ontología ENVO.
- Se utilizará la Herramienta CMAPS – COE que permite crear modelos conceptuales y después convertirlos en ontologías OWL, las cuales serán refinadas posteriormente con otra herramienta.

- Se utilizará Protege como herramienta de ontologías para alinear las ontologías seleccionadas.
- Se utilizará infraestructura de terceros para los procesos de indexación de la información de los objetos del servidor IoT. La infraestructura es: Lucene.NET, Json.NET, NGeo, RestSharp, StemmersNet, APIs de Xively y APIs de WordNet.
- Se realizará un enriquecimiento de los archivos de metadatos e información de los objetos provistos por el servidor IoT de acuerdo a las ontologías utilizadas y luego se indexará utilizando el cálculo de TF –IDF (frecuencia de términos – frecuencia inversa del documento), donde se mide la frecuencia de los conceptos candidatos en los documentos que fueron recuperados, estos resultados son posteriormente guardados en una matriz.
- Para el cálculo de la similitud en el índice se hará la asignación de pesos a los conceptos de la ontología, teniendo en cuenta la jerarquía de la misma. Los pesos fueron asignados consecutivamente dando más peso a los hijos que los padres esperando tener mejores resultados en las búsquedas. La estrategia es mediante una expansión de consulta de usuario.

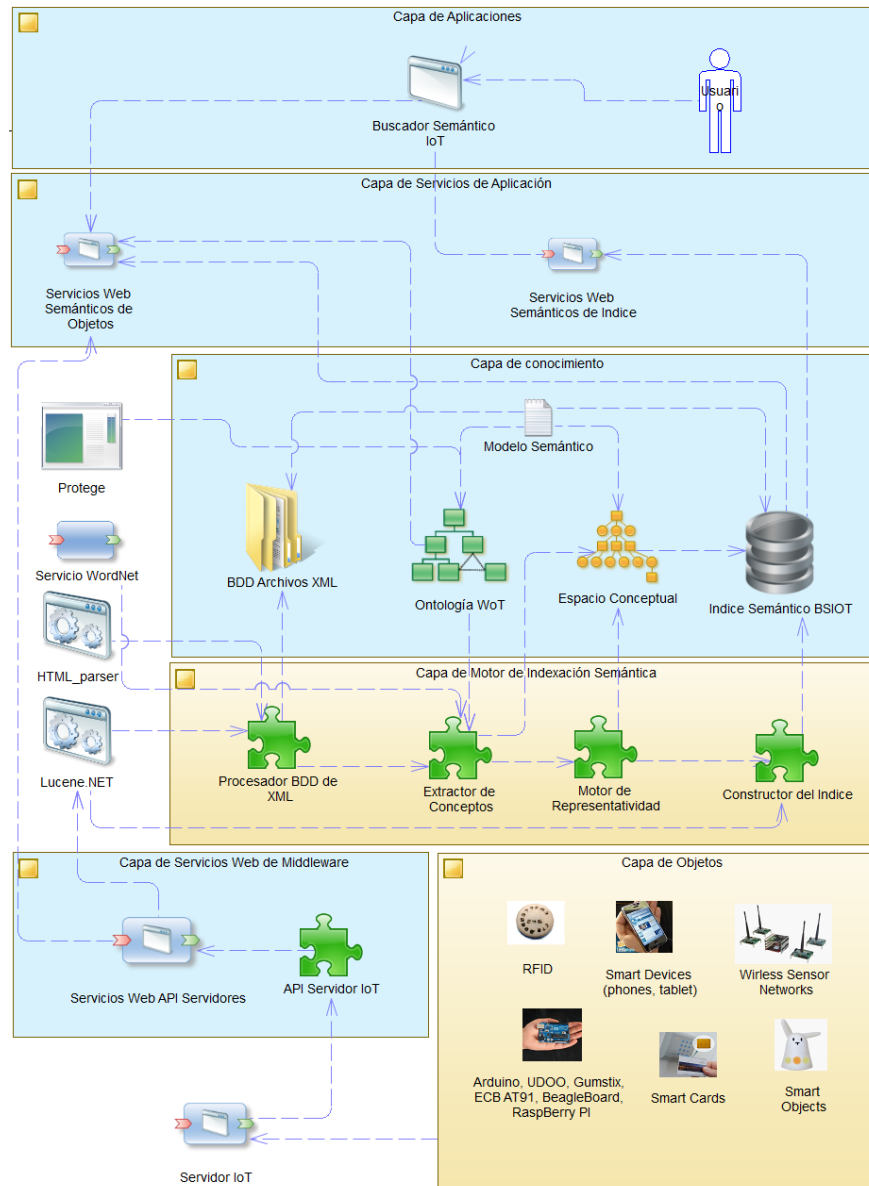
#### **4.3.4 Vistas Arquitectónicas**

En este punto se presenta la vista lógica del sistema, como una personalización de la arquitectura general propuesta en el método definido y una vista de casos de uso generales, los cuales definen los requisitos específicos de la aplicación de búsqueda semántica a desarrollar.

##### **4.3.4.1 Vista Lógica de la Arquitectura**

Esta vista describe la estructura y el comportamiento de las partes componentes del sistema. Se reutiliza la arquitectura propuesta en el método y se personaliza a las decisiones tomadas. En la Figura 12 se puede apreciar en azul las capas que se han personalizado con respecto de la arquitectura general propuesta en el método.

**Figura 12: Arquitectura Lógica del Buscador Semántico IoT**



Las herramientas software y servicios que son ajenos al sistema se han colocado por fuera de las capas. Se pueden apreciar las herramientas específicas y en que etapas de proceso son utilizadas en incluidas en el sistema.

Siguiendo con el análisis de la figura en la capa de servicios está el servidor IoT, el cuál será seleccionado en el estudio de evaluación de servidores. Este servidor debe ofrecer un API de desarrollo que permite acceder a todos los objetos de la IoT que tenga conectados a su base de datos centralizada. Dado que el servidor

IoT no permite la interacción hacia los objetos el modelo está realizado sólo para obtener datos de los mismos.

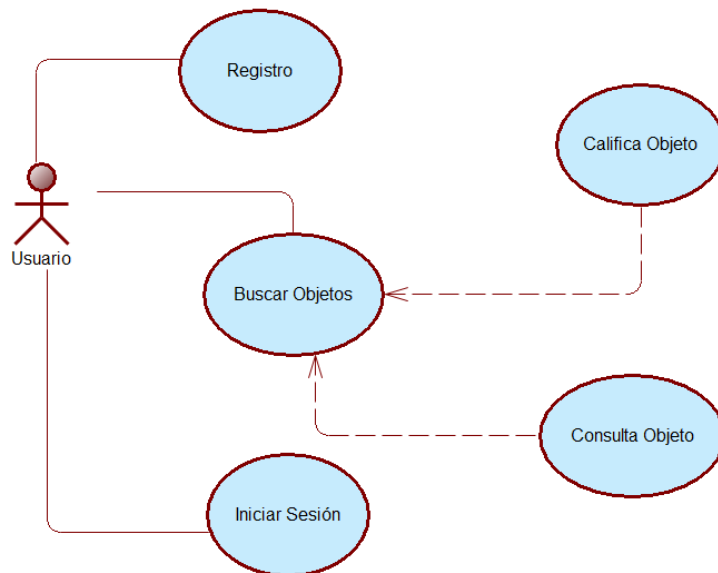
La capa de indexación es igual a la propuesta en el modelo general, sin embargo cuando se realice la implementación de cada una de ellas se podrá ver las decisiones específicas de indexación semántica que se reúnen en el documento del modelo semántico, el cual afecta cada uno de los productos de la indexación que se encuentran en la capa de conocimiento.

Capa de conocimiento, los productos más importantes son el índice semántico y la personalización de la ontología, ésta se debe desarrollar con el caso de estudio particular escogido que fue la contaminación medioambiental y la alineación a las estandarizadas existentes.

Finalmente, la capa de servicios, exponen servicios web para ser utilizados en la capa de aplicaciones. Estos servicios pueden acceder directamente al índice, a la ontología y al servicio creado sobre el API del servidor de IoT seleccionado, con el fin de obtener la información necesaria de primera mano.

#### 4.3.4.2 Vista de Caso de Uso General

Figura 13. Modelo General de Casos de Uso



En la Figura 13 se pueden ver los casos de uso generales del sistema. El usuario puede realizar en general tres acciones cuando ve la página web por primera vez:

Registro, Buscar Objetos o Iniciar Sesión. La versión final incluyó un segundo actor (Administrador) y opciones de gestión del índice y la ontología.

Si no se ha registrado puede hacerlo, sin embargo no es necesario que se registre para poder realizar las búsquedas, por defecto el usuario es iniciado como anónimo y se le permite también calificar y consultar objetos.

Se inicia sesión cuando se necesite identificar al usuario ya sea para que califique los objetos recuperados y poderlos asociar a un usuario particular y también ofrecer en el futuro mejores funcionalidades a los usuarios registrados.

#### **4.3.4.1 Ubicación de los Productos**

El producto “Visión del Sistema” contiene la arquitectura definida para crear el índice semántico. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Índice IoT\Fase 1\Actividad 1\Tarea 3.

### **4.4 Fase I – Actividad 1: Tarea 4: Modelo Semántico BSIoT**

#### **4.4.1 Introducción**

El presente documento presenta el modelo desarrollado para la construcción del índice semántico en la IoT. Este modelo incorpora los cuatro modelos que requiere la arquitectura general propuesta en la metodología de desarrollo. Primero se realiza una respuesta general a las preguntas guías para el desarrollo del modelo y posteriormente se profundizan con la presentación de los submodelos desarrollados para el caso de estudio particular.

#### **4.4.2 Preguntas de Referencia del Modelo**

Debido a que este modelo se deriva de la propuesta metodológica de desarrollo de índices semánticos para la IoT, en esta primera parte se introduce a como se responden las preguntas del modelo semántico para el caso particular de desarrollo:

1. **¿Cuáles son los usuarios del índice semántico?:** Los usuarios del índice son personas de la Web que estén interesadas en buscar objetos de la WoT que midan condiciones ambientales con el fin de reutilizar su información.
2. **¿Cuáles son los tipos de consultas que va resolver el índice semántico?:** Ya que el dominio del índice semántico se relaciona con el medioambiente se definen primero el rango de mediciones medioambientales cómo lo siguiente: temperatura, niveles de ruido, humedad, calidad del aire (CO2), presión

atmosférica, calidad del agua, radiactividad, movimiento. Las consultas que puede resolver son:

- a. ¿Qué dispositivos cercanos a <<lugar>> pueden entregarme <<información medioambiental>>?
- b. ¿Qué <<información medioambiental>> puedo obtener de <<lugar>>?
- c. ¿Cuál es el histórico de la <<información medioambiental>> en el periodo de tiempo <<periodo>>?
- d. ¿Qué <<variable medioambiental>> se puede medir en <<lugar>>?
- e. ¿Cuál es la tendencia de la <<variable medioambiental>> en <<lugar>>?
- f. ¿Cuáles <<variable medioambiental>> presentan <<contaminación ambiental>> en <<lugar>>?

Los tipos de consultas a), b), d) y e) se pueden obtener directamente de la información almacenada en el índice, a su vez el índice se crea en una primera etapa con consultas directas a al servidor IoT a través del servicio web de API. Las preguntas c) y f) se deben consultar en tiempo real al servidor IoT por datos y luego calcular las mediciones solicitadas a través del modelo de servicios que se desarrolle.

3. **¿Cómo se van a manejar las consultas que hacen los usuarios para que se pueda entregar resultados relevantes?:** Las consultas de los usuarios se realizará un preprocesamiento similar a la de los documentos, con el fin de expandir su consulta y alinearla con los conceptos de la ontología medioambiental seleccionada y adaptada a las necesidades del proyecto.
4. **¿Qué cantidad de información se va a indexar y cual se va a consultar directamente al sensor?:** Se va a realizar consultas sobre el servidor IoT acerca de todos los conceptos relacionados con la información medioambiental, que se encuentran en la ontología que ha sido adaptada para la creación del índice. Por cada sensor que coincida con la búsqueda se recupera la información en formato XML de sus datos y metadatos. Cuando se necesite información de los datos actuales o datos históricos se realizará una consulta directa al servidor IoT, recuperándolos y presentándolos o reprocesándolos en caso de que sea necesario.
5. **¿Cómo se va a manejar la concurrencia de consultas a un mismo sensor?:** La concurrencia se manejará en dos niveles, el primero es que el API del servidor IoT maneja la concurrencia a través del servidor Web en el cual se alojen los servicios. El acceso a los datos del índice se van a realizar en una base de datos SQL Server, con el fin de manejar la concurrencia con el mismo motor. Dado que los dispositivos no reciben instrucciones para actuadores, ya que el sistema es de sólo consulta, por ello no existe problemas de instrucciones conflictivas al mismo tiempo.
6. **¿Cómo se va a modelar el contexto que necesariamente está asociado a los sensores?:** El contexto que se va a manejar inicialmente se cierra a los datos primarios (Perera, Zaslavsky et al. 2013) de localización, identidad, tiempo y actividad de los sensores, los cuales en el servidor IoT deben estar

asociados a una ubicación geográfica y unos metadatos opcionales que coloca el creador del objeto. Si el tiempo lo permite se puede explorar datos secundarios en las mismas categorías. Los datos secundarios corresponden a un preprocesamiento de los datos de acuerdo a necesidades específicas. Sin embargo el principal dato secundario es el cálculo de cercanía de sensores respecto de un lugar particular.

7. **¿Cómo se va a manejar la frecuencia de actualización del índice?:** Con respecto a la frecuencia se pretende hacer dos etapas. La primera antes de liberar el buscador se iniciará el proceso de poblar el índice con los sensores actuales en el servidor IoT seleccionado. Posteriormente se libera el Buscador y este trabajara con los sensores que fueron indexados en esa primera etapa. Si es necesario hacer una actualización se procederá de la misma manera pero de forma manual y programada.
8. **¿Cuál es la estructura de datos más adecuada para almacenar el índice?:** Se decidió adaptar una ontología de dominio específico en información medioambiental (ENVO<sup>8</sup>) a una ontología de contaminación ambiental desarrollada para este proyecto. Adicionalmente estas ontologías serán alineadas a su vez con la Ontología “Objeto Semántico” aportada por Niño, con el fin de tener una ontología que permita incluir el conocimiento necesario a los sensores para que se puedan responder las preguntas realizadas en el buscador.
9. **¿Cómo se va a compartir el índice para su correcto aprovechamiento?:** El índice creará unos servicios Web semánticos que se publicarán en un servidor Web público. Los resultados de las búsquedas sobre este servicio permitirán exponer información de búsqueda en el dominio medioambiental y enlaces a servicios web de los objetos específicos para que sean aprovechados por aplicaciones de terceros.

#### **4.4.3 Modelos del Índice Semántico a Construir**

En los siguientes apartados se determinan los modelos encargados de establecer las estructuras de conocimiento a utilizar, establecer el dominio conceptual de objetos, estructura de datos y algoritmos para la recuperación y actualización de los conceptos.

##### **4.4.1.1 Modelo de Conceptos**

Este modelo presenta las decisiones sobre cómo manejar la información del contexto de los dispositivos, es decir la información que se interrelaciona a los datos generados por el sensor o dispositivos y se presentan decisiones de cómo se va a manejar la información semántica de los objetos. En el **ANEXO C-1.1** se

---

<sup>8</sup> Environment Ontology (<http://environmentontology.org/home>), consultado, 28/06/2013.

presenta el caso de estudio el cual es la “Información Medio Ambiental y la Contaminación Producida por el hombre”.

- **Alineación con Ontologías de Dominio Existentes:** El modelo conceptual de información medioambiental y contaminación del hombre después de definir las variables, conceptos y contextos se genera un mapa conceptual el cual se convertirá en una ontología OWL. Esta ontología debe ser adaptada y formalizada para que pueda llevarse a la producción. La construcción de la ontología también contempla la utilización de otra ontología realizada por expertos la cual abarcaba conceptos de tipificación ambiental de área o lugar se realizó para estar en la línea de estandarización y aumentar la posibilidad de que la ontología desarrollada en el proyecto sea aceptada por la comunidad científica. En el **ANEXO C-1.1.1** se presenta la alineación de la ontología desarrollada de Contaminación medio ambiental con la ontología estándar ENVO.
- **Ontología de Mediciones en Contaminación Medioambiental:** Finalmente y después de las alineaciones se presenta la Ontología que se utilizará para anotar los documentos de los dispositivos o sensores que se encuentre en el servidor IoT para su posterior indexación. Esta ontología será usada en esta primera versión con propósitos de anotar semánticamente los documentos analizados del servidor IoT, para que posteriormente se indexen para su consulta. En el momento de la consulta también será utilizada para presentar ayuda de autocompletado al usuario y así guiar mejor el proceso de búsqueda. La ontología no será usada para instanciar sensores y razonar sobre la misma las consultas. Esta opción se deja para la segunda versión del proyecto o trabajo futuro.

#### **4.4.4 Modelo de Consultas Personalizadas**

Se establece la información del contexto (Zhang, Cheng et al. 2012) teniendo en cuenta que los usuarios generalmente están interesados inicialmente en las entidades (cosas, lugares, personas) y posteriormente por las cualidades o estados de esas cosas (vacío, libre, sentado, caminando, etc.) y no en sensores individuales y sus datos de salida.

En el **ANEXO C-1.2** se presenta un modelo para establecer las consultas que el buscador responde en función del caso de estudio de contaminación medioambiental.

#### 4.4.5 Modelo de Servicios

Es un modelo que establece las interfaces a desarrollar con respecto a los servicios web de los objetos y los servicios Web del índice mismo. Los servicios básicos del índice son:

1. **Buscar (consulta, idioma):** Esta función recibe un conjunto de caracteres a los cuales les hace un proceso de ampliación de consulta y posteriormente busca en el índice las coincidencias con respecto a toda la información que ha sido encontrada en el repositorio de información en la etapa anterior de indexación. Retorna un conjunto de resultados ordenados por relevancia. También recibe el idioma de manera opcional por el cual realizará la búsqueda y los resultados. Por defecto es español. Retorna un *DataSet* con un conjunto restringido de información de los sensores, el cual es traído directamente de la información almacenada en el índice y por esto es más rápida.
2. **BuscarFeeds (consulta, idioma):** Es igual a la anterior, pero se diferencia en que retorna un listado de objetos denominados *FeedXively*, los cuales reúnen la información de los json anotados semánticamente que previamente fueron almacenados en la base de datos documental en el proceso de indexación. No trae *Datapoints*, o información medida por los sensores sólo los metadatos. Es más lento que su búsqueda predecesora ya que ha lectura de archivos local.
3. **BuscarFeedsDataPoints (consulta, fechaInicio, fechaFin idioma):** Esta función realiza lo mismo que las predecesoras, pero se diferencia en que trae toda la información del dispositivo incluido los *Datapoints* o mediciones directamente desde el servidor *Xively*. Es muy costosa en términos de tiempo.
4. **RetornarConceptos (termino, idioma):** Esta función recibe un término y lo compara en la ontología con el fin de establecer si es un concepto. Si es concepto retorna los conceptos más relacionados al mismo (relaciones directas a otros conceptos), desde el punto de vista del dominio de la ontología cargada. Adicionalmente, puede especificar el idioma el cual puede ser: español, inglés o dejar vacío para traer datos en ambos idiomas.
5. **RetornarMapaLugar (latitud, longitud, consulta, idioma, radio):** Esta función recibe un lugar en términos de latitud y longitud, una consulta al índice. Devuelve la información relacionada a los sensores que tienen influencia en dicho lugar (determinado por el radio) y sus relaciones a conceptos medioambientales. El idioma y el radio son opcionales. Por defecto son español y 100 km. Esto lo realiza aplicando la fórmula de *Haversine* para calcular las distancias desde el punto provisto y los datos de latitud y longitud que poseen los sensores. Los sensores que no poseen estos datos son eliminados de los resultados finales.
6. **RetornarMapaLugarDataPoints (latitud, longitud, consulta, idioma, radio):** Realiza la misma acción que la función predecesora, sin embargo

utiliza para la búsqueda la función *BuscarFeedsDataPoints*, permitiendo traer las mediciones de los sensores, esto la hace más costosa en términos de tiempo de respuesta.

7. ***RetornarMapaLugarListaSensores (latitud, longitud, ListaSensores, idioma, radio)***: Esta función retorna la misma información que las anteriores, pero a diferencia de ellas recibe como parámetro un listado de objetos o sensores FeedXively en vez de la consulta, los cuales serán comparados con el lugar de búsqueda para retornar los que se encuentran ubicados en dicho lugar.
8. ***ExpandirConsultaConceptosOntologia (Consulta, idioma)***: Esta función recibe una consulta y la expande con conceptos de la ontología. Realiza un modelo de expansión teniendo en cuenta una similitud de cercanía entre conceptos. Añade conceptos hijos y relacionados a los conceptos identificados para hacer la consulta más precisa al dominio de la ontología. Los términos no encontrados los compara con Wordnet para traer sinónimos, finalmente los no encontrados se añaden al final de la consulta.

Los servicios de Geolocalización se desarrollaron basados en el API de Geonames y la información que poseen en sus bases de datos, los servicios desarrollados en el índice son:

1. ***ObtenerLocalizacion(lugar)***: Permite Buscar un lugar geográfico por su nombre en el servicio Web de GeoNames. Retorna una lista objetos que corresponden con el lugar buscado y la información GeonameNode relacionada.
2. ***ObtenerCiudadesConsulta(Consulta)***: Esta función busca en la consulta, cuales conceptos corresponden a un lugar geográfico. Esta función se debe manejar con cuidado ya que términos comunes como aire, que también es del dominio en este caso, es un lugar geográfico con ese nombre.
3. ***ObtenerLugardeCoordenadas(latitud, longitud)***: Este método obtiene el Lugar más cercano a las coordenadas enviadas.
4. ***ObtenerListaLugarCoordenadas(geonodesSerializados)***: Esta función realiza el mismo proceso de ubicar un lugar específico, sin embargo lo hace a cada GeonameNode que ha sido provisto a través de una lista.
5. ***ObtenerListaLugarCoordenadasJerarquica(geonodesSerializados)***: Obtiene las coordenadas de una lista de GeonameNode. Adicionalmente obtiene la jerarquía de lugares de cada nodo, con el fin de establecer con mayor precisión el lugar geográfico encontrado.
6. ***ObtenerListaLugarCoordenadas\_AM1(geonodesSerializados)***: Obtiene las coordenadas de una lista de GeonameNode. Adicionalmente Obtiene de la jerarquía de lugares de cada nodo la primera división política que corresponde a dicho lugar. Esta función en otras palabras permite ubicar una región o (biotipo) para el caso de estudio.
7. ***ObtenerLugardeCoordenadasJerarquia(latitud, longitud)***: Obtiene la

jerarquía de lugares de un lugar.

8. **GeoNames\_Hierarchy(GeonameId)**: Obtiene la jerarquía dado un GeonameNode a través de su identificador.

Los servicios web de los sensores son:

1. **RetornarMetadatosSensor (IdSensor)**: Este servicio permite retornar en formato de objetos FeedXively la información relacionada al sensor particular. El enriquecido con la ontología a través de metadatos y se extraen de la base de datos documental almacenada localmente.
2. **RetornarDatosSensor(IdSensor, [Fecha Inicio], [Fecha Fin])**: Esta función permite obtener información de los datos que mide el sensor entre el rango de fechas definido. En caso de no existir datos retorna vacío.
3. **RetornarJsonSensor(idSensor)**: Esta función permite obtener información de metadatos en formato JSON directamente del servidor Xively.
4. **RetornarDatapointsFeed(string IdSensor, string DatastreamId, string fechaInicio, string fechaFin)**: Esta función retorna los *datapoints* de un canal específico del dispositivo o sensor. Los datos son traídos directamente desde el servidor Xively con respecto al rango de fechas provisto.

Servicios del Caso se estudió de Información medioambiental:

1. **OntenerBiotiposConsulta (ListaSensores)**: Después de realizar una consulta al índice semántico, se puede obtener los biotipos relacionados a dichos resultados. Para esto utiliza los datos de geo posicionamiento de los sensores y la obtención de la división política de primer orden que se encuentra en los servicios de Geonames.
2. **ObtenerInfoBiotipoMedioAmbiente (luga, radio)**: Corresponde a un cuadro elaborado de conceptos medio ambientales (Edatopo, Hidrotopo, Climatopo) y las mediciones que se pueden obtener de los sensores del área de impacto (Biotipo) y la información de las variables definidas en la ontología. El área de influencia por ahora se maneja como el área circular alrededor del lugar y definida en el concepto de área de impacto del biotipo que se identifique.
3. **ObtenerInfoBiotipoContaminacion (luga, radio)**: Igual a la anterior función, sin embargo centra su atención a los conceptos relacionados con la contaminación ambiental.
4. **HistóricosSensor (IdSensor, periodo de tiempo)**: A partir del periodo de tiempo retorna la información de los datos medidos y almacenados en el servidor IoT. Utiliza la función RetornarDatos de los servicios de sensores, pero debe implementar un mecanismo adecuado de presentación de la información y relación con la información medioambiental con el fin de que

sea fácilmente entendible para el usuario.

#### **4.4.6 Ubicación de los Productos**

El producto “Modelo Semantico del Buscador BSIOT v8” contiene el modelo a seguir para la construcción del índice semántico. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 1\Actividad 1\Tarea 4

### **4.5 Fase I – Actividad 2: Tarea 1: Evaluar el middleware / servidor IoT a utilizar (AEMS)**

#### **4.5.1 Introducción**

El presente documento presenta el análisis realizado para la selección del Middleware de datos a utilizar para obtener la información de los sensores que permitirán satisfacer las necesidades de información de las consultas hechas por los usuarios.

#### **4.5.2 Marco de Referencia de Servidores IoT**

En la elaboración del método para crear índices semánticos en la IoT se hizo un estudio de los principales middleware existentes y de los cuales se podían utilizar para obtener información de los sensores que actualmente están conectados a la WoT (Tabla 11).

En la Tabla 10 presenta las características evaluadas y sus convenciones definidas con el fin de poder resumir adecuadamente la tabla comparativa. Las principales clasificaciones que interesaban al proyecto eran los tipos de metadatos que ofrecían de los sensores, formatos y la posibilidad de que se pudiera acceder a ellos de manera gratuita y en lo posible si restricciones, ya que la idea es que la herramienta y el índice se pensaron también crear de uso público.

**Tabla 10: Convenciones de Servidores Mediadores de la IoT**

Abreviatura	Definición de las abreviaturas	Abreviatura	Definición de las abreviaturas
<b>Tipo de Arquitectura implementada</b>		<b>Tipos de Formatos de Estructura de Datos Soportados</b>	
R	REST (Representational State Transfer)	J	JSON (JavaScript Object Notation)
Sk	Socket	X	XML (eXtensible Markup Language)
So	SOAP (Simple Object Access Protocol)	C	CVS (Comma Separated Values)
M	MQTT (Message Queuing Telemetry Transport)	Xd	XDR (eXternal Data Representation)
<b>Tipos de Metadatos soportados</b>		<b>Disponibilidad de la funcionalidad</b>	
G	OGC (Open Geospatial Consortium)	√	Quiere decir que si lo soporta
Em	EML (Environmental Markup Language)	-	Quiere decir que no se encontro información
Tg	Tags (Etiquetas definidas por los usuarios)	<b>Tipos de Usuarios</b>	
Xp	XMPP (Extensible Messaging and Presence Protocol)	T	Todo Tipo de Usuarios
Co	COAP (Constrained Application Protocol)	Ep	Empresariales
Lp	6LoWPAN (IPv6 over Low power Wireless Personal Area Networks)	<b>Tipos de Costos</b>	
<b>Tipos de Redes</b>		O	Open: Abierta sin costo
I	Internet	P	Paga: Se debe pagar por sus servicios
M	Móviles	O/P	Servicios Open y otros son Pagos
St	Satélite		
<b>Formato de sindicación</b>			
At	ATOM (Atom Publishing Protocol)		
Rs	RSS (Really Simple Syndication)		

El resultado del estudio comparativo se puede ver en la Tabla 11. Inicialmente se clasificaron las características en Soporte y desarrollo de aplicaciones, cobertura y servicios que ofrece. Lo anterior permitió seleccionar el servidor Xively (antes se llamaba Cosm y PachuBe) como el más adecuado para el estudio particular por su API de desarrollo abierto y de fácil utilización tipo REST, la gran cantidad de objetos conectados, los diferentes sormatos en los que ofrece su funcionalidad y la posibilidad de crear aplicaciones sin costos.

**Tabla 11: Comparación de Características de Servidores Mediadores de la IoT**

CARACTERISTICAS ANALIZADAS																				
PLATAFORMAS	Soporte Desarrollo Aplicaciones					Cobertura			Servicios											
	Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	Uso de Estándares de metadatos	Uso de formatos de Estructura de Datos	Arquitecturas que Soporta IoT	API de desarrollo para diferentes Lenguajes	Ciclo de vida y Despliegue de Productos	Redes que da soporte	Cantidad de Objetos Conectados*	Cantidad de Empresas Conectadas	Gestión de Metadatos	Gestión Datos en Tiempo Real	Directorio de Objetos - Búsqueda	Servicios de Negocio	Servicios de Personalización y Aplicaciones	Seguridad, Privacidad y Autorización	Usuarios en los que se focaliza	Costos	Escalabilidad	Extensa Documentación	Formato de sindicación	Trigger
AMEE	G,Tg	J	R	√	√	I	342	0,26	√	√			√	√	Ep					
Arkessa				√		I,M,St			√	√	√	√	√	√	T	O/P				
Axeda			R,S	√	√	I,M,St	1.2	150	√	√	√	√	√	√	T	P				
Bugswarm			R	√		I		300				√	√		T					
Carriots cloud		J,X	R	√		I				√				√	T,Ep	O/P		√		√

Eart Cam						I,M						√		√	E	P				
Evrythng		J	R	√		I				√		√		√	T,E p	O		√		
GroveStreams		J	R	√		I,M				√		√	√	√	T,E p	O/ P		√	Rs	
Sensinode	Co,Lo				√	I,M,St				√		√	√	√	Ep	P				
Nimbits	Xp	J,X	R	√		I				√	√				T	O	√	√		√
Exosite		J		√	√	I,M,St				√	√	√	√	√	T,E p	O/ P	√	√		
Open.Sen.se		J	R	√		I				√	√				T	O		√		
Paraimpu		J	R			I									T	O				
NewAer					√	I,M,St							√		Ep	P				
SensorCloud		C,X d	R	√		I,M,St				√	√		√	√	Ep	P		√		
SensorPedia	G	J	R	√		I,M,St				√	√	√		√	Ep	O	√	√	At	
Sensormap	Tg			√		I,M,St				√	√	√		√	T		√	√		
SenseWeb	Tg			√		I,M,St				√	√		√	√	T		√	√		
SensorBase	Em			√		I,M,St				√	√		√	√	T		√	√	Rs	
ThingSpeak		J		√		I	0.01			√	√	√		√	T,E p	O/ P		√	Rs	
Thingworx				√	√	I,M,St				√	√	√	√	√	Ep	P	√			
Xively Cloud Services***	G,Tg	J,X, C	R,S k,M	√	√	I	250	55		√	√	√	√	√	T	O	√	√	Rs	√
Yaler		J		√		I				√				√		O			Rs	

\* Millones de dispositivos.

A parte de la clasificación de las características realizadas en la Tabla 11 para soporte al desarrollo, cobertura y servicios, adicionalmente, se agruparon los diferentes portales web de objetos de acuerdo a sus características interrelacionadas, con el fin de poder tipificar los portales relacionados a sus funcionalidades. En la Tabla 12 se presenta esta clasificación:

Tabla 12: Tipos de Servidores Mediadores de la IoT

Nº de característica	Tipo de Servidor
2, 9, 17	<b>Orientado a Gestión de Datos:</b> Apoya el fácil acceso a las observaciones de sensor, el reusó de los datos y el peso ligero de los archivos.
1, 2, 4, 5, 9, 10, 11, 15, 16, 18	<b>Orientados al Desarrollo:</b> Apoyan el suministro de datos por medio de mecanismos sencillos, desarrollo rápido de aplicaciones para el usuario y bajo costo de implementaciones en las plataformas abiertas.
1, 3, 11, 14, 19, 20	<b>Orientados a Compartir Datos:</b> Apoya las mejores prácticas y las soluciones comprobadas provenientes de la web 2.0 para vinculación y control del contenido.
6, 7, 8, 12, 13	<b>Orientados a Usuarios:</b> Apoyo al desarrollo automático de productos que beneficie el reusó y aplicación de los datos.

Aunque los servidores o portales de objetos pueden cumplir varias funcionalidades, es posible establecer cómo algunos privilegian un tipo de servicio particular. Este estudio permitió definir parte los indicadores por los cuales se seleccionó el servidor IoT para el caso de estudio, lo cual se realizó en la siguiente sección, ubicada en el **ANEXO C-2** en el cual se presenta el proceso de evaluación estableciendo los indicadores de selección y para escoger el middleware más adecuado para el desarrollo de un índice semántico y un buscador semántico.

#### **4.5.3 Ubicación de los Productos**

El producto “Evaluación de Middleware y API de Desarrollo” contiene un análisis de los portales del IoT presentando un proceso de evaluación para seleccionar el más adecuado a las especificaciones buscadas. Se encuentra en la siguiente ruta del CD de entrega:

C: \CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 1\Actividad 2\Tarea 1

#### **4.6 Fase I – Actividad 2: Tarea 2: Alineación de Conceptos del Servidor IoT y Fuentes Externas**

En este apartado se analizan los formatos provistos por el servidor de objetos Xively que fue seleccionado en la tarea anterior, del cual se obtiene conceptos que serán equiparados en la ontología desarrollada. El portal Xively tiene un modelo de datos encargado de describir el dominio de conocimiento con el que representara la información. Este formato se procede a actualizar la ontología Objeto Semántico ya sea encontrando las similitudes con los conceptos existentes o ampliando los nuevos conceptos.

##### **4.6.1 Ubicación de los Productos**

El producto “Identificación de Categorías de Conceptos en los Formatos Estructurados”. Se encuentra en la siguiente ruta del CD de entrega:

C: \CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 1\Actividad 2\Tarea 2

#### **4.7 Fase II – Actividad 1: Tarea 1: Seleccionar una metodología de desarrollo adecuada**

##### **4.7.1 Introducción**

El presente documento establece la metodología que se sigue para la implementación de la ontología para el buscador de contaminación

medioambiental. Aunque en la fase del modelo semántico ya se han definido las ontologías conceptualmente, en esta fase se procede con la construcción – reutilización de las mismas.

#### **4.7.2 Implicaciones y Relaciones con el Modelo Semántico Propuesto**

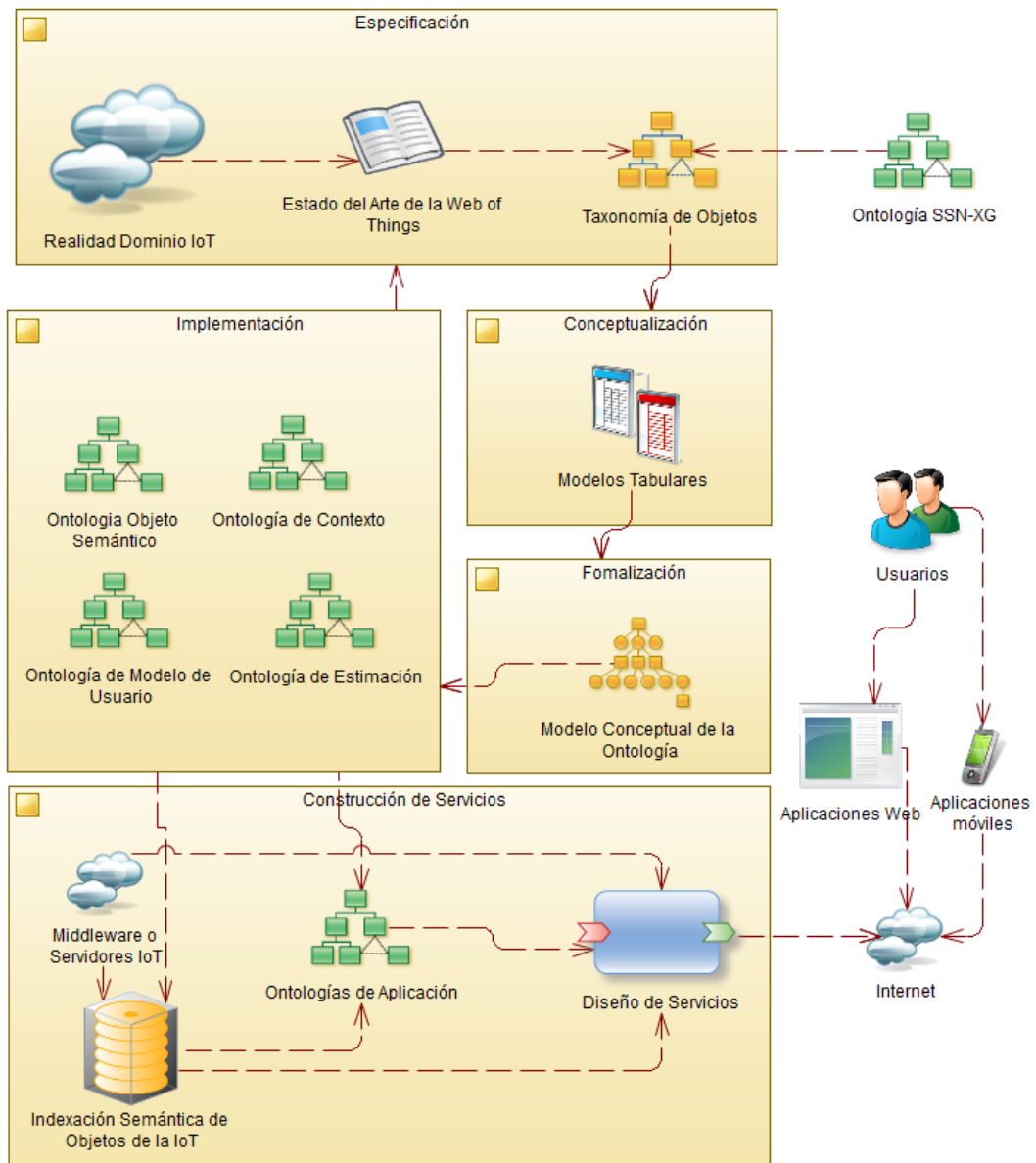
Con el fin de establecer el proceso de desarrollo de las ontologías, en la Figura 14 se presenta el ciclo de construcción basado en la fase de desarrollo de la metodología METHONTOLOGY (Corcho, Fernández-López et al. 2005). Adicionalmente se presentan los componentes principales de las ontologías a construir y su posterior utilización en una creación de servicios web.

El método inicia con una especificación. Para esto, además de definir los objetivos de la ontología se toma como principal insumo el estado del arte desarrollado para tal fin, reutilizando las taxonomías presentadas como la de (Mathew, Atif et al. 2011) y los estándares ontológicos existentes como SSN-XG (Compton, Barnaghi et al. 2012), las cuales serán aportadas a la fase de conceptualización con el fin de identificar su reutilización. Estos primeros pasos fueron realizados por los expertos en conocimiento en la definición del modelo semántico para el buscador propuesto.

Para la conceptualización se realizarán las tareas y formatos tabulares propuestas por METHONTOLOGY. En la formalización se utilizará una herramienta como CMAPS para representar semi-formalmente la especificación de las ontologías. En la construcción se trabaja con la herramienta Protégé (Noy, Sintek et al. 2001) y en formato OWL-DL se almacenaran las ontologías con el fin de su utilización y evaluación.

Finalmente, se plantea una etapa de construcción de servicios, en la cual se expondrán los servicios de la ontología a través de servicios web, para ser consumidos por aplicaciones y usuarios de Internet. Los servicios fundamentan su fuente de información en un índice semántico construido a partir de la interacción con los middleware y las ontologías construidas. Así mismo, los middleware, el mismo índice y las ontologías de aplicación soportan los servicios, como fuentes directas para hacer más ágil el funcionamiento de los mismos.

**Figura 14. Diagrama Metodológico y de Componentes de la Ontología de Interacción Semántica WoT**



Las ontologías de aplicación aparecen a partir de la personalización de las ontologías desarrolladas para la interacción semántica, estas ontologías reutilizan la información de las ontologías de dominio y definen tareas específicas con los elementos reales de un caso de estudio particular.

#### 4.7.3 Principios de diseño de la ontología

Es importante definir unos principios de diseño de la ontología con el fin de mantener una coherencia con los objetivos de la misma. Ya que la ontología va de la mano con el conocimiento de la IoT y la WoT, es necesario tener presente que se necesitan estructuras que faciliten su aprovechamiento, equilibrando

adecuadamente el impacto que produzcan estas infraestructuras con los requerimientos propios de los objetos de la IoT. Así, el trabajo de (Wei, De et al. 2012) en el cual proponen una ontología para esta área, definen unos principios de diseño, los cuales compartimos, adecuamos y ampliamos de la siguiente manera:

- **Ligera:** un modelo ontológico ligero que equilibre adecuadamente la complejidad en la expresividad y la inferencia, tiene más posibilidades de ser utilizado y adaptado.
- **Complejidad:** la posibilidad de reutilizar los modelos anteriores y hacerlos en lo posible completos con respecto del dominio de la IoT garantizarán también su importancia.
- **Compatibilidad:** la ontología debe ser coherente con las ontologías existentes bien diseñadas, así se pueden alinear con las existentes fácilmente.
- **Modularidad:** la ontología diseñada se desarrolla con un enfoque altamente modular para facilitar su evolución, la extensión y la integración con ontologías externas.
- **Abstracción:** La ontología debe permitir diferentes niveles de abstracción con el fin de ofrecer vistas adecuadas del conocimiento de los objetos de la IoT.
- **Adaptable:** La ontología debe proveer un mecanismo de personalización, de tal forma que permita a sus usuarios adecuarla, ampliarla y reutilizarla como mejor les parezca.

El enfoque para la construcción de la ontología que permita interacción semántica debe seleccionar los elementos importantes de los estándares actuales para la descripción y manejo de los sensores como las especificaciones XML provista por la Open Geospatial Consortium (OGC) (Botts, Percivall et al. 2008) y las ontologías de sensores como SSN-XG (Compton, Barnaghi et al. 2012). Además del aprovechamiento de propuestas ontológicas como: clasificación de los objetos propuesto por (Mathew, Atif et al. 2011), manejo de contexto social de (Biamino 2012), modelo de servicios de (Kostelník, Sarnovský et al. 2011) y de representación de conocimiento IoT propuesto por (Wei, De et al. 2012).

El reto de alinear estas ontologías y seleccionar los elementos que más enriquecen el modelo aporta a la complejidad, pero puede ir en detrimento de una propuesta ligera, por tal motivo se utilizará el principio de abstracción y modularidad con el fin de hacer más fácil de implementar y describir la ontología resultante.

#### 4.7.4 Metodología de Diseño de Ontologías

Para el diseño de las ontologías se ha decidido realizar una adaptación de la metodología METHONTOLOGY (Corcho, Fernández-López et al. 2005), la cual originalmente propone las siguientes fases:

1. **Especificación:** ¿Cuál es su Nombre?, ¿Por qué se construye?, ¿Cuál es su uso?, ¿Quiénes son sus usuarios?, ¿Cuál es el tipo de ontología?
2. **Conceptualización:** Convertir la percepción en representaciones semiformales y tabulares, del cual se puede obtener el modelo conceptual de la ontología.
3. **Formalización:** Transforma el modelo conceptual en uno formal o semicomputacional.
4. **Implementación:** Construye el modelo formal de ontologías en formatos estándar.
5. **Mantenimiento:** Actualización y corrección de la Ontología.

Se ha realizado una versión resumida y adecuada de dicha metodología con el fin de permitir construir las ontologías en versiones rápidas y evolutivas. Así se soporta la metodología en herramientas que permiten conceptualizar y formalizar automáticamente las mismas. Los pasos adecuados son:

1. **Abstracción Formal:** Se corresponde a especificación, conceptualización y formalización
  - a. Se responden las preguntas de la especificación de la ontología.
  - b. Con el fin de tener una idea clara del conocimiento a modelar se puede hacer una descripción textual o gráfica en compañía de los expertos o de una fuente de conocimiento válida. Se diligencian la plantilla tabular de glosario de términos propuesta en METONTOLOGY.
  - c. Se utiliza la herramienta CmapTools COE<sup>9</sup>, las cuales permiten general el modelo conceptual gráficamente y siguiendo la especificación de patrones de diseño de COE con el fin de facilitar el proceso de implementación. Esta especificación evita tener que definir todas las plantillas de METHONTOLOGY.

---

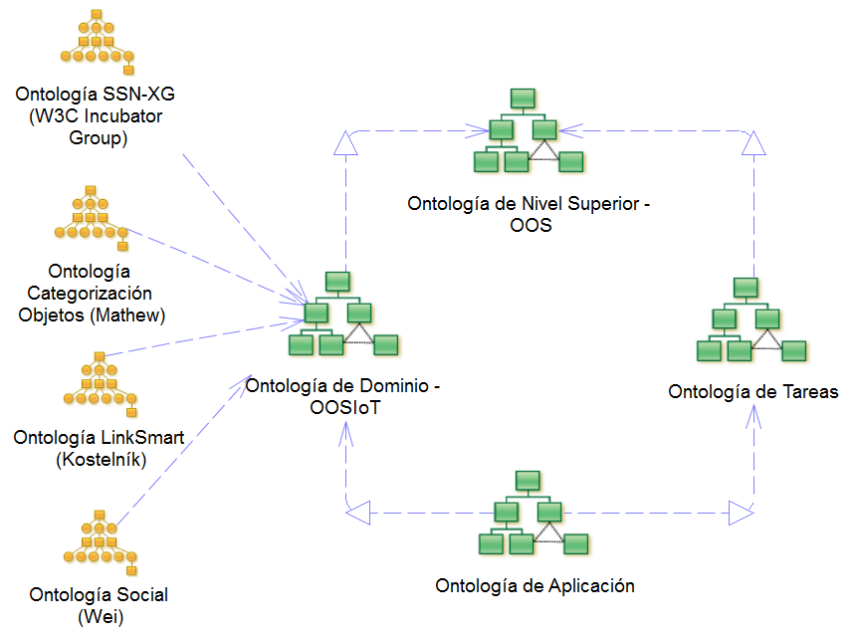
<sup>9</sup> <http://www.ihmc.us/groups/coe/>, consultado 05/07/2013. Es un conjunto de herramientas software que permiten construir, compartir y visualizar ontologías basados en CmapTools. Estas herramientas son desarrolladas por Florida Institute for Human & Machine Cognition – IHMC y bajo el proyecto Concept-map Ontology Environment - COE

## 2. Implementación:

- a. Exportar la Ontología en formato OWL a través de la utilidad de exportación de la herramienta CmapTools.
- b. Abrir la ontología en la herramienta Protégé <sup>10</sup>, con la cual se revisa su correspondencia con la conceptualización y en caso de necesitar ajustes adicionales se realiza en esta etapa.
- c. Verificar el funcionamiento de la ontología teniendo en cuenta las preguntas que se espera que conteste, en lo posible utilizando el razonador integrado en Protégé.

Teniendo en cuenta que el proyecto debe reutilizar las ontologías existentes en el tema de la Web de las cosas se decidió crear y adaptar ontologías de nivel superior, que posteriormente se reutilicen en ontologías de dominio específico y de tarea. Así será posible desarrollar ontologías de aplicación para el desarrollo de servicios en la WoT Figura 15.

Figura 15: Modelo de Dependencia de Ontologías



<sup>10</sup> <http://protege.stanford.edu/>, consultado 05/07/2013. Es un editor de ontologías de código abierto y gratuito que permite crear y editar ontologías en diversos formatos estándar como RDF(S), OWL y XML entre otros. Es modular y permite incorporar herramientas de visualización y razonamiento.

Para este trabajo se propone una ontología de nivel superior que modela cualquier objeto denominada ontología objeto semántico – OOS. La cual define los elementos primordiales para que exista un objeto capaz de interactuar en cualquier contexto. Posteriormente, se alinearán o adaptarán las ontologías de la IoT relacionadas al presente trabajo (Kostelník, Sarnovský et al. 2011; Mathew, Atif et al. 2011; Biamino 2012; Compton, Barnaghi et al. 2012; Wei, De et al. 2012), creando una ontología de dominio llamada ontología de objeto semántico IoT – OOSIoT.

En el caso de estudio se crea una ontología de aplicación que adapta la OOSIoT a los requerimientos específicos y de tareas del dominio particular. Cada caso de estudio debe crear su propia ontología de aplicación para que pueda recopilar el conocimiento necesario para crear servicios adecuados.

En fase de reutilización de ontologías se aplicará el método propuesto por (Kavouras and Kokla 2000; Kavouras 2005) en el cual presentan la factorización semántica y los látices de conceptos como elementos para integrar ontologías y tres procesos de integración de las mismas.

#### **4.7.5 Ubicación de los Productos**

El producto “Modelo Semántico del Buscador BSIOT v8” metodología que se sigue para la implementación de la ontología para el buscador de contaminación medioambiental. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 2\Actividad 1\Tarea 1

#### **4.8 Fase II – Actividad 1: Tarea 2: Crear la estructura de conocimiento**

En esta fase se llevó a cabo el plan de desarrollo de la estructura del conocimiento utilizando la herramienta PROTEGE, se construyó una ontología de Contaminación Ambiental alineando el modelo semántico con ontologías estandarizadas existentes como la (ENVO, Objeto Semántico, SSW-XG, AWS y CF-Features).

##### **4.8.1 Ontología de Contaminación Ambiental**

El producto de esta tarea es un documento OWL, primero se creó un modelo conceptual que fue realizado con la herramienta CMAP y luego fue adaptado con la herramienta PROTEGE.

#### **4.8.2 Ubicación de los Productos**

El producto “ontologiaambiental.owl” que contiene la ontología de Contaminación Ambiental. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 2\Actividad 1\Tarea 2

#### **4.9 Fase II – Actividad 2: Tarea 1: Crear los servicios del API de la Fuente de datos**

En este apartado se desarrollaron los servicios web encargados de utilizar el portal de la IoT como fuente de datos, el portal seleccionado fue Xively, que sirve como repositorio de información de dispositivos que en tiempo real están subiendo información, la interfaz que dispone Xively, compartiendo servicios web de transferencia de estado representacional (Representational State Transfer – REST) los cuales puede ser invocados utilizando el protocolo HTTP por medio de línea de comandos. Los servicios REST para ser invocados a través de un servicio web REST cliente requiere conocer los métodos que implementaron y ya que no poseen metadatos WSDL no se puede crear un cliente REST que consuma el servicio, pero REST ha sido pensado para poder ser consumido también por peticiones utilizando el protocolo HTTP los cuales fueron encapsuladas en métodos por el buscador semántico que implementa el índice y así lograr la comunicación con la interfaz REST de Xively. En el **ANEXO C-3** se presentan la definición de las funciones encargadas de cumplir las dos fases funcionales Primera: Obtención de los documentos del portal web IoT Xively y crear una base documental. Segundo: Indexar los conceptos de dicha base documental.

##### **4.9.1 Ubicación de los Productos**

El producto “clases Xively.PNG” y “97mplementación servicios Xively.doc” que contiene que contiene el modelo de clase desarrollas para xively y la declaración de los métodos. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemanticoIoT\BuscadorSemanticoIoT\Modelo Semántico Indice IoT\Fase 2\Actividad 2\Tarea 1

#### **4.10 Fase II – Actividad 2: Tarea 2: Realimentación de la funcionalidad**

En los servicios desarrollados y por motivos de alcance del proyecto se desarrollaron pruebas internas de funcionamiento, las cuales permitieron realimentar el método y los programas, por ello en la documentación entregada se encuentran las diferentes versiones por cada realimentación hecha.

#### **4.11 Fase III – Actividad 1: Tarea 1: Extraer Información de Objetos**

La implementación del índice semántico se realizaron dos maneras de indexación, una de ellas es crear extrayendo los documentos JSON directamente desde Xively lo que aumenta en gran medida el tiempo de creación del índice, el otro método es generar el índice a partir de la base de datos documental que previamente fue extraída desde el servicio Xively en una indexación previa, el tiempo de creación disminuye drásticamente.

##### **4.11.1 Ubicación de los Productos**

El producto “clases Xively.PNG” y “implementación servicios Xively.doc” que contiene el modelo de clase desarrollada para Xively y la declaración de los métodos. Se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\Modelo Semántico Indice I\T\Fase 2\Actividad 2\Tarea 1

#### **4.12 Fase III – Actividad 1: Tarea 2, 3 y 4: Análisis Léxico, Eliminación de Palabras vacías y Aplicación de Lematización**

Se implementó un módulo, el cual usando los servicios web del API Xively, hace un escaneo sistemático de toda la base de datos de Xively, extrayendo la información de cada dispositivo que coincide con los conceptos almacenados en la ontología desarrollada. Con esto se crea la base de datos documental - BDD, la cual es un directorio que está compuesto por documentos en formato JSON con el nombre del FEED que es el identificador único que tiene Xively a cada documento. En el proceso de extracción se van anotando los JSON con respecto a los conceptos encontrados en su contenido.

Las Tarea 2, Tarea 3 y Tarea 4 corresponde directamente a la segunda fase de funcionalidad que es procesamiento de los documentos que fueron recopilados de Xively, es decir se procesan los documentos para crear el índice semántico.

Una vez ya se tienen los documentos en la base documental se procede a hacer el análisis léxico. El formato en el que se obtuvo los documentos es con JSON, como este ya es un formato estructurado entonces se va extrayendo la información estructurada y se analiza. Se toma el documento asociado a un FEED y se extrae la información del documento analizando los documentos. Los procesos principales son realizados a través de la herramienta LUCENE, la cual al momento de realizar la indexación del documento particular realiza el análisis léxico, la eliminación de palabras vacías y aplicación de Lematización, dependiendo del analizador relacionado (español, inglés).

Una vez recopilados los documentos y analizados el documento ahora si se pueden indexar. Este apartado requiere proveer una clase de realice la funcionalidad en español la cual se creó directamente de cero. Este paso constituye un aporte adicional en la aplicación ya que se implementó no solamente el analizador en inglés existente en Lucene, sino también el analizador en español, con el objetivo de adecuar la herramienta a lenguajes natural castellano para ampliar el espectro de usuarios que puedan utilizar la misma.

#### **4.12.1 Ubicación de los Productos**

La base de datos documental se encuentra en la siguiente ruta del CD de entrega:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\BuscadorSemantico\WSSemanticSearch\App\_Data\Json\_Data.

Adicionalmente, se creo un documento de nombre: “Preprocesamiento de la Base de datos Documental.doc” en el cual se puede precisar al método de extracción y anotación desarrollado:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\Modelo Semántico Indice lo\T\Fase 3\Actividad 1

Se pueden consultar también las líneas de código encargadas de procesar los documentos se encuentra con el nombre de “Colocar el codigo que extrae los documentos.txt” que contiene el contenido del archivo de nombre “WSSemanticSearch.asmx.cs” de Visual Studio ubicado en:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\Modelo Semántico Indice lo\T\Fase 3\Actividad 1\Tarea 1

### **4.13 Fase III – Actividad 2 – Tareas 1, 2 y 3 – Selección de Términos, Obtención de los Conceptos, Desambiguar Conceptos**

#### **4.13.1 Introducción**

En esta actividad se hace una selección de términos candidatos, es decir, los términos relevantes en los documentos y que son candidatos para extraer sus conceptos. Estos conceptos aparecerán a partir de un proceso de consulta en las estructuras de información diseñadas para almacenar conocimiento en la primera fase, como los tesauros y las ontologías u otras aproximaciones de la teoría cognitiva

### **4.13.1 Extracción de Conceptos**

#### **4.13.1.1 Selección de Términos**

Dependiendo de los objetivos del índice se deben realizar algunas funcionalidades, en este apartado se hace algo diferente normalmente la indexación se realiza tomando los documentos anotarlo con los términos, sino que toman los documentos y se anotan con los conceptos relacionados a la ontología de contaminación ambiental y por lo tanto fue recuperado desde el servidor. El resultado es una lista de conceptos relacionados al JSON o al sensor recopilado.

Por una decisión de diseño no se aplicaron Frases Nominales pero si se anota el concepto con frases nominales, es decir no se tomó cada concepto del documento y se busca en la ontología sino lo que se realizo fue tomar el concepto de la ontología y se estableció que documento lo contiene y por último se anota.

#### **4.13.1.2 Obtención de los Conceptos**

Con la lista de conceptos relacionados a cada documento JSON se procede a realizar indexación con la librería de LUCENE, esta herramienta utiliza el modelo de Espacio Vectorial. En el directorio de productos se muestra el método que implementa la indexación de los documentos. El producto final es una lista de conceptos que fueron anotados en el JSON en la etapa anterior.

#### **4.13.1.3 Desambiguar Conceptos**

De acuerdo al método de anotación elegido no hay necesidad de desambiguar los conceptos ya que la lista de conceptos anotados son tomados directamente de la ontología en un dominio particular y específico y por otro lado la librería de Lucene se encarga de hacer un proceso similar al momento de indexar el documento con los términos de los campos abiertos.

### **4.14 Fase III – Actividad 3 – Tareas 1, 2 y 3 – Análisis de Coocurrencia y Análisis de Representatividad**

#### **4.14.1 Análisis de coocurrencia**

Este proceso se decidió utilizar la similitud semántica que ofrece el modelo vectorial, con el fin de reutilizar la librería Lucene, la cual ya internamente desarrolla dicho modelo con el método *MoreLikeThis*. Sin embargo, aquí se puede utilizar cualquier otro modelo o algoritmo personalizado y en este caso los desarrolladores debería utilizar otra librería o crear sus propios algoritmos.

#### **4.14.1.1 Análisis de Representatividad**

El análisis de representatividad se realizó con el mismo método de modelo vectorial, Sin embargo, el hecho de anotar los documentos con listas de términos, hace que este modelo pondere más alto los documentos que más conceptos tengan relacionados en la ontología. Los resultados son muy precisos para búsquedas que se relacionan directamente con el dominio particular, pero sacrifica las búsquedas que no encuentran coincidencias con los conceptos de la ontología, por esta razón a veces los lugares geográficos salen varias posiciones después en la salida del buscador. En el **ANEXO C-4** se presenta el código fuente necesario para realizar este apartado.

#### **4.14.2 Ubicación de los Productos**

Esta es la ubicación donde se encuentra el producto correspondiente a toda la actividad 2 en el CD de entrega con el nombre de “Extracción de conceptos.doc”:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\Modelo Semántico Indice lo\T\Fase 3\Actividad 2

#### **4.15 Fase III – Actividad 4: Tarea 1: Implementación del Índice Semántico**

Ya que la creación del índice se realizó con la librería de LUCENE, estos módulos se encuentran en la implementación interna de la misma.

#### **4.15.1 Ubicación de los Productos**

LUCENE utiliza arboles invertidos y los archivos del índice se pueden encontrar en la ruta del CD:

C:\CD del Proyecto\IndiceSemantico\T\BuscadorSemantico\T\BuscadorSemantico\WSSemanticSearch\SemanticIndex

#### **4.16 Fase III – Actividad 4: Tarea 2: Implementación de Servicios Web Semántico de Objetos**

Para el caso particular se creó un conjunto de servicios Web que exponen toda la funcionalidad del índice semántico y teniendo en cuenta el dominio particular y el diseño realizado del modelo semántico de la fase 1 del desarrollo del índice. En el **ANEXO C-5** se puede encontrar el código fuente de este apartado.

#### **4.17 Fase III – Actividad 4: Tarea 3: Implementación de Servicios de Índice**

La implementación de los servicios del índice se encuentra en el **ANEXO C-6**, este anexo sirve como guía para comprender la disposición de los archivos en Visual Studio.

#### **4.18 Fase III – Actividad 4: Tarea 4: Evaluación del índice construido**

La evaluación del índice construido se realizó utilizando las medidas actuales de recuperación de la información, ya que lo que se deseaba en el momento era establecer su capacidad para extraer información relevante del WoT. En el siguiente capítulo de la monografía se puede evidenciar este proceso.

Un elemento importante a mencionar es que cuando se hizo el sondeo de la información específica a recuperar en el servidor Xively, nos encontramos que no había suficientes sensores relacionados a dicho dominio sobre todo en Colombia, por tal razón se procedió a crear un conjunto de sensores virtuales en dos regiones de Colombia (Cauca y Santander), con el fin de poder tener datos más relacionados en nuestro entorno particular.

#### **4.19 Fase IV – Actividad 1: Tarea 1, 2: Elaboración del Plan de Restitución y Ejecutar el despliegue**

En esta última fase se realizó el despliegue en un Servidor Windows Server 2008, con base de datos SQL Server 2008 y Microsoft Framework 4, en un servidor web IIS 7.5 alojado en la siguiente dirección: <http://semanticsearchiot.net/sswot/SearchWoT/>.

#### **4.20 Conclusiones del Capítulo**

- La instanciación del método permitió desarrollar un índice semántico para la IoT en el campo del dominio de contaminación mediambiental. Sin embargo, se tuvo que tomar decisiones de diseño en ciertos pasos con el fin de solventar adecuadamente el índice, entre ellos, la construcción de una ontología en el campo de contaminación ambiental, lo cual llevo un tiempo significativo. El método utilizado para crear la ontología basado en herramientas CASE existentes, disminuyo el tiempo de desarrollo de la misma y permitió hacer una buena aproximación.
- La selección del servidor IoT fue otro paso que demando bastante trabajo, ya que se tuvo que estudiar cada uno y revisar completamente sus documentaciones, crear clasificaciones e indicadores de selección. Sin embargo, este trabajo puede facilitar la selección del mismo para posteriores instancias del método.

- La documentación desarrollada pasa a paso y en carpetas instanciando el método propuesto del índice, permitió organizar adecuadamente la construcción del mismo y realimentar el método a la vez que se iban tomando las decisiones de diseño. Para el caso particular esto llevo a eliminar dos actividades y a unir otras dos de las propuestas inicialmente.
- Aunque el método propuesto en la fase III, se definen tareas específicas de indexación semántica (análisis léxico, eliminación de palabras vacías, Lematización, indexación), para el caso de estudio, la decisión de reutilizar API's de desarrollo como Lucene.Net y modelos estandarizados como en vectorial, permitió resolver éstos procesos de construcción con dichas herramientas, facilitando un desarrollo rápido y eficiente. Para un estudio que quiera aplicar sus propios métodos y se enfoque más en ello, sería necesario construir sus propios módulos aumentando el proceso de construcción.
- La decisión de realizar una expansión de consulta y un proceso de anotación semántica de listas de conceptos en los documentos de los sensores, permitió que el modelo vectorial arrojara resultados bastante precisos con sensores que más conceptos tienen del dominio de la ontología aplicada al índice. Sin embargo, castiga la ubicación del lugar ya que éste queda al final de la consulta, lo cual se podría mejorar haciendo un ranking posterior por sitio, esto se solucionó en parte con la detección de Biotipos.
- El proceso de detección de Biotipos fue apoyado con el API de Geonames. Un aporte interesante fue como se pudo determinar el mismo a partir de las ubicaciones de latitud y longitud de los sensores existentes en los resultados de la consulta del usuario.
- La posibilidad de cambiar la ontología de dominio y generar un índice semántico que expone su funcionalidad mediante servicios web, permite que se puedan reutilizar los mismos sensores en diferentes ámbitos, conduciendo a una reutilización de su información por cualquier aplicación y objetivos.
- Finalmente, aunque el uso de las herramientas CASE facilitó varios procesos de implementación, la curva de aprendizaje es alta para utilizarlas adecuadamente y es un factor que se debe tener en cuenta al momento de crear un índice con respecto a los tiempos de desarrollo.

## VALIDACIÓN DEL PROTOTIPO

La relevancia es un concepto subjetivo, en relación que es una necesidad humana, por ello las personas tienen una percepción diferente acerca de la relevancia de un documento (Barragán 2008).

Para la validación del prototipo se propusieron la medida de Precisión-Recuerdo (Manning, Raghavan et al. 2007) e Índice MAP (Croft, Metzler et al. 2009; Yepes 2009) en una segunda instancia se aplican las medidas de Precisión at K, de nuevo el índice MAP y por último las estadísticas Kappa (Manning, Raghavan et al. 2007) Cohen y Fleiss. Sin embargo, la evaluación de los buscadores Web y en otros dominios diferentes a la WoT generalmente utilizan *datasets de prueba*, los cuales permiten determinar la relevancia (precisión – recuerdo) para cada documento de acuerdo a una consulta específica que ha sido previamente evaluada por uno o más expertos en el dominio; en este caso, en el dominio del WoT no se han encontrado evaluaciones (*datasets*) y menos de las del dominio específico de dispositivos de contaminación del aire en la IoT, por este motivo no se pudo implementar la medida Precisión-Recuerdo debido a que era costoso disponer de los expertos que determinarían que documentos del dominio eran relevantes, además que por ser una tarea que conlleva mucho tiempo en tener resultados fiables saliéndose del alcance en tiempo del proyecto.

### 5.1 Modelo de Evaluación de SIMIOT

Se planteó un modelo de evaluación en donde se tienen en cuenta unos pasos encaminados para realizar una evaluación satisfactoria de la herramienta SIMIOT, en el **ANEXO D-1** se amplía el proceso que se tuvo en cuenta, como el escenario de ambientación, escenario de motivación y el escenario de aplicación, clasificación y planteamiento de las consultas según el dominio de aplicación como lo es la Contaminación del Aire, por último se relacionan los datos recopilados de los usuarios y las métricas más usadas para la evaluación de buscadores en la web.

### 5.2 Desarrollo de Dispositivos Virtuales para las Pruebas

Dado que en la exploración hecha al servidor Xively reportaba que Colombia tenía muy pocos sensores de los cuales se pudiera recuperar información de contaminación medioambiental, fue necesario crear una batería de sensores virtuales, desplegados en dos regiones de Colombia (Cauca y Santander). Estos sensores se desarrollaron teniendo en cuenta la normatividad actual de Colombia

con respecto a la contaminación y calidad del aire y los dispositivos reales, así como sus mediciones y rangos, de tal forma que los mismos se comportaran como los dispositivos físicos reales, asegurando que los datos obtenidos se acerquen en lo posible a la realidad. Para una mayor información sobre el diseño y despliegue de los mismos refiérase al **ANEXO D-4.4**.

### 5.3 Curva de Precisión-Recuerdo (Precision-Recall):

Permite evaluar la eficacia de los resultados entra varias consultas. Es una relación entre la cantidad de documentos recuperados y la cantidad de documentos relevantes para una consulta ver Ecuación 4 y Ecuación 5 y en la Tabla 13 se plantean las convenciones.

Ecuación 4. Formula para la Precisión

$$Precisión = \frac{\# \text{ documentos relevantes}}{\# \text{ total documentos recuperados}}$$

Ecuación 5. Formula para el Recuerdo

$$Recuerdo = \frac{\# \text{ documentos relevantes}}{\# \text{ total documento relevantes}}$$

Tabla 13: Explicación de medidas de Precisión - Recuerdo

Convención	Descripción
# documentos relevantes	Son los documentos que fueron catalogados como relevantes y están asociados a una consulta.
# total de documentos recuperados	Son todos los documentos que se recuperaron de la consulta.
# total documentos relevantes	Son los documentos relevantes que se deberían recuperar.

- **Precisión Promedio (Average-Precision):** Promedio entre los valores de la Precisión para varias consultas asociadas.

- **Indice MAP (Mean Average -Precisión):** Promedio entre valores de precisión media de varias consultas. Los valores del índice MAP varían entre 0.1 y 0.7 esto significa un alto grado de precisión general asociada a las consultas realizadas.

#### 5.4 Pruebas Alfa

Es necesario precisar, que son aquellas que se realizan en un entorno controlado y son guiadas por los desarrolladores de la aplicación que se pretende evaluar (Jenkins 2008). En este punto es importante medir la relevancia de la información recuperada por el buscador IoT implementado, para abordar esta cuestión, se han definido ciertas métricas encaminadas a determinar la relevancia de un sistema de recuperación de información, entre ellas están la precisión y la precisión at k resultados (Manning, Raghavan et al.) y estadísticas Kappa.

En Kappa Cohen se tienen en cuenta dos usuarios y en Kappa Fleiss se tienen en cuenta (m) usuarios.

**Ecuación 6. Kappa Cohen**

$$Kappa_c = \frac{P_a - P_e}{1 - P_e}$$

Alternativa para Kappa Cohen:

**Ecuación 7. Otra forma de Kappa Cohen**

$$Kappa_c = \frac{n_a - n_e}{1 - n_e}$$

**Ecuación 8. Kappa Fleiss**

$$Kappa_F = \frac{P_a - P_e}{1 - P_e}$$

En esta ronda de medidas tienen un enfoque a usuarios y fueron realizadas en un grupo de estudiantes de Ingeniería Ambiental de sexto semestre de la Universidad Autónoma de Popayán, se tomó este grupo de personas debido a que el dominio en el que el buscador fue desarrollado es la contaminación ambiental, lo que hace a estos usuarios sean los más idóneos para las pruebas.

A continuación se presentan las condiciones generales de la prueba:

- **Condiciones de la recopilación de datos:** En la Tabla 14 se muestran las 4 consultas que fueron seleccionadas teniendo en cuenta el dominio de la Contaminación Ambiental y el modelo propuesto de lo que debería poder contestar mismo. Participaron 13 jueces que emitieron juicios de relevancia sobre cada uno de los documentos encontrados por el buscador y cuya muestra se determinó que fuera limitada a 25 documentos.

Tabla 14: Preguntas del dominio Contaminación

Consulta	Descripción
1	Calidad del aire en Santander
2	Agentes contaminantes en el Cauca
3	Monóxido de carbono en Santander
4	Contaminación en new york

- **Interpretación de la recopilación de datos:** En la recopilación de datos se identificaron 3 clases de resultados (1=Relevante), (0=NO relevante), (Vacío=NO reporto), estas características deben ser tomadas en cuenta para realizar las medidas de precisión at K y estadísticas Kappa.
- **Condiciones de la prueba:** Para realizar las pruebas de Precisión at K y estadísticas Kappa se realizaron las 4 consultas y participaron 13 usuarios, se identificaron 25 documentos recuperados por el buscador en cada consulta. Se recopilieron los datos de manera automática y los resultados generales de las mismas se pueden apreciar en la Tabla 15.

Tabla 15: Resumen de Precisión Promedio para cada consulta

Consulta	Índice MAP
1	37%
2	39,8%
3	36,8%
4	18,3%

- **Resultados de la Prueba:** Comparando el índice MAP para cada uno de las consultas realizadas en la prueba se puede verificar que en la consulta 2 el valor de 39,9% fue el más alto, de lo que podemos inferir que se determinaron como relevantes un mayor número de documentos en las primeras posiciones de respuesta del buscador con relación a las otras tres consultas.

Un valor cercano al 40% determina una precisión optima ya se asume que este índice es excelente cuando se acerca al 70% por este motivo no se puede referenciar como un resultado negativo de la precisión de la primera posiciones k. Para mirar en detalle los resultados específicos refiérase al **ANEXO D-4.9**.

## 5.5 Métricas Kappa Cohen

Kappa es una medida de concordancia, pero la Kappa Cohen determina la concordancia entre dos Jueces, hallando la frecuencia y las combinaciones en que cada documento fue calificado. El análisis de los resultados se puede guiar con la Tabla 16.

Tabla 16: Tabulación de resultados Kappa

Kappa	Grado de Acuerdo
< 0,00	sin acuerdo
>0,00 - 0,20	insignificante
0,21 - 0,40	discreto
>0,41 - 0,60	moderado
0,61 - 0,80	sustancial
0,81 - 1,00	casi perfecto

- **Condiciones de la prueba:** En la prueba de Kappa Cohen se establece como parámetro que solo pueden participar 2 usuarios, en nuestra prueba controlada se realizaron 4 consultas, se tomó una muestra de 25 dispositivos con los resultados presentados en la Tabla 17.

Tabla 17: Resumen de Kappa Cohen para cada consulta

Consulta	Kappa Cohen
----------	-------------

1	0,651
2	0,705
3	0,705
4	0,336

- **Resultados de la prueba:** En las tres primeras consultas se presentó una concordancia (Sustancial), esto indica que las preguntas son consistentes con los resultados y le permite a los usuarios identificar el dispositivo que se encuentran en el dominio de búsqueda, estas consultas retornaron en las primeras posiciones los resultados de los sensores virtuales desarrollados y posteriormente sensores relacionados, permitiendo evidenciar que efectivamente la semántica recopilada permite identificar claramente el tipo de sensores que coinciden con la búsqueda. En la última consulta se evidencia que el grado de concordancia es (Discreta) y esto es evidente, ya que el área (lugar) de la pregunta no se relacionaba con los sensores virtuales desarrollados y retorno los pocos sensores que encontró en el servidor Xively, esto implica que aunque no se desarrollaron sensores de calidad del aire y de contaminación en New York el buscador es capaz de retornar los existentes así no estén bien documentados.

## 5.6 Métricas Kappa Fleiss

Esta métrica de concordancia tiene un enfoque más específico y el cual tiene en cuenta todas las (Clases) de respuestas que se presentan en la prueba.

- **Condiciones de la prueba:** Para Kappa Fleiss si toma importancia las clases o número de posibles juzgamientos que pueden emitir los usuarios. Para este caso se identifican 3 clases de resultados (1=Relevante), (0=NO relevante), (Vacío=NO reporto), la muestra fue de 25 dispositivos (n), un número de jueces 13 (m), y se emitieron 4 consultas. Los resultados se presentan en la
- Tabla 18.

Tabla 18: Resumen de Kappa Fleiss para cada consulta

Consulta	Capa Fleiss
1	0,402
2	0,554

3	0,355
4	0,052

- **Resultados de la Prueba:** En las dos primeras consultas se presentó una concordancia (Discreto), esto indica que los dispositivos son fáciles de determinar sus relevancias y que con un número relativamente bueno de jueces el resultado es consistente. En las dos últimas consultas se evidencia el grado de concordancia con tendencia (Insignificante) ya que esta métrica da especial interés a las clases y en estas últimas consultas la clase adicional de (Vacío) esto se debe a que el buscador puede retornar sensores a unos usuarios y a otros no en la misma consulta y la última consulta retorna pocos sensores relacionados, pero es debido más a la existencia real de los sensores que a la capacidad del buscador en recuperarlos. Por otro lado, al aumentar los posibles resultados disminuye la concordancia entre los usuarios, esto último porque es más difícil relacionarlos con el lugar buscado y el buscador retorna sensores con menos relevancia.

## 5.7 Prueba de usabilidad

### 5.7.1 Preguntas de usabilidad

Se realizó una encuesta y se definieron evaluar aspectos y preguntas de estos aspectos como se ve en la Tabla 19 y así establecer la conformidad de los usuarios con la usabilidad de la aplicación.

Tabla 19: Aspectos y Preguntas de Usabilidad

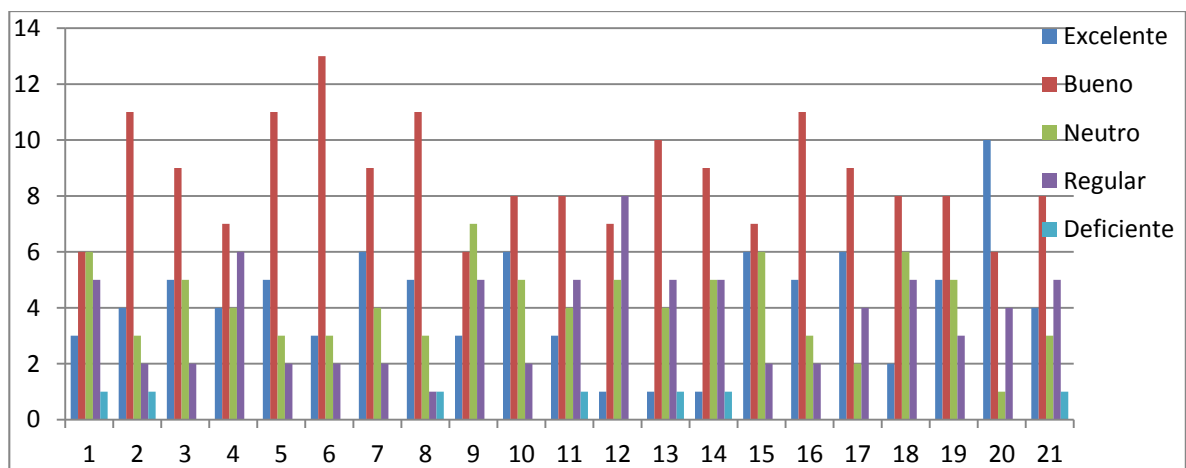
# Aspecto que evalúa	# Preguntas
a. Visibilidad del estado del sistema	1. El sitio muestra claramente dónde se encuentra el usuario?
	2. Los enlaces posibles de explorar están claramente señalados?
b. Relación entre sistema y mundo real	3. El lenguaje es claro?
	4. Los conceptos utilizados son entendibles?
	5. Las palabras son de significado conocido?
c. Consistencia y estándares	6. Existe coherencia entre el nombre de un enlace y el sitio al que apunta?
	7. Todos los enlaces tienen contenido?
d. Reconocer en	8. Los enlaces pueden identificarse claramente?

lugar de recordar	9. Es posible reconocer dónde se encuentra el usuario?
e. Recuperación de Información	10. El buscador es fácil de usar?
	11. Los resultados arrojados cumplieron las expectativas de su búsqueda?
f. Ayuda y documentación	12. Existe algún tipo de ayuda o indicación en el sitio?
	13. Cuando existe ayuda, ésta es específica?
	14. La ayuda es asequible?
g. ¿Cómo califica globalmente el sitio Web analizado?	15. Visibilidad del estado del sistema.
	16. Relación entre sistema y mundo real.
	17. Control del usuario y libertad.
	18. Prevención de errores.
	19. Flexibilidad y eficiencia de uso
	20. Estética y diseño
	21. Ayuda y documentación

### 5.7.2 Tabla de resultados de la prueba de usabilidad

Los resultados se graficaron en la siguiente Figura 16:

Figura 16. Resultado encuesta de Usabilidad



La prueba se realizó para 21 usuarios y se realizaron 21 preguntas cuyas respuestas son (Excelente, Bueno, Neutro, Regular, Deficiente).

Como se puede observar en la figura hay una alta frecuencia de la respuesta (Bueno) calificadas por todos los usuarios por pregunta, en la pregunta 6 se

alcanzo una frecuencia de 13 observandose la misma tendencia en el resto de preguntas. Esto significa que todos los aspectos evaluados en general se consideraron buenos, conclusión que se retifica en la pregunta 20 donde repunta (Excelente) y el aspecto 7 de la Tabla 19.

En la pregunta 9 la respuesta (Neutro) se encuentra un poco por encima de (Bueno). Esto significa que solo predomina una pregunta neutra en un total de 21 preguntas, de lo que se puede concluir que los resultados son positivos.

La encuesta se llevo a cabo con herramientas de Google en el enlace<sup>11</sup>, y de la cual se pueden verificar los resultados en el archivo “Evaluaciones.xlsx” del CD en la ruta:

C:\CD del Proyecto\Validacion Buscador

## 5.8 Conclusiones de las Pruebas

Después de realizar las pruebas y analizarlas podemos concluir lo siguiente:

- Es necesario que el (los) servidor(es) IoT que se seleccionen o se utilicen para indexar los objetos, tengan en lo posible varios nodos de información existentes en el dominio y lugares particulares, ya que esto puede afectar los resultados retornados por el buscador semántico. Para el caso particular la solución de generar sensores virtuales permitió reportar buenos resultados en el dominio y áreas seleccionadas en el caso de estudio particular.
- Para realizar las pruebas de precisión – recuerdo es necesario generar un dataset de prueba con expertos, esto requiere tiempo y disposición de recursos, pero podría mejorar la forma de evaluar este tipo de buscadores en la IoT.
- Las pruebas realizadas con los usuarios reportaron en general buenos resultados, sobre todo en las tres primeras consultas, ya que las mismas lograron recuperar en las primeras posiciones los sensores virtuales desarrollados y posteriormente los sensores relacionados así fueran de otros lugares.
- En la última consulta, fue consistente los resultados bajos ya que se preguntaba por sensores en otro lugar geográfico y al retornar en las primeras posiciones los del cauca y Santander, estas quedaban como no relevantes, pero esto sucede porque el lugar tiene menor relevancia en el modelo de

---

<sup>11</sup>  
JgfaQRV1nqfHJhT\_i7OGa0RKv4/viewform

[https://docs.google.com/forms/d/1pCmmtF6V2\\_6\\_8Fet-](https://docs.google.com/forms/d/1pCmmtF6V2_6_8Fet-)

búsqueda que las variables de contaminación ambientales que se presentaban con mayor relevancia al expandir la consulta.

- Aunque el buscador es capaz de agrupar por biotipos y buscar información clasificada, esto no se evaluó ya que las medidas tradicionales de relevancia no aplican a este tipo de resultados. Es necesario idear un mecanismo que permita evaluar estos resultados adicionales del buscador en el dominio particular.

## **CREACIÓN DE SERVICIOS WEB SEMÁNTICOS DEL BUSCADOR SEMANTICO PARA LA IOT**

Este capítulo presenta un diseño de servicios web semánticos, que utilizan los servicios web desarrollados en el índice semántico del buscador web para la IoT con el fin de hacer una demostración en como se implementarían los mismos, buscando una automatización y reuso por parte de aplicaciones y usuarios. No se realizó la implementación de los mismos ya que se salían de los alcances y recurso del presente estudio. A continuación se presentan unos conceptos clave para su mejor comprensión y posteriormente unas tablas que resumen los principales productos desarrollados. Para mayor información se puede referir al los anexos del final del capitulo y el cd que se entrega con la monografía.

### **6.1 Introducción**

El primer objetivo de los SWS(Studer, Grimm et al. 2007) es que una maquina pueda entender los descripciones y les permita razonar sobre sus componentes, el segundo es describir coreografías que las partes usan con el fin de crear de manera automática nuevos servicios.

#### **6.1.1 Coreografía de servicios**

La interacción entre el proveedor del servicio y los solicitantes debe contar con ciertas restricciones para que tengan sentido para ambas partes, la conversación entre servicios participantes se ajusta a un protocolo multipartidista que significa que las restricciones son establecidas para ambas partes. La coreografía se generaliza entre dos participantes pero pueden existir más, los extremos de los servicios se le denomina la tierra de la coreografía.

La descripción de las coreografías deben representar los mensajes a intercambiar y también representar el modelo de intercambio de mensajes. Esto permite a las entidades razonar sobre el modelo explícito de una coreografía para tomar decisiones dinámicas o saber qué mensaje es el siguiente por enviar.

#### **6.1.2 Orquestación de servicios**

Para determinar qué mensaje se debe enviar también hay que tener en cuenta la orquestación, esta determina que capacidades tiene cada parte, que mensajes envía y cuando los envía, mientras que la coreografía específica que capacidades se permiten de cada parte.

### **6.1.3 Composición de servicios**

Un participante de servicios toma el rol de solicitante en cierto momento y en otro momento el rol de proveedor de servicios, hay casos en que una parte se comunica con los proveedores para coordinar servicios más complejos coordinando los mensajes componiendo los servicios.

### **6.1.4 Enfoques de Coreografía, Orquestación y Composición de servicios**

BPEL selecciona proveedores para realizar la orquestación pero cuando un proveedor no está disponible falla, WSMF incluye descripciones de la funcionalidad de los proveedores descubriendo y seleccionando dinámicamente los servicios, pero OWL-S encarga a cada proveedor la tarea de generar las formas en que se puede combinar los servicios y así cada solicitante asume la orquestación.

### **6.1.5 Mediación de servicios**

Una interacción conlleva razonamiento y recursos para llevar a cabo un proceso, la mediación reconcilia estos procesos entre las partes de una interacción. Este tipo de mediación solo es posible de forma automática si los mensajes y las coreografías son anotados semánticamente.

La interacción entre dos partes puede llevar a cabo mediación de datos (transformación de formato sintáctico de datos), mediación ontologías (razonamiento para traducir a vocabulario y términos en ontologías de servicios), mediación coreografías o protocolo (se traducen las secuencias de mensajes de una parte por las secuencias de la otra parte para lograr el mismo fin es decir para que las coreografías sean idénticas) y por último mediación de procesos (alineación de procesos con diferentes modelos conceptuales donde se busca que el resultado de la ejecución de los procesos puedan ser compatibles).

## **6.2 Etapas para el diseño de SWS**

Las herramientas para SWS soportan la construcción y diseño de SWS, las 4 etapas principales para crear SWS son las siguientes.

### **6.2.1 Descubrimiento del Servicio:**

Las coincidencias de una o más descripciones del servicio con una descripción de servicio requisito. El solicitante del servicio puede coincidir o no con las descripciones de los servicios, en caso negativo puede que tenga que reformular la descripción, la mayoría de las herramientas no dan soporte para el proceso de descubrimiento de en múltiples etapas. WSMX proporciona instalación para las capacidades que necesita y el proceso de adaptación para realizar la selección más adecuada. Primero se realiza el proceso Matchmaker y el proceso Selector,

además pueden invocar mediadores para desajustes entre objetivos del servicio y los requisitos del solicitante.

### **6.2.2 Definición del Servicio**

Esta etapa es una de las más complejas y no se considera como componente de WSMO y sucede después de que se hace el descubrimiento del servicio; el cliente debe decidir cuál servicio seleccionar, los clientes utilizan las descripciones del servicio. El problema es la especificación del modelo de proceso, que define el orden de espera de información y que tipo de información se requiere, algunas herramientas soportan generación automática de interfaces y solicitan al cliente la información a enviar y las opciones no deterministas a realizar. Esta etapa la descripción del servicio elegido es representada formalmente en un contrato entre el proveedor del servicio y el solicitante.

### **6.2.3 Entrega del Servicio**

En esta etapa se contempla el intercambio de múltiples mensajes entre el proveedor y el solicitante. La entrega del servicio requiere apoyo para la coreografía esta es responsabilidad del solicitante, el WSMX da apoyo para realiza suposiciones fuertes acerca de la interacción y el despliegue, su estructura centralizada entrega la capacidad de controlar la composición de servicios y así poder aplicar la coreografía representando en WSMO el servicio, lo que es muy difícil o casi directamente imposible; no queda muy claro como los supuestos en WSMX se trasladan a WSMO. Para la invocación del servicio en WSMX hay un componente llamado (Communication Manager) que invoca servicios seleccionados, levantando y bajando transformaciones entre la semántica que describen los servicios. En la invocación del servicio también puede implementar mediadores de datos que transforman los datos entre las ontologías del proveedor y el solicitante.

## **6.3 Herramientas de Desarrollo de un SWS**

Un servicio web semántico es aquel que es impulsado por una ontología, la ontología presenta anotaciones para servicios. Inicialmente lo que se hace es encontrar ontologías adecuadas para crear la descripción semántica del servicio web.

El servicio solicitante y el proveedor ofrecen descripciones de los requisitos del servicio. Los archivos WSDL describen las operaciones y mensajes intercambiados, un archivo WSDL no poseen descripciones semánticas de los datos de entrada y salida y además no imponen ningún orden en la invocación de las operaciones. Lo que se busca es que a partir del archivo WSDL crear una descripción completa del Modelo de Proceso y la Conexión a Tierra del servicio

(puntos de entrada) en el WSMO y luego convertir este archivo al WSML y por último se define una coreografía para el SWS.

Existen muchas más se han relacionado las siguientes herramientas:

1. Herramienta WSMT<sup>12</sup> se basa en el lenguaje WSML.
2. herramienta WSMX<sup>13</sup> servidor de SWS.

#### **6.4 Prototipo de Servicios Web Semánticos para SIMIOT**

En el **ANEXO E** aparece cómo se creó el prototipo donde se creó en la herramienta WSMT los servicios que fueron creados para el buscador de objetos, es necesario decir que aunque se implementaron los servicios utilizando en las herramientas la puesta en producción es un trabajo que excede los objetivos del presente proyecto y serán planteados para futuros trabajos.

#### **6.5 Diseño general del Servicio Web Semántico SIMIOT**

De la Tabla 24 se presentan la clasificación y los nombres del diseño de los SWS para diseñados para SIMIOT.

#### **6.6 Ubicación del diseño**

Los archivos fuente desarrollados en la herramienta WSMT, con el nombre de "WSMT\_SIMIOT.zip" se pueden encontrar en el CD del proyecto en la siguiente ruta:

C:\CD del Proyecto\Diseño Servicio Web Semántico

---

<sup>12</sup> <http://www.antepedia.com/detail/p/4864081.html>

<sup>13</sup> <http://sourceforge.net/projects/wsmx/files/distribution/WSMX-0.5/>

## 6.6.1 Ontologías

Tabla 20: Diseño de Ontologías SIMIOT

ESPACIO DE NOMBRES	ONTOLOGIA	ONTOLOGIA IMPORTADA	CONCEPTOS	ATRIBUTOS	RELACIONES	AXIOMAS	MEDIADOR USADOS
http://BusquedaEntidad/	BusquedaEntidad	http://ResultadosConsulta/	Entidad	entidadConsulta	EnviaEntidad	SetConsulta	http://EntidadConceptosMediador/
		http://BusquedaConcepto/		entidadConsulta	RetornaConsultas	GetConsulta	http://EntidadConsultaMediador/
		http://BusquedaTermino/		entidadTermino	SolicitaConceptos	setConceptos	http://EntidadTerminosMediador/
				Descripcion	RetornaConceptos	getConcepto	
					SeleccionConcepto	getSeleccion	
					EnviaTerminos	setTerminos	
					RefinaConsultas	getRefina	
http://ResultadosConsulta/	ResultadosConsulta	http://BusquedaEntidad/	Consulta	Titulo			
				Enlace			
				ID_Consulta			
http://BusquedaTermino/	BusquedaTermino	http://BusquedaEntidad/	Termino	PalabraTermino			
http://BusquedaConcepto/	BusquedaConcepto	http://BusquedaEntidad/	Concepto	PalabraConcepto			
				SeleccionaConcepto			

## 6.6.2 Servicios

Tabla 21: Diseño de Servicios SIMIOT

ESPACIO DE NOMBRES	SERVICIO	CAPACIDAD	ASUNCION	PRECONDICION	POSTCONDICION	EFEECTO	ONTOLOGIA IMPORTADA
http://sw_Solicitante/	sw_Solicitante	SolicitanteCapacidad	SolicitanteAsuncion	SolicitantePrecondicion	SolicitantePostcondicion	SolicitanteEfecto	http://BusquedaEntidad/BusquedaEntidad
http://sw_Busqueda/	sw_Busqueda	BusquedaCapacidad	BusquedaAsuncion	BusquedaPrecondicion	BusquedaPrecondicion	BusquedaEfecto	
http://sw_Conceptos/	sw_Conceptos	ConceptosCapacidad	ConceptoAsuncion	ConceptosPrecondicion	ConceptosPostcondicion	ConceptoEfecto	

### 6.6.3 Metas

Tabla 22: Diseño de Metas SIMIOT

ESPACIO DE NOMBRES	META	CAPACIDAD	POSTCONDICION
http://RefinarConsultaConConceptosMeta/	BusquedaConceptoMediador	RefinarConsultaConConceptosMetaCapability	RefinarConsultaConConceptosPostcondicion

### 6.6.4 Mediadores entre Ontologías

Tabla 23: Diseño de Mediador de Ontologias SIMIOT

ESPACIO DE NOMBRES	ooMEDIADOR	ONTOLOGIA SOURCE	ONTOLOGIA TARGET
http://EntidadConceptosMediador/	entcop	http://BusquedaConcepto/BusquedaConcepto	http://BusquedaEntidad/BusquedaEntidad
http://EntidadConsultaMediador/	entcon	http://ResultadosConsulta/ResultadosConsulta	http://BusquedaEntidad/BusquedaEntidad
http://EntidadTerminosMediador/	entter	http://BusquedaTermino/BusquedaTermino	http://BusquedaEntidad/BusquedaEntidad

### 6.6.5 Mediadores entre Servicios

Tabla 24: Diseño de Mediador de Conceptos SIMIOT

ESPACIO DE NOMBRES	wwMEDIADOR	ONTOLOGIA SOURCE	ONTOLOGIA TARGET
http://BusquedaConceptoMediador/	BusquedaConceptoMediador	http://sw_Busqueda/sw_Busqueda	http://sw_Conceptos/sw_Conceptos
http://SolicitanteBusquedaMediador/	SolicitanteBusquedaMediador	http://sw_Solicitante/sw_Solicitante	http://sw_Busqueda/sw_Busqueda

## 7 CONCLUSIONES

Finalmente, a partir del trabajo realizado podemos condensarlo en las siguientes conclusiones:

- Se ha realizado un **marco de referencia**, que identifica las tecnologías para la creación de índices semánticos en la IoT. Se aportaron estudios propios en las tecnologías middleware de la IoT que pueden ser reutilizados en estudios relacionados.
- Se realizó la propuesta de una **Arquitectura Inicial** para la recuperación de información semántica de los objetos y procesos en la IoT. Esta arquitectura reúne los principales elementos identificados en otros estudios y la visión de solución propuesta por el modelo conceptual semántico del presente trabajo.
- Se propuso un **Método de indexación semántica en la IoT**, guiado por un modelo semántico conceptual bien definido y formalizado en el estándar de SPEM 2.0, el cual puede ser seguido por un grupo de desarrollo de software, haciéndolo reutilizable, ampliable y entendible para su posterior apropiación en otros estudios.
- Se creó una **ontología en el campo de la contaminación ambiental** y alineada con otras ontologías estándar en el tema ambiental y de la IoT, con el fin de poder desarrollar un índice en el caso de estudio medioambiental de dos regiones de Colombia, específicamente Cauca y Santander.
- Se creó un **índice semántico en la IoT** en el área de la contaminación medioambiental, el cual expone su funcionalidad mediante un **conjunto de servicios web** para implementar aplicaciones que utilicen dichos servicios. El índice es reutilizable y adaptable al permitir cambiar la ontología de dominio que utiliza para realizar la indexación semántica.
- El índice semántico fue probado mediante la implementación de un **buscador web semántico**, que permite buscar sensores en la IoT que pueden traer información de variables de contaminación ambiental. Adicionalmente, es capaz de realizar análisis por biotipo y por contaminantes del aire.
- Se realizó la **validación** del buscador Web Semántico de contaminación medioambiental que implementa el método propuesto. Los resultados

preliminares denotan una recuperación de información relevante y la posibilidad de utilizar los servicios del índice en otras aplicaciones.

- Se creó un conjunto de **sensores virtuales** de contaminación del aire en dos regiones de Colombia (Cauca y Santander), teniendo en cuenta las reglamentaciones nacionales y las características y rangos de los sensores reales, esto permitió asegurar resultados consistentes cuando dichos dispositivos se reemplacen por los físicos.
- Se presentó un **diseño de servicios Web semánticos** con los servicios web desarrollados en el índice creado, con el fin de demostrar la manera de reutilizar estos servicios y llevarlos a un entorno automatizado.

Las recomendaciones propuestas son:

- Es necesario realizar más pruebas del método propuesto, con el fin de afirmar su utilidad y mejorarlo.
- Es importante que el servidor de objetos o el middleware utilizado, efectivamente tenga sensores con metadatos adecuados en el dominio particular de indexación, con el fin de que las búsquedas de información reporten resultados adecuados.
- Para poder validar adecuadamente los índices creados y su recuperación de información, es importante desarrollar una dataset de prueba, que permita evaluar la precisión – recuerdo con mayor fiabilidad.
- Para reutilizar el índice semántico es importante desarrollar y utilizar ontologías ajustadas al dominio particular de indexación. Un elemento importante es que la ontología tenga anotaciones en español e inglés con el fin de aprovechar la funcionalidad en los dos idiomas que ofrece el índice desarrollado.

El trabajo futuro se resume en:

- Realizar más pruebas del método de indexación semántica en otros dominios de aplicación, para realimentar el método mismo y crear nuevos servicios de información en la WoT.
- Realizar un despliegue de sensores reales en un caso de estudio particular con el fin de establecer los servicios de información que se podrían crear al implementar un índice semántico en el dominio.
- Reutilizar los desarrollos creados con el fin de desarrollar un entorno de interoperabilidad semántica entre objetos de la IoT.

## BIBLIOGRAFIA

- Aggarwal, C., N. Ashish, et al. (2013). The Internet of Things: A Survey from the Data-Centric Perspective. *Managing and Mining Sensor Data*. C. C. Aggarwal, Springer US: 383-428.
- Akkaya, K. and M. Younis (2005). "A survey on routing protocols for wireless sensor networks." *Ad Hoc Networks* 3(3): 325-349.
- Al-Karaki, J. N. and A. E. Kamal (2004). "Routing techniques in wireless sensor networks: a survey." *Wireless Communications, IEEE* 11(6): 6-28.
- Atzori, L., A. Iera, et al. (2010). "The Internet of Things: A survey." *Computer Networks* 54(15): 2787-2805.
- Avello, D. G. (2005). *BlindLight- Una nueva técnica para procesamiento de texto no estructurado mediante vectores de n-gramas de longitud variable con aplicación a diversas tareas de tratamiento de lenguaje natural*. Departamento de Informática. Oviedo, Universidad de Oviedo: 241.
- Bandyopadhyay, D. and J. Sen (2011). "Internet of Things: Applications and Challenges in Technology and Standardization." *Wireless Personal Communications* 58(1): 49-69.
- Bandyopadhyay, S., M. Sengupta, et al. (2011). "Role of middleware for internet of things: A study." *International Journal of Computer Science & Engineering Survey (IJCSES)* 2(3): 94-105.
- Barragán, M. A. R. (2008). *Un Método para Recuperación de Información en Documentos Orales basado en Codificación Fonética*. Tonantzintla, Puebla, Instituto Nacional de Astrofísica, Óptica y Electrónica.
- Baziz, M., M. Boughanem, et al. (2005). "Evaluating a Conceptual Indexing Method by Utilizing WordNet." 8.
- Benavides, K. D. R. (2011). "Índices de RI." Retrieved 14 April de 2011, From [http://www.kramirez.net/RI\\_Maestria/Material/Presentaciones/Indices%20de%20RI.pdf](http://www.kramirez.net/RI_Maestria/Material/Presentaciones/Indices%20de%20RI.pdf).
- Biamino, G. (2012). "A Semantic Model for Socially Aware Objects." *Advances in Internet of Things*.
- Botts, M., G. Percivall, et al. (2008). "OGC® sensor web enablement: Overview and high level architecture." *GeoSensor networks*.
- Bouros, P. (2005). *Semantic Web Services: A conceptual comparison of*

- OWL-S, WSMO and METEOR-S approaches, Technical report, Department of Informatics and Telecommunications-National and Kapodistrian University of Athens (NKUA), Panepistimiopolis, TYPA Buildings, GR-157 84 Ilisia, Athens, Greece.
- Bröring, A., J. Echterhoff, et al. (2011). "New generation sensor web enablement." *Sensors* 11(3): 2652-2699.
- Broring, A., P. Maue, et al. (2012). Semantic mediation on the Sensor Web. Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International.
- Cardoso, J. and A. P. Sheth (2006). *Semantic Web Services, Processes and Applications (Semantic Web and Beyond: Computing for Human Experience)*, Springer-Verlag New York, Inc.
- Ceri, S. and A. Bozzon (2013). *Web Information Retrieval*. Springer Berlin Heidelberg: 3-11.
- Compton, M., P. Barnaghi, et al. (2012). "The SSN ontology of the W3C semantic sensor network incubator group." *Web Semantics: Science*.
- Corcho, O., M. Fernández-López, et al. (2005). "Building legal ontologies with METHONTOLOGY and WebODE." *Law and the semantic Web*.
- Corcho, O., Ra, et al. (2010). "Five challenges for the Semantic Sensor Web." *Semant. web* 1(1,2): 121-125.
- Cord, V., P. Lombardi, et al. (2005). "An Ontology-Based Similarity between Sets of Concepts."
- Croft, B., D. Metzler, et al., Eds. (2009). *Search Engines: Information Retrieval in Practice*. USA, Addison-Wesley.
- Chang, C. and B. Schatz (1999). Performance and Implications of Semantic Indexing in a Distributed Environment. Proceedings of the eighth international conference on Information and knowledge management Kansas City, Missouri, United States, ACM, New York: 391-398.
- Chaqfeh, M. A. and N. Mohamed (2012). Challenges in middleware solutions for the internet of things. Collaboration Technologies and Systems (CTS), 2012 International Conference on.
- Chung, Y.-M., Q. He, et al. (1999). "Semantic Indexing for a Complete Subject Discipline." Proceedings of the fourth ACM conference on Digital libraries 39-48.
- Daconta, M. and L. Obrst (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Indianapolis, Indiana and Canada Wiley.
- Delbru, R., S. Campinas, et al. (2012). "Searching web data: An entity

- retrieval and high-performance indexing model." *Web Semantics: Science, Services and Agents on the World Wide Web* 10(0): 33-58.
- Desmontils, E. and C. Jacquin (2002). "Indexing a Web Site with a Terminology Oriented Ontology." 181-198.
- Fensel, D., F. M. Facca, et al. (2011). *Semantic web services*, Springer.
- Fernández-López, M. and A. Gomez-Perez (2003). "Overview and analysis of methodologies for building ontologies." *The Knowledge Engineering Review*: 129-156.
- Fernández, M., I. Cantador, et al. (2011). "Semantically enhanced Information Retrieval: An ontology-based approach." *Web Semantics: Science, Services and Agents on the World Wide Web* 9(4): 434-452.
- Fortino, G., A. Guerrieri, et al. (2012). Agent-oriented smart objects development. *Computer Supported Cooperative Work in Design (CSCWD)*, 2012 IEEE 16th International Conference on.
- Frakes, W. (1992). *Information retrieval: data structures and algorithms*, citeulike.org.
- Friedewald, M. and O. Raabe. (2011). "Ubiquitous computing: An overview of technology impacts." *Telematics and Informatics* Retrieved 2, 28, from <http://www.sciencedirect.com/science/article/pii/S0736585310000547>
- Gigli, M. and K. S. (2011). "'Internet of Things: Services and Applications Categorization," *Advances in Internet of Things.*" *Scientific Research* 1 No. 2: 27-31.
- Gluhak, A., S. Krco, et al. (2011). "A survey on facilities for experimental internet of things research." *IEEE Communications Magazine* 49(11): 58-67.
- Gubbi, J., R. Buyya, et al. (2013). "Internet of things (IoT): A vision, architectural elements, and future directions." *Future Generation Computer Systems*.
- Guha, R. V. and D. B. Lenat (1990). "Cyc: a mid-term report." *AI magazine*.
- Handong, Z. and Z. Lin (2011). Internet of Things: Key technology, architecture and challenging problems. *Computer Science and Automation Engineering (CSAE)*, 2011 IEEE International Conference on.
- Henricksen, K. and R. Robinson (2006). A survey of middleware for sensor networks: state-of-the-art and future directions. *Proceedings of the international workshop on Middleware for sensor networks*. Melbourne, Australia, ACM: 60-65.
- Hernández, J. P. R. and G. A. Hernández (2008). *Indización y Búsqueda a*

- través de Lucene. Veracruz, Sinaloa, Universidad Autónoma de Sinaloa, Instituto Tecnológico de Orizaba.
- Herrera, A. G. L. (2006). Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada. Doctor en Informática: 255.
- Hoang, L. N. (2007). Middlewares for Home Monitoring and Control. Seminar on Internetworking (March 2007).
- Hock Beng, L., M. Iqbal, et al. (2009). A Large-Scale Service-Oriented Sensor Grid Infrastructure. Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE.
- IPSO. (2013). "IPSO Alliance." Retrieved Recovered 29/07/2013, from <http://www.ipso-alliance.org/>.
- James, A. and J. Cooper (2009). Challenges for Database Management in the Internet of Things.
- Jenkins, N. (2008). A Software Testing Primer, [nickjenkins.net](http://nickjenkins.net).
- Kara, S., Ö. Alan, et al. (2012). "An ontology-based retrieval system using semantic indexing." *Information Systems* 37(4): 294-305.
- Kashyap, V. and C. Bussler (2008). *Semantic Web Services*, Springer Berlin Heidelberg: 249-286.
- Kavouras, M. (2005). A unified ontological framework for semantic integration.
- Kavouras, M. and M. Kokla (2000). Ontology-based fusion of geographic databases, [ntua.gr](http://ntua.gr).
- Kent, A., M. M. Berry, et al. (1955). "Machine literature searching VIII. Operational criteria for designing information retrieval systems." *American Documentation* 6(2): 93-101.
- Kiritsis, D. (2011). "Closed-loop PLM for intelligent products in the era of the Internet of things." *Computer-Aided Design* 43(5): 479-501.
- Kostelník, P., M. Sarnovský, et al. (2011). "The Semantic Middleware for Networked Embedded Systems Applied in the Internet of Things and Services Domain." *Scalable Computing: Practice and Experience* 12(3).
- Le-Phuoc, D., H. Q. Nguyen-Mau, et al. (2012). "A middleware framework for scalable management of linked streams." *Web Semantics: Science, Services and Agents on the World Wide Web*(0).
- Lin, D. (1998). "An Information-Theoretic Definition of Similarity." *Proc 15th International Conference on Machine Learning*: 296-304.

- Lv, G., C. Zheng, et al. (2009). "Text Information Retrieval Based on Concept Semantic Similarity." 2009 Fifth International Conference on Semantics, Knowledge and Grid: 356-360.
- Manning, C. D., P. Raghavan, et al. (2008). Introduction to Information Retrieval. Cambridge, Cambridge University Press.
- Mathew, S. S., Y. Atif, et al. (2011). Web of Things: Description, Discovery and Integration. Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing.
- Mathew, S. S., Y. Atif, et al. (2011). "Web of things: Description, discovery and integration." Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing: 9-15.
- Mazuel, L. and N. Sabouret (2008). Semantic Relatedness Measure Using Object Properties in an Ontology. Proceedings of the 7th International Conference on The Semantic Web. Karlsruhe, Germany, ACM: 681-694.
- Mendiburu, H. (2003). Automatización Medioambiental. Lima, Perú, INDECOPI.
- Miao, Y. and Y. Bu (2010). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. Advances in Energy Engineering (ICAEE), 2010 International Conference on.
- Miller, G. A., R. Beckwith, et al. (1990). Introduction to WordNet: An On-line Lexical Database, Oxford University Press.
- Mingxia Gao, C. L., Furong Chen (2008). "An Ontology Search Engine Based on Semantic Analysis, Extraído el 20 de octubre de 2010." IEEE.
- Niño-Zambrano, M. A. (2013). Interacción Semántica de Objetos en la Web de las Cosas. Departamento de Ingeniería Telemática. El trabajo aún no ha sido publicado y se encuentra en proceso., Universidad del Cauca. PhD.
- Niño, M. A., I. D. López, et al. (2012). "Modelo Semántico de Expansión de Consultas para la Búsqueda Web (MSEC)." UIS Ingenierías 11(1): 10.
- Novoa, D. and L. Ballen (2007). "La Indexación Semántica Latente en la recuperación de información." Preprint.
- Noy, N. F., M. Sintek, et al. (2001). "Creating semantic web contents with protege-2000." IEEE Intelligent Systems.
- OMG (2008) "Software & Systems Process Engineering Meta-Model

- Specification Version 2.0." 236.
- OMG (2008, Abril). "Metamodel specification (SPEM) V2.0." 218.
- org., E. (2013). "OpenUp." Retrieved 30/05/2013, from <http://epf.eclipse.org/wikis/openup/>.
- Ostermaier, B., Ro, et al. (2010). A real-time search engine for the Web of Things. Internet of Things (IOT), 2010.
- Patni, H., C. Henson, et al. (2010). Linked sensor data. Collaborative Technologies and Systems (CTS), 2010 International Symposium on.
- Perera, C., A. Zaslavsky, et al. (2013). "Context Aware Computing for The Internet of Things: A Survey." IEEE Communications Surveys & Tutorials.
- Pezo, D. M., D. J. Perez, et al. (2013). "Procedimiento para la Construcción de Índices Semánticos Basados en Ontologías de Dominio Específico." Entramado 9: 27.
- Presser, M., P. M. Barnaghi, et al. (2009). "The SENSEI project: integrating the physical world with the digital world of the network of the future." Communications Magazine, IEEE 47(4): 1-4.
- Qin, X. and Y. Gu (2011). "Data fusion in the Internet of Things." Procedia Engineering 15(0): 3023-3026.
- Rada, M. and D. I. Moldovan (2000). An Iterative Approach to Word Sense Disambiguation. Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference Orlando, FL, AAAI Press: 219 - 223.
- Resnik, P. (1999). "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language." Journal of Artificial Intelligence Research 11: 36.
- Romer, K., B. Ostermaier, et al. (2010). "Real-Time Search for Real-World Entities: A Survey." Proceedings of the IEEE 98(11): 1887-1902.
- Römer, K., B. Ostermaier, et al. (2010). "Real-Time Search for Real-World Entities: A Survey."
- Salton, G. and M. J. McGill (1986). Introduction to Modern Information Retrieval. New York, McGraw-Hill: paginas 400.
- Samaneh Chagheri, C. R., Sylvie Calabretto, Cyril Dumoulin (2009). "Semantic Indexing of Technical Documentation." Laboratoire d'InfoRmatique en Image et Systèmes d'information: 12.
- Sánchez, D. and A. Moreno (2008). "Learning non-taxonomic relationships from web documents for domain ontology construction." Data & Knowledge Engineering 64(3): 600-623.

- Sánchez, D. M., J. M. Cavero, et al. (2007). The road toward ontologies. *Ontologies*, Springer: 3-20.
- Sanderson, M. (1996). Word sense disambiguation and information retrieval. Department of Computing Science. Glasgow, University of Galsgow. PhD: 136.
- Sangers, J., F. Frasincar, et al. (2013). "Semantic Web service discovery using natural language processing techniques." *Expert Systems with Applications* 40(11): 4660-4671.
- Schatz, B. R. (1997). "Information Retrieval in Digital Libraries: Bringing Search to the Net." *Science - Bioinformática* 275: 327 - 334.
- Sheth, A., C. Henson, et al. (2008). "Semantic Sensor Web." *Internet Computing*, IEEE 12(4): 78-83.
- Song, W., C. H. Li, et al. (2009). "Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures." *Expert Systems with Applications* 36: 9095–9104.
- Studer, R., S. Grimm, et al. (2007). *Semantic Web Services: Concepts, Technologies, and Applications*, Springer-Verlag New York, Inc.
- Swets, J. (1969). "Effectiveness of information retrieval methods." *American Documentation*.
- Tao, L. and L. Dongxin (2012). The application and development of IOT. *Information Technology in Medicine and Education (ITME)*, 2012 International Symposium on.
- Thomas, C. and A. Sheth (2006). Chapter 1 On the expressiveness of the languages for the semantic web — Making a case for 'a little more'. *Capturing Intelligence*. S. Elie, Elsevier. Volume 1: 3-20.
- Vetter, J., J. Hamard, et al. (2007). Physical mobile interaction with dynamic physical object. *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*. Singapore, ACM: 339-340.
- Wei, W., S. De, et al. (2012). A Comprehensive Ontology for Knowledge Representation in the Internet of Things. *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-2012)*.
- Yates, R. B. and B. R. Neto (1999). *Modern Information Retrieval*. New York, ACM Press.
- Yepes, J. J. (2009). *Ontology Refinement for Improved Information Retrieval in the Biomedical Domain*. Castellón, Universitat Jaume. PhD Thesis.
- Yuxi, L. and Z. Guohui (2012). *Key Technologies and Applications of*

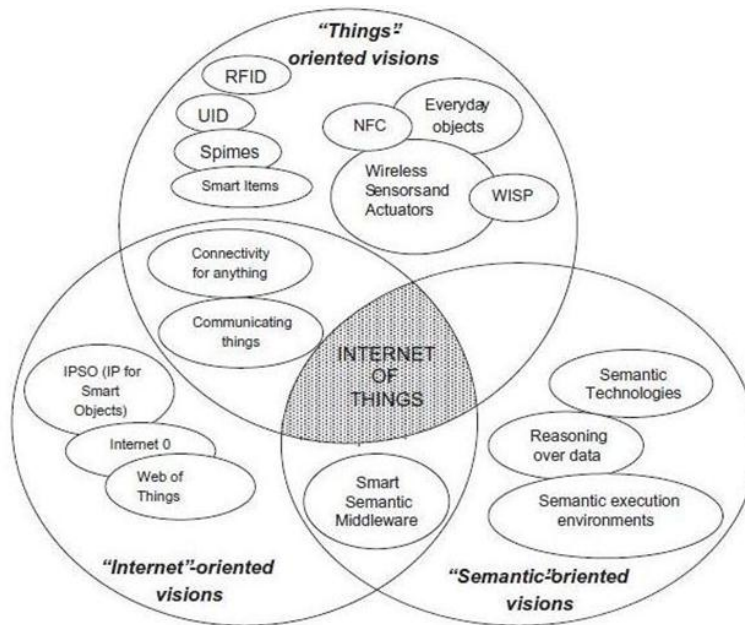
- Internet of Things. Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on.
- Zhang, C., C. Cheng, et al. (2012). Architecture design for social web of things. Proceedings of the 1st International Workshop on Context Discovery and Data Mining. Beijing, China, ACM: 1-7.
- Zhou, H. (2012). Architecture Standardization for WoT. The Internet of Things in the Cloud, CRC Press: 217-254.
- Zhou, H. (2012). Cloud Computing. The Internet of Things in the Cloud, CRC Press: 257-278.
- Zhou, H. (2012). The Cloud of Things. The Internet of Things in the Cloud, CRC Press: 279-300.
- Zhou, H. (2012). The Internet of Things in the Cloud: A Middleware Perspective. The Internet of Things in the Cloud, CRC Press.
- Zobel, J. (2006). "Inverted files for text search engines." ACM Computing Surveys (CSUR).

## ANEXO A: MARCO REFERENCIA

### 1 Convergencia de Visiones en la IoT

La IoT se puede ver como una intersección de tres visiones diferentes (Aggarwal, Ashish et al. 2013). Según (Atzori, Iera et al. 2010) , con la Visión orientada a las cosas, visión orientada a la semántica y la visión orientada a la Internet.

Figura 17. Internet de Objetos convergencia de visiones



Tomado de: The Internet of Things: A survey, (Atzori, Iera et al. 2010)

- **Thing Oriented Visions / Visión orientada a las cosas:** La visión orientada a las cosas corresponde a las tecnologías desarrolladas para identificar (*etiquetar*) y comunicarse con los objetos como: RFID, sensores, actuadores y teléfonos móviles entre otros, los cuales pueden operar e interactuar entre si y buscar objetivos comunes.
- **Internet Oriented Visions / Visión Orientada a la Internet:** La visión orientada a la Web se basa en protocolos de internet (*Internet Protocol – IP*) para el desarrollo de servicios de comunicación con los objetos, éstos pueden ser introducidos en elementos del mundo real como vallas publicitarias, automóviles e interruptores de luz entre otras.
- **Semantic Oriented Visions / Visión Orientada a la Semántica:** La tercera visión es la orientada a la semántica se basa en la separación entre datos y el significado de los mismos, almacenando por separado los significados semánticos de los datos y la

capacidad de razonar sobre ellos mismos. Estandarizar las descripciones de recursos heterogéneos es clave para interoperabilidad del (IoT).

## 2 Aplicaciones del Internet de Objetos IoT

Las aplicaciones (Atzori, Iera et al. 2010) del IoT podrían mejorar la calidad de vida de las personas los objetos equipados con inteligencia primitiva como: autos, electrodomésticos, dispositivos médicos, máquinas para el trabajo y el deporte, si se les da la capacidad de razonar y comunicarse entre ellos, permite generar muchos escenarios de aplicación, en la siguiente tabla se presentan ejemplos de aplicaciones del (IoT).

### 1.1 Transporte

- **Logística:** Wal-Mart y Metro en pocos días puede trabajar el stock con seguridad debido al seguimiento en tiempo real que realiza a la cadena de suministros con tecnologías RFID y de campo de comunicación cercana (*Near Field Communication – NFC*) en compra de materias primas, producción y transporte, almacenamiento, distribución y comercialización. El software (Enterprise Resource Planning – ERP) en tiempo real informa a los clientes sobre la disponibilidad de productos.
- **Conducción asistida:** Los vehículos equipados con sensores, actuadores y capacidad de procesamiento, pueden proporcionar información importante al conductor y pasajeros. Sistemas para evitar accidentes, información sobre el estado de los productos transportados.
- **Entradas y los celulares:** Los carteles y paneles informativos pueden ser equipados con etiquetas (NFC) o marcadores visuales (Quick Response Code - QR), por medio del celular se recoge la información de los servicios asociados.
- **Parámetros ambientales:** El seguimiento de productos perecederos como frutas carnes verduras, monitoreando el estado de conservación y control de temperatura y humedad.
- **Aumentar los mapas:** Es identificar un mapa con un (NFC) y relacionarlo con un servicio web en el cual se entrega información sobre carreteras, hoteles, restaurantes, monumentos y eventos. Las técnicas (PMI) (Vetter, Hamard et al. 2007) pueden emplearse para aumentar la información de etiquetas.

### 1.2 Cuidado Médico

- **Seguimiento:** Es la autenticación de los pacientes y objetos, además de la monitorización de sus signos vitales. El rastreo está destinado a la identificación de una persona u objeto en movimiento.
- **Identificación y autenticación:** Es usado para cumplir los requisitos y procedimientos de seguridad y evitar los robos y pérdidas.
- **Recolección:** Es adquisición de la información de los pacientes como historial clínico, automatización en el procesamiento de formularios atención, gestión de inventarios y procesos de auditoría.
- **Sintiendo:** Es el diagnóstico de las condiciones del paciente, un ejemplo es la telemedicina que supervisa el cumplimiento de los pacientes con la toma de medicamentos, otro ejemplo es la atención ambulatoria por medio de acceso remoto y por último monitorizar la movilidad de los pacientes.

### 1.3 Entornos Inteligentes

La inteligencia contenida en los objetos permitirá facilitar el trabajo en la oficina, la industria, el hogar o para la recreación.

- **Viviendas y oficinas confortables:** Es poder hacer la vida más cómoda por medio de sensores y actuadores, por ejemplo: la calefacción puede adaptarse a nuestras preferencias y clima, la iluminación de la habitación puede variar de acuerdo a los momentos del día. Los accidentes pueden evitarse con sistemas de seguimiento y alarma, además se puede ahorrar energía con sistemas apagado automático.
- **Industria:** Se etiquetan elementos, a la etiqueta se le asigna un evento que es activado cuando se le inalámbricamente por la unidad de procesamiento maquina o robot, así mismo se puede controlar la calidad del proceso por medio de sensores por ejemplo de vibración el cual se conecta a una pieza y dependiendo del umbral de vibración se activa el evento para detener el proceso. Seguimiento en tiempo real el Planeamiento de Recursos Empresariales (ERP) de pérdidas y el progreso de la producción.
- **Museos y gimnasios inteligentes:** En los museos se pueden interactuar con la información con mayor calidad en exposiciones y en el gimnasio a las maquinas se les puede programar la rutina de las personas.

#### 1.4 Personal y Social

- **Dominio Personal y social:** El establecimiento de relaciones sociales por medio de mecanismos de interacción, como programar mensajes a los amigos a través de etiquetas RFID, pueden generar la creación de eventos con personas y lugares de manera intuitiva.
- **Redes Sociales:** Actualización automática de la información de las redes sociales a través de eventos programados por el usuario utilizando etiquetas RFID, puede actualizar la información en tiempo real.
- **Consultas históricas:** Para realizar consultas sobre los datos y eventos históricos de los objetos permite desarrollar aplicaciones que soportan actividades a largo plazo, como una agenda digital en Google calendar, donde el usuario puede consultar en que ha gastado su tiempo. También generando lugares con API grafica de Google, que hace seguimiento de sitios y fechas.
- **Pérdidas y Robos:** la utilidad puede ser el seguimiento de objetos que no recordamos donde ha quedado por medio de aplicaciones de búsqueda.

#### 1.5 Futuristas

En el campo de comunicaciones se pueden desarrollar sistemas de detección, desarrollo de materiales y tecnologías de procesos industriales, un ejemplo es el proyecto SENSEI(Presser, Barnaghi et al. 2009).

- **Taxi Robo:** Ideado para las ciudades del futuro donde la prestación de este servicio sea de manera eficiente, reduciendo la congestión de tráfico y sin la necesidad de un conductor humano, y el usuario se ubica por GPS.
- **Modelo de información de ciudad (City Information Model - CIM):** Es una aplicación donde se sigue el estado y rendimiento de cada uno de los elementos de una ciudad como edificios, calles, cloacas, vías férreas, estaciones de autobuses y pasos peatonales.
- **Sala de juego:** Mejorar la sala de juego equipándolas con una cantidad de dispositivos para detectar la ubicación, movimiento, aceleración, información visual, ruido, voz, temperatura y humedad.

## 2. Protocolos de comunicación para la IoT

Los protocolos que se deben tener en cuenta para la IoT a nivel de capa de borde se enfocan en el enrutamiento de la redes de nodos y la redes inalámbricas de sensores WSN, a diferencia de los protocolos a nivel de capa internet, de capa middleware y la capa aplicación se enfocan en la interoperabilidad de dispositivos por ejemplo los protocolos de comunicación para el dominio del hogar.

### 2.1 Protocolo de Enrutamiento de Nodos

Los protocolos IP de las redes Ad Hoc no son eficientes para redes de sensores que son muy cambiantes y de numerosos nodos, por este motivo las arquitecturas de los protocolos deben utilizar técnicas para dividir las redes en grupos y para integración de datos por medio de algoritmos de fusión(Qin and Gu 2011) de datos apoyando las capacidades limitadas de energía, ancho de banda, procesamiento, almacenamiento. Los protocolos de enrutamiento a nivel de red utilizan estrategias para el descubrimiento de la topología de red como Inundaciones (Flooding) esta es la difusión de mensajes a todos los vecinos, otra es una versión mejorada es el (Gossiping) que selecciona al azar los vecinos a los que le envía mensajes para establecimiento de rutas(Akkaya and Younis 2005).

Comparado con las redes inalámbricas tradicionales tales como Red móvil ad hoc (MANET), y la red de sistemas celulares algunas características del diseño de redes de sensores inalámbricos se presentan en (Al-Karaki and Kamal 2004).

### 2.2 Protocolos Para Redes de Dispositivos

Para los conceptos a nivel de red se han creado abstracciones a nivel de aplicación desarrollando protocolos que permiten interoperabilidad entre dispositivos, lenguajes y aplicaciones, posteriormente presentamos una relación de protocolos utilizados en dispositivos del dominio del hogar(Hoang 2007), ver Tabla 25.

Tabla 25: Protocolo para dominio Hogar

Protocolo de Interfaz	Objetivos
Lonwork, X10	Protocolos de bajo nivel son enfoque de normas eléctricas también conocido como LonTalk ANSI / EIA 709.1 para control de red eléctrica, sigue el modelo (Open Systems Interconnection - OSI). Introduce el concepto de variable de red como abstracción por ejemplo de temperatura

	o estados de un interruptor.
<b>Plug and Play (UPnP)</b>	Estándar fácil de usar y flexible a los consumidores es un conjunto de protocolos y servicios para permitir control simultáneo de electrodomésticos, permite la creación de APIs propias es independiente del sistema operativo, lenguaje y hardware. Utiliza TCP/IP. Unión dinámica de red (AutoIP) y aprende de las capacidades de los dispositivos conectados. Los servicios se describen en formato XML a través de un puente se puede comunicar con otras no (UPnP) aprovecha muchos protocolos estándares como el TCP/IP. Con servicios como (Simple Service Discovery Protocol – SSDP) localiza recursos y anuncia disponibilidad, (Generic Event Notification Architecture – GENA) envía y recibe notificaciones, (Simple Object Protocol de Acceso – SOAP) llama a procedimientos remotos.
<b>Jini</b>	Plataforma para sistemas distribuidos complejos basados en java y java (Java Remote Method Invocation - RMI) está compuesto por servicios jini, clientes jini y servicios de búsqueda jini (LUS) que es equivalente a un directorio o índice de servicios disponibles, es considera una plataforma adecuada para la integración de protocolos a bajo nivel y mecanismos de descubrimiento.
<b>OSGi</b>	Proporciona estándares para programación de dispositivos remotamente es compatible con LonWorks, Jini, UPnP sobre una puerta de enlace de servicio permite múltiples servicios, los servicios tienen una interfaz java el modelo arquitectónico asegura interoperabilidad entre middleware heterogéneos usando las APIs los usuarios cargan los servicios bajo demanda independiente del middleware proveedor. La especificación no es los suficientemente flexibles para manejar los complejos requisitos de seguridad entre módulos. Hay paquetes que se pueden descargar y uso libre por ejemplo el servicio http y el servicios registro.
<b>oBIX - Open Building Information Xchange</b>	Permite la integración de las aplicaciones empresariales utiliza estándares XML y SW servicios web y el estándar REST http es decir sus servicios son accesibles a través de la web y compatible con otra tecnologías es considerado el middleware de próxima generación para la automatización del hogar.

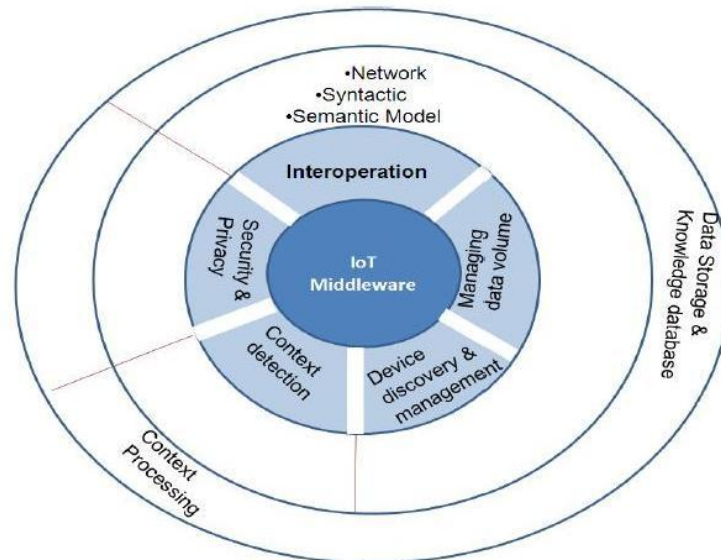
Tomado de: Middlewares for Home Monitoring and Control, (Hoang 2007)

### 3. Tipos de Taxonomía de Middleware

#### 3.1 Taxonomía de Componentes Funcionales de Middlewares

Según (Bandyopadhyay, Sengupta et al. 2011) para lograr la interoperabilidad en el IoT los middleware aportan abstracciones para ofrecer servicios múltiples, las principales funciones de los middleware se pueden clasificar en taxonomías de enfoques de adaptabilidad (adaptability), enfoques de conciencia del contexto (context-aware) y enfoques de descubrimiento de servicios (discovery service), además funciones en dominios específicos tales como WSN y RFID con bases de datos espaciales, tuplas (*tupla space*) y basado en eventos.

**Figura 18. Componentes Funcionales de Middlewares**



Tomado de: Role of middleware for internet of things: A study, (Bandyopadhyay, Sengupta et al. 2011)

En la Figura 18 se puede apreciar de adentro hacia afuera que en el primer círculo se encuentra los bloques esenciales requeridos como: interoperación, gestión del volumen de datos, descubrimiento y gestión de dispositivos, detección de contexto y seguridad y privacidad. En el segundo círculo la división presenta porciones más grandes, abarcando varios bloques funcionales básicos del primer círculo, pero orientado a los modelos sintácticos, semánticos y de red. Finalmente el círculo externo no forma parte esencial de todos los middlewares y se encuentran procesos complejos como: almacenamiento de datos (*Data Storage*) y base de datos de conocimiento (*Knowledge Database*) y procesamiento del contexto (*Context Processing*). La descripción más detallada se hace a continuación:

- **Interoperation / Interoperación:** Usa los mismos métodos de información sobre diversas interfaces de comunicación, los protocolos soportan intercambios de datos en redes diferentes sin tener en cuenta el contenido de la información. Existe interoperabilidad en la red sintáctica (network sintáctica) y red semántica (network semantic), las interacciones sintácticas son a nivel de métodos y la semántica a nivel de descripciones o significados. La interoperabilidad semántica tiene en cuenta normas de comprensión, significados y contenidos, API's de programación de servicios SOA y Ubicuos, también lenguajes para modelado de sensores como SensorML para la conexión entre diversos sensores.
- **Context Detection / Detección del Contexto:** El contexto identifica los factores de una situación. Una situación puede ser una persona, lugar y objetos, se caracteriza la situación y se genera una respuesta o decisión.
- **Context Processing / Procesamiento del Contexto:** Los identificadores de los servicios cambian dependiendo del contexto y se gestionan por medio de estructuras de datos con información de los identificadores y la extracción de la información se realiza con minería de datos.

- **Device Discovery and Management / Detección de dispositivos y gestión:** La interoperabilidad semántica y sintáctica se logra a través de la combinación de técnicas Ontológicas y Servicios Web Semánticos. Un Dispositivo Semántico es la representación física de un dispositivo almacenada en ontologías de dispositivos, el mapeo de físico a semántico incluye información del dispositivo como: nombre, proveedor, tipo, capacidades, funcionamiento y propiedades de seguridad, descripción de recursos software y hardware. En este módulo existen estrategias de descubrimiento y gestión las cuales soportan: identificación, descubrimiento, secuencias y propiedades de ciclo de vida (*life-cycle*). Utilizando Agentes es capaz de soportar roles y escenarios para descubrimiento de agentes y el uso de protocolos P2P, un ejemplo es (S-APL). Otros módulos que pertenecen a la misma categoría son el *Monitoring and Inventory* el cual soporta un repositorio de dispositivos (*Device Repository*) definiendo: las clases de dispositivos, Monitor de dispositivo (*Device Monitor*) el cual gestiona los procesos e invocación de métodos, el historial y descubrimiento (*History and Discovery*) para la correlación de eventos y lleva un historial de los eventos invocados y mensajes de depuración, el repositorio de dispositivo encuentra dispositivos y servicios a través de: un modo pasivo (escuchar y notificar) y un modo activo(búsqueda dinámica). En servicios ubicuos se tiene el descubrimiento de servicios (sw-Addressing, ws-discovery, ws-metadata exchange, ws-evening).
- **Security and Privacy / Seguridad y Privacidad:** La seguridad se maneja a alto nivel entre pares y nivel de gestión de topologías de red con autenticación, permisos de acceso y protección de información de enrutamiento, también se presenta seguridad en todos los bloques funcionales o confianza entre otros.
- **Managing Data Volume / Volúmenes de datos:** Los datos producidos en exabytes requiere de métodos de datos para buscar (Indexación), recuperar (modelo de procesos) y transferir (gestión de transacciones). Otros módulos son el módulo de gestión del almacenamiento (*Storage Manager Module*) y el módulo de apuntalamiento (*Shoring Module*) el cual recopila, filtra y finalmente el módulo de consultas (*Query Manager*).

En la Tabla 26 se pueden observar algunos middleware de la IoT clasificados teniendo en cuenta sus componentes funcionales.

**Tabla 26: Middleware por Componentes Funcionales**

Middleware	Device Management	Interoperation	Platform Portability	Context Awareness	Security and Privacy
HYDRA	✓	✓	✓	✓	✓
ISMB	✓		✓		
ASPIRE	✓		✓		
UBIWARE	✓		✓	✓	
UBISOAP	✓	✓	✓		
UBIROAD	✓	✓	✓	✓	✓
GSN	✓		✓		✓

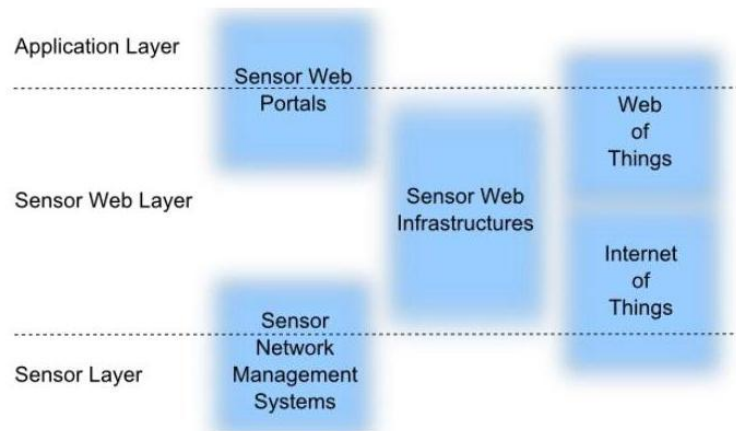
<b>SMEPP</b>	√		√	√	√
<b>SOCRADES</b>	√	√	√		√
<b>SIRENA</b>	√	√	√		√
<b>WHERE</b>	√	√	√		

### 3.2 Taxonomía de Enfoques de Middleware

Las funciones de los middlewares es poner los recursos de sensores a disposición de las aplicaciones, se puede diferenciar cuatro clases de acuerdo a su función. Las funciones pueden superponerse por eso las fronteras de la imagen están difuminados, las tres capas principales pueden tener subcapas dependiendo de la arquitectura de cada middleware. Según Bröring A. (Bröring, Echterhoff et al. 2011) en la Tomado de: New generation sensor web enablement, (Bröring, Echterhoff et al. 2011)

se presenta los cuatro enfoques que pueden abarcar varias capas.

Figura 19. Enfoques Middleware



Tomado de: New generation sensor web enablement, (Bröring, Echterhoff et al. 2011)

En la Tabla 27 se establecen las capas de la IoT:

Tabla 27: Descripción de las Capas del IoT

Capa	Descripción
------	-------------

<b>Sensor Layer / Capa sensor</b>	En la capa de sensor se encuentran los dispositivos conectados y protocolos de comunicación que pueden ser propietario o estándares por ejemplo el protocolo WPAN y el IEEE 1451.
<b>Sensor Web Layer / Capa web sensor</b>	En la capa intermedia es el puente entre los sensores y aplicaciones de sensores.
<b>Application Layer / Capa aplicación</b>	En la capa de aplicación se encuentran las interfaces de interacción con los usuarios o con las maquinas como dispositivos finales.

En la Tabla 28 se establecen los enfoques de la IoT:

Tabla 28: Descripción de los Enfoques del IoT

<b>Enfoque</b>	<b>Funciones</b>	<b>Descripción</b>
<b>Sensor network manager system / Sistema de gestión de red de sensores</b>	Sistemas de administración de red o bajo nivel a comunicación de red.	Protocolos de enrutamiento Optimización de comunicación Optimización de rutas de recolección Optimización de cobertura de las redes Localización de los sensores dentro de la red
<b>Sensor web infraestructuras / Infraestructuras Sensor Web</b>	Fabricación de sensores disponibles en la red.	Acceso a nivel de aplicación Abstracción de la red No proporciona nivel de gestión de red de sensores.
<b>Sensor web Portals / Portales web sensores</b>	APIs que presentan funcionalidades de sensores	Funciones básicas de sensores como: Registro, carga, consulta, descubrimiento. Funciones avanzadas de sensores como: Central de programación de sensores.
<b>Frameworks for (Internet of Things - IoT)/(Web of Things - WoT)</b>	(Web of Things - WoT) es como una evolución para la (IoT)	(WoT) utiliza protocolos de aplicación como (Web Services - SW). Servicios de nombres para las cosas (ONS) e identificación por códigos (RFID).

En la Tabla 29 se pueden observar algunos middleware de la IoT clasificados teniendo en cuenta los enfoques.

Tabla 29: Middleware por Enfoques

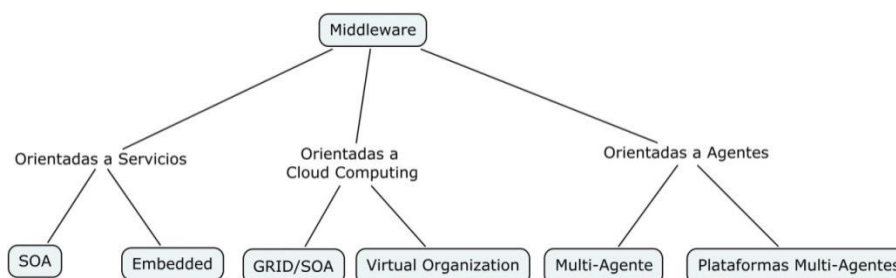
<b>Enfoque</b>	<b>Middleware y Propósito</b>
<b>Sensor Network Manager System</b>	MundoCore, Mires, Milan(Henricksen and Robinson 2006).

<b>Sensor Infraestructures</b>	<b>Web</b>	52Nort: Consultas, inserción de datos. GeoSwift: Consulta espacial. PulseNet: Reúsa 52 <sup>o</sup> Norte. Web NASA. Hourglass: Localización y Proceso. GSN(Cardoso and Sheth 2006) Consulta y filtrado, Abstracción sensor virtual, base de datos SQL. SNSP: Red de sensores e Interfaz de programación. SOCRATES: Red de sensores, acceso, descubrimiento, integración por puerto de enlace. IrisNet: Recopilación de datos. SWAP: Paradigma de Servicios y Agentes.
<b>Sensor Web Portals</b>		SensorPedia: Interfaz de Programación. Sensormap, SenseWeb, SensorBase, Xively(Broring, Maue et al. 2012; Zhou 2012), Weater Underground, Eart Cam.
<b>Frameworks (IoT)/(WoT)</b>	<b>for</b>	Optimización IP, (Web Services - SW) para las cosas, (Web of Things - WoT)(Zhou 2012), (Semantic sensor web - SSW)(Sheth, Henson et al. 2008): Mejorar el descubrimiento y la integración con anotaciones y razonamiento, sensor semántico. (Linked Open Data - LOD)(Patni, Henson et al. 2010): Uso de URIS y formato RDF.

### 3.3 Taxonomía de Arquitecturas Middleware

Según Pileggi S. F. (Sangers, Frasinca et al. 2013) se presenta una clasificación para las redes de sensores inalámbricos dependiendo de la arquitectura, adaptamos esta clasificación a la IoT y presentamos todos los middleware que cumplen esta taxonomía, en la Figura 20 se observa.

Figura 20. Arquitecturas Middleware



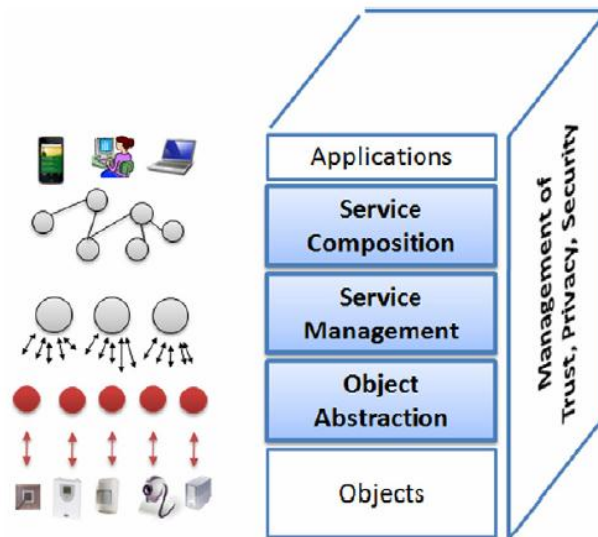
A continuación se detalla las arquitecturas de middleware más importantes para la IoT y sus características generales.

#### 3.3.1 Orientada a Servicios (Services Oriented Architecture – SOA)

Es una de las arquitecturas que han adoptado algunos estudios como en(Atzori, Iera et al. 2010). Esta arquitectura descompone servicios complejos y monolíticos en aplicaciones simples y por componentes con interfaces comunes. El proceso de negocio requiere lenguajes de ejecución de

procesos de negocio (*Business Process Execution Language* - BPEL) para crear flujos de trabajo y servicios coordinados que apoyan las interacciones entre los componentes agilizando la adaptación de las aplicaciones a los cambios del mercado, la interacción con las entidades externas requiere de lenguajes como lenguajes de descripción de servicios web (*Web Services Description Language* - WSDL) para llamar flujos de trabajo desde el interior de otros flujos de trabajo es decir anidados, todo esto facilita el desarrollo y la comunicación con otras aplicaciones. Ver **Figura 21**

. **Figura 21. Orientado a SOA**



Tomado de: The Internet of Things: A survey, (Atzori, Iera et al. 2010)

Las características de cada capa son:

- **Applications / Aplicaciones:** No hace parte del middleware sino que aprovecha de todas las funcionalidades prestadas por el middleware, estas funciones se utilizan por medio de protocolos estándar de servicio y tecnologías de composición de servicios.
- **Service Composition / Composición de servicios:** Crea un entorno de servicio con un repositorio de instancias de servicio de los objetos conectados, construyen servicios complejos por composición de servicios, los cuales soportan procesos complejos en secuencias de acciones que son coordinadas por componentes software individual.
- **Service Management / Gestión de servicios:** Ofrecen funciones a la capa superior, funciones para los objetos como descubrimiento, monitoreo, configuración, gestión de calidad (QoS) de tolerancia a fallos, gestión de bloqueo y funciones semánticas como administración del contexto.
- **Object Abstraction / Objeto Abstracto:** Los objetos heterogéneos implementa funciones en su propio lenguaje por tal motivo es necesaria una abstracción por medio de una capa

de envoltura que implemente una capa de interfaces y otra de comunicación, la capa de interfaz web implementa (operaciones de mensajería - outcoming) y la capa de comunicación es la lógica o los comandos para comunicarse con los objetos por ejemplo las pilas TCP/IP como TinyTCP, MIP, lwIP y los servidores web embebidos que son una abstracción de los objetos.

- **Management of Trust, Privacy, Security / Gestión de confianza, privacidad, seguridad:** La comunicación automática implica un peligro inherente de seguridad la cual requiere de mecanismos de vigilancia por medio de middleware que implementen servicios seguros para el IoT. El papel de la seguridad es primordial para el desarrollo comercial de IoT un ejemplo el (Fosstrak) que implementa tecnologías RFID que establece aplicaciones destino, un conjunto específico de objetos y pone limitaciones geográficas, otro ejemplo pero que no tiene arquitectura SOA como (e-SENSE) que implementa tecnología de (Web Sensor Network - WSN y establece protocolos de control de entidades, un último ejemplo es el Proyecto (UbiSec & Sens) también WSN que implementa mecanismos de seguridad como largo plazo de registro de datos con (Tiny PEDS), abstracción de memoria compartida con (Tiny DSM) y almacenamiento de memoria distribuida con el protocolo (DISC) para WSN.

### 3.3.2 Orientada a Servicios (Middleware Embebidos – Embedded)

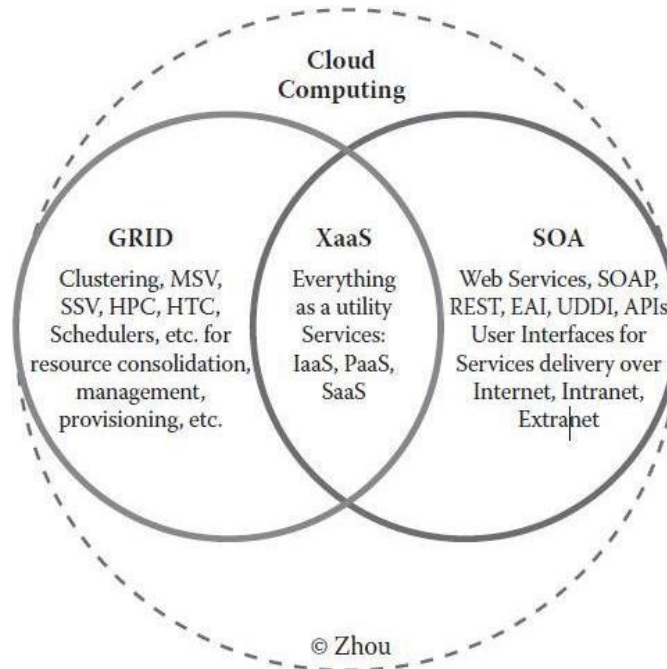
Son capas de software entre los sistemas operativos y las aplicaciones, tales como CORBA, EJB y DCOM. El objetivo es crear una abstracción del sistema operativo a las aplicaciones a través de servicios bien definidos. Adicionalmente hacen un desarrollo especial en mecanismos de comunicación. También se pueden encontrar subtipos de middleware embebidos como: orientados a mensajes, orientados a objetos, basados en llamadas a procedimientos remotos, de base de datos y de transacciones.

### 3.3.3 Orientada a Cloud Computing (Grid/SOA)

Según Zhou, H. (Zhou 2012) es la fusión de las arquitecturas SOA (SOAP, REST) y Grid Computing, el nombre Grid es una analogía de las redes eléctricas debido a consistencia (consistent), Omnipresente (permasive), Confiable (dependable) y acceso transparente. El Grid Computing y la tecnología de Grupo de estaciones de trabajo son los antepasados de Cloud Computing, las granjas de servidores de Google se basa también en esta filosofía. La IoT se parece a Cloud Computing en que están constituidas por sistemas distribuidos de redes de comunicaciones los servicios de Cloud Computing, esta no requiere que los usuarios conozcan la ubicación física de los datos y configuraciones del sistema distribuido.

El Cloud Computing encierra los siguientes conceptos de servicios: Software como servicio (*Software as Service* - SaaS), Plataforma como servicio (*Platform as Service* - Paas) e Infraestructura como servicio (*Infrastructure as Service* - IaaS). Ver **Figura 22**.

Figura 22. Orientado a Grid/SOA



Tomado de: Cloud Computing, (Zhou 2012)

El Cloud Computing permitirá crear una gran red de objetos a nivel mundial para crear una arquitectura de Nube de las Cosas (*Cloud of Things*) que está en proceso de definición y especificación, esta especificación debe tener en cuenta los siguientes aspectos según Zhou, H.(Zhou 2012), ver Tabla 30.

Tabla 30: Cloud of Things

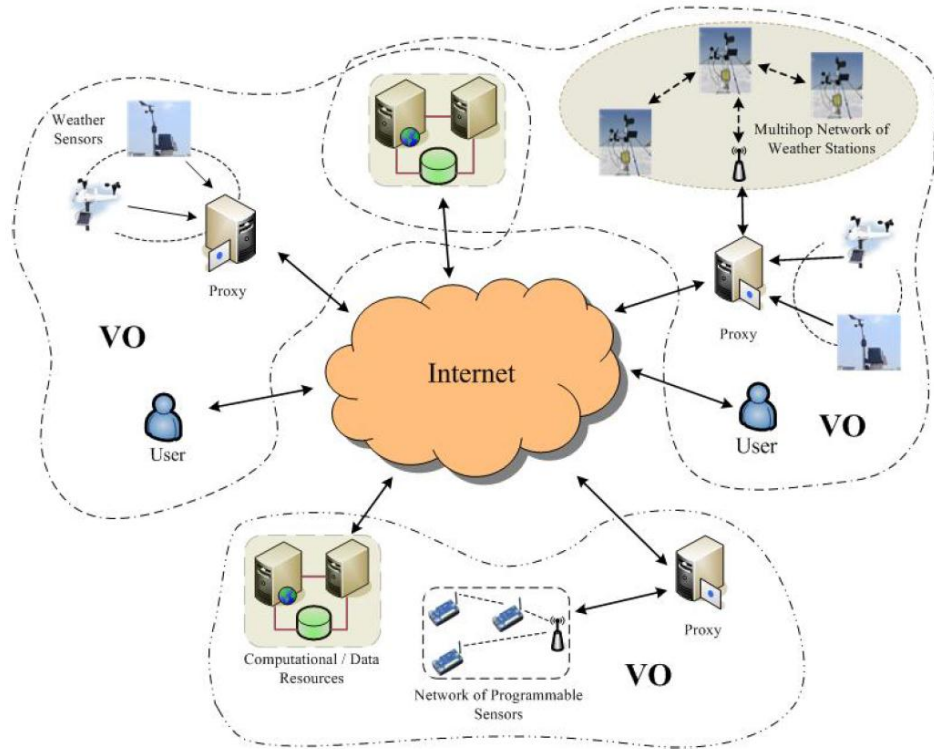
Aspectos	Descripción
<b>Conectivity / Conectividad</b>	Conexión de objetos móviles o restringidos.
<b>Content / Contenido</b>	Datos masivos producidos por los objetos.
<b>Cloud / Nube</b>	Servicios de la nube.
<b>Context / Contexto</b>	Sensible al contexto para mejorar el rendimiento.
<b>Collaboration / Colaboración</b>	Comunicaciones cooperativas e intercambio de servicios.
<b>Cognition / Cognición</b>	Minería de conocimiento y mejora de sistemas autónomos.

### 3.3.4 Orientada a Cloud Computing (Virtual Organization – VO)

Según B. Hock (Hock Beng, Iqbal et al. 2009) define que los portales de despliegue de sensores y la infraestructura de alojamiento grid son considerados como (Organizaciones virtuales - VO) y se caracterizan por tener diferentes proveedores, diferentes plataformas, conectividad de red variada y nodos de sensores programables, como ejemplos tenemos: SensorScope, Crossbow Motes, el proyecto estudio nacional de metodología (*National Weather Study Project - NWSP*) con

estaciones meteorológicas en Singapur. En la Figura 23 se presentan los tipos de middleware de este tipo de arquitectura.

**Figura 23. Organizaciones Virtuales**



Tomado de: A Large-Scale Service-Oriented Sensor Grid Infrastructure, (Hock Beng, Iqbal et al. 2009)

Proporciona servicios orientados a integrar servicios y compartir recursos entre diferentes (VO) proporciona interfaces de acceso web y sockets, servicios de composición de sensor y aplicación, también autenticación de usuarios. Los middleware implementan arquitecturas SOA y ontologías para sensores. Los componentes de diseño del modelo (VO) ver Tabla 31:

**Tabla 31: Middleware por Capas VO**

Capas	Descripción
<b>Capa Plataforma</b>	Programación de nodos existen productos comerciales como Crossbow Motes y Gumstix.
<b>Capa Software</b>	Gestiona las capacidades de detección, almacenamiento, encaminamiento utiliza fireware como el entorno de programación middleware Tiny OS.

<b>Capa Semántica</b>	Compuesto por ontología de datos: Proporcionan semántica a los datos. Ontología de procesos: Proporciona interfaces orientados a servicios de los componentes de la capa de software.
-----------------------	---

### 3.3.5 Orientada a Agentes (Multi-Agente)

La (IoT) es una red de objetos a los cuales se le han proporcionado capacidades de detección, procesamiento, almacenamiento y comunicación a estos objetos se les denomina Objetos Inteligentes (Smart Object - SO) los cuales tienen el propósito de ser autónomos, los middleware desarrollados con la arquitectura de agentes se ajusta perfectamente al concepto de (SO) en la Tabla se muestran los aspectos en que se ajustan las agentes y (SO) según Fortino G (Fortino, Guerrieri et al. 2012) como se muestra en la Tabla 32.

**Tabla 32: Aspectos de concordancia entre Agentes y SO**

Aspectos	Descripciones
<b>Autonomy / Autonomía</b>	Solución de problemas sin intervención humana o de otros agentes, controla las acciones y estados internos.
<b>Social ability / Habilidades sociales</b>	Interactúa cuando se requiera con humanos u otros agentes para ayudar a resolver problemas.
<b>Responsiveness / Capacidad de respuesta</b>	Al percibir el mundo físico, usuarios, otros agentes y responde de manera oportuna a los cambios.
<b>Proactiveness / Proactividad</b>	Es oportunista es decir no se limita a responder al entorno sino que toma la iniciativa por ejemplo dirigir una conducta a un objeto.

La alianza (IP Smart Objects - IPSO)(IPSO 2013) determina middleware para dominios determinados de aplicación tales como (Smart SO) para conciencia del contexto, (2WEAR) portátiles SO, middleware para arquitecturas preinstaladas de servicios como (Aura), (Gaia), (IRoom). Middleware con arquitecturas más generales como (UbiComp GAS) y (FedNet) centrado en datos.

### 3.3.6 Orientada a Agentes (Plataformas Multi-Agente)

La arquitectura orientada a agentes establece un nodo coordinador el cual puede ser dispositivos con mayores capacidades como un PC, Smartphone el cual pueda realizar el proceso solicitado por nodos internos de menores capacidades como sensores y actuadores con el protocolo (Internal Smart Object Protocol - ISPO), el nodo coordinador establece comunicación con nodos estratégicos en la red que pueden ser remotos con el protocolo (External Smart Object Protocol). Los middleware facilitan la interacción entre los actores externos como dispositivos, usuarios y servicios de internet proporcionando servicios de descubrimiento de tipo de objetos, servicios ofrecidos, ubicación y atributos estáticos y dinámicos. En la Tabla 33 se muestran las capas con enfoque de agentes.

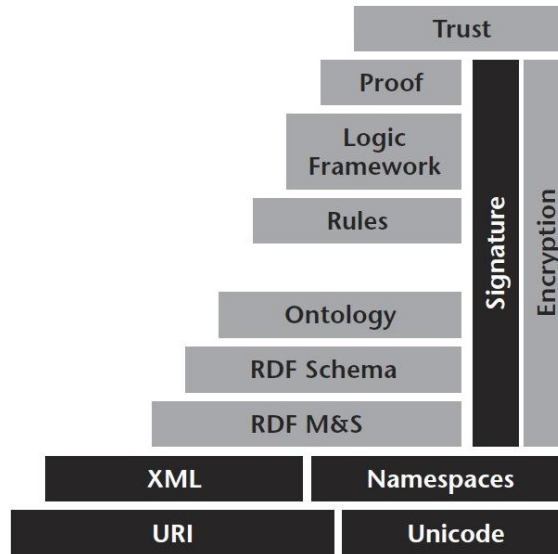
**Tabla 33: Middleware por Plataformas Multi-Agente**

Capa	Objetivos Principales	Descripción
<b>Arquitectura de Alto nivel</b>	Gestor de entrada /salida	Recibe y transmite desde y para el mundo exterior datos de comando de comunicación e interrupciones.
	Gestor de comunicación	Comunicación de (SO) con el mundo exterior a través de middleware.
	Comunicación middleware	Comunicaciones con objetos coordinadores remotos.
	Módulo de descubrimiento	Servicios de descubrimiento de (SOs).
	Adaptador de base de conocimiento	Modificación y consulta de bases de conocimiento.
	Gestor de contexto	Interpretación de los datos de usuarios, (SO), internet según el contexto donde se encuentran.
	Razonador	Inferencias a la base de conocimiento y proactivo es decir que tareas ejecutar.
	Adaptador sensor y actuador	Abstracciones de sensores y actuadores, oculta el tipo y la ubicación.
<b>Middleware de Agentes</b>	Externos sensor actuador	No pertenecen al (SO).
	JADE Plataformas coordinador	Arquitecturas de alto nivel middleware derivados JADEX, JADEX Android, JADE LEAP.
	Mobile Agent Platform for SunSPOT – MAPAS	Descubrimiento dinámico de servicios de agente búsqueda de (SOs) con metadatos que contienen tipo, servicios, ubicación, propiedades hardware etc..
<b>Gestión WSAN</b>	Building Management Framework – BMF	Presenta soporte multiplataforma, interfaz de programación, rápida configuración, algoritmos de procesamiento, abstracciones dinámicas. (Signal Processing In-Node Environment - SPINE) propone bibliotecas de alto nivel en java para emitir nodos y solicitar servicios de sensores.

### 3.4 Tecnologías de la Web Semántica

En la Figura Daconta M.(Daconta and Obrst 2003) explica las capas de la web semántica en la cual se encuentran las tecnologías que la hicieron realidad.

Figura 24. Arquitectura de la Web Semántica



Tomado de: The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, (Daconta and Obrst 2003)

En la Tabla 34 se describen las tecnologías de la web semántica.

Tabla 34: Descripción de Tecnologías de la Web Semántica

Tecnología	Descripción
XML	Es un lenguaje de definición de marcado que define los nombres legales de los elementos y atributos y la estructura legal del documento para que sea legible y comprensible para todos. XML no es en realidad un lenguaje sino un conjunto de sintaxis para crear semánticamente ricos lenguajes de marcado en un dominio particular. La razón por la que XML tiene tanto éxito es porque ha logrado 4 aspectos principales. Primero: Crea documentos de datos independientes de las aplicaciones. Segundo: Tiene sintaxis estándar para metadatos. Tercero: Tiene una estructura estándar para documentos y datos. Cuarto: NO es una tecnología nueva, esta última es la más importante ya se lleva 20 años trabajando con XML y en este momento las capacidades computacionales son lo suficientemente baratas para utilizarlo.
RDF	Es utilizado para crear metadatos sobre un recurso como una entidad independiente es decir otorga una identidad al recurso. Se compone de Sujeto: En gramática es el sintagma nominal o sustantivo el cual realiza la acción por ejemplo en la frase "la compañía vende baterías" el sujeto es la empresa para el cual le asignamos un URI. Predicado: En la gramática es la parte de la oración que modifica el tema y es la relación entre el sujeto y el objeto, la parte de la frase "vende baterías" se le asigna un URI único para el concepto "vende". Objeto: En la lógica se actúa sobre el por medio del predicado, en muestra frase la palabra "baterías" recibe un URI. Declaración: Es la combinación de Sujeto-Predicado-Objeto son una sola entidad. Si las palabras tienen deferente significado dependiendo en las frases en que se encuentren al darle un identificador único a

---

cada significado se puede hacer referencia a cada significado independientemente.

**RDFS** Es un lenguajes en capas y pilar de la web semántica; en la capa base se encuentra el concepto de **URI** y el conjunto de caracteres universales **UNICODE**, en la siguiente capa la sintaxis XML como elementos, atributos y corchetes angulares, esta capa contiene los espacios de nombres para evitar conflictos de vocabulario. En la siguiente capa las afirmaciones basadas en Triples RDF, si se utiliza un triple para definir una clase y sus propiedades y valores se puede crear jerarquías y clasificaciones de objetos; este es el objetivo de RDFS. RDFS es un conjunto de recursos y propiedades estándar RDF para permitir crear vocabularios propios, cuyo modelo de datos es igual a modelo orientado a objetos. Las capas se pueden observar en la Figura 25.

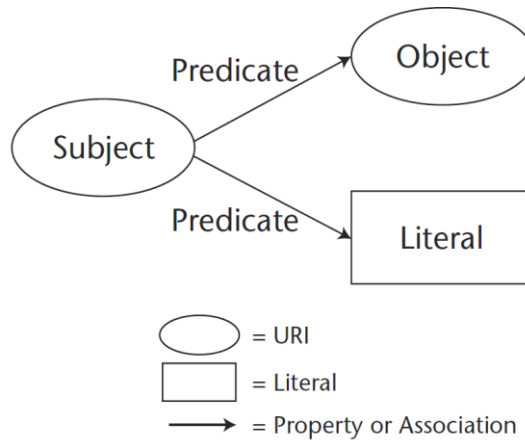
**Ontologías** Cuando los datos son suficientemente inteligentes son descritos por relaciones concretas y formalismos sofisticados donde se pueden hacer cálculos lógicos realizando análisis atómico o más detallado de los datos un ejemplo puede ser la traducción automática de un documento en un dominio a un documento equivalente tan cerca como sea posible a otro dominio. Este tipo de datos inteligentes se pueden definir como datos independientes de la aplicación que los procesa que pertenecen a un conjunto de datos más general. El desarrollo de ontologías se realiza por medio de lenguajes robustos como **OWL**. Por encima de la capa de las ontologías encontramos la Reglas: Son reglas lógicas sobre las ontologías, un lenguaje de reglas nos permite inferir nuevos conocimientos y tomar decisiones. Marco Lógico: El marco lógico es la lógica avanzada, permite pruebas lógicas que serán compartidas. Confianza: El enfoque colaborativo establece la confianza con firmas XML y cifrado XML. Estos componentes se pueden observar en la Figura 25.

---

Tomado de: The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, (Daconta and Obrst 2003)

Presenta un diagrama de los elementos que componen a un triple RDF:

Figura 25. Triple RDF



Tomado de: The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, (Daconta and Obrst 2003)

### 3.5 Aplicaciones de Búsqueda en la IoT

#### 3.5.1 Tipos de datos en el Internet de Objetos IoT

Los modelos de representación del conocimiento y los tipos de datos (James and Cooper 2009) ver Tabla 35 provenientes de las entidades del mundo real son factores que deben ser tenidos en cuenta para implementar aplicaciones que conformen servicios en distintas áreas. Los datos continuos y discretos se pueden definir en distintas clases y hacen parte de las transacciones de datos en el IoT.

Tabla 35: Tipos de datos del IoT

Tipos de Datos	Descripción
<b>Identificación por Radio Frecuencia</b>	Las etiquetas RFID emiten y reciben señales, los datos tienen como objetivo la identificación y seguimiento de los objetos del IoT, las ondas de radio superan obstáculos físicos en el proceso de realizar la lectura de las etiquetas, evitando el contacto directo de otros dispositivos como los infrarrojos.
<b>Direcciones y Identificadores únicos</b>	- El IoT presenta la necesidad de un enorme aumento en direcciones disponibles para conectar lugares, cosas y personas por tal motivo es idóneo el protocolo de direccionamientos IPv6, existen otras formas de identificación como los ( <i>Identificadores Universales Únicos – UUID</i> ) que son utilizadas por aplicaciones basadas en Bluetooth y otras tecnologías.
<b>Datos descriptivos acerca de los Objetos, Procesos y Sistemas</b>	El metadato es una forma auto descriptiva para almacenar, representar y validar los datos para garantizar la máxima eficiencia y no repudio en la recuperación y actualización de la información, por ejemplo gato, negro, Fluffy sus metadatos podrían ser animal, color, nombre. Los metadatos son datos acerca de los datos, procesos y sistemas. Los metadatos requieren de modelo de representación de la información adecuada.

<b>Datos Posicionales y Datos generalizados del Medio Ambiente</b>	El objetivo de los datos posicionales es dar la ubicación de los objetos etiquetados dentro de un sistemas de posicionamiento Global o Local, por ejemplo estaciones GPS o celulares, Wi-Fi, los datos de posición pueden venir de sensores y de transmisores que envíen señales a dispositivos inteligentes los cuales resuelven su ubicación, ya que los objetos físicos pueden ser estáticos o móviles. La aparición de un nuevo concepto es el <i>Internet de los Sitios Internet of Places</i> donde la información es dependiente del sitio de donde se accede o se produzca.
<b>Datos de Sensores - Datos de Serie de Tiempo Multidimensional</b>	El IoT utiliza Redes de Sensores Inalámbricos por ejemplo Zigbee, para el seguimiento de fenómenos ambientales, en la captura aparecen cuestiones como la frecuencia y cantidad de datos almacenados para constituir una muestra representativa del evento estudiado. La consulta y exploración es problemática debido a que es información en tiempo real.
<b>Datos Históricos</b>	Los datos de los sensores llegan a las cantidades de los Petabytes a medida que pasa el tiempo, las aplicaciones deben solucionar problemas acerca de cómo y qué datos almacenar, que pueden ser en almacenes de datos activos para consultas frecuentes y también en estructuras de datos menos accesibles.
<b>Modelos Físicos - Plantillas la Realidad</b>	Son modelos de representación que permiten la simulación de escenarios físicos, son plantillas de la realidad por ejemplo de gravedad, fuerza, luz, sonido y magnetismo.
<b>Estado de Actuadores para datos y Comandos de Control</b>	La IoT se utiliza para controlar dispositivos remotos, los avances de la miniaturización electrónica, reducido consumo de energía en microcomputadoras, y la mejora en micro baterías han sido necesarios para el aumento de los actuadores en la IoT. Este tipo de datos es considerado como comandos para el control de dispositivos un ejemplo es emitir un comando para que se encienda la calefacción 30 minutos antes de llegar a casa.

### 3.5.2 Dimensiones en el espacio de diseño para aplicaciones en el IoT

La búsqueda de entidades del Internet de Objetos IoT es más difícil debido a que estas entidades presentan diferentes estados en el tiempo es decir *Estado Dinámico*, el descubrimiento de estas entidades abarca distintas dimensiones de diseño (Romer, Ostermaier et al. 2010) ver Tabla 36.

Tabla 36: Dimensiones de Diseño para aplicaciones en la IoT

Dimensión	Descripciones
<b>Tipo de consulta / Query Type</b>	Se clasifican en consultas <i>Ad Hoc</i> que son las que se ejecutan solo en el momento en el que se realiza la consulta, y las consultas <i>Continuas</i> que se encuentran activas por un periodo de tiempo.
<b>Lenguaje de Consulta / Query Language</b>	Lenguajes basados en palabras claves que el usuario proporciona, y lenguajes complejos que soporta predicados más grandes que describe el estado de las entidades.
<b>Ámbito de Consulta /</b>	Los motores de consulta están diseñados para dominios locales y otros globales que son escalables para el IoT.

<b>Query Scope</b>	
<b>Plazo consultas / Query Time</b>	<b>de</b> Hace referencia al <i>estado en tiempo real</i> de las entidades y al <i>estado histórico</i> que se presentan en un intervalo de tiempo.
<b>Precisión Consulta / Query Accuracy</b>	<b>de</b> Para hacer coincidir las consultas con las entidades se utilizan algoritmos <i>Exactos</i> , también algoritmos <i>Heurísticos</i> donde la coincidencia no es perfecta entre entidades y consulta.
<b>Consulta Contenido / Query Content</b>	<b>de</b> Está relacionado con la velocidad de cambio de los datos, se clasifica en contenido <i>Estático</i> por ejemplo el tipo de entidad, contenido <i>Pseudo estático</i> por ejemplo los metadatos del propietario, y por último el contenido <i>Dinámico</i> por ejemplo los datos de sensores que describen el estado de entidad, una subclase apropiada es el cambio poco a poco del estado dinámico.
<b>Movilidad Entidad / Entity Mobility</b>	Con respecto a la ubicación de las entidades los sistemas pueden asumir que las entidades son estáticas si no cambian de lugar y móviles si no permanecen en el mismo sitio.
<b>Estado Entidad / Entity State</b>	El estado se puede representar por un conjunto <i>finito</i> o <i>discreto</i> que se pueda clasificar en categorías por ejemplo el conjunto caliente, frío o también se representa el estado por un valor perteneciente a una escala continua por ejemplo grados Celsius.
<b>Los usuarios afectados / Target Users</b>	Existen aplicaciones que utilizan sistemas de descubrimiento diseñados por expertos en el dominio en los cuales el usuario final no participa en el proceso de descubrimiento de servicios, y sistemas que para uso directo como por ejemplo los motores de búsqueda en la web.

### 3.5.3 Técnicas y Estrategias fundamentales para el descubrimiento de entidades

Un aspecto importante de los sistemas en tiempo real del IoT son las técnicas y estrategias para el descubrimiento (Romer, Ostermaier et al. 2010) de entidades, donde los datos consultados describen los estados de las entidades dinámicas. Ver Tabla 37.

Tabla 37: Técnicas para el Descubrimiento de entidades

Estrategia de Descubrimiento	Descripción
<b>Empujar y Tirar / Push and Pull</b>	Esta técnica almacena en un mismo lugar los datos y las consultas hechas en un sistema distribuido para realizar las consultas de una forma local, utilizando entidades de avance de datos hacia los usuarios, las entidades almacenan los datos que luego son extraídos por los mediadores, los datos más consultados son empujados hacia los usuarios y los menos consultados son retirados.
<b>Publicar-Suscribir / Publish-</b>	Es una técnica donde los usuarios se suscriben a ciertas entidades, es muy útil cuando se realizan consultas continuas ya que existen relaciones

---

<b>Subscribe</b>		explicitas entre usuarios y entidades.
<b>Mediadores / Mediators</b>	/	Técnica que introduce un elemento conceptual entre entidades y usuarios, un mediador puede responder a consultas sin necesidad de extraer los datos de las entidades debido a que crean una vista agregada de entidades. Se pueden implementar en un sistema centralizado o distribuido, puede haber jerarquías de mediadores donde el Super mediador tiene una vista global de los Sub mediadores.
<b>Índice Invertido / Inverted Index</b>	/	Esta técnica es capaz de seguir de manera eficiente entidades que presentan un estado específico y son requeridas en una consulta, es utilizada por los mediadores para crear una vista de conjunto de las entidades.
<b>Compresión / Compression</b>	/	Reduce la cantidad de datos para almacenar y comunicar en las consultas por ejemplo los mediadores puede crear una compresión de la vista agregada de entidades. La compresión <i>Sin Pérdida</i> no se afecta el funcionamiento del sistema y la compresión <i>Con Pérdida</i> es un resultado aproximado de las consultas y sirve para establecer un subconjunto de entidades o usuarios para posterior empuje o tire. Un ejemplo extremo de compresión con pérdida es comunicar o almacenar un solo bit para identificar la entidad que ha cambiado de estado y así realizar operaciones de extracción sobre estas entidades.
<b>Modelos / Models</b>		El objetivo de los modelos es <i>deducir información</i> de una entidad o usuario, la deducción no requiere comunicación con la entidad o usuario, se crean modelos del estado de una entidad basándose en sus estados anteriores por ejemplo modelos estadísticos del estado de las entidades. Se pueden utilizar para la resolución heurística de consultas donde las entidades tienen modelos para establecer que usuarios estarían interesados en los datos. Otra manera de uso es la identificación de entidades para posterior empuje o tire.
<b>Puntuación y Clasificación / Scoring and Ranking</b>	y / and	Se trata de dar un valor escalar a las entidades, este valor es proporcional a su relevancia en la consulta, estas entidades se clasifican en base a la puntuación obtenida. El objetivo de la puntuación y clasificación es presentar las entidades más relevantes, aumentar la eficiencia en el proceso de empuje y tire, comparación de entidades normalizando las puntuaciones por último la ponderación de palabras claves respecto a la frecuencia de aparición.
<b>Top-k Consulta / Top-k Query</b>	/	Es una alternativa a la resolución de consultas por <i>Bruteforce</i> que compara el conjunto completo de entidades, Top-k establece un subconjunto de las k entidades más importantes que coincidan, las cuales a menudo son suficientes para responder a la consulta.

---

### 3.6 Servicios para construir aplicaciones del IoT

A la hora de desarrollar aplicaciones para la IoT nos encontramos con el inconveniente de la interoperabilidad ya que la IoT está constituida por una gran variedad de dispositivos y redes de

comunicaciones la cual no está representada por una arquitectura común. La construcción de aplicaciones sobre servicios disponibles (Gigli and S. 2011) en el IoT constituye un apoyo al desarrollo e implementación y la solución ha sido las arquitecturas orientadas a servicios. Ver Tabla 38.

**Tabla 38: Servicio para construir aplicaciones del IoT**

Servicios	Descripción
<b>Relacionados con la Identidad</b>	Servicios que proporcionan información acerca de recursos, la tecnología más usada es la RFID, compuesto por una etiqueta con información codificada y un dispositivo de lectura el cual realiza una solicitud al servidor para acceder a información detallada del recurso. Existen dos categorías los servicios activos y pasivos, el activo tiene una fuente de energía y envía señales y el pasivo no tiene baterías y requiere de un campo electromagnético externo para emitir la señal. Los servicios de identidad son comunes en el transporte marítimo, la producción y la cadena de suministros.
<b>Agregación de Información</b>	Servicio encargado de la adquisición y transmisión de datos de varios sensores, las pasarelas de acceso access gateways comparte información de distintos tipos de sensores por ejemplo etiquetas RFID y la transmisión de datos de sensores a través de una red ZigBee los cuales pueden ser ofrecidos a través de un servicio web JSON y XML . El servicio de agregación se incorpora a través redes inalámbricas de sensores WSN el cual se puede extiende a través de la red celular. Los servicios de agregación son útiles en seguimiento de fenómenos naturales o eventos en la empresa o el hogar.
<b>Colaboración Cociente</b>	Servicios que utilizan información agregada para tomar decisiones y realizar acciones. Este servicio se apoya en la seguridad de la red, la velocidad y la potencia del procesamiento de las terminales, además de recogerse la información debe haber dispositivos en la red para procesar de los datos, por esta razón el IPv6 se conviene como una tecnología usada ya que amplía el número de direcciones para dispositivos. Existen pocas aplicaciones y servicios disponibles de colaboración cociente en el IoT.
<b>Ubicuos</b>	Servicios más especializados de la IoT por que no solo es un servicio de colaboración cociente por que debe relacionar a todos los nodos de la red todo instante, para eso se necesita superar la barrera de distintos protocolos y tecnologías, se espera acceso y control completo de todo lo que nos rodea. No se han logrado implementar pero se requiere de una arquitectura única para integrar todas las tecnologías en una aplicación para poder comunicarse e interoperar.

### 3.7 Lenguajes para construir Servicios Web Semánticos

#### 3.7.1 OWL-S

Es el producto de la investigación de DAML y OIL que inicialmente se llamó DAML-S pero posteriormente para hacer referencia al lenguaje de origen OWL

cambio de nombre por OWL-S, debido a algunos efectos secundarios no deseados de OWL-S se creó una variante del mismo llamado OWL-DL que impone restricciones de razonamiento. El modelo conceptual de OWL-S se divide en tres sub-ontologías:

- **Service Profile:** Es la funcionalidad del servicio incluyendo limitaciones y propiedades no funcionales y proporciona un medio para exponerlo. En OWL-S no distingue entre petición del servicio y provisión del servicio lo que lleva utilizar para Service Profile para publicar como proveedor y buscar como solicitante. Esta sub-ontología establece entradas y salidas del servicio, condiciones previas a la ejecución, efectos y condiciones no funcionales del servicio.
- **Service Model:** Define el comportamiento del servicio presentando un flujo de trabajo en donde el solicitante proporciona los requisitos para la ejecución del servicio. Esta sub-ontología permite la descripción de diferentes tipos de servicios de procesos Atómicos, Abstractos y Compuestos.
- **Service Grounding:** Proporciona un enlace entre el Service Model y la descripción del servicio proporcionada por WSDL, asignando procesos atómicos a las entradas y salidas de las operaciones en el WSDL, para asignar procesos de composición se requiere de un motor de procesos OWL-S para interpretar el control de flujo de las interacciones definidas.

Los lenguajes semánticos para condiciones previas y resultados del servicio en el Service Profile se pueden usar el Semantic Web Rules Language - SWRL para escribir las expresiones lógicas, RDF para el tratamiento de expresiones como literales y por último Knowledge Interchange Format - KIF o el Planning Domain Description Language - PDDL para tratar expresiones como cadenas.

### 3.7.2 SWSF

Se crea gracias a la necesidad de describir no solo a los servicios web sino que también el modelo de procesos de servicios, ya que OWL-S no es adecuado para esta tarea. El modelo conceptual se divide en dos partes. Primera: (Semantic Web Service Ontology - SWSO) que usa la lógica de primer orden. Segunda: (Semantic Web Service Language – SWSL).

Los axiomas de lógica de primer orden en SWSO se denominan FLOWS o flujos, estos se basan en el (Process Specification Language - PSL) que es una ontología normalizada internacional que describe procesos en el ámbito de los negocios, la ingeniería y la fabricación, su objetivo era proporcionar interoperabilidad entre los

muchos lenguajes de procesos. Los flujos se expresan en (SWSL for First Orden Logic – SWSL-FOL), e implementar la programación lógica en SWSO, existe una segunda ontología llamada (Rules Ontology for Web Service - ROWS) análogamente se expresa con el lenguaje SWSL-Rules. Los tres conceptos principales de SWSO se derivan de OWL-S son los siguientes:

- **Service Description:** Conjunto de elementos no funcionales los cuales constituyen elementos simples cuando se especifica un flujo, como el nombre el autor y la descripción textual del servicio.
- **Process Model:** El objetivo de PSL era de proveer la expresividad ontológica y de lenguaje para representar todos los lenguajes de procesos, esto hace realidad la integración de procesos independientemente del modelo de procesos en que fueron creados. FLOWS extiende a PSL describiendo las limitaciones y sub actividades del servicio.
- **Grounding:** Es muy parecido al de OWL-S v1.1 para WSDL; en SWSO se establecen cuatro cosas. Las dos primeras son las asignaciones entre patrones de mensajes y tipos de mensajes entre SWSO y WSDL, y las dos últimas serialización y deserialización entre tipos de mensajes.

### 3.7.3 WSDL-S

A diferencia de los modelos anteriores WSDL-S no es una ontología para SWS sino que hace una descripción de datos y operaciones de servicio utilizando conceptos ontológicos a través de anotaciones semánticas. Las anotaciones se realizan con XML y extienden a WSDL por medio de un modelo conceptual de cinco elementos:

- **modelReference:** Este atributo establece una correlación uno a uno entre el un tipo de esquema XML y un concepto ontológico.
- **SchemaMapping:** Donde no hay correlación uno a uno este atributo señala la transformación entre el elemento del esquema XML con el concepto de la ontología. Un ejemplo puede ser que el valor del atributo schemaMapping es una URI que identifica una transformación XSLT.
- **Precondition:** En una operación WSDL solo acepta una precondición, esta apunta a un conjunto de expresiones lógicas en un lenguaje ontológico.
- **Effect:** A diferencia de las precondiciones en una operación WSDL puede contener sub-elementos de efecto.

- **Category:** Es una extensión de la interfaz de WSDL y el objetivo es que una categoría pueda ser recogida por un UDDI.

### 3.7.4 SAWSDL

Son anotaciones semánticas basadas en una forma simplificada de WSDL-S, los elementos de los documentos WSDL son anotados con atributos por las extensiones válidas para el estándar WSDL. SAWSDL limita sus atributos a SchemaMapping, ListingSchemaMapping y LoweringSchemaMapping y utiliza el atributo ModelReference en anotaciones XSD.

### 3.7.5 WSMO, WSML Y WSMX

Tiene un modelo conceptual más complejo, plantea modelos separados entre Metas y Servicios, los elementos de WSMO resuelve el problema de la heterogeneidad. Una diferencia con WSMO con OWL-S es que OWL-S no modela mediadores sino que los considera como un tipo de servicios específico. Los elementos de WSMO están identificados por un URI con espacios de nombre serializables en XML. Algunas características son:

- **Basado en ontologías:** Usa ontologías para modelar cada elemento de WSMO.
- **Desacoplamiento estricto:** Los recursos WSMO se define aisladamente uno de otro, NO supone que todos los recursos se deben definir utilizando la misma ontología.
- **Fuerte mediación:** Es posible gracias a al desarrollo de la mediación. Los mediadores son elementos de modelado de alto nivel que se utilizan para interoperabilidad entre recursos heterogéneos.
- **Separación Ontológica de Roles:** Los puntos de vista de solicitante y proveedor son claramente representados sobre los conceptos de las Metas y Servicio.
- **Servicio vs Servicio Web:** El servicio es definido en WSMO como un súper conjunto de servicios web esto quiere decir que WSMO está diseñado para abarcar todos los tipos de servicios que están disponibles en la web.

**WSML:** Es un lenguaje que presenta capas para proporcionar diferentes niveles de expresividad semántica en WSMO dependiendo de los requisitos de razonamiento.

**WSMX:** Es un middleware diseñado para anotaciones semánticas en WSMO, WSMX proporciona un enfoque para el descubrimiento, composición automática, mediación e invocación de SWS existen otras herramientas basados en WSMO pero ninguna aborda todos los aspectos de ejecución semántica para servicios web.

## ANEXO B: METODO DE INDEXACIÓN SEMANTICA EN LA IOT - SIMIOT

### 1. Retos de los Índices Semánticos en IoT

Teniendo en cuenta las características que debe poseer un índice semántico como la multidimensionalidad, adaptabilidad y autocontenido, los retos para la construcción del índice semántico se resumen en la Tabla 39:

Tabla 39: Retos vs. Características de los Índices Semánticos en la IoT

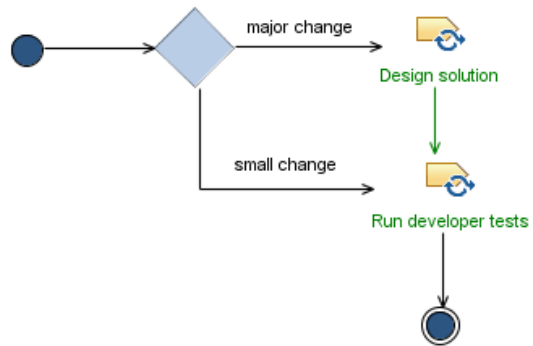
Reto	Característica de los Índices Semánticos		
	Multidimensionalidad	Adaptabilidad	Autocontenido
Eficacia en la extracción semántica	X		X
Mecanismo Eficiente de captura de datos		X	
Correcto uso de estándares		X	X
Selección adecuada del modelo semántico	X		X
Exposición de Servicios	X	X	X

Lo que se propone en la Tabla 39, presenta como al resolver los retos, se aporta a la característica correspondiente, la cual está señalada con la "X".

### 2. Modelado de Procesos con EPFC Composer

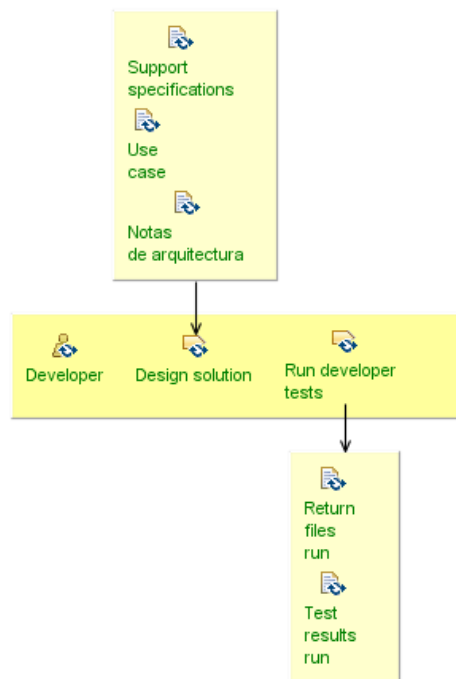
Las fases, iteraciones y actividades son definidas con base en los ladrillos de procesos es decir las tareas, para mostrar como EPFC Composer presenta esta información, se toma como ejemplo la actividad (APCEC - Prepare or Create Knowledge Structure) de la fase 2, iteración 2 de la Figura 26 y en la Figura 27, se muestra el desglose de las tareas correspondiente a la actividad.

**Figura 26. Diagrama de actividad con desglose de tareas**



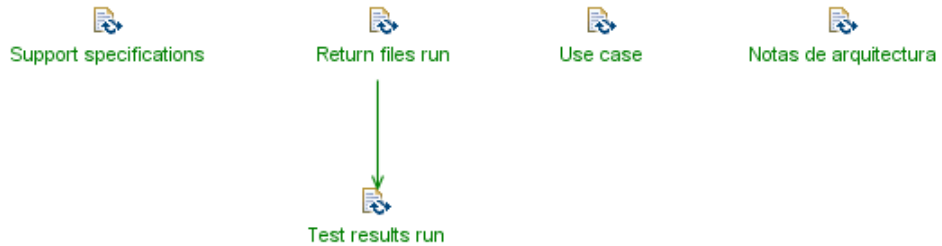
En la Figura 27 se muestra el diagrama detallado de actividades el cual tiene en cuenta los roles, tareas y productos de trabajo especificando cuales son insumos y cuales son resultados de la actividad correspondiente.

**Figura 27. Diagrama detallado de actividad con roles, tareas y productos de trabajo**



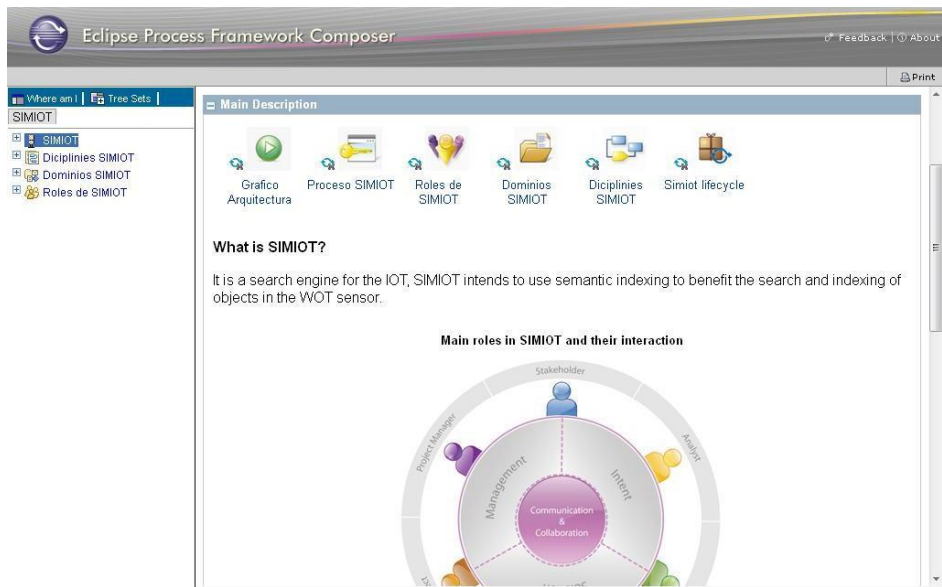
En la Figura 28, se muestra el diagrama de dependencias de productos de trabajo correspondiente a la actividad APCEC.

Figura 28. Diagrama de dependencias de productos de trabajo.



En la Figura 29 se muestra publicación web de EPF Composer provee diferentes vistas de los flujos de procesos y sus elementos dentro del método SIMIOT, en la vista general está la introducción del método, en la vista de disciplinas están las tareas que colaboran en un aspecto metodológico como por ejemplo la arquitectura o los requerimientos, en resumen las tareas conforman actividades y disciplinas estas dos últimas están asociadas distintos a procesos. La vista de dominios de roles presenta una perspectiva detallada de cada rol dentro del método. Por último la vista de productos compila y da acceso fácil a toso los productos de trabajo.

Figura 29. Publicación web de SIMIOT.



El EPF Composer da soporte a los miembros del proyecto para tener un acceso fácil y centralizado a la información del proceso, la cual se actualiza constantemente como una herramienta de aprendizaje de las mejores prácticas de la ingeniería de procesos.

## ANEXO C: IMPLEMENTACIÓN DEL BUSCADOR SEMÁNTICO EN LA IOT

### 1. Modelos del Índice Semántico a Construir

#### 1.1 Modelo de Conceptos

Esta sección que establece las estructuras a utilizar (ontologías, tesauros), cómo utilizarlas y las decisiones de modelos conceptuales y algoritmos a utilizar en cada tarea desarrollada para la obtención de los conceptos. En esta capa se toman decisiones de cómo manejar el contexto y demás información semántica de los objetos. Es un modelo que permite establecer el dominio de objetos que se deben indexar y como se va a caracterizar dicho dominio al momento de seleccionar la información y los objetos a indexar.

El caso de estudio es la información medioambiental y la contaminación producida por el hombre. Para poder establecer claramente las variables y contextos que serán utilizados se ha decidido tomar el libro de (Mendiburu 2003), en el cual hace una explicación muy detallada de lo que supone la información medioambiental y como construir sistemas para automatizarla. Así en la Tabla 40 resumen tenemos la definición de los principales conceptos que se reutilizarán en una ontología que recogerá dicha información.

Tabla 40: Tabla de conceptos del Modelo Medio Ambiental del Buscador SIMIOT

Concepto	Descripción
Ambiente:	Es el hábitat físico y biótico que nos rodea, lo que podemos ver, oír, tocar, oler, saborear.
Medio Ambiente	Es el conjunto de factores físico-químicos (clima, topografía, mares, etc.), y de factores bióticos o factores trópicos (parasitismo, prelación, competencia, etc.), que regulan el desarrollo de los seres vivos (seres humanos, animales, plantas) dentro del ambiente donde se desarrollan.
Ecosistema	Es el ambiente físico-químico donde radican seres abióticos (no vivientes, como el agua y el suelo) y asociaciones de seres bióticos (vivientes, como las plantas y animales) formando un sistema interrelacionado.
Biotipo	El espacio que encierra todos los recursos necesarios para el mantenimiento de la vida, ocupado por las comunidades biológicas. Generalmente tienen unas características ambientales uniformes que se pueden medir. Por ello se ha relacionado con los Biomas. Adicionalmente, para el caso se han elegido algunas variables ambientales que se pueden medir.
Bioma	Un bioma (del griego «bios», vida), también llamado paisaje bioclimático o áreas bióticas (y que no debe confundirse con una ecozona o una ecorregión), es una determinada parte del planeta que comparte el clima, flora y fauna. <sup>14</sup>
Contaminación	Es un cambio indeseado en las características físicas, químicas o biológicas de un sistema.
Contaminación	Esta referido a la contaminación del agua, aire, suelo, y demás componentes

<sup>14</sup> Concepto tomado de: <http://es.wikipedia.org/wiki/Bioma>, accedido el 01/07/2013

Ambiental	de un hábitat, que puede afectar de manera adversa la salud, y la supervivencia de los seres que viven en dicho hábitat.
Desarrollo Sostenido	Es el desenvolvimiento del accionar humano de modo tal que satisfaga sus necesidades presentes, pero sin comprometer la capacidad de las futuras generaciones para satisfacer sus propias necesidades y desarrollarse libremente.
Ecología	Es la ciencia que estudia la relación entre los organismos y el ambiente en que se desarrollan, y la interacción de los seres vivos entre sí.
Población	Conjunto de individuos o seres vivos de la misma especie.
Biocenosis	Conjunto de poblaciones que habitan en el mismo ambiente.
Daño ambiental	Efecto negativo que ocurre en el medio ambiente producido por un agente externo, y como este afecta el desarrollo social, económico y político del ser humano.
Contaminación del Agua	La contaminación del agua corresponde a la alteración de las características que naturalmente debe tener el agua, tornándose inadecuada para el consumo humano, consumo de animales, o riego de plantas.

Se realizó un mapa conceptual (diseño de ontologías con IHMC CMAPS Tools<sup>15</sup>) del modelo medioambiental, el cual se presenta en la Figura 30. El modelo conceptual está dividido en dos grandes partes:

- La primera es el **concepto de medio ambiente**, el cual está relacionado con el ecosistema y este a su vez con todos los seres vivos y no vivos (biocenosis) que interactúan en un biotipo. El **biotipo** relaciona las **condiciones medibles** de: la tierra (Edafotopo), el agua (Hidrotopo) y el aire (Climátopo). Así cada tipo tiene variables medibles como: la calidad del aire, humedad, magnetismo, presión atmosférica y demás variables que caracterizan un clima de una región particular. Al biotipo se le ha relacionado información del área de impacto con el fin de poder relacionarlo posicionalmente con los sensores y también un **Bioma**, el cual servirá para alinear por este concepto la ontología ENVO<sup>16</sup>, la cual posee el mismo concepto (**Biome**) como una de las raíces de la misma y la caracteriza de manera experta. El ecosistema mismo puede tener unas alteraciones ambientales causadas principalmente por las actividades humanas que hace que cambie sus componentes y equilibrio.
- La segunda, corresponde al **concepto de contaminación** y todas las formas de contaminación de los ecosistemas. Esta contaminación puede ser producida por el hombre o por la naturaleza y se conoce como **contaminación ambiental**, ésta contaminación se produce por seis tipos de contaminación: agua, aire, suelo, sonora, visual y térmica. A cada tipo de contaminación se le ha desglosado los principales agentes contaminantes y a algunos como el agua y suelo los **tipos de alteraciones producidas**. Estos elementos terminan en eventos físicos, químicos, mecánicos que son posibles de medir a través de sensores. Los cuales se esperan encontrar en las consultas que se realicen a las bases de datos de servidores IoT. Finalmente, La contaminación ambiental puede producir un **daño ambiental** en el cual se relacionan

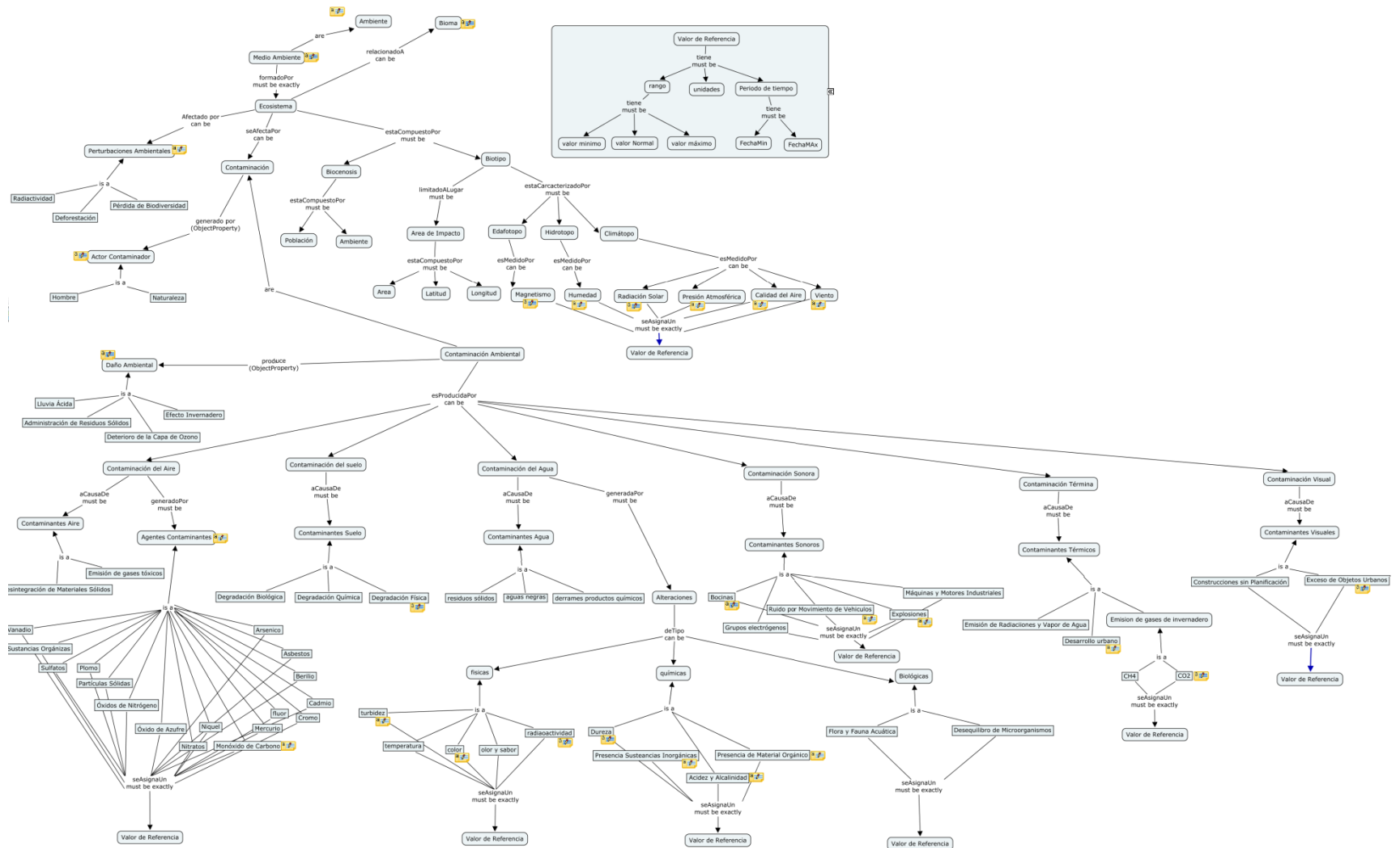
<sup>15</sup> El software y la especificación de creación de ontologías con COE se pueden descargar de <http://cmap.ihmc.us/download/>, accedido el 01/08/2013

<sup>16</sup> Consorcio para la creación de ontologías medioambientales: <http://www.environmentontology.org/Browse-EnvO>, consultado 01/07/2013. Proponen la ontología ENVO.

las preocupaciones actuales más importantes de lo que ocurre a nivel ambiental en el mundo como: administración de residuos sólidos, deterioro de la capa de ozono, efecto invernadero y lluvia ácida.

Para la construcción del modelo de ontología de contaminación medioambiental se siguió un método rápido de construcción de ontologías y mediado con herramientas computacionales. En la Figura 30 se muestra el mapa conceptual de la ontología medioambiental.

**Figura 30: Mapa Mental del Modelo Medio Ambiental del proyecto**



### 1.1.1 Alineación con Ontologías de Dominio Existentes

El mapa conceptual de la Figura 30 se convertirá en ontología en formato OWL. La cual debe ser revisada y adaptada para que quede formalizada de forma correcta. Sin embargo, se encontró una ontología en el área medioambiental desarrollada por expertos, la cual se puede aprovechar para obtener más conocimiento con respecto de la tipificación ambiental del área o lugar que se esté midiendo, adicionalmente aumenta la posibilidad de aceptación y uso por parte de la comunidad interesada. La alineación está dada de la siguiente manera (Kavouras and Kokla 2000; Kavouras 2005):

1. El concepto **Bioma** corresponde con el concepto (**Biome**) de ENVO. Así se conecta el **biotipo** a **biome** creando una clase equivalente a "**bioma**" del modelo conceptual. El Biome es un concepto raíz de la ontología ENVO y este desarrolla todos los conceptos relacionados por expertos al medioambiente, caracterizando de mejor forma el biotipo.
2. El concepto **Ambiente** corresponde con el concepto (**Habitat**) de ENVO. Así se conecta el Medio Ambiente a hábitat de ENVO. Este concepto desarrolla todos los tipos de hábitat disponibles en la ontología y clasificados de forma experta.
3. Los niveles de **materiales del ambiente (environmental material)** y las **características ambientales (environmental feature)** están asociadas a los biomas, a través de ellos se puede encontrar más información de los biotipos relacionados a determinados Biomas. Sin embargo para esta versión estos conceptos no se alinean ya que interesa más el trabajo en contaminación que en la parte caracterización medioambiental.
4. Finalmente, la condición ambiental (**environmental condition**) permite clasificar los biomas en categorías mayores, los cuales permiten entender mejor el tipo de hábitat. También se pueden llegar a estos a través del Biome.

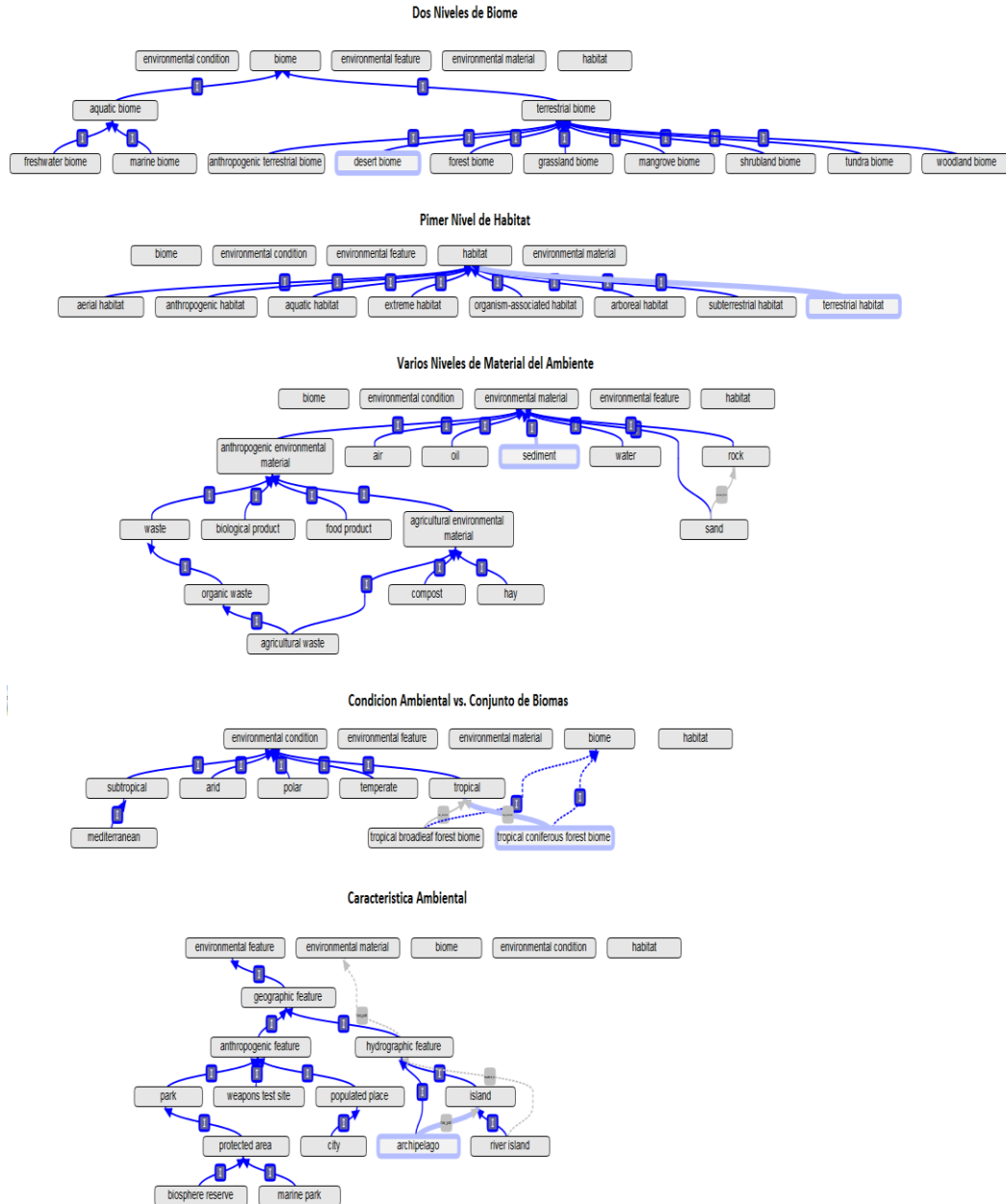
El elemento interesante el cual aporta la ontología desarrollada es la modelización correspondiente a la contaminación del medio ambiente y los posibles sensores que se pueden relacionar en cada elemento contaminante. Adicionalmente, se relacionan al Biotipo sensores que pueden caracterizar el mismo y se pueden definir medidas normalizadas, que posteriormente se pueden comparar con las medidas reales. Para este caso se utilizará la tabla de la **Global 200**, la cual es la lista de las ecorregiones globales o biorregiones identificadas como prioritarias para la conservación por el World Wide Fund for Nature (WWF), específicamente los datos relacionados con Colombia.

Cuando se relacionen los sensores a cada uno de los elementos conceptuales medioambientales, los sensores se representarán a través de la ontología de objeto semántico, desarrollada en el proyecto de doctorado de (Niño-Zambrano 2013) al cual aporta también este proyecto. Esta ontología permitirá definir los aspectos relevantes de los dispositivos aplicables al caso de estudio particular como: propiedades, elementos de contexto (conocimiento) y servicios a ofrecer (métodos). En la Figura 31 siguiente se presenta el objeto semántico atemperado a este trabajo:

Este modelo ya está en su segunda versión y ha sido alineado con las ontologías de observación de sensores SSN-XG (Compton, Barnaghi et al. 2012). Además del

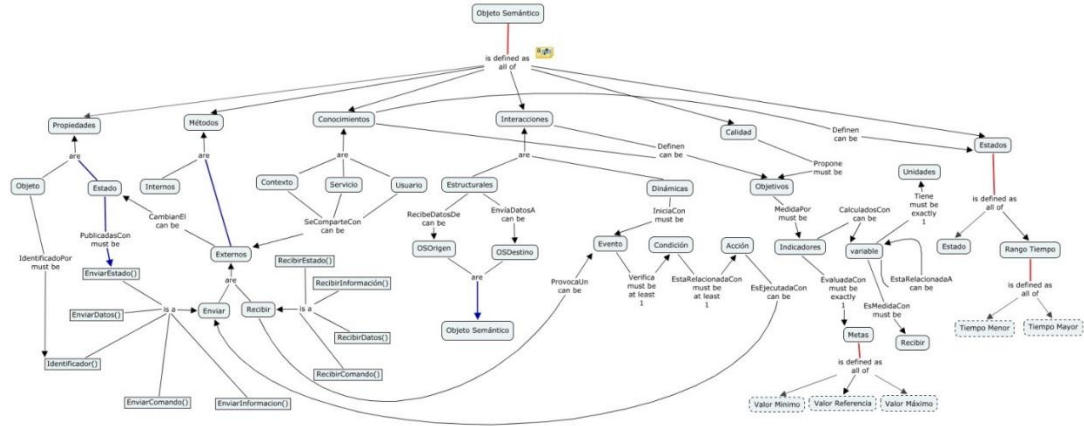
aprovechamiento de propuestas ontológicas como: clasificación de los objetos propuesto por (Mathew, Atif et al. 2011), manejo de contexto social de (Biamino 2012), modelo de servicios de (Kostelnik, Sarnovský et al. 2011) y de representación de conocimiento IoT propuesto por (Wei, De et al. 2012). Esta alineación no se presenta en este documento por políticas de propiedad intelectual.

**Figura 31: Vista Rápida de Alineación a la Ontología ENVO**



En la Figura 32 se muestra el modelo conceptual de la ontología de objeto semántico.

**Figura 32: Modelo Conceptual del Objeto Semántico Tomado Tesis doctoral Miguel Angel Niño Zambrano**



De la Ontología SSN-XG se utiliza el módulo de medición y el módulo de Observación. El primero permite describir un sensor y el segundo establecer que está midiendo. El módulo de capacidades se deja para una versión posterior del índice. Así tenemos:

**Tabla 41: Alineación de Ontologías SSN-XG y AWS**

Concepto de SSN-XG	Conceptos e Instancias Alineadas
ssn:SensingDevice	Se utiliza la ontología de <b>“Automatic Weather Station - AWS”</b> para crear subclase de conceptos relacionados con los tipos de sensores de las estaciones climáticas. Estos conceptos se relacionan a los tipos de sensores definidos en la ontología de contaminación medioambiental.
ssn:FeatureofInterest y ssn:Property	Se utiliza el aporte del <b>“Climate and Forecast Metadata Conventions”</b> , que ofrece una ontología con <b>“cf-features”</b> que aporta una colección de nombres estándar para variables climáticas e incluida en <b>“AWS”</b> . A través de la relación <b>ssn:hasProperty</b> se enlaza a <b>ssn:Property</b> , ésta última crea instancias de <b>“cf:property”</b> . Las propiedades son relacionadas a las variables de contaminación definidas en la ontología de contaminación medioambiental del proyecto.
ssn:Platform y Area de Impacto	Para relacionar los lugares geográficos, que para este caso de estudio corresponde a las ciudades de Colombia, se utilizará la información existente en el servicio <b>GEONAMES</b> ( <a href="http://www.geonames.org">http://www.geonames.org</a> ), la cual a partir del nombre de la ciudad se obtiene las coordenadas geoespaciales de su localización en el planeta. Estas instancias se crean bajo el concepto <b>Area de Impacto</b> , el cual a su vez relaciona una plataforma con <b>ssn:Platform</b> y este a su vez a un <b>ssn:SensingDevice</b> . La plataforma que se crea es virtual, la cual agruparía los sensores distribuidos y cercanos a dicha ciudad. Para el

---

caso particular se colocaran las instancias de dos ciudades: “Popayán” y “Bucaramanga”.

---

En la Tabla 41, se presenta adicionalmente el uso de otra ontología denominada Automatic Weather Station – AWS. La cual permite utilizar nombre estándar para los tipos de sensores y variables medioambientales, asegurando en mejor medida la inclusión de conocimiento experto a la ontología y de estándares que pueden ser utilizados por varios propietarios de dispositivos sensores.

De la clase SensingDevice se relaciona al Biotipo para medir cada una de las siguientes características con sus respectivos sensores a través de estrategia de clases equivalentes:

- 1) Humidity = Humedad
- 2) Atmospheric Pressure = Presión Atmosférica
- 3) Radiation = Radiación Solar
- 4) Temperature = Temperatura
- 5) Wind = Viento
- 6) Precipitation = Precipitación
- 7)

Del proyecto de Sensores Virtuales se adicionan los siguientes conceptos:

- 1) Monóxido de Carbono = CO. Se adiciona etiqueta.
- 2) Partículas Suspendidas Totales = PST. Se crea el individuo de la clase “Agentes contaminantes” y se agrega una etiqueta con el acrónimo.
- 3) Material Particulado Menor a 10 micrómetros = MP10
- 4) Óxido de Azufre = SO<sub>2</sub>. Se crea una etiqueta con el acrónimo SO<sub>2</sub>.
- 5) Óxidos de Nitrógeno = NO<sub>2</sub>. Se crea una etiqueta con el acrónimo NO<sub>2</sub>.
- 6) Ozono = Ozone = O<sub>3</sub>. Aunque el ozono no es un agente contaminante, se ha adicionado a esta clase con el fin de tomar sus mediciones, como lo establece la norma colombiana de Sistemas de Calidad del Aire.

Adicionalmente, se utiliza el servicio de GEONAMES con la finalidad de caracterizar los lugares (ciudades) e identificarlos geoespacialmente, así también se asegura una correcta referenciación de dichos sitios lo más cercanos a la realidad espacial.

### **1.1.2 Ontología de Mediciones en Contaminación Medioambiental en dos Ciudades de Colombia**

Finalmente y después de las alineaciones se presenta la Ontología que se utilizará para anotar los documentos de los dispositivos o sensores que se encuentre en el servidor IoT para su posterior indexación.

Esta ontología será usada en esta primera versión con propósitos de anotar semánticamente los documentos analizados del servidor IoT, para que posteriormente se indexen para su consulta. En el momento de la consulta también será utilizada para presentar ayuda de autocompletado al usuario y así guiar mejor el proceso de búsqueda.

La ontología no será usada para instanciar sensores y razonar sobre la misma las consultas. Esta opción se deja para la segunda versión del proyecto o trabajo futuro.

En la siguiente página se presenta el modelo conceptual de ontología final.

## 1.2 Modelo de Consultas Personalizadas

Establecen la información del contexto (Zhang, Cheng et al. 2012) teniendo en cuenta que los usuarios generalmente están interesados inicialmente en las entidades (cosas, lugares, personas) y posteriormente por las cualidades o estados de esas cosas (vacío, libre, sentado, caminando, etc.) y no en sensores individuales y sus datos de salida. El presente proyecto comparte la misma idea y por ello las preguntas van más enfocadas a entidades medioambientales (lugar) y sus estados. En las preguntas de referencia se había definido cuatro preguntas iniciales (a-e):

- a. ¿Qué dispositivos *cercanos* a <<lugar>> pueden entregarme <<información medioambiental>>?
- b. ¿Qué dispositivos se encuentran cerca de Popayán que entreguen información medioambiental?
  - ¿Qué dispositivos miden información medioambiental aquí?
- c. ¿Qué <<información medioambiental>> puedo *obtener* de <<lugar>>?
  - ¿Qué información medioambiental hay en Colombia?
  - ¿Qué información medioambiental hay en latitud ( ° , ' , ") y longitud ( ° , ' , ")?
- d. ¿Cuál es el *histórico* de la <<información medioambiental>> en un <<lugar>> en el <<periodo de tiempo>>?
  - ¿Cuál es el histórico de la información medioambiental en Popayán entre el 01/01/2010 y el 31/12/2010?
- e. ¿Qué <<variable medioambiental>> hay en <<lugar>> y en que [<<periodo de tiempo>>]?
  - ¿Qué temperatura hay en Madrid?
  - ¿Qué humedad hay en New York entre el 30/03/2013 hasta hoy?

Aquí vamos a detallar en que consiste cada ítem encerrado entre paréntesis angulares:

- <<lugar>>: Corresponde en el sentido más simple a un punto geográfico con latitud y longitud, el cual está caracterizado en el área de impacto en la ontología. Sin embargo, como los biotipos están relacionados al biome en la ontología ENVO, podemos a partir de allí extender el concepto a todas las tipificaciones desarrolladas pudiendo caracterizar más explícitamente el lugar y ampliar el rango de preguntas en este sentido. Adicionalmente, la información de metadatos de los objetos de la WoT contiene su posición en latitud y longitud, lo cual se convierte en una ventaja para interrelacionar sensores con biotipos, ecosistemas y hábitat. Los lugares geográficos adicionalmente se relacionan al servicio de Google Maps y Geonames con el fin de establecer a partir de los nombres de ciudades o lugares los datos geo-posicionales.
- <<información medioambiental>>: Esta información se refiere a cualquier concepto debajo de este nodo de la ontología. Así podemos dar una información clasificada y bien definida de lo que caracteriza cierta ubicación geográfica y relacionarla con las mediciones actuales de los sensores encontrados en dicha ubicación. Si el concepto se encuentra en niveles superiores, el reporte puede incluir varios niveles de medición

de sensores y sus posibles interacciones, dependiendo de los modelos de servicios desarrollados.

- <<periodo de tiempo>>: Compuesta de dos variables, una de fecha inicio y otra de fecha fin, con la condición que la fecha fin debe ser mayor que la fecha inicio. En los casos en que no se provee el rango se asume el periodo de tiempo puntual del momento en el que se hizo la consulta.
- <<variable medioambiental>>: Corresponde a las variables relacionadas al biotipo y en las cuales, dependiendo de la ubicación geográfica, podemos relacionar ciertos sensores existentes. Estas variables deben tener previamente un valor de referencia para poder realizar las comparaciones correspondientes con los datos medidos. El valor de referencia puede ser un rango. Por ejemplo: la temperatura en verano en España normalmente oscila entre 30 y 40 grados, lo que esté por debajo o por encima en ese periodo no es normal.
- <<agentes contaminantes>>: Corresponde a todas las variables medidas de elementos que contaminan el medio ambiente y de la cual se encontraron sensores relacionados. Cuando su medición está por fuera de los rangos normales se considera como agente que está incidiendo en el medio ambiente.
- <<daños ambientales>>: Corresponde a las principales preocupaciones de daños al medioambiente generados por el hombre y que se podría acercar a esta conclusión tomando como base las mediciones de los agentes contaminantes del lugar particular. Esto es un proceso complejo y aquí sólo pretendemos presentar una forma útil en que podríamos encontrar este tipo de respuestas, pero se espera que los modelos de servicios sean desarrollados por expertos del tema.

Los verbos (resaltados en cursiva) utilizados en los tipos de preguntas: cercanía, históricos, tendencia, diferente, causando y generar, se definirán en el modelo de servicios de este mismo documento, ya que corresponden a operaciones que se deben desarrollar en el dominio de la aplicación.

Con el fin de que las entradas sean más precisas se implementara un sistema de autocompletado en línea, en el cual cuando el usuario este digitando una consulta se buscaran palabras clave en la misma ontología y se le guiara en la forma correcta de hacer la estructura de la pregunta con el fin de obtener mejores resultados.

### 1.3 Algoritmos de Similitud

Es un modelo con un conjunto de decisiones alrededor de los algoritmos que se van a utilizar para el análisis de coocurrencia y representatividad.

Para la primera versión del modelo semántico se utilizará un algoritmo de similitud básico con el cálculo de TF –IDF (frecuencia de términos – frecuencia inversa del documento), donde medimos la frecuencia de los conceptos candidatos en los documentos sensor que fueron recuperados, estos resultados son posteriormente guardados en una matriz.

Para el cálculo de la similitud en el índice se realizará con la asignación de pesos a los conceptos de la ontología, teniendo en cuenta la jerarquía de la misma. Los pesos fueron asignados consecutivamente dando más peso a los hijos que los padres. La función de similitud escoge los conceptos relacionados de mayor peso con el fin de ampliar y relacionar los conceptos a la ontología.

## 2. Evaluación de Middleware Seleccionados

### 2.1 Indicadores de Selección

El caso de estudio pretende indexar información de los sensores que provean datos importantes con respecto al medio ambiente y la contaminación. Supone que un usuario cualquiera puede hacer una consulta y obtener información de los sensores más relacionados a dicha consulta, retornando los sensores en un orden de relevancia particular y con la posibilidad de acceso a sus datos en tiempo real.

Dado lo anterior, y teniendo en cuenta el análisis citado en el marco de referencia, se seleccionan y definen los indicadores para seleccionar el mejor servidor IoT a utilizar en el caso de estudio. Se identifican dos tipos de indicadores:

1. **Obligatorios:** Son indicadores que debe cumplir el servidor, por ser una restricción de la arquitectura del caso de estudio. El servidor que no lo cumpla queda descartado por defecto.
2. **Opcionales:** Son indicadores deseables y que aportan al cálculo de puntuación para la selección del servidor.

Para la evaluación primero se descartan los Servidores IoT que no cumplan con los indicadores obligatorios y finalmente los que quedan se les calcula un puntaje de selección – PS. El PS suma la prioridad de cada indicador opcional que cumpla un servidor IoT particular, tal como presenta la Ecuación 9:

**Ecuación 9: Puntaje de Selección del Servidor IoT como fuente de indexación**

$$PS_j = \sum_{i=1}^n PIndOpc_i$$

Así el Servidor j que tenga mayor PS se convierte en la primera opción para ser utilizado en la indexación y sucesivamente con los siguientes en el orden descendente.

En la Tabla 42 definen los indicadores en cuatro columnas así:

- El indicador: es nombre dado al mismo.
- La prioridad: establece un peso de mayor o menor importancia, para el caso entre más grande el número mayor es el peso de dicho indicador, los pesos se ha decidido manejar en un rango de números enteros de 1 a 3.
- El Cálculo: la forma de calcular y/u observar el mismo en determinado servidor IoT.
- La Descripción: una definición conceptual del indicador.

**Tabla 42: Tabla de indicadores para selección de Middleware**

Indicador	Prioridad	Cálculo	Descripción
Directorio de Objetos - Búsqueda - DOB	3	Existencia (1) o No (0) de la funcionalidad.  Es Obligatorio	Es la principal funcionalidad que debe proveer el servidor IoT para poder utilizarlo como base de un meta-buscador, es decir el caso de estudio desarrollará una indexación sobre los objetos que retorne una consulta sobre la base de datos del servidor IoT. Por esta razón los servidores que no cumplan con este indicador son descartados directamente.
Gestión de Datos en Tiempo Real - GTR	3	Existencia (1) o No (0) de la funcionalidad.  Es Obligatorio	El servidor permite hacer consultas de los datos que se producen en tiempo real por parte de los sensores. Esta funcionalidad es requisito primordial del caso de estudio según el documento de requisitos y arquitectura, por esta razón los servidores IoT que no cumplan este requisito son descartados directamente.
API de desarrollo - API	3	Existencia (1) o No (0) de API.  Es Obligatorio	El servidor debe proveer una librería de desarrollo o "Application Program Interface" - API de desarrollo con acceso a datos y metadatos de los objetos y sensores. Es un requisito esencial y por ello los que no provean el API quedan descartados directamente.
Gestión de Metadatos - GMD	3	Existencia (1) o No (0) de la funcionalidad.  Es Obligatorio	El API permite gestionar metadatos relacionados a los objetos que contiene el Servidor IoT. Es importante tener los metadatos porque brindan más información al índice semántico.
Licencia Open - LOP	2	Existencia (1) o No (0) de la licencia.  Obligatorio.	El servidor IoT permite el uso de todos sus servicios bajo un licencia abierta, reutilizable, modificable y compatible.
Servidor de Objetos en el Dominio Medioambiental - SDM	3	Existencia (1) o No (0) de la restricción de dominio.  Opcional.	Es un servidor IoT que almacena especialmente objetos de información medioambiental y de contaminación. Este es un requisito importante pero no primordial para descartar el servidor IoT. Por lo general todos son de propósito general aunque algunos admitan configurar dominios.
Uso de Estándares y Formatos de Datos - EFD	2	Existencia (1) o No (0) de formatos Estándar.  Opcional.	El servidor se sirve de estándares definidos en la IoT para gestionar los datos de los objetos. Además de utilizar formatos ampliamente aceptados para estructura y datos.
Documentación - DOC	1	Existencia (1) o No (0) de la restricción de dominio.  Opcional.	El servidor IoT presenta una excelente documentación para el uso del API y de sus servicios.

Número Objetos - NOB	3	Por encima de la media (3), igual (2) y debajo (1).  Opcional.	Es el número de objetos conectados actualmente al servidor IoT. Entre más objetos, más posibilidades de encontrar diferentes tipos de sensores y mediciones medioambientales.
----------------------	---	--	---

## 2.2 Resultados de las Pruebas

Para obtener los resultados de la aplicación de los indicadores se ha procedido a puntuar cada uno con respecto a cada servidor en la Tabla 43, se procede a descartar los servidores que no cumplen con los indicadores obligatorios.

Tabla 43: Evaluación del Servidor IoT a utilizar en el caso de estudio

Servidor IoT	Indicadores Obligatorios					Cumpl e
	DOB	GTR	API	GM D	LOP	
<b>AMEE</b>		√	√	√		No
<b>Arkessa</b>	√	√	√	√	√	Si
<b>Axeda</b>	√	√	√	√		No
<b>Bugswarm</b>			√			No
<b>Carriots cloud</b>		√	√		√	No
<b>Eart Cam</b>						No
<b>Evrythng</b>		√	√		√	No
<b>GroveStreams</b>		√	√		√	No
<b>Sensinode</b>		√				No
<b>Nimbits</b>	√	√	√		√	No
<b>Exosite</b>	√	√	√		√	Si
<b>Open.Sen.se</b>		√	√	√	√	No
<b>Paraimpu</b>					√	No
<b>NewAer</b>						No
<b>SensorCloud</b>		√	√	√		No
<b>SensorPedia</b>	√	√	√	√	√	Si
<b>Sensormap</b>	√	√	√	√		No
<b>SenseWeb</b>	√	√	√			No
<b>SensorBase</b>	√	√	√			No
<b>ThingSpeak</b>	√	√	√	√	√	Si
<b>Thingworx</b>	√	√	√			No
<b>Xively Cloud Services</b>	√	√	√	√	√	Si
<b>Yaler</b>		√	√		√	No

Los servidores que cumplen los indicadores obligatorios son: Arkessa, Exosite, SensorPedia, ThingSpeak y Xively.

En la Tabla 44 se presenta el cálculo del PS para cada servidor. En la casilla se coloca el valor de la prioridad en caso de que el servidor cumpla con dicho indicador. Para el caso del indicador SDM, para que el servidor obtenga su puntaje debe estar especializado en ese dominio, en caso de ser de dominio general no se le aportará puntaje. Para el indicador de NOB, algunos servidores no reportan el número de objetos conectados, en

este caso se asume que tienen la media con un valor de 1, para el caso el promedio de objetos conectados es de 62,5 millones, teniendo en cuenta los siguientes datos: Arkessa (¿No hay datos?), Exosite (¿No hay datos?), SensorPedia (¿No hay datos?), ThingSpeak(0.01 millones) y Xively (250 millones).

**Tabla 44: Evaluación del PS para cada servidor IoT seleccionado**

Servidor IoT	Cálculo = Suma Indicadores.				
	SDM	EFD	DOC	NOB	
Arkessa	0	0	0	1	1
Exosite	0	2	1	1	4
<u>SensorPedia</u>	0	2	1	1	4
ThingSpeak	0	2	1	1	4
Xively Cloud Services	0	2	1	3	6

De acuerdo a la evaluación realizada el primer servidor a indexar sería el de Xively Cloud Services y después en el orden cualquiera de: Exosite, SensorPedia y ThingSpeak, finalmente y de último Arkessa. Sin embargo es necesario realizar las pruebas de programación, con el fin de determinar cuál API se adapta más fácilmente al caso de estudio.

### 3. Código Fuente de Creación de los servicios del API de la Fuente de datos

```

class XivelyManager
{
    //Objeto de clase que obtiene los JSON dada una Ontología OWL
    private RecolectorDocumentosXively _RecolectorXively = null;

    //Objeto de la clase ClienteXively para operar sobre los feed
    private ClienteXively _ClienteXively = null;

    //Variable que almacena el directorio en el cual se almacenará finalmente cada uno de los JSON
    //relacionados a la búsqueda en Xively con los conceptos de la ontología
    private String _DirectorioJSON =string.Empty;

    //Variable que almacena la ruta de la Ontología que se utilizará para traer los JSON
    private String _RutaOntologia = string.Empty;

    public XivelyManager(String RutaOntologia,String DirectorioJSON)
    {
        _RutaOntologia = RutaOntologia;
        _DirectorioJSON = DirectorioJSON;
        _RecolectorXively = new RecolectorDocumentosXively(_RutaOntologia, ConfigurationManager.AppSettings["urlBaseXively"],
        DirectorioJSON);

        //this.ProcesarDocumentos();
    }

    //Funcion que realiza la obtención de los documentos de xively buscando por cada concepto de la ontología
    //Almacena los JSON localmente en DirectorioJSON y retorna la lista de urlfeed correspondientes
    public List<FeedXively> ProcesarDocumentos(Boolean fullJson)
    {
        _RecolectorXively.ObtenerUrlsFeedConceptos(fullJson);
        return _RecolectorXively.ListaUrlsFeed;
    }

    //Carga los FeedXively (json) de la BDD con el fin de reindexar
    public List<FeedXively> LoadJSONfromDiskDocumentsURL()

```

```

    {
        return _RecolectorXively.LoadJSONfromDiskDocumentsURL();
    }

    //Este procedimiento retorna el JSON del feed estipulado de la BDD
    public FeedXively ObtenerJsonFeedBDD(string Idfeed)
    {
        return _RecolectorXively.ObtenerJsonFeedBDD(Idfeed);
    }

    //Esta funcion retorna el Json desde el servidor Xively
    public string ObtenerJsonFeed(string APIkey, string feedID)
    {
        _ClienteXively = new ClienteXively(APIkey, feedID);
        return _ClienteXively.ObtenerJsonFeed();
    }

    //Define un objeto ClienteXively con el fin de traer datos directamente del servidor
    public string RetornarDatosSensor(string feedID,DateTime fechaInicio, DateTime fechaFin)
    {
        _ClienteXively = new ClienteXively(ConfigurationManager.AppSettings["APIkey"], feedID);

        return _ClienteXively.LeerMedicion(fechaInicio, fechaFin);
    }

    //Retorna los datapoints
    public string RetornarDatapointsFeed(string feedID, string DatastreamId, DateTime fechaInicio, DateTime fechaFin)
    {
        _ClienteXively = new ClienteXively(ConfigurationManager.AppSettings["APIkey"], feedID);

        return _ClienteXively.RetornarDatapointsFeed(DatastreamId, fechaInicio, fechaFin);
    }

    //Almacena el JSON en la BDD y lo Anota semánticamente
    public List<FeedXively> ProcesarDocumento(string feedId)
    {
        _RecolectorXively.AdjuntarJSONFeedBDD(feedId);
        return _RecolectorXively.ListaUrlsFeed;
    }
}

```

#### 4. Código Fuente del Análisis de Representatividad

```

//Realiza la búsqueda de un texto "original" contra los documentos indexados utilizando TFIDF
4 public static List<Document> moreLikeThisAnalyzer(String original, ISet<string> stopWords,
Lucene.Net.Analysis.Analyzer analyzer)
5 {
6     Trace.WriteLine("Realizando la Búsqueda");
7     List<Document> DocumenResult = new List<Document>();
8
9     IndexReader indexReader = IndexReader.Open(_directory, true);
10    IndexSearcher indexSearcher = new IndexSearcher(indexReader);
11
12    MoreLikeThis mlt = new MoreLikeThis(indexReader);
13
14    mlt.SetFieldNames(DEFAULT_FIELD_NAMES);
15    mlt.MinDocFreq = DEFAULT_MIN_DOC_FREQ;
16    mlt.MinTermFreq = DEFAULT_MIN_TERM_FREQ;
17    mlt.MaxQueryTerms = MAX_QUERY_TERMS;
18    mlt.MinWordLen = DEFAULT_MIN_WORD_LENGTH;
19    mlt.Analyzer = analyzer;
20    mlt.SetStopWords(stopWords);
21
22    Query query = mlt.Like(new System.IO.StringReader(original));
23
24    int topCount = DEFAULT_DOCUMENT_TO_SEARCH;
25
26    TopScoreDocCollector collector = TopScoreDocCollector.Create(topCount, true);
27    indexSearcher.Search(query, collector);
28    ScoreDoc[] hits = collector.TopDocs().ScoreDocs;
29    var result = new List<string>();
30    //Hits hits = indexSearcher.Search(query);
31
32    int len = hits.Length;
33
34    Trace.WriteLine("Entering");

```

```

35         Trace.WriteLine("-----");
36         Trace.WriteLine("original : " + original);
37         Trace.WriteLine("query: " + query);
38         Trace.WriteLine("found: " + len + " documents");
39         for (int i = 0; i < Math.Min(25, len); i++)
40         {
41             int d = hits[i].Doc;
42             Trace.WriteLine("score   : " + hits[i].Score);
43             Trace.WriteLine("name    : " + d.ToString());
44             //Colocar los datos en el arreglo de resultados
45             Document doc = indexSearcher.Doc(hits[i].Doc);
46             DocumenResult.Add(doc);
47         }
48         Trace.WriteLine("-----");
49         Trace.WriteLine("Exiting");
50         return DocumenResult;
51     }
52

```

## 5. Código Fuente de Implementación de Servicios Web Semántico de Objetos

```

4     ///

```

```

59     ///relacionada a los sensores que tienen influencia en dicho lugar (determinado por el radio) y sus relaciones a
60     ///conceptos medioambientales. El idioma y el radio son opcionales. Por defecto busca en: Español y 100 km.
61     ///</summary>
62     ///<returns>
63     ///DataSet con una lista de Sensores que estan en el área de onfluencia circular del biotipo
64     /// </returns>
65     /// <param name="lat">Latitud del lugar en el cual se buscarna los sensores</param>
66     /// <param name="lng">Longitud del lugar en el cual se buscarna los sensores</param>
67     /// <param name="dtsResultados">Dataset con los sensores que se encontraron en la consulta del usuario</param>
68     /// <param name="idioma">Idioma de preferencia del usuario. Por defecto busca Español</param>
69     /// <param name="radio">Área circular al rededor del punto geografico enviado en kilometros en los cuales se buscaran
    los sensores</param>
70     public List<FeedXively> RetornarMapaLugar(string lat, string lng, string consulta, string idioma, string radio)
71     {
72         List<FeedXively> feedList = new List<FeedXively>();
73
74         feedList.AddRange(srvIndex.RetornarMapaLugar(Convert.ToDouble(lat, culture),
75             Convert.ToDouble(lng, culture),
76             consulta, idioma, Convert.ToDouble(radio, culture)));
77         return feedList;
78     }
79
80     ///<summary>
81     ///Esta función recibe un Biotipo(lugar) en términos de latitud y longitud, una consulta (buscar) y devuelve la
    información
82     ///relacionada a los sensores que tienen influencia en dicho lugar (determinado por el radio) y sus relaciones a
83     ///conceptos medioambientales. El idioma y el radio son opcionales. Por defecto busca en: Español y 100 km.
84     ///La variante con respecto al anteriopr es que busca en Xively los datapoints
85     ///</summary>
86     ///<returns>
87     ///DataSet con una lista de Sensores que estan en el área de onfluencia circular del biotipo
88     /// </returns>
89     /// <param name="lat">Latitud del lugar en el cual se buscarna los sensores</param>
90     /// <param name="lng">Longitud del lugar en el cual se buscarna los sensores</param>
91     /// <param name="consulta">Consulta del usuario</param>
92     /// <param name="fechaFin">Fecha inicial del rango de datapoint a recuperar</param>
93     /// <param name="fechaInicio">FEcha Final del rango de datapoint a recuperar</param>
94     /// <param name="idioma">Idioma de preferencia del usuario. Por defecto busca Español</param>
95     /// <param name="radio">Área circular al rededor del punto geografico enviado en kilometros en los cuales se buscaran
    los sensores</param>
96     public List<FeedXively> RetornarMapaLugarDatapoints(string lat, string lng, string consulta, DateTime fechaInicio,
    DateTime fechaFin, string idioma, string radio)
97     {
98         List<FeedXively> feedList = new List<FeedXively>();
99
100        srvIndex.Timeout = -1;
101        feedList.AddRange(srvIndex.RetornarMapaLugarDatapoints(Convert.ToDouble(lat, culture),
102            Convert.ToDouble(lng, culture),
103            consulta,
104            fechaInicio,
105            fechaFin,
106            idioma,
107            Convert.ToDouble(radio, culture)));
108        return feedList;
109    }
110
111    ///<summary>
112    ///Esta función recibe un Biotipo(lugar) en términos de latitud y longitud, una lista de sensores y devuelve la
    información
113    ///relacionada a los sensores que tienen influencia en dicho lugar (determinado por el radio) y sus relaciones a
114    ///conceptos medioambientales. El idioma y el radio son opcionales. Por defecto busca en: Español y 100 km.
115    ///</summary>
116    ///<returns>
117    ///DataSet con una lista de Sensores que estan en el área de onfluencia circular del biotipo
118    /// </returns>
119    /// <param name="lat">Latitud del lugar en el cual se buscarna los sensores</param>
120    /// <param name="lng">Longitud del lugar en el cual se buscarna los sensores</param>
121    /// <param name="dtsResultados">Dataset con los sensores que se encontraron en la consulta del usuario</param>
122    /// <param name="idioma">Idioma de preferencia del usuario. Por defecto busca Español</param>
123    /// <param name="radio">Área circular al rededor del punto geografico enviado en kilometros en los cuales se buscaran
    los sensores</param>
124    public List<FeedXively> RetornarMapaLugarListaSensores(string lat, string lng, List<FeedXively> feedList, string idioma,
    string radio)
125    {
126        FeedXively[] feedsarr = feedList.ToArray();
127        feedsarr = srvIndex.RetornarMapaLugarListaSensores(Convert.ToDouble(lat, culture),
128            Convert.ToDouble(lng, culture),
129            feedsarr, idioma,
130            Convert.ToDouble(radio, culture));
131        return feedsarr.Cast<FeedXively>().ToList();
132    }
133
134    ///<summary>
135    ///Esta función busca los Biotipos realcionados a una lista de sensores encontrados en una búsqueda realizada al indice.
136    ///</summary>
137    ///<returns>
138    ///DataSet con una lista de Ssensores que cumplen con la consulta realizada
139    /// </returns>

```

```

140     /// <param name="ListaSensoresEncontrados">Lista de sensores encontrados despues de la consulta de usuario</param>
141     public List<GeonameNode> ObtenerBiotiposConsulta(List<GeonameNode> ListaSensoresEncontrados)
142     {
143         //Serializar la lista de GeonameNodes
144         string xmlListGeonodes = ListaSensoresEncontrados.SerializarToXml();
145
146         //Obtenemos la División Administrativa de Primer orden para la lista de sensores
147         //que tiene localización
148         srvIndex.Timeout = -1;
149         xmlListGeonodes = srvIndex.ObtenerListaLugarCoordenadas_AM1(xmlListGeonodes);
150
151         //Deserializamos los objetos retornados
152         List<GeonameNode> gnCercanos = xmlListGeonodes.DeserializarTo<List<GeonameNode>>();
153
154         return gnCercanos;
155     }
156
157     ///<summary>
158     ///Esta función recibe un término y lo compara en la ontología con el fin de establecer si es un concepto.
159     ///Si es concepto retorna los conceptos más relacionados al mismo (relaciones directas a otros conceptos),
160     ///desde el punto de vista del dominio de la ontología cargada.
161     ///</summary>
162     ///<returns>
163     ///Lista de conceptos relacionados semánticamente al concepto buscado
164     /// </returns>
165     /// <param name="concepto">Concepto que se esta buscando en la ontología</param>
166     /// <param name="idioma">Idioma de preferencia del usuario.</param>
167     public List<string> RetornarConceptos(string concepto, string idioma)
168     {
169         string res = srvIndex.RetornarConceptos(concepto, idioma);
170         return res.DeserializarTo<List<string>>();
171     }
172
173     ///<summary>
174     ///Esta función retorna un FeedXively con los datapoints correspondientes entre un rango de tiempo
175     ///</summary>
176     ///<returns>
177     ///Feed completo
178     /// </returns>
179     /// <param name="concepto">Concepto que se esta buscando en la ontología</param>
180     /// <param name="idioma">Idioma de preferencia del usuario.</param>
181     public FeedXively HistoricosSensor(string feedID, DateTime fechaInicio, DateTime fechaFin)
182     {
183         FeedXively fx = new FeedXively();
184
185         fx = srvIndex.RetornarDatosSensor(feedID, fechaInicio, fechaFin);
186
187         return fx;
188     }
189
190     //Aquí vienen los demás servicios del negocio
191     //5. TendenciaVariableAmbiental (variable medioambiental): La tendencia debe implementar un sistema de análisis
192     correlacional básico con el fin de proveer información de tendencias de variables. Se utilizará lo más básico pero aquí se
193     pueden colocar análisis más elaborados que se dejan para futuros estudios.
194     //6. VariablesDiferentesBiotipo (lugar): Este método debe comparar los valores de referencia almacenados en el
195     Biotipo de ciertos lugares y posteriormente con las mediciones de los sensores establecer si se encuentran en los rangos
196     establecidos o se han salido de sus valores esperados. Para esto hay que buscar la información de referencia y poblar la
197     ontología con la misma en un lugar seleccionados para demostrar sus posibilidades de uso.
198     //7. CausandoDañoAmbiental (lugar): Este método devuelve la información de los posibles agentes contaminantes que
199     están afectando cierto lugar. Para esto debe tomar los sensores de dicho lugar y posteriormente utilizar la función de
200     VariablesDiferentesBiotipo, las variables identificadas se relacionan a sus agentes contaminantes y se retorna una respuesta
201     en una estructura de conceptos provista por la ontología en la cual se identifican dichos contaminantes.
202     //8. DañosAmbientales (lugar, AgentesContaminantes): A partir de un conjunto de agentes contaminantes y su magnitud
203     se desarrolla un modelo matemático sencillo para relacionarlo a un daño medioambiental. La respuesta son los posibles daños
204     causados el ecosistema por estos agentes.

```

## 6. Código Fuente de Implementación de Servicios de Índice

```

1.     #region "Servicios Básicos del Índice semántico"
2.
3.     //Realiza la Búsqueda de la consulta dada, con la ontología y las opciones de configuración
4.     //realizadas previamente y almacenada en el Web.config
5.     [WebMethod]
6.     public DataSet Buscar(string stringBuscar, string idioma = "Español")
7.     {
8.         if (!string.IsNullOrEmpty(stringBuscar))
9.             return SearchSemanticIndex(stringBuscar, idioma);
10.        else
11.            return null;
12.    }

```

```

13.
14. //Igual que la anterior pero no expande consulta y retorna una lista de FeedXively Serializados
15. [WebMethod]
16. public List<FeedXively> BuscarFeeds(string consulta, string idioma = "Español")
17. {
18.     if (!string.IsNullOrEmpty(consulta))
19.     {
20.         //Crea el Objeto que guarda todo el conocimiento del negocio
21.         SemanticIndexManager semanticIndexManager = new SemanticIndexManager();
22.
23.         //Crea la lista de documentos que serán el resultado de la búsqueda
24.         List<UrlDocument> allReponse = new List<UrlDocument>();
25.
26.         //Se busca en español o el ingles. Si es nulo busca por ambos.
27.         if (idioma == "Español")
28.         {
29.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, true);
30.         }
31.         else if (idioma == "Ingles")
32.         {
33.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, false);
34.         }
35.         else
36.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, true); //Por defecto
37. busca español
38.
39.         //Armar la estructura de datos a retornar como resultado
40.         List<FeedXively> SensoresResult = new List<FeedXively>();
41.
42.         //Traemos todos los metadatos del sensor que estan almacenados en la BDD local
43.         foreach (UrlDocument urldoc in allReponse)
44.         {
45.             FeedXively xtemp = semanticIndexManager.ObtenerJSONFeedBDD(urldoc.Id);
46.             SensoresResult.Add(xtemp);
47.         }
48.         return SensoresResult;
49.     }
50.     else
51.         return null;
52. }
53.
54. //Igual que la anterior pero retorna una lista de FeedXively Serializados
55. //Además de los datapoints correspondientes consultados en Xively.
56. [WebMethod]
57. public List<FeedXively> BuscarFeedsDataPoints(string consulta, DateTime fechaInicio, DateTime
58. fechaFin, string idioma = "Español")
59. {
60.     if (!string.IsNullOrEmpty(consulta))
61.     {
62.         //Crea el Objeto que guarda todo el conocimiento del negocio
63.         SemanticIndexManager semanticIndexManager = new SemanticIndexManager();
64.
65.         //Crea la lista de documentos que serán el resultado de la búsqueda
66.         List<UrlDocument> allReponse = new List<UrlDocument>();
67.
68.         //Se busca en español o el ingles. Si es nulo busca por ambos.
69.         if (idioma == "Español")
70.         {
71.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, true);
72.         }
73.         else if (idioma == "Ingles")
74.         {
75.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, false);
76.         }
77.         else
78.             allReponse = semanticIndexManager.BuscarSemanticIndex(consulta, true); //Por defecto
79. busca español
80.
81.         //Armar la estructura de datos a retornar como resultado
82.         List<FeedXively> SensoresResult = new List<FeedXively>();
83.
84.         //Traemos todos los metadatos del sensor qdesde le servidor Xively, junto con sus mediciones
85.         foreach (UrlDocument urldoc in allReponse)
86.         {

```

```

85.         FeedXively xtemp = RetornarDatosSensor(urldoc.Id, fechaInicio, fechaFin);
86.         SensoresResult.Add(xtemp);
87.     }
88.
89.     return SensoresResult;
90. }
91. else
92.     return null;
93. }
94.
95. //Retorna un listado de conceptos directamente relacionados al concepto. Estos de obtienen
96. //de la ontología almacenada. Descarta términos no encontrados y no busca en wordnet.
97. //Si el concepto no esta en la ontología retorna una cadena vacia.
98. [WebMethod]
99. public string RetornarConceptos(string Concepto, string idioma = "Español")
100. {
101.     return BuscarConceptosOntologia(Concepto, idioma);
102. }
103.
104. //Retorna un listado de sensores y su localización con respecto a la localización dada y una Consulta
105. [WebMethod]
106. public List<FeedXively> RetornarMapaLugar(double Latitud, double Longitud, string Consulta, string
idioma = "Español", double radio = 100)
107. {
108.     string expansion = ObtenerConfig("utilizarExpansion");
109.
110.     //Crea el Objeto que guarda todo el conocimiento del negocio
111.     semanticIndexManager = new SemanticIndexManager();
112.
113.     //Crea una copia de la lista para eliminar los sensores que no se les puede calcular distancia
114.     List<FeedXively> FeedsConPosicion = new List<FeedXively>();
115.
116.     //Lista de los resultados de la búsqueda
117.     List<FeedXively> dtsResult = new List<FeedXively>();
118.
119.     //Buscar la consulta en el índice
120.     if (!string.IsNullOrEmpty(Consulta))
121.         dtsResult = BuscarFeeds(Consulta, idioma);
122.     else
123.         return null;
124.
125.     //Si encuentro sensores entonces validar los que entran en el radio de acción
126.     if(dtsResult != null)
127.     {
128.         List<string> clavesBorrar = new List<string>();
129.
130.         foreach (FeedXively dtr in dtsResult)
131.         {
132.             if (dtr.feed.location != null)
133.             {
134.                 //La funcion establece que el sensor este en el rango de N kms del lugar
especificado. Si esta devuelve la distancia, sino devuelve cero
135.                 double distancia = semanticIndexManager.VerificarSensoresenLugar(Latitud,
136.                                         Longitud, Convert.ToDouble(dtr.feed.location.lat,
culture),
137.                                         Convert.ToDouble(dtr.feed.location.lon, culture),
radio);
138.                 if (distancia > 0)
139.                 {
140.                     //ListaResultados.Add(indice, distancia);
141.                     GeonameNode nodolugardistancia = new GeonameNode();
142.                     nodolugardistancia.Latitud = Latitud.ToString();
143.                     nodolugardistancia.Longitud = Longitud.ToString();
144.                     lugar ld = new lugar();
145.                     ld.nodoGeoname = nodolugardistancia;
146.                     ld.distancia = distancia;
147.                     //Creamos el espacio en la variable de lista de lugares relacionada al sensor
148.                     dtr.lugares = new List<lugar>();
149.                     dtr.lugares.Add(ld);
150.                     FeedsConPosicion.Add(dtr);
151.                 }
152.             }
153.         }
154.     }
155. }

```

```

156.         //Finalmente retorna la lista con los sensores correspondientes
157.         return FeedsConPosicion;
158.     }
159.
160.     //Retorna un listado de sensores y su localización con respecto a la localización dada y una Consulta
161.     //Adicional retorna los datapoints de la fechas dadas
162.     [WebMethod]
163.     public List<FeedXively> RetornarMapaLugarDatapoints(double Latitud, double Longitud, string Consulta,
DateTime fechaInicio, DateTime fechaFin, string idioma = "Español", double radio = 100)
164.     {
165.         string expansion = ObtenerConfig("utilizarExpansion");
166.
167.         //Crea el Objeto que guarda todo el conocimiento del negocio
168.         semanticIndexManager = new SemanticIndexManager();
169.
170.         //Crea una copia de la lista para eliminar los sensores que no se les puede calcular distancia
171.         List<FeedXively> FeedsConPosicion = new List<FeedXively>();
172.
173.         //Lista de los resultados de la búsqueda
174.         List<FeedXively> dtsResult = new List<FeedXively>();
175.
176.         //Buscar la consulta en el índice
177.         if (!string.IsNullOrEmpty(Consulta))
178.             dtsResult = BuscarFeedsDataPoints(Consulta, fechaInicio, fechaFin, idioma);
179.         else
180.             return null;
181.
182.         //Si encuentro sensores entonces validar los que entran en el radio de acción
183.         if (dtsResult != null)
184.         {
185.             List<string> clavesBorrar = new List<string>();
186.
187.             foreach (FeedXively dtr in dtsResult)
188.             {
189.                 if (dtr.feed.location != null && dtr.feed.location.lat != null && dtr.feed.location.lon
190.                 != null)
191.                 {
192.                     //La funcion establece que el sensor este en el rango de N kms del lugar
193.                     //especificado. Si esta devuelve la distancia, sino devuelve cero
194.                     double distancia = semanticIndexManager.VerificarSensoresenLugar(Latitud,
195.                     Longitud, Convert.ToDouble(dtr.feed.location.lat,
196.                     culture),
197.                     Convert.ToDouble(dtr.feed.location.lon, culture),
198.                     radio);
199.
200.                     if (distancia > 0)
201.                     {
202.                         //ListaResultados.Add(indice, distancia);
203.                         GeonameNode nodolugardistancia = new GeonameNode();
204.                         nodolugardistancia.Latitud = Latitud.ToString();
205.                         nodolugardistancia.Longitud = Longitud.ToString();
206.                         lugar ld = new lugar();
207.                         ld.nodoGeoname = nodolugardistancia;
208.                         ld.distancia = distancia;
209.                         //Creamos el espacio en la variable de lista de lugares relacionada al sensor
210.                         dtr.lugares = new List<lugar>();
211.                         dtr.lugares.Add(ld);
212.                         FeedsConPosicion.Add(dtr);
213.                     }
214.                 }
215.             }
216.
217.         //Finalmente retorna la lista con los sensores correspondientes
218.         return FeedsConPosicion;
219.     }
220.
221.     //Retorna un listado de Sensores y su localización con respecto a una localización dada y un conjunto
222.     //se sensores
223.     [WebMethod]
224.     public List<FeedXively> RetornarMapaLugarListaSensores(double Latitud, double Longitud,
List<FeedXively> dtsResult, string idioma = "Español", double radio = 100)
225.     {
226.         //Crea el Objeto que guarda todo el conocimiento del negocio
227.         semanticIndexManager = new SemanticIndexManager();
228.

```

```

224.         //Crea una copia de la lista para eliminar los sensores que no se les puede calcular distancia
225.         List<FeedXively> FeedsConPosicion = new List<FeedXively>();
226.
227.         //Si encuentro sensores entonces validar los que entran en el radio de acción
228.         if (dtsResult != null)
229.         {
230.             List<string> clavesBorrar = new List<string>();
231.
232.             foreach (FeedXively dtr in dtsResult)
233.             {
234.                 if (dtr.feed.location != null)
235.                 {
236.                     //La funcion establece que el sensor este en el rango de N kms del lugar
237.                     //especificado. Si esta devuelve la distancia, sino devuelve cero
238.                     double distancia = semanticIndexManager.VerificarSensoresenLugar(Latitud,
239.                                             Longitud, Convert.ToDouble(dtr.feed.location.lat,
240.                                             Convert.ToDouble(dtr.feed.location.lon, culture),
241.                                             radio);
242.
243.                     if (distancia > 0)
244.                     {
245.                         //ListaResultados.Add(indice, distancia);
246.                         GeonameNode nodolugardistancia = new GeonameNode();
247.                         nodolugardistancia.Latitud = Latitud.ToString();
248.                         nodolugardistancia.Longitud = Longitud.ToString();
249.                         lugar ld = new lugar();
250.                         ld.nodoGeoname = nodolugardistancia;
251.                         ld.distancia = distancia;
252.                         //Creamos el espacio en la variable de lista de lugares relacionada al sensor
253.                         dtr.lugares = new List<lugar>();
254.                         dtr.lugares.Add(ld);
255.                         FeedsConPosicion.Add(dtr);
256.                     }
257.                 }
258.             }
259.         }
260.         //Finalmente retorna la lista con los sensores correspondientes
261.         return FeedsConPosicion;
262.     }
263.
264.     //Realiza un proceso de expansión de la consulta del usuario utilizando la ontología para hacerla más
265.     exacta
266.     //Para esto primero Retorna conceptos, posteriormente busca en wordnet conceptos no encontrados y
267.     finalmente
268.     //enlaza los términos no identificados al final de la consulta por considerarse importantes para el
269.     usuario.
270.     [WebMethod]
271.     public string ExpandirConsultaConceptosOntologia(string Consulta, string idioma)
272.     {
273.         return ExpandirConsulta(Consulta, idioma);
274.     }
275.     #endregion
276.
277.     #region "Servicios de Configuración del Índice Semántico"
278.
279.     //Método Adicional: Permite recuperar los valores de las claves almacenadas en el Web.Config
280.     //con el fin de establecer la configuración actual del índice
281.     [WebMethod]
282.     public string ObtenerConfiguracion(string key)
283.     {
284.         if (!string.IsNullOrEmpty(key))
285.             return ObtenerConfig(key);
286.         else
287.             return "La clave no puede ser vacía o nula";
288.     }
289.
290.     //Método Adicional: Permite asignar un valor a una clave específica en el Web.config con el fin
291.     //de establecer una nueva configuración de indexación
292.     [WebMethod]
293.     public string AsignarConfiguracion(string key, string valor)
294.     {
295.         if (!string.IsNullOrEmpty(key) && !string.IsNullOrEmpty(valor))
296.             return AsignarConfig(key, valor);
297.     }

```

```

293.         else
294.             return "La clave o el valor no pueden ser nulos o vacios";
295.     }
296.
297.     //Método Adicional: Una vez se han cambiado las configuraciones o se desea inicia una nueva ontología
de dominio
298.     //Es necesario crear el índice semántico. Este método crea el índice teniendo en cuenta si lo hace
desde el servidor
299.     //de objetos Xively o desde los archivos locales almacenados en una indexación previa.
300.     [WebMethod]
301.     public string CrearIndiceSemantico()
302.     {
303.         //Crea el Objeto que guarda todo el conocimiento del negocio
304.         semanticIndexManager = new SemanticIndexManager();
305.         if (ObtenerConfig("BDDfuente") == "Xively")
306.         {
307.             semanticIndexManager.CrearIndiceSemantico(true);
308.             return "El índice se ha creado correctamente desde el servidor Xively.";
309.         }
310.         else
311.         {
312.             semanticIndexManager.CrearIndiceSemantico(false);
313.             return "El índice se ha creado correctamente desde la Base de Datos Local de Archivos JSON.";
314.         }
315.     }
316.
317.     //Método Adicional: Permite cambiar y cargar los conceptos a la BD del servicio de una nueva
ontología.
318.     //Por defecto crea nuevamente el índice desde el servidor de objetos Xively.
319.     [WebMethod]
320.     public string CargarOntologiaeIndexar(string nuevaontologia)
321.     {
322.         string NombreArchivoNuevaOntologia = Path.GetFileName(nuevaontologia);
323.         string camino = SemanticIndexManager.RutaOntologiaSinArchivo;
324.         string conceptoscargados = string.Empty;
325.
326.         //Verificar ue el archivo de la ontologia existe
327.         if (File.Exists(nuevaontologia))
328.         {
329.             //Copia la ontologia al directorio de ontologías
330.             File.Copy(nuevaontologia, camino + NombreArchivoNuevaOntologia, true);
331.             //Se debe cargar los conceptos de la ontologia para el proceso de expansión de consulta
332.             conceptoscargados = CargarConceptosOntologia(nuevaontologia);
333.
334.             //Si el proceso fue Exitoso se cambia la nueva ruta de la Ontología
335.             AsignarConfig("FileOntology", NombreArchivoNuevaOntologia);
336.
337.             //Recupera el objeto de logica de negocio de la aplicación para que el cambio de ontologia
tenga efecto
338.             semanticIndexManager = new SemanticIndexManager();
339.
340.             //Finalmente Crea el índice semántico Nuevamente desde Xively
341.             semanticIndexManager.CrearIndiceSemantico(true);
342.
343.             //Retorna los conceptos cargados
344.             return conceptoscargados;
345.         }
346.         return "ERROR - La ruta del archivo de la ontología no es correcta o el archivo no existe";
347.     }
348.
349.     //Método Adicional: Permite cargar un feed de Xively específico a la BDD
350.     //Posteriormente lo indexa en el índice semántico
351.     [WebMethod]
352.     public string CargarFeedXivelyBDD(string feedId)
353.     {
354.         try
355.         {
356.             //Recupera el objeto de logica de negocio de la aplicación para que el cambio de ontologia
tenga efecto
357.             semanticIndexManager = new SemanticIndexManager();
358.             semanticIndexManager.CargarFeedXivelyBDD(feedId);
359.             return "El índice se ha Actualizado correctamente desde el servidor Xively.";
360.         }
361.         catch(Exception ex)
362.         {

```

```

363.         return "Ocurrió un error para cargar e indexar el Feed: " + ex.Message;
364.     }
365. }
366.
367. #endregion
368.
369. #region "Servicios de Geolocalización de Sensores del Índice Semántico"
370.
371. //Método Adicional: Permite Buscar un lugar geográfico por su nombre en el servicio Web de GeoNames
372. [WebMethod]
373. public List<GeonameNode> ObtenerLocalizacion(string lugar)
374. {
375.     //Crea el Objeto que guarda todo el conocimiento del negocio
376.     semanticIndexManager = new SemanticIndexManager();
377.
378.     return semanticIndexManager.ObtenerLocalizaciones(lugar);
379. }
380.
381. //Método Adicional: Permite determinar y retornar los conceptos que son lugares geográficos y con sus
    posiciones
382. //geolocalizadase a través del servicio de Geonames.
383. [WebMethod]
384. public List<GeonameNode> ObtenerCiudadesConsulta(string Consulta)
385. {
386.     //Crea el Objeto que guarda todo el conocimiento del negocio
387.     semanticIndexManager = new SemanticIndexManager();
388.
389.     return semanticIndexManager.ObtenerCiudadesConsulta(Consulta);
390. }
391.
392. //Obtiene un solo lugar dadas sus coordenadas
393. [WebMethod]
394. public GeonameNode ObtenerLugardeCoordenadas(string latitud, string longitud)
395. {
396.     //Crea el Objeto que guarda todo el conocimiento del negocio
397.     semanticIndexManager = new SemanticIndexManager();
398.
399.     return semanticIndexManager.ObtenerLugardeCoordenadas(latitud, longitud);
400. }
401.
402. //Obtiene Los nombres de los lugares de una lista de GeonameNodes que recibe
403. [WebMethod]
404. public List<GeonameNode> ObtenerListaLugarCoordenadas(string geonodesSerializados)
405. {
406.     //Crea el Objeto que guarda todo el conocimiento del negocio
407.     semanticIndexManager = new SemanticIndexManager();
408.
409.     //Deserializamos el objeto enviado
410.     List<GeonameNode> geonodes = geonodesSerializados.DeserializarTo<List<GeonameNode>>();
411.
412.     //Objeto copia para las operaciones
413.     List<GeonameNode> geonodestmp = new List<GeonameNode>();
414.
415.     foreach (GeonameNode geotmp in geonodes)
416.     {
417.         geonodestmp.Add(semanticIndexManager.ObtenerLugardeCoordenadas(geotmp.Latitud,
    geotmp.Longitud));
418.     }
419.
420.     return geonodestmp;
421. }
422.
423. //Obtiene las coordenadas de una lista. Adicionalmente Obtiene la jerarquía de lugares de cada nodo
424. [WebMethod]
425. public List<GeonameNode> ObtenerListaLugarCoordenadasJerarquica(string geonodesSerializados)
426. {
427.     //Crea el Objeto que guarda todo el conocimiento del negocio
428.     semanticIndexManager = new SemanticIndexManager();
429.
430.     //Deserializamos el objeto enviado
431.     List<GeonameNode> geonodes = geonodesSerializados.DeserializarTo<List<GeonameNode>>();
432.
433.     //Objeto copia para las operaciones
434.     List<GeonameNode> geonodestmp = new List<GeonameNode>();
435.

```

```

436.         foreach (GeonameNode geotmp in geonodes)
437.         {
438.             List<GeonameNode> geonodestmpj = new List<GeonameNode>();
439.             geonodestmpj = ObtenerLugardeCoordenadasJerarquia(geotmp.Latitud, geotmp.Longitud);
440.             //Añadimos el último nodo que contiene el lugar buscado
441.             if(geonodestmpj != null)
442.                 geonodestmp.Add(geonodestmpj[geonodestmpj.Count-1]);
443.         }
444.
445.         return geonodestmp;
446.     }
447.
448.     //Obtiene las coordenadas de una lista. Adicionalmente Obtiene de la jerarquia de lugares de cada
nodo
449.     //La primera división política que corresponde a dicho lugar
450.     [WebMethod]
451.     public string ObtenerListalugarCoordenadas_AM1(string geonodesSerializados)
452.     {
453.         //Crea el Objeto que guarda todo el conocimiento del negocio
454.         semanticIndexManager = new SemanticIndexManager();
455.
456.         //Desserializamos el objeto enviado
457.         List<GeonameNode> geonodes = geonodesSerializados.DeserializarTo<List<GeonameNode>>();
458.
459.         //Objeto copia para las operaciones
460.         List<GeonameNode> geonodestmp = new List<GeonameNode>();
461.
462.         foreach (GeonameNode geotmp in geonodes)
463.         {
464.             List<GeonameNode> geonodestmpj = new List<GeonameNode>();
465.             geonodestmpj = semanticIndexManager.ObtenerLugardeCoordenadasJerarquia(geotmp.Latitud,
geotmp.Longitud);
466.
467.             if (geonodestmpj != null)
468.                 foreach (GeonameNode getmpj in geonodestmpj)
469.                 {
470.                     if (getmpj.fcode == "ADM1")
471.                         geonodestmp.Add(getmpj);
472.                 }
473.             }
474.             //Serializar la lista de salida
475.             return geonodestmp.SerializarToXml();
476.         }
477.
478.         //Obtiene la jerarquía de lugares de un punto en el mapa
479.         [WebMethod]
480.         public List<GeonameNode> ObtenerLugardeCoordenadasJerarquia(string latitud, string longitud)
481.         {
482.             //Crea el Objeto que guarda todo el conocimiento del negocio
483.             semanticIndexManager = new SemanticIndexManager();
484.
485.             return semanticIndexManager.ObtenerLugardeCoordenadasJerarquia(latitud, longitud);
486.         }
487.
488.         //Obtiene la jerarquía dado un geonameId
489.         [WebMethod]
490.         public List<GeonameNode> GeoNames_Hierarchy(int GeonameId)
491.         {
492.             //Crea el Objeto que guarda todo el conocimiento del negocio
493.             semanticIndexManager = new SemanticIndexManager();
494.
495.             return semanticIndexManager.GeoNames_Hierarchy(GeonameId);
496.         }
497.
498.     #endregion
499.
500.     #region "Métodos de los Sensores"
501.
502.     //Este método devuelve en formato FeedXively los Metadatos del Sensor
503.     [WebMethod]
504.     public FeedXively RetornarMetadatosSensor(string IdSensor)
505.     {
506.         //Crea el Objeto que guarda todo el conocimiento del negocio
507.         semanticIndexManager = new SemanticIndexManager();
508.

```

```

509.         //Obtener el objeto de la BDD
510.         FeedXively xtemp = semanticIndexManager.ObtenerJSONFeedBDD(IdSensor);
511.
512.         return xtemp;
513.     }
514.
515.     //Este método devuelve en formato json los Metadatos del Sensor desde Xively
516.     [WebMethod]
517.     public string RetornarJsonSensor(string IdSensor)
518.     {
519.         //Crea el Objeto que guarda todo el conocimiento del negocio
520.         semanticIndexManager = new SemanticIndexManager();
521.
522.         //Obtener el JSON de la BDD
523.         string xtemp = semanticIndexManager.ObtenerJsonFeed(ConfigurationManager.AppSettings["APIkey"],
IdSensor);
524.
525.         //Retornar el json
526.         return xtemp;
527.     }
528.
529.     //Este método obtiene información de los datos que mide el sensor.
530.     //Se retorna todo el feed con los datapoints
531.     [WebMethod]
532.     public FeedXively RetornarDatosSensor(string feedID, DateTime fechaInicio, DateTime fechaFin)
533.     {
534.         //Crea el Objeto que guarda todo el conocimiento del negocio
535.         semanticIndexManager = new SemanticIndexManager();
536.
537.         //Obtener el JSON de la BDD
538.         return semanticIndexManager.RetornarDatosSensor(feedID, fechaInicio, fechaFin);
539.     }
540.
541.     //Este método retorna los datapoints de un datastream específico, de un feed específico
542.     [WebMethod]
543.     public string RetornarDatapointsFeed(string feedID, string DatastreamId, string fechaInicio, string
fechaFin)
544.     {
545.         //Crea el Objeto que guarda todo el conocimiento del negocio
546.         semanticIndexManager = new SemanticIndexManager();
547.
548.         //Obtener el JSON de la BDD
549.         string datapoints = semanticIndexManager.RetornarDatapointsFeed(feedID, DatastreamId,
Convert.ToDateTime(fechaInicio),
550.                                     Convert.ToDateTime(fechaFin));
551.
552.         return datapoints;
553.     }
554.
555.     #endregion
556.
557.     #region "Métodos Internos del Servicio de Índice Semántico"
558.
559.     private DataSet SearchSemanticIndex(string Consulta, string idioma)
560.     {
561.         //string idioma = ObtenerConfig("idiomaBusqueda");
562.         string expansion = ObtenerConfig("utilizarExpansion");
563.
564.         //Crea el Objeto que guarda todo el conocimiento del negocio
565.         semanticIndexManager = new SemanticIndexManager();
566.
567.         //Crea la lista de documentos que serán el resultado de la búsqueda
568.         List<UrlDocument> allReponse = new List<UrlDocument>();
569.
570.         if (expansion == "Si")
571.         {
572.             //Realiza la expansión de la consulta a través del servicio.
573.             Consulta = ExpandirConsulta(Consulta, idioma);
574.         }
575.
576.         //Se busca en español o el ingles. Si es nulo busca por ambos.
577.         if (idioma == "Español")
578.         {
579.             allReponse = semanticIndexManager.BuscarSemanticIndex(Consulta, true);
580.         }
581.         else if(idioma == "Ingles")

```

```

582.         {
583.             allReponse = semanticIndexManager.BuscarSemanticIndex(Consulta, false);
584.         }
585.         else
586.             allReponse = semanticIndexManager.BuscarSemanticIndex(Consulta, true); //Por defecto busca
español
587.
588.         //para DATASET que se devolviera como resultado
589.         UriDocument objUrld = new UriDocument();
590.         DataSet nuevoDS = new DataSet();
591.         DataTable table1 = new DataTable("Objetos Encontrados");
592.
593.         table1.Columns.Add("Id");
594.         table1.Columns.Add("Title");
595.         table1.Columns.Add("Titulo_HTML");
596.         table1.Columns.Add("Url");
597.         table1.Columns.Add("URL_Sensor");
598.         table1.Columns.Add("Resumen");
599.         table1.Columns.Add("Tags");
600.         table1.Columns.Add("Localizacion");
601.         table1.Columns.Add("Dominio");
602.         table1.Columns.Add("Datastreams_feed");
603.         table1.Columns.Add("Website");
604.         table1.Columns.Add("Elevacion");
605.         table1.Columns.Add("Latitud");
606.         table1.Columns.Add("Longitud");
607.         //Campos para el PageRangkin
608.         table1.Columns.Add("Conceptos"); //Conceptos por los cuales fue encontrado en Xively
609.         table1.Columns.Add("Consulta"); //Consulta por la cual fue seleccionado el sensor
610.         table1.Columns.Add("Distancia"); //Distancia para efectos de un lugar específico
611.         table1.PrimaryKey = new DataColumn[] { table1.Columns["Id"] };
612.
613.         foreach (UriDocument urldoc in allReponse)
614.         {
615.             table1.Rows.Add(urldoc.Id, urldoc.Tittle, urldoc.TituloHTML(), urldoc.URL,
urldoc.URLMostrar(), urldoc.Resume,
616.                 urldoc.Tags, urldoc.Localizacion_name, urldoc.Domain,
urldoc.Datastreams_feed, urldoc.Website,
617.                 urldoc.Elevacion, urldoc.Latitud, urldoc.Longitud, urldoc.ConceptosLista(),
Consulta);
618.         }
619.
620.         nuevoDS.Tables.Add(table1);
621.         return nuevoDS;
622.     }
623.
624.     private string ExpandirConsulta(string Consulta, string idioma)
625.     {
626.         //Realiza la expansión de la consulta a través del servicio.
627.         ExpansionConsulta ex = new ExpacionConsulta();
628.         return ex.ExpandirConsulta(SemanticIndexManager.RutaOntologia, Consulta, idioma);
629.     }
630.
631.     private string BuscarConceptosOntologia(string Concepto, string idioma)
632.     {
633.         //Realiza la expansión de la consulta a través del servicio.
634.         ExpacionConsulta ex = new ExpacionConsulta();
635.         return ex.RetornarConceptosOntologia(SemanticIndexManager.RutaOntologia, Concepto, idioma);
636.     }
637.
638.     private string CargarConceptosOntologia(string nuevaontologia)
639.     {
640.         srvExpansionConsulta.ExpansionConsulta ex = new srvExpansionConsulta.ExpacionConsulta();
641.         string conceptos = ex.CargarConceptos(nuevaontologia);
642.         return conceptos;
643.     }
644.
645.     private string ObtenerConfig(string key)
646.     {
647.         return ConfigurationManager.AppSettings[key];
648.     }
649.
650.     private string AsignarConfig(string key, string valor)
651.     {
652.         try

```

```

653.         {
654.             // Get the application configuration file.
655.             System.Configuration.Configuration config =
656.                 System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration("~/");
657.
658.             // Update the configuration file appSettings section.
659.             config.AppSettings.Settings.Remove(key);
660.             config.AppSettings.Settings.Add(key, valor);
661.
662.             // Save the configuration file.
663.             config.Save(System.Configuration.ConfigurationSaveMode.Modified);
664.
665.             //Configuración guardada correctamente
666.             return "ok";
667.         }
668.         catch (Exception ex)
669.         {
670.             return ex.Message;
671.         }
672.     }
673.
674.     #endregion

```

## 7. Diagramas de Despliegue, Componentes e Interfases del Índice y Buscador Semántico

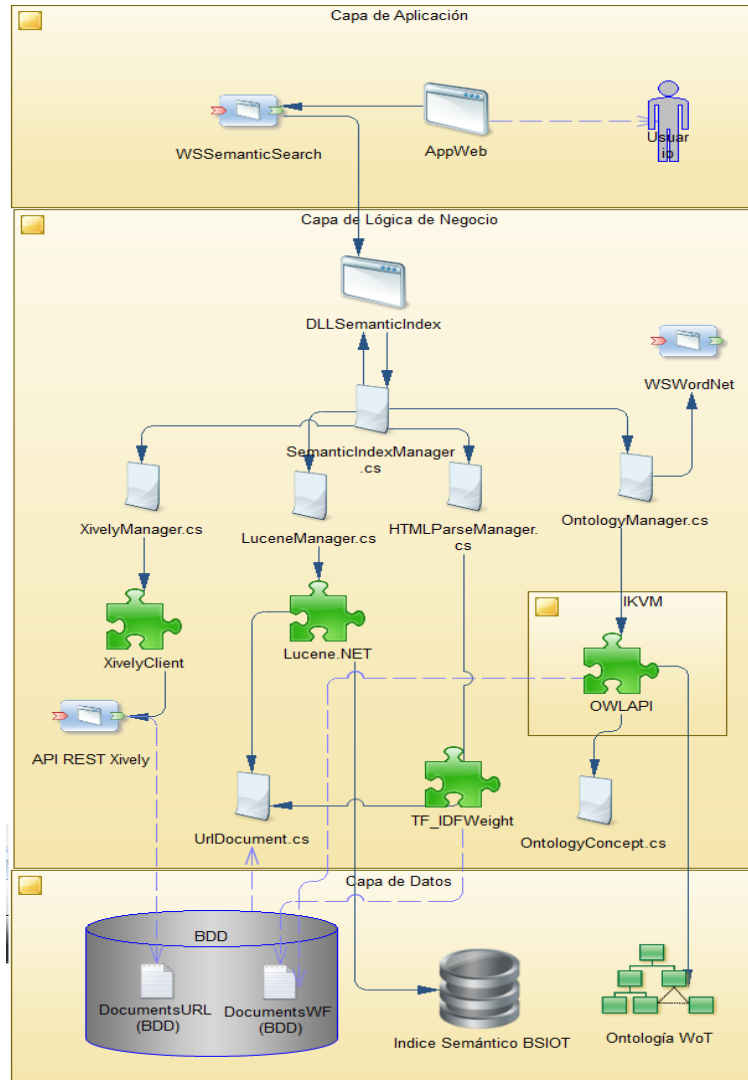
Para la descripción estructural del prototipo se dispone de un diagrama de componentes en la Figura 33, el cual representa los elementos funcionales principales y sus relaciones, con esto se pretende mostrar una vista estática del sistema.

En este diagrama se puede ver como se utilizó la arquitectura propuesta por el método en varias capas. Por otro lado se puede apreciar cómo en la capa de aplicación queda el servicio web del índice el cual es usado por el buscador. Así cualquier aplicación puede hacer uso del índice semántico construido.

En la capa de lógica de negocio se puede apreciar el uso de ciertos patrones de diseño de software como la clase proxy que termina su nombre en “manager”, las cuales se encargan de hacer la interfaz para las funcionalidades específicas y especializadas de cada componente utilizado (API's). La funcionalidad del índice queda encapsulada en una librería dll (DLLSemanticIndex), la cual puede ser reutilizada en otros proyecto con el framework de .Net para desarrollar otras aplicaciones de indexación semántica en la IoT.

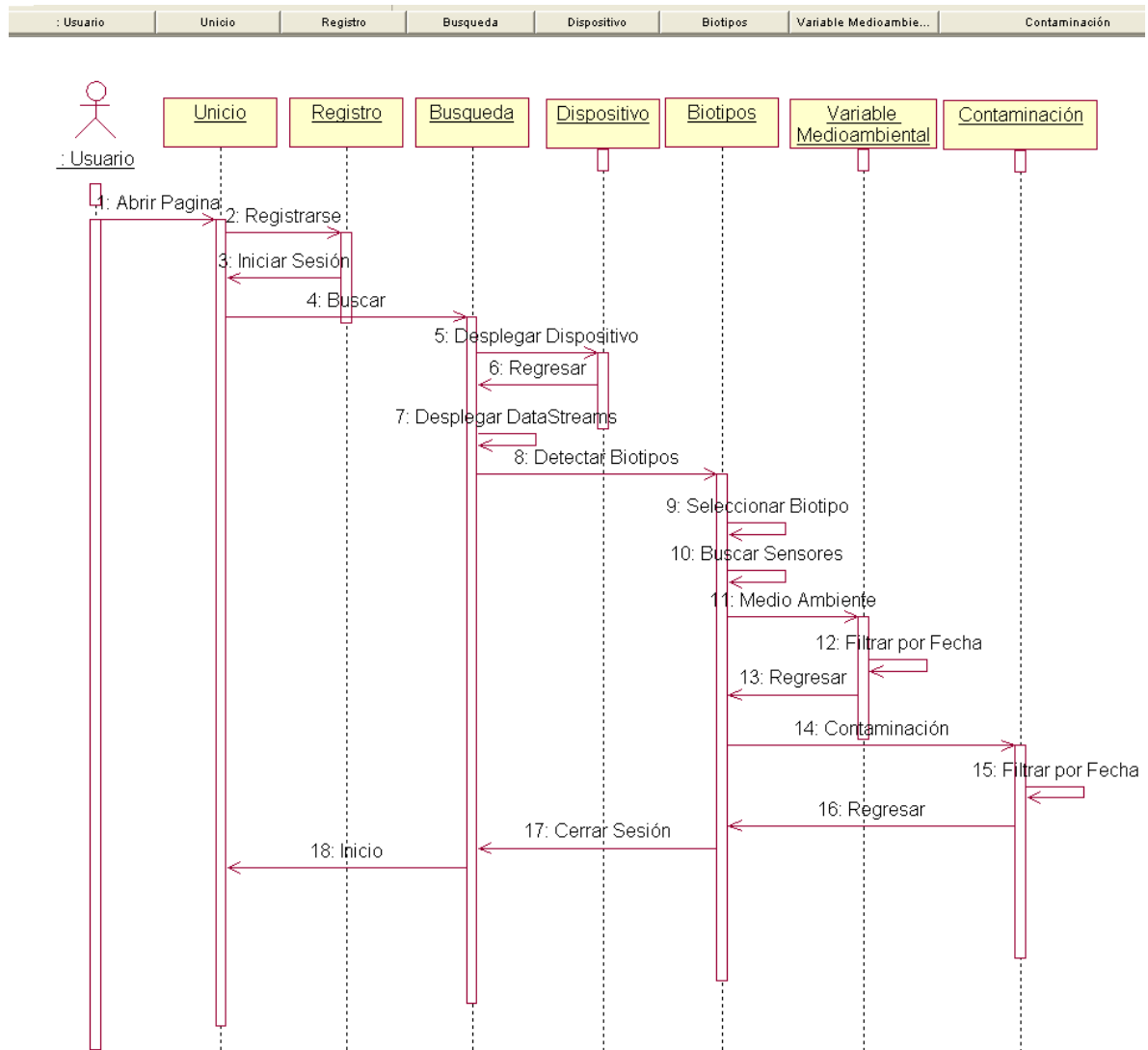
Finalmente, en la capa de datos se puede encontrar los archivos de BDD, dos archivos para carga rápido de metadatos, la ontología y el propio índice semántico que administrado con Lucene.Net.

Figura 33. Diagrama de Componentes para SIMIOT



Para la descripción funcional a nivel de interfaces de usuario en el sitio web del prototipo SIMIOT, se presenta un diagrama de secuencia de interfaces web en la Figura 34, que pretende mostrar una vista funcional del sistema. En el diagrama se puede ver la vida del usuario, la cual además de permitirle realizar búsquedas, puede detectar biotipos y realizar una búsqueda avanzada de variables medioambientales y de Contaminación ambiental. Pudiendo obtener incluso los datos en tiempo real de los sensores encontrados.

**Figura 34. Diagrama de Secuencia de Interfaces para SIMIOT**



Por ultimo se muestran las interfaces más significativas del prototipo buscador SIMIOT. La interfaz de Inicio, Busqueda, Biotipos. Se pueden visualizar en las Figura 35, Figura 36, Figura 37.

Figura 35. Interfaz de Inicio para SIMIOT



En la Figura 35 se puede apreciar como el usuario puede registrarse o ir al módulo de administración en caso de tener los privilegios. Una vez registrado e iniciado sesión se muestra su nombre debajo del cuadro de búsqueda de texto. Antes de realizar la búsqueda el usuario puede seleccionar su idioma preferido o una combinación de ambos, esta selección afectara fundamentalmente la expansión de consulta con el fin de obtener sensores en dichos idiomas en las primeras posiciones.

Una vez que el usuario ha comprendido que tipo de preguntar realizar y en que dominio, puede hacer la misma en lenguaje natural. Es importante que realice la consulta en el lenguaje seleccionado, ya que esto permitirá realizar la expansión más adecuadamente.

Figura 36. Interfaz de Búsqueda para SIMIOT

SEMANTIC SEARCH ON  
INTERNET OF THINGS

INICIO • REGISTRARSE • ADMINISTRAR • ACERCA...

---

Soportado en el Servidor de Objetos [Xively](#) .

Presión

Cerrar sesión swuswo - Idioma Preferido:  Español  Inglés  Ambos

**Resultados de la búsqueda ( + &Presión). Total 12 - Tiempo Consulta: 0,2496004 segundos**

Tabla Resultados Sensores de la Consulta:

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
100184	<a href="#">Prueba Estación Meteorológica Oviedo</a>	Oviedo	Comprobando estabilidad del firmware y lecturas de presión atmosférica.	43.3561397098407	-5.8751106262207
97233	<a href="#">prueba Arduino</a>	Miranda de Ebro	pruebas termostato caldera	42.6855270287249	-2.93124675750732
101626	<a href="#">WeatherDog</a>		Información meteorológica		
102976	<a href="#">Temperatura, Humedad y Presión ambientales</a>	Tica HQ	Sensor Sension SHT11 midiendo temperatura y humedad en edificio La Concepción 81, piso 17 a la intersección. Construido alrededor de un Arduino Ethernet. Actualizado 8 de febrero: Agregado sensor de presión atmosférica y temperatura MS5607, ubicado al interior de la oficina.	-33.4246787875303	-70.6152275204659
126734	<a href="#">Estación meteorológica</a>	Torrejón de Ardoz	Medición de luz, humedad, temperatura y presión atmosférica.	40.458507307739	-3.44850540161133
118635	<a href="#">MyWeatherDog</a>			38.3425271297471	-0.497453212738037
8809	<a href="#">ESTACIÓN METEOROLÓGICA Y CONTROL AEROGENERADOR SAVONIUS-3P</a>	Medina de Pomar	PRUEBA: condiciones de temperatura, humedad, presión atmosférica, velocidad del viento, rpm turbina, consumo estación, temperatura estator de turbina y Voltaje Baterías. Añadidos sensores de seconwind para viento... EN PRUEBAS AÚN!!! de momento pospuesto hasta final del verano 2012 por motivos personales...	42.9311648577218	-3.48549842834473
11279	<a href="#">Estación meteorológica - Miguelitis</a>	Huércal-Overa Almería Spain	Estación meteorológica.	37.4084525878418	-1.92991375923157
55882	<a href="#">VEGAMET</a>	CHILE	Meteorología aficionada Chile Creador: Carlos Vega www.vegamet.cl vortexchile.byethost5.com	-33.4788069229576	-70.5666795372963
83960	<a href="#">Meteo A Rua</a>	A Rua de Valdeorras		42.3955337869864	-7.10002183914185
83984	<a href="#">Meteo A Gudiña</a>	A Gudiña	Datos meteorológicos en A Gudiña.	42.0604931122227	-7.11976289749146
76711	<a href="#">El tiempo en Leandro N. Alem</a>	Leandro N. Alem, Misiones	Estación Meteorológica Alem I - Información meteorológica de la ciudad de Leandro N. Alem, Misiones.	-27.603889	-55.323611

Universidad del Cauca / Universidad Industrial de Santander © Copyright © 2014  
 Designed by: Your Provider of [Website Templates](#)  
 Para mayor información: [manzamb@unicauca.edu.co](mailto:manzamb@unicauca.edu.co) - [ninochucho@hotmail.com](mailto:ninochucho@hotmail.com)

En la **Figura 36**, se puede apreciar el resultado de la búsqueda realizada por el usuario, la cual queda en una lista de sensores ordenados por relevancia, de los cuales se espera cumplan con las necesidades de información del usuario. Debajo de la consulta se puede apreciar la expansión realizada, la cual es enviada al motor de búsqueda. En el identificador de la tabla corresponde al identificador del servidor Xively y dando clic en el enlace los usuarios pueden ver los metadatos de los mismos directamente en el portal de Xively.

En la parte derecha del cuadro de consulta se puede apreciar que aparece un botón llamado "Detectar Biotipos", este botón permite realizar una búsqueda organizada por los lugares geográficos en los que se encuentran los sensores y los cuales podrían en conjunto proveer una información más completa del caso de contaminación medioambiental.

Figura 37. Interfaz de Biotipo para SIMIOT

**SEMANTIC SEARCH ON  
INTERNET OF THINGS**

INICIO • REGISTRARSE • ADMINISTRAR • ACERCA...

---

Soportado en el Servidor de Objetos [Xorely](#).

Contaminación del aire

Cerrar sesión swuswo - Idioma Preferido:  Español  Inglés  Ambos

**Resultados de la búsqueda en el Biotipo: (Departamento de Santander, Colombia). Total 5 - Tiempo Consulta: 37,22166 segundos**

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
1155676551	Floridablanca		Simulador Estación Meteorológica del Departamento de Santander. Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire. Escala de Monitoreo SVCA: Tipo Regional: Radio 7.06095774037291-73.0876421928406 de más de 20km hasta el área total de la jurisdicción.		
1942345911	Lebrija	Lebrija	Simulador Estación Meteorológica del Departamento de Santander. Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire. Escala de Monitoreo SVCA: Tipo Regional: Radio 7.11471566995142-73.2183408737183 de más de 20km hasta el área total de la jurisdicción.		
576461558	Girón	Girón	Simulador Estación Meteorológica del Departamento de Santander. Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire. Escala de Monitoreo SVCA: Tipo Regional: Radio 7.07221397324112-73.1683874130249 de más de 20km hasta el área total de la jurisdicción.		
452485984	Bucaramanga	Bucaramanga	Simulador Estación Meteorológica del Departamento de Santander. Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire. Escala de Monitoreo SVCA: Tipo Regional: Radio 7.11331	-73.120468	
140284357	Piedecuesta	Piedecuesta	Simulador Estación Meteorológica del Departamento de Santander. Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire. Escala de Monitoreo SVCA: Tipo Regional: Radio 6.98679819277429-73.048267364502 de más de 20km hasta el área total de la jurisdicción.		

Biotipos y Ubicación de los Sensores


Departamento de Santander, Colo Lat: 7, Lng: -73.25 Radio (km): 100

En la Figura 37 se puede apreciar el resultado de hacer un análisis por Biotipo. La interfaz cambia presentando un control que permite seleccionar el área geográfica específica detectada y posteriormente ubicar en el mapa los sensores que pueden retornar información medioambiental. Así el usuario puede refinar su consulta y empezar a relacionar la información de los sensores que más se ajustan a su necesidad de información. Es posible aumentar o disminuir el radio de acción del Biotipo, por defecto se tomaron 100 kilómetros, pero el usuario puede cambiarlo y realizar una nueva búsqueda de biotipos con el botón “Buscar Sensores”.

Finalmente, buscando hacer una implementación más personalizada del dominio de aplicación, se colocaron dos funcionalidades adicionales que se ejecutan con los botones: “Medio ambiente” y “Contaminación”, los cuales se presentan en la siguiente figura.

Figura 38: Búsqueda de Sensores por Variables Ambientales y Biotipo

**SEMANTIC SEARCH ON  
INTERNET OF THINGS**



[INICIO](#) • [REGISTRARSE](#) • [ADMINISTRAR](#) • [ACERCA...](#)

---

Selección del Biotipo y Area de Acción:

Lista de Biotipos Encontrados:  Radio (km):  [Analizar Biotipo](#)

Rango de Análisis de Datos (MM-dd-yyyy hh:mm:ss):

Desde   hasta

### DATOS GENERALES DEL BIOTIPO

**Id de Biotipo:** 3336900  
**Nombre:** Castille and León  
**País:** Spain  
**Latitud:** 41.66667  
**Longitud:** -4.25  
**Radio:** 500

### MEDICIÓN DE VARIABLES MEDIOAMBIENTALES

**EDATOPO**

Magnetismo

---

**HIDROTOPO**

Humedad

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
48102	Twitter		En este feed, son presentadas las mediciones en tiempo real de las variables: Temperatura (°C) Humedad(%) Luminosidad(%) Punto de Rocío(°C) Lugar de medición: Montilla, (Córdoba) España	37.5903971879045	-4.64159488677979
8478	Acuario v2.0	alcorcon	Control de Acuario Marino	40.3468057815893	-3.82839202880859
126834	Estación Meteorologica Arduino	Leon	Estación meteorológica en Arduino, sensores SHT15, BMP-085, DS18B20	42.5941452082974	-5.58636498956298
60027	Melero Sala Ordeño	Luanco		43.6076058804288	-5.83090804100038
62901	Temper			40.4519191897175	-3.69501650333405
499634275	Acabados Calcesteria	12150 Villafranca del Cid, Castellón , Spain		40.429787	-0.247986
81786	Yoctopuce	Majadahonda		40.4764633838589	-3.84774684906006
920954703	Gelotemp			40.5633673	-3.6128291

---

**CLIMATOPO**

Temperatura

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
95360	Remota Queso	Zaragoza	Medida Temperatura Humedad	41.6425113062437	-0.878804535605013
46381	Maison	Bordeaux	Enter a description of the Feed	44.7691618526244	-0.57403564453125
72792	Temperaturas Estanque	estanque		40.8678662948583	-4.38071250915527

Viento

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
29272	123456	Logroño		42.2935641921701	-2.5048828125
55741	ESTACIÓN METEOROLÓGICA ARDUINO	SALAMANCA	Estación meteorológica remota usando la plataforma open source arduino	40.94671366508	-5.64697265625

En la Figura 38 se puede apreciar el resultado por análisis de variables ambientales, las cuales son contenidas en la ontología de dominio. En este caso ya no utiliza el resultado de los sensores retornados sino el biotipo detectado. Así, se puede cambiar el radio de acción del Biotipo, las fechas de sensado y presentar una página categorizada con cada una de las variables ambientales, los sensores encontrados y la posibilidad de consultar sus mediciones históricas y/o actuales. Lo mismo se hace con la opción de “Contaminación”, pero este se enfoca en encontrar las variables relacionadas con la contaminación del aire en dicho Biotipo.

Figura 39: Búsqueda de Sensores por Contaminación del Aire y por Biotipo

INICIO • REGISTRARSE • ADMINISTRAR • ACERCA...

SEMANTIC SEARCH ON INTERNET OF THINGS

Selección del Biotipo y Area de Acción:

Buscador Semántico en Internet de Objetos - Google Chrome

localhost:3610/AppWeb/AnalizarContaminacion.aspx?id=0

INICIO • REGISTRARSE • ADMINISTRAR • ACERCA...

Selección del Biotipo y Area de Acción:

Lista de Biotipos Encontrados: Departamento de Santander, C Radio (km): 100 [Analizar Biotipo]

Rango de Análisis de Datos (MM-dd-yyyy hh:mm:ss):

Desde 05/10/2014 04:12:55 hasta 05/10/2014 14:12:55

### DATOS GENERALES DEL BIOTIPO

Id de Biotipo: 3668578  
 Nombre: Departamento de Santander  
 País: Colombia  
 Latitud: 7  
 Longitud: -73.25  
 Radio: 100

### MEDICIÓN DE CONTAMINACION

#### CONTAMINACION DEL AIRE

Agentes Contaminantes

Vanadio

Plomo

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
576461558	Girón	Girón	Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción.	7.07221397324112-73.1683874130249	

Dispositivo	Datastream	Fecha	Valor Actual	Valor Mínimo	Valor Máximo
576461558	Dioxido_de_Azufre_SO2	03/30/2014 05:00:22	461.00	50.0	496.0
576461558	Dioxido_de_nitrogeno_NO2	03/30/2014 18:00:08	361.00	50.0	983.0
576461558	Monoxido_de_Carbono_CO	03/28/2014 21:00:04	10.00	10.0	50.0
576461558	Ozono_O3	03/30/2014 19:00:11	292.00	50.0	992.0
576461558	PM10_Materia_Particulado	03/30/2014 05:00:16	169.00	50.0	401.0
576461558	PST_Materia_Particulado_Total	03/30/2014 05:00:19	283.00	50.0	648.0

452485984 Bucaramanga Bucaramanga Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción. 7.11331 -73.120468

1942345911 Lebrija Lebrija Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción. 7.11471566995142-73.218340873183

1155676551 Floridablanca Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción. 7.06095774032791-73.0876421928406

Material Particulado Menor a 10 micrómetros

Monóxido de Carbono

ID Dispositivo	Nombre del Sensor	Localización	Descripción	Lat	Lng
576461558	Girón	Girón	Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción.	7.07221397324112-73.1683874130249	
452485984	Bucaramanga	Bucaramanga	Simulador Estación Meteorológica del Departamento de Santander, Monitoreo de SVCA Sistema de Vigilancia de Calidad del Aire, Escala de Monitoreo SVCA: Tipo Regional: Radio de más de 20Km hasta el área total de la jurisdicción.	7.11331	-73.120468

En la Figura 39 se puede apreciar como realiza una clasificación de los sensores por cada agente contaminante encontrado en la ontología.

## ANEXO D: VALIDACIÓN DEL PROTOTIPO

### 1. Metodología de validación

- Escenario de Ambientación.
- Escenario de Motivación.
- Escenario de la Aplicación
- Clasificación de las preguntas.
- Estadísticas.
- Encuesta de Usabilidad.

### 2. Escenario de Ambientación

Los buscadores de internet tradicionales están asociados a documentos que son tratados en una sola dimensión esto significa se busca información concreta que no se está esperando que cambie en función del tiempo u otra variable. En Internet de Objetos los documentos representan algún fenómeno físico o algunos sucesos que desean medir, los ejemplos van desde eventos meteorológicos seguidos por sensores como también existencias de productos en almacenes monitoreadas por artilugios tecnológicos tales como etiquetas RFID.

Los documentos que se desea recuperar están asociados a lo se denomina Entidades en el IoT, estos documentos se representan por un flujo de información que cambia en función del tiempo, adicionalmente también puede cambiar el lugar geográfico o sitio donde está ubicada la entidad estudiada, a esto se le denomina Dinamismo de Datos.

En conclusión a este flujo de datos se debe realizar análisis dimensional de espacio y tiempo, por ejemplo el monitorea un fenómeno físico, un suceso o un evento determinado, se le realiza un modelo estadístico, la complejidad del modelo puede ser tan simple como tomar una muestra discreta, por ejemplo el clima de la ciudad de Houston en un día determinado, y 0.7 es la probabilidad de encontrar una habitación desocupada en el hotel Caesars Palace de las Vegas los días martes. En este orden de ideas se decidió clasificar las preguntas que hemos planeado se van a realizar en el buscador.

### 3. Escenario de Motivación

La necesidad de buscar información de la IoT aplicando diversas tecnologías que automatizan el análisis y búsqueda de la información para poder cumplir las nuevas necesidades de información por las personas o aplicaciones relacionadas con los dispositivos que pueden percibir y monitorear fenómenos físicos y climáticos o eventos relevantes para el proceso de desarrollo de la humano como las cadenas de suministros u operaciones y transacciones comerciantes o de bolsa. Se están descubriendo diversas necesidades de información y se están planteando soluciones que deben ser probadas y estandarizadas para su uso a nivel global. Los datos generados por la IoT están relacionados con todas las características físicas para lo que un sensor fue diseñado para medir y todos los artilugios técnicos desarrollados para medir eventos que desea monitorear como por ejemplo etiquetas RFID. Algunos ejemplos pueden ser las características de resistencia de los

puentes, las mediciones de un globo climatológico, las condiciones subterráneas de las maquina perforadoras de petróleo, los niveles de contaminación ambiental de una ciudad.

## **4. Escenario de la Aplicación**

### **4.1 Características del Buscador**

- Un buscador semántico realiza las consultas en función de los conceptos o significados, extrayendo toda esta información de los recursos, esto lo diferencia de los buscadores tradicionales que se basan en palabras claves.
- Extrayendo los conceptos de los documentos se puede analizar y clasificar los conceptos y utilizados por diversos controles para ser mostrarlos a los usuarios de una forma más conveniente para su posterior uso o reutilización.
- La visualización de los datos se puede valer de mapas, graficas, calendarios para comprender la información más fácilmente, en comparación con los buscadores tradiciones que solo presentan un pequeño texto que contiene la palabra clave de la consulta para tratar de entender cuál es el contenido del recuso.
- Los conceptos en el buscador están asociados a la variable de tiempo y a los sitios o lugares donde se encuentran los fenómenos o eventos que se quiere monitorear por esta motivo la información producida es un flujo que se debe diagramar, a diferencia de los datos continuos que tradicionalmente se busca en los buscadores de propósito general.

### **4.2 Determinación de la Relevancia**

La información de los conceptos extraídos de los recursos se clasifica de una manera idónea para colaborarle a al usuario a tomar la decisión si un recurso es relevante en acuerdo con su necesidad de información y también mostrando información muy relacionada a la información requerida, para que el usuario descubra fácilmente información que le puede ser de ayuda al proceso de búsqueda. La información se presenta utilizando las siguientes utilidades:

#### **4.2.1 Dispositivos**

Los documentos que el buscador SIMIOT muestra son los dispositivos que se encuentran en lugares y espacios físicos reales, también pueden ser dispositivos virtuales, todos estos dispositivos están generando información y subiéndola a la IoT para su posterior búsqueda y análisis.

#### **4.2.2 Acceso a flujos**

Es una pestaña por la cual se muestra las últimas actualizaciones de información para cada dispositivo, esta se hace para lograr que el usuario ahorre tiempo entrando individualmente a dispositivos con información que se hace mucho tiempo se actualizo por última vez, y el usuario se vea en la necesidad de comenzar a buscar en los histogramas la información relevante para su necesidad de consulta perdiendo tiempo en el proceso.

#### **4.2.3 Biotipo**

Se refiere al lugar o sitio físico donde se encuentran desplegados los dispositivos, como las necesidades de información del IoT también contempla los espacios físicos por medio de esta

utilidad de SIMIOT se agrupan todos los dispositivos en función de la región geográfica donde se encuentran y se muestra un mapa mostrando todos los dispositivos que están emitiendo información a partir de un radio de acción inicial de 100 Km el cual se puede ampliar.

#### 4.2.4 Medición de Ambiente

Este aplicativo agrupa a todos los dispositivos tomando las variables medioambientales y clasificándolas en función de la caracterización del Biotipo como lo son Edatopo (Magnetismo), Hidrotopo (Humedad), Climatopo (Temperatura, Viento, Presión Atmosférica, Precipitación, Calidad del Aire, Radiación Solar) los cuales adicionalmente se le establecen un rango de análisis de datos determinado por las fechas de inicial y final.

#### 4.2.5 Contaminación del Ambiente

Este aplicativo agrupa a todos los dispositivos tomando las variables medioambientales y clasificándolas en función de la medición de la contaminación ambiental como la contaminación del Aire (Contaminantes como Vanadio, Plomo, etc...), Suelo (Contaminantes, Degradación Química, Física y Biológica), Agua (Contaminantes, Residuos sólidos, Derrames de petróleo, Aguas negras y Alteraciones), Sonora (Ruido de Vehículos, Industria, Grupos Electrónicos, Explosiones y Bocinas), Térmica (Contaminantes, Emisiones de gases de invernadero, Emisión de Radiaciones y Vapor de Agua, Desarrollo Urbano), Visuales (Contaminantes, Exceso de objetos urbanos, Construcción sin planificación) los cuales adicionalmente se le establecen un rango de análisis de datos determinado por las fechas de inicial y final.

### 4.3 Clasificación de las Preguntas:

#### 4.3.1 Capacidad de la Respuesta

Las necesidades de información que puede responder el buscador son las siguientes:

1. ¿Qué dispositivos *cercanos* a <<lugar>> pueden entregarme <<información medioambiental>>?
  - a. ¿Qué dispositivos se encuentran cerca de Popayán que entreguen información medioambiental?
  - b. ¿Qué dispositivos miden información medioambiental aquí?
2. ¿Qué <<información medioambiental>> puedo *obtener* de <<lugar>>?
  - a. ¿Qué información medioambiental hay en Colombia?
  - b. ¿Qué información medioambiental hay en latitud ( °, ', ") y longitud ( °, ', ")?
3. ¿Qué <<variable medioambiental>> hay en <<lugar>> y en que [<<periodo de tiempo>>]?
  - a. ¿Qué temperatura hay en Madrid?
  - b. ¿Qué humedad hay en New York entre el 30/03/2013 hasta hoy?

4. ¿Cuál es la *tendencia* de la <<variable medioambiental>> en <<lugar>>?
  - a. ¿Cuál es la tendencia de la humedad relativa en Popayán?
  - b. ¿Cuál es el comportamiento de las temperaturas en Bucaramanga?

#### 4.3.2 Planteamiento de las consultas:

Las características de las preguntas deben ser analizadas con especial cuidado para garantizar resultados fiables de relevancia de los documentos evitando vicios que pudieran desviar las mediciones.

- Primera Pregunta (Calidad del aire en Santander): En esta pregunta se tuvo cuidado en seleccionar una región donde se pueda encontrar más de un documento, teniendo en cuenta que si la región es muy pequeña puede que no se encuentren documentos.
- Segunda Pregunta (Agentes contaminantes en el Cauca): En esta pregunta asume que la necesidad de información del usuario es más de un elemento, con el objetivo de que en la consulta recopile varios documentos asociados a cada elemento del concepto Agentes contaminantes.
- Tercera Pregunta (Monóxido de carbono en Santander): En esta consulta si se plantea un elemento específico, el objetivo es verificar la capacidad del buscador para encontrar unidades de información muy detallada que se encuentren en una región o diferentes regiones.
- Cuarta Pregunta (Contaminación en new york): En esta pregunta se presenta un concepto muy común que se puede encontrar en cualquier región del mundo para aumentar la probabilidad de que se hayan desplegados muchos dispositivos.

#### 4.4 Creación de sensores Virtuales de la Calidad del Aire

La página web de la Alcaldía de Bogotá<sup>17</sup>, cuya información fue recuperada el 16 de Noviembre del 2013 establece que corresponde al Ministerio de Ambiente, Vivienda y Desarrollo Territorial de acuerdo con los numerales 10, 11 y 14 del artículo 5º de la Ley 99 de 1993, determinar las normas ambientales mínimas y las regulaciones de carácter general aplicables a todas las actividades que puedan producir de manera directa o indirecta daños ambientales y dictar regulaciones de carácter general para controlar y reducir la contaminación atmosférica en el territorio nacional. En conformidad con los artículos 6º, 10 y 12 del Decreto 948 de 1995, corresponde a este Ministerio establecer la norma nacional de calidad del aire, o nivel de inmisión, para todo el territorio nacional en condiciones de referencia y establecer la concentración y el tiempo de exposición de los contaminantes para cada uno de los niveles de prevención, alerta y emergencia.

**Resolución del Min. De Ambiente 650 de 2010. CAPITULO II. Artículo 4º. Niveles Máximos Permisibles para Contaminantes Criterio:** Se establecen los niveles máximos permisibles en condiciones de referencia para contaminantes criterio, contemplados en la

---

<sup>17</sup> <http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=19983>

Tabla N° 1 de la presente resolución, los cuales se calcularán con el promedio geométrico para PST y aritmético para los demás contaminantes.

#### 4.4.1 Tipos de contaminantes contemplados en las normas de calidad del aire:

Una descripción de los tipos de contaminantes y sus propiedades que se tienen en cuenta para medir la calidad del aire se recogen en la página web (SIAC)<sup>18</sup> del Sistema de contaminación Ambiental de Colombia. En la Tabla 45 se muestran los contaminantes más referenciados y en la Tabla 46 se muestran los niveles máximos permitidos por la ley, y el tiempo de exposición para recoger los datos.

Tabla 45: Contaminantes mas referenciados

Contaminante	Nombre Común
<b>PST</b>	Material Particulado Total de tamaño relativo de la partícula hasta 1000 µm (micrómetros) igual a 1 mm (milímetro).
<b>PM10</b>	Material Particulado de tamaño relativo de la partícula hasta 10 µm (micrómetros).
<b>SO2</b>	Dióxido de Azufre.
<b>NO2</b>	Dióxido de Nitrógeno
<b>O3</b>	Ozono
<b>CO</b>	Monóxido de Carbono

Tabla 46: Limite maximo en tiempo de exposición

Contaminante	Unidad	Límite máximo permisible	Tiempo de exposición
<b>PST</b>	µg/m3	100	Anual
		300	24 horas
<b>PM10</b>	µg/m3	70	Anual
		150	24 horas
<b>SO2</b>	ppm (µg/m3)	0.031 (80)	Anual
		0.096 (250)	24 horas
		0.287 (750)	3 horas
<b>NO2</b>	ppm (µg/m3)	0.053 (100)	Anual
		0.08 (150)	24 horas
		0.106 ( 200)	1 hora
<b>O3</b>	ppm (µg/m3)	0.041 (80)	8 horas
		0.061 (120)	1 hora
<b>CO</b>	ppm (mg/m3)	8.8 (10)	8 horas
		35 (40)	1 hora

**Nota:** mg/m3 o µg/m3: a las condiciones de 298,15°K y 101,325 KPa. (25°C y 760 mm Hg).

<sup>18</sup> <https://www.siac.gov.co/contenido/contenido.aspx?catID=582&conID=615>

Contaminantes del aire: ppm se refiere a partes de vapor o gas por cada millón de partes de aire contaminado;  $\text{cm}^3/\text{m}^3$ . Otra forma de expresarlo es en  $\text{mg}/\text{m}^3$ , de lo que surge un factor de conversión que depende de las propiedades físicas de cada contaminante. Por ejemplo para el (Benceno) el factor de conversión es  $1 \text{ ppm} = 3,19 \text{ mg}/\text{m}^3$ . La Tabla 47 muestra unidades presentan las siguientes convenciones:

Tabla 47: Unidades utilizadas

$10^n$	Prefijo	Símbolo
$10^{-3}$	mili	m
$10^{-6}$	micro	$\mu$

**Resolución del Min. De Ambiente 650 de 2010. CAPITULO V. Artículo 10. Declaración de los niveles de prevención, alerta y emergencia por contaminación del aire.** La concentración y el tiempo de exposición bajo los cuales se debe declarar por parte en la Tabla 48 de las autoridades ambientales competentes los estados excepcionales de Prevención, Alerta y Emergencia:

Tabla 48: Niveles por contaminante

Contaminante	Tiempo de Exposición	Unidades	Prevención	Alerta	Emergencia
PST	24 horas	$\mu\text{g}/\text{m}^3$	375 $\mu\text{g}/\text{m}^3$	625 $\mu\text{g}/\text{m}^3$	875 $\mu\text{g}/\text{m}^3$
PM10	24 horas	$\mu\text{g}/\text{m}^3$	300 $\mu\text{g}/\text{m}^3$	400 $\mu\text{g}/\text{m}^3$	500 $\mu\text{g}/\text{m}^3$
SO2	24 horas	ppm ( $\mu\text{g}/\text{m}^3$ )	0.191 (500)	0.382 (1.000)	0.612 (1.600)
NO2	1 hora	ppm ( $\mu\text{g}/\text{m}^3$ )	0.212 (400)	0.425 (800)	1.064 (2.000)
O3	1 hora	ppm ( $\mu\text{g}/\text{m}^3$ )	0.178 (350)	0.356 (700)	0.509 (1.000)
CO	8 horas	ppm ( $\text{mg}/\text{m}^3$ )	14.9 (17)	29.7 (34)	40,2 (46)

**Nota:**  $\text{mg}/\text{m}^3$  o  $\mu\text{g}/\text{m}^3$ : a las condiciones de 298,15°K y 101,325 KPa. (25°C y 760 mm Hg).

#### 4.5 Herramientas Utilizadas

Los simuladores de desarrollaron en Visula Studio 2012 al que se le instalo la librería para desarrollar aplicaciones para dispositivos embebidos Microframework versión 4.0.

#### 4.6 Regiones de Colombia para las cuales se implementó sensores virtuales para la medición de la calidad del aire

- Sistema de calidad del aire del Departamento del Cauca se muestra en la Tabla 49:

Tabla 49. Regiones Dpto Cauca con Sensores virtuales

Distritos y Municipios	Ubicación
------------------------	-----------

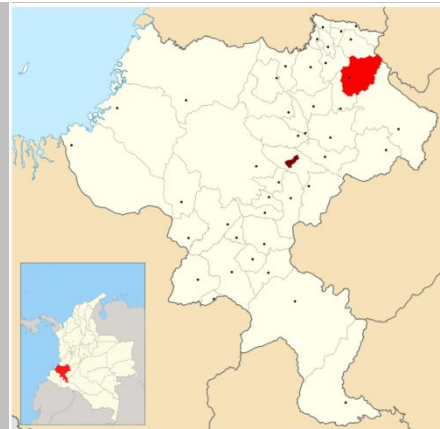
**Popayán:** es una ciudad colombiana, capital del departamento del Cauca, entre la Cordillera Occidental y Central al occidente del país, en las coordenadas 2°26'39"N 76°37'17"O. Tiene 265.702 habitantes, de acuerdo al censo del DANE (Departamento Administrativo Nacional de Estadísticas) elaborado en el año 2005. Su extensión territorial es de 512 km<sup>2</sup>, su altitud media es de 1760 m sobre el nivel del mar.



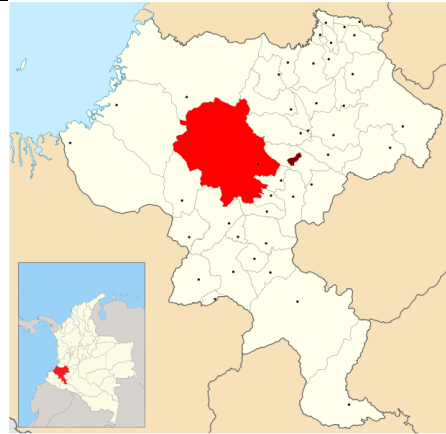
**Timbío:** es un municipio de Colombia ubicado en el departamento de Cauca situado a 13 kilómetros de Popayán, la capital del departamento. Perteneciente al Área Metropolitana de Popayán. Extensión total: 205 kilómetros cuadrados Km<sup>2</sup>. Altitud de la cabecera municipal (metros sobre el nivel del mar): entre 1.000 y 2.000 metros.



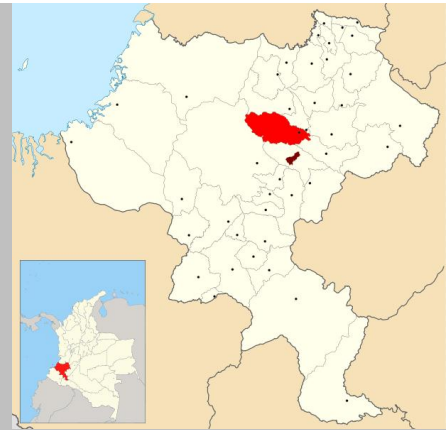
**Toribío:** es un municipio colombiano ubicado en el departamento de Cauca. La cabecera municipal se encuentra localizada a los 02° 57' 29" de latitud norte y 76° 16' 17" de longitud oeste, a una altura de 1.800 m.s.n.m.



**El Tambo:** es un municipio colombiano ubicado en el departamento de Cauca. Perteneciente al Área Metropolitana de Popayán. Ubicado aproximadamente 33 km de su capital, Popayán. Cuenta con una población aproximada de 50.000 habitantes. La altura sobre el nivel del mar que es de 1.745 m. Tiene una extensión de 3.280 kilómetros cuadrados.



**Cajibío:** es un municipio colombiano en el departamento del Cauca a una distancia de 29 km al norte de Popayán, con una población aproximada de 32.965 habitantes. Perteneciente al Área Metropolitana de Popayán.



- Sistema de calidad del aire del Departamento de Santander se muestra en la
- Tabla 50:

**Tabla 50: Regiones Dpto Santander con Sensores virtuales**

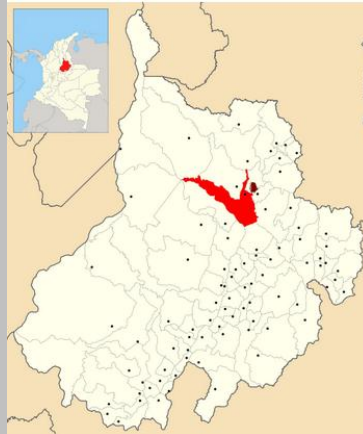
Distritos y Municipios	Ubicación
<p><b>Bucaramanga:</b> es una Ciudad de Colombia, capital del departamento de Santander. Con unos 526.940 habitantes (Proyección DANE, 2013) ocupa el octavo puesto (8) por población a nivel nacional. Tiene una longitud de 10 km de Sur a Norte y 6 km de Oriente a Occidente.</p>	

---

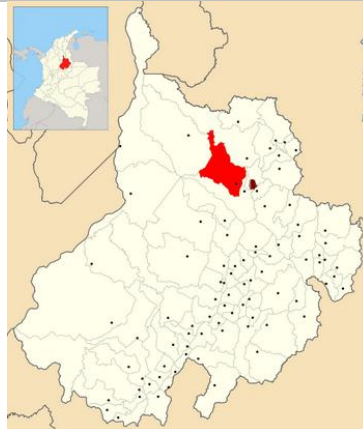
**Floridablanca:** es un municipio y la segunda ciudad más importante del Departamento de Santander, noreste de Colombia, con 263.095 habitantes (2012). Tiene una extensión aproximada de 97 kilómetros cuadrados.



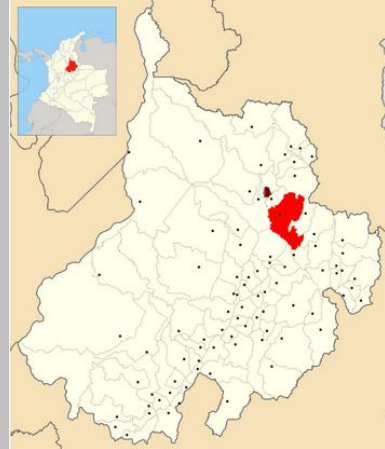
**Girón:** es un municipio del departamento de Santander, que forma parte del área Metropolitana de Bucaramanga al noreste de Colombia. Ubicación 7°4'23"N 73°10'5"O. Altitud 777 msnm. Población 170.770 hab.



**Lebrija:** es un municipio del departamento de Santander (Colombia). Se ubica en el a 15 kilómetros de Bucaramanga. Ubicación 7°6'47"N 73°13'8"O. Altitud 1.055 msnm.



**Piedecuesta:** es un municipio del departamento de Santander. Ubicación 6°59'19"N 73°3'1"O. Población 129.000 hab.



#### 4.7 Simulación de los canales de Xively por contaminante y parámetros establecidos por la ley

La Tabla 51 muestra la simulación de los canales se puede apreciar como las curvas de fjujo cumplen los parámetros de tiempo y magnitudes para los tres niveles en los que se clasifican los posibles valores, los valores no son randomicos sino que para cada nivel se asigno una probabilidad, siendo la probabilidad más pequeña para el nivel de emergencia, asi los posibles valores se parecen más a los emitidos por estaciones de monitoreo reales.

Tabla 51: Simulación de Canales Xively

Parametros	Canales
------------	---------

<p><b>Contaminante:</b> PST  <b>Tiempo de Exposición:</b> 24 horas  <b>Prevención:</b> 375 µg/m3  <b>Alerta:</b> 625 µg/m3  <b>Emergencia:</b> 875 µg/m3</p>	<div style="text-align: right;"> <b>PST_Material_Particulado_Total</b>  <b>193.00</b>  <small>ug/m3</small> </div> <p style="text-align: right;"><small>Last updated 12 hours ago</small></p> <p>Activos Semiautomáticos</p> <p>Beta Gauge Cadmio Calcio</p> <p>Carbono Elemental</p> <p>Carbono Orgánico</p> <p>diámetro aerodinámico Hi Vol</p> <p>Hierro Low Vol</p> <p>Material Particulado Total</p> <p>Muestra cada 24 horas Nitratos</p> <p>Plomo Silicio Sulfatos Zinc</p> <p style="text-align: center;"><small>7 days averaged datapoints</small></p> <p><a href="#">Edit</a> <a href="#">Delete</a></p>
<p><b>Contaminante:</b> PM10  <b>Tiempo de Exposición:</b> 24 horas  <b>Prevención:</b> 300 µg/m3  <b>Alerta:</b> 400 µg/m3  <b>Emergencia:</b> 500 µg/m3</p>	<div style="text-align: right;"> <b>PM10_Material_Particulado</b>  <b>192.00</b>  <small>ug/m3</small> </div> <p style="text-align: right;"><small>Last updated 12 hours ago</small></p> <p>10 micrómetros Cadmio Calcio</p> <p>Carbono Elemental</p> <p>Carbono Orgánico</p> <p>diámetro aerodinámico</p> <p>gotas líquidas Hierro</p> <p>Material Particulado</p> <p>Muestra cada 24 horas Nitratos</p> <p>Plomo Silicio Sulfatos</p> <p>velocidad de sedimenta terminal</p> <p>Zinc</p> <p style="text-align: center;"><small>7 days averaged datapoints</small></p> <p><a href="#">Edit</a> <a href="#">Delete</a></p>

<p><b>Contaminante:</b> SO2  <b>Tiempo de Exposición:</b> 24 horas  <b>Prevención:</b> 500 µg/m3  <b>Alerta:</b> 1.000 µg/m3  <b>Emergencia:</b> 1.600 µg/m3</p>	<div data-bbox="769 233 1382 737"> <p>Dioxido_de_Azufre_SO2 <span style="float: right;">354.00 ug/m3</span></p> <p>bronconstricción <span style="float: right;">Last updated 8 hours ago</span></p> <p>Dióxido de Azufre lluvia ácida</p> <p>Muestra cada 24 horas SO2</p> <p>SOx sulfatos</p> <p>7 days averaged datapoints</p> <p>Edit Delete</p> </div>
<p><b>Contaminante:</b> O3  <b>Tiempo de Exposición:</b> 1 hora  <b>Prevención:</b> 350 µg/m3  <b>Alerta:</b> 700 µg/m3  <b>Emergencia:</b> 1.000 µg/m3</p>	<div data-bbox="769 766 1382 1270"> <p>Ozono_O3 <span style="float: right;">260.00 ug/m3</span></p> <p>contaminante secundario <span style="float: right;">Last updated an hour ago</span></p> <p>irritante pulmonar</p> <p>Muestra cada 1 horas O3</p> <p>Ozono Ozono contaminante</p> <p>smog troposfera</p> <p>6 hours averaged datapoints</p> <p>Edit Delete</p> </div>
<p><b>Contaminante:</b> CO  <b>Tiempo de Exposición:</b> 8 horas  <b>Prevención:</b> 17 µg/m3  <b>Alerta:</b> 34 µg/m3  <b>Emergencia:</b> 46 µg/m3</p>	<div data-bbox="769 1299 1382 1782"> <p>Monoxido_de_Carbono_CO <span style="float: right;">11.00 mg/m3</span></p> <p>carboxihemoglobina cefalea <span style="float: right;">Last updated 8 hours ago</span></p> <p>CO COHb hemoglobina</p> <p>hipoxia hipoxia anémica</p> <p>intoxicación</p> <p>Monóxido de Carbono</p> <p>Muestra cada 8 horas náuseas</p> <p>1 day averaged datapoints</p> <p>Edit Delete</p> </div>

## 4.8 Recopilación de Datos

### 4.8.1 Condiciones de la recopilación de datos

- Se realizaron 4 consultas
- Participaron 13 usuarios
- Se identificaron 25 muestras por cada consulta

En la Tabla 52 se muestran los datos recolectados.

Tabla 52: Datos prueba de relevancia

CST	Dispositivo	Juez 1	Juez 2	Juez 3	Juez 4	Juez 5	Juez 6	Juez 7	Juez 8	Juez 9	Juez 10	Juez 11	Juez 12	Juez 13
1	576461558	1	1	0	0	1	1	1	0	1	1	1	1	1
1	452485984	1	1	0	0	1	1	1	0	1	0	1	1	1
1	1942345911	1	1	0	0	1	0	1	0	1	1	1	1	1
1	1155676551	1	1	0	0	1	0	1	1	1	0	1	1	1
1	1422637955	0	1	0	1	0	0	0	0	0	1	1	0	0
1	1509142040	0	1	0	0	0	0	0	0	0	0	1	0	0
1	1898258902	0	0	0	0	0	0	0	0	0	0	1	0	0
1	541602029	0	0	0	1	0	0	0	0	0	0	1	0	0
1	2003665973	0	0	0	0	0	0	0	0	0	0	0	0	0
1	140284357	1	0	0	0	1	0	0	0	1	0	0	0	1
1	80276	0	0	0	0	0	0	0	0	0	0	0	0	0
1	80204	0	0	0	0	0	0	0	0	0	0	0	0	0
1	80277	0	0	0	0	0	0	0	0	0	0	0	0	0
1	880147192	0	0	0	0	0	0	0	0	0	0	0	0	0
1	54117	0	0	0	0	0	0	0	0	0	0	0	0	0
1	53887	0	0	0	0	0	0	0	0	0	0	0	0	0
1	399177680	0	0	0		0	0	0	0	0	0	0	0	0
1	105488	0	0	0	0	0	0	0	0	0	0	0	0	0
1	118311	0	0	0	0	0	0	0	0	0	0	0	0	0
1	130758	0	0	0		0	0	0	0	0	0	0	0	0
1	50147	0	0	0	0	0	0	0	0	0	0	1	0	0
1	72225	0	0	0		0	0	0	0	0	0	1	0	0
1	63645	0	0	0		0	0	0	0	0	0	0	0	0

1	118388	0	0	0	0	0	0	0	0	0	0	0	0
1	7262	0	0	0	0	0	0	0	0	0	0	0	0
2	1422637955	1	0	1	0	1	1	1	1	1	1	1	1
2	1509142040	1	0	1	0	1	0	1	1	1	1	1	1
2	1898258902	1	0	1	0	1	1	1	1	1	0	1	1
2	541602029	1	0	1	0	1	0	1	1	1	0	0	1
2	2003665973	1	0	1	1	1	0	1	1	1	0	0	1
2	576461558	0	0	0	1	0	0	0	0	0	0	0	0
2	452485984	0	0	0	0	0	0	0	0	0	0	1	0
2	1942345911	0	0	0	0	0	0	0	0	0	0	1	0
2	1155676551	0	0	0	0	0	0	0	0	0	0	0	0
2	140284357	0	0	0	0	0	0	0	0	0	0	0	0
2	880147192	0	0	0	0	0	0	0	0	0	0	1	0
2	54117	0	0	0	0	0	0	0	0	0	0	1	0
2	53887	0	0	0	0	0	0	0	0	0	0	0	0
2	80276	0	0	0	0	0	0	0	0	0	0	0	0
2	105488	0	0	0	0	0	0	0	0	0	0	0	0
2	80204	0	0	0	0	0	0	0	0	0	0	0	0
2	80277	0	0	0	0	0	0	0	0	0	0	0	0
2	118311	0	0	0	0	0	0	0	0	0	0	0	0
2	118388	0	0	0	0	0	0	0	0	0	0	1	0
2	7262	0	0	0	0	0	0	0	0	0	0	0	0
2	1196	0	0	0	0	0	0	0	0	0	0	1	0
2	46386	0	0	0	0	0	0	0	0	0	0	0	0
2	77591	0	0	0	0	0	0	0	0	0	0	0	0
2	41032	0	0	0	0	0	0	0	0	0	0	0	0
2	50147	0	0	0	0	0	0	0	0	0	0	0	0
3	576461558	1	0	0	1	1	1	1	1	1	1	1	1
3	452485984	1	0	0	0	1	1	1	1	1	1	1	1
3	1942345911	1	0	0	1	1	0	1	1	1	1	1	1
3	1155676551	1	0	0	0	1	1	1	1	1	1	1	1
3	140284357	1	0	0	1	1	0	1	1	1	0	1	1
3	1422637955	0	0	0	0	0	0	0	0	0	0	0	1
3	1509142040	0	0	0	0	0	0	0	0	0	0	0	0
3	1898258902	0	0	0	0	0	0	0	0	0	0	0	0
3	541602029	0	0	0	0	0	0	0	0	0	1	0	0
3	2003665973	0	0	0	0	0	0	0	0	0	0	0	0

3	399177680	0	0	0	0	0	0	0	0	0	0	0
3	49761	0	1	0	0	0	0	0	0	0	0	0
3	75856	0	0	0	0	0	0	0	0	0	0	0
3	54177	0	0	0	0	0	0	0	0	0	0	0
3	54179	0	0	0	0	0	0	0	0	0	0	0
3	22587	0	0	0	0	0	0	0	0	0	0	0
3	17468	0	0	0	0	0	0	0	0	0	0	0
3	17475	0	0	0	0	0	0	0	0	0	0	0
3	17473	0	0	0	0	0	0	0	0	0	0	0
3	37353	0	0	0	0	0	0	0	0	0	0	0
3	119509	0	0	0	0	0	0	0	0	0	0	0
3	72068	0	0	0	0	0	0	0	0	0	0	0
3	21544	0	0	0	0	0	0	0	0	0	0	0
3	46307	0	0	0	0	0	0	0	0	0	0	0
3	26118	0	0	0	0	0	0	0	0	0	0	0
4	17468	0	1	1	1	1	1	1	1	0	1	0
4	88950	0	0	0	0	0	0	0	0	0	0	0
4	121471	1	0	0	1	1	1	1	1	0	0	0
4	3929	0	1	1	0	0	0	1	0	1	1	0
4	49761	0	0	0	0	0	0	0	1	0	0	0
4	50719	0	0	0	0	0	0	0	1	0	0	0
4	50721	0	0	0	0	0	0	0	1	0	0	0
4	50720	0	0	0	0	0	0	0	0	0	0	0
4	1407542821	0	0	0	0	0	0	0	0	0	0	0
4	57363	1	1	0	1	0	0	1	0	1	1	0
4	172881863	0	0	0	0	0	0	0	0	0	0	0
4	117981	0	0	0	1	0	0	0	0	0	0	0
4	34917	0	0	0	0	0	0	0	0	0	0	0
4	102976	0	0	0	0	0	0	0	0	0	0	0
4	140284357	0	0	0	0	0	0	0	0	0	0	0
4	576461558	0	0	0	0	0	0	0	0	0	0	0
4	452485984	0	0	0	0	0	0	0	0	0	0	0
4	1422637955	0	0	0	0	0	0	0	0	0	0	0
4	1942345911	0	0	0	0	0	0	0	0	0	0	0
4	1155676551	0	0	0	0	0	0	0	0	0	0	0
4	1509142040	0	0	0	0	0	0	0	0	0	0	0
4	1898258902	0	0	0	0	0	0	0	0	0	0	0

4	541602029	0	0	0	0	0	0	0	0
4	2003665973	0	0	0	0	0	0	0	0
4	121272	0	0	0	0	0	0	0	0

## 4.8.2 Interpretación de la recopilación de datos

En la recopilación de datos se identificaron los siguientes resultados.

- Se identifican 3 clases de resultados (1=Relevante), (0=NO relevante), (Vacío=NO reporto).
- Existen relevancias en los primeras posiciones de aparición de dispositivos lo que indica buena Precisión vs Recuerdo porque esta métrica le da gran ponderación a los documentos relevantes de las primeras posiciones, aunque los usuarios no hallan identificado pocas relevancias hacia las últimas posiciones de la tabla.

## 4.9 Métricas Precisión at K

El objetivo de estas métricas es calcular la relevancia mirando las frecuencias en que los usuarios marcaron como relevante un documento determinado por medio de la exactitud, verificando el efecto acumulativo por medio de la precisión y hallando el punto medio de este efecto.

### 4.9.1 Condiciones de la prueba

- Se realizaron 4 consultas
- Participaron 13 usuarios
- Se identificaron 25 muestras por cada consulta

Para la Tabla 53 de Precisión at K se presenta una guía de como se realizaron los cálculos con Excel y una descripción de las formulas.

**Tabla 53: Excel Precisión at K**

	A	B	C	D	E	F	G
1	CST	DPT	R	TOTAL	EXACTITUD	PRECISION	P. MEDIA
2							
3							
4							
5							

En la Tabla 54 se muestra como se realizaron los cálculos de la Precisión at K.

**Tabla 54: Formulas Excel Precisión at K**

Calculo	Celda	FORMULA EXCEL
R	C2	=CONTAR.SI(Muestras!C2:Muestras!O2;"1")

TOTAL	D2	Número de Jueces
EXACTITUD	E2	=C2/D2
PRECISION	F2	=SUMA(C\$2:C2)/SUMA(D\$2:D2)
P. MEDIA	G2	=PROMEDIO(F2:F26)

La Tabla 55 se encuentra en la hoja de nombre "Muestras" la cual se accede desde la formula de la celda (C2) y la columna (R) que es el conteo de los usuarios que juzgaron como relevante un dispositivo.

**Tabla 55: Recopilación de datos en la hoja Muestra en Excel**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	CST	DPT	Juez 1	Juez 2	Juez 3	Juez 4	Juez 5	Juez 6	Juez 7	Juez 8	Juez 9	Juez 10	Juez 11	Juez 12	Juez 13
2															

Para la formula de la (Presición) se dezplaza las celdas que abarcan los 25 dispositivos, para la (P. Media) se dezplaza la formula del PROMEDIO de Excel hasta donde abarcan los 25 dispositivos de cada consulta.

En la Tabla 56se presenta el calculo de la Presición at K.

**Tabla 56: Calculo de Presición at K**

CST	DPT	R	TOTAL	EXACTITUD	PRECISION	P. MEDIA
1	1	10	13	76,9%	76,9%	37,0%
	2	9	13	69,2%	73,1%	
	3	9	13	69,2%	71,8%	
	4	9	13	69,2%	71,2%	
	5	4	13	30,8%	63,1%	
	6	2	13	15,4%	55,1%	
	7	1	13	7,7%	48,4%	
	8	2	13	15,4%	44,2%	
	9	0	13	0,0%	39,3%	
	10	4	13	30,8%	38,5%	
	11	0	13	0,0%	35,0%	
	12	0	13	0,0%	32,1%	
	13	0	13	0,0%	29,6%	
	14	0	13	0,0%	27,5%	
	15	0	13	0,0%	25,6%	
	16	0	13	0,0%	24,0%	
	17	0	13	0,0%	22,6%	
	18	0	13	0,0%	21,4%	
	19	0	13	0,0%	20,2%	
	20	0	13	0,0%	19,2%	
	21	1	13	7,7%	18,7%	

	22	1	13	7,7%	18,2%	
	23	0	13	0,0%	17,4%	
	24	0	13	0,0%	16,7%	
	25	0	13	0,0%	16,0%	
2	1	11	13	84,6%	84,6%	39,8%
	2	10	13	76,9%	80,8%	
	3	10	13	76,9%	79,5%	
	4	8	13	61,5%	75,0%	
	5	8	13	61,5%	72,3%	
	6	1	13	7,7%	61,5%	
	7	1	13	7,7%	53,8%	
	8	1	13	7,7%	48,1%	
	9	0	13	0,0%	42,7%	
	10	0	13	0,0%	38,5%	
	11	1	13	7,7%	35,7%	
	12	1	13	7,7%	33,3%	
	13	0	13	0,0%	30,8%	
	14	0	13	0,0%	28,6%	
	15	0	13	0,0%	26,7%	
	16	0	13	0,0%	25,0%	
	17	0	13	0,0%	23,5%	
	18	0	13	0,0%	22,2%	
	19	1	13	7,7%	21,5%	
	20	0	13	0,0%	20,4%	
	21	1	13	7,7%	19,8%	
	22	0	13	0,0%	18,9%	
	23	0	13	0,0%	18,1%	
	24	0	13	0,0%	17,3%	
	25	0	13	0,0%	16,6%	
3	1	10	13	76,9%	76,9%	36,8%
	2	9	13	69,2%	73,1%	
	3	9	13	69,2%	71,8%	
	4	9	13	69,2%	71,2%	
	5	8	13	61,5%	69,2%	
	6	1	13	7,7%	59,0%	
	7	0	13	0,0%	50,5%	
	8	0	13	0,0%	44,2%	
	9	1	13	7,7%	40,2%	
	10	0	13	0,0%	36,2%	
	11	0	13	0,0%	32,9%	
	12	1	13	7,7%	30,8%	
	13	0	13	0,0%	28,4%	
	14	0	13	0,0%	26,4%	
	15	0	13	0,0%	24,6%	
	16	0	13	0,0%	23,1%	
	17	0	13	0,0%	21,7%	
	18	0	13	0,0%	20,5%	
	19	0	13	0,0%	19,4%	
	20	0	13	0,0%	18,5%	
	21	0	13	0,0%	17,6%	
	22	0	13	0,0%	16,8%	
	23	0	13	0,0%	16,1%	
	24	0	13	0,0%	15,4%	
	25	0	13	0,0%	14,8%	
4	1	7	13	53,8%	53,8%	18,3%
	2	0	13	0,0%	26,9%	
	3	5	13	38,5%	30,8%	
	4	5	13	38,5%	32,7%	
	5	1	13	7,7%	27,7%	
	6	1	13	7,7%	24,4%	
	7	1	13	7,7%	22,0%	
	8	0	13	0,0%	19,2%	

9	0	13	0,0%	17,1%
10	6	13	46,2%	20,0%
11	0	13	0,0%	18,2%
12	1	13	7,7%	17,3%
13	0	13	0,0%	16,0%
14	0	13	0,0%	14,8%
15	0	13	0,0%	13,8%
16	0	13	0,0%	13,0%
17	0	13	0,0%	12,2%
18	0	13	0,0%	11,5%
19	0	13	0,0%	10,9%
20	0	13	0,0%	10,4%
21	0	13	0,0%	9,9%
22	0	13	0,0%	9,4%
23	0	13	0,0%	9,0%
24	0	13	0,0%	8,7%
25	0	13	0,0%	8,3%

#### 4.9.2 Interpretación de Resultados Precisión at K

- En la consulta 2 se obtuvo el mejor resultado con un 39.8% debido a los mejores resultados en el comienzo de la tabla es decir en la primeras posiciones.
- Aunque en la consulta 1 está muy cerca con un 37% los usuarios no acumularon relevancias en las primeras posiciones.
- En la consultas 3 y 4 los valores están por debajo debido al alto número de documentos que no recibieron calificación. debido a que los mismos documentos no aparecen para todos los usuarios, esto es algo que determina el proceso automático de búsqueda.

#### 4.10 Métricas Kappa Cohen

Kappa es una medida de concordancia pero la Kappa Cohen determina la concordancia entre dos Jueces, hallando la frecuencia y las combinaciones en que cada documento fue calificado.

##### 4.10.1 Condiciones de la recopilación de datos

- Se seleccionaron sin orden 2 usuarios
- Se realizaron 4 consultas
- Se identificaron 25 muestras por cada consulta
- Se realiza por dos cálculos distintos para verificar los resultados

En la Tabla 57 se muestra el modelo en Excel para calcular la métrica Kappa Cohen.

Tabla 57: Modelo Excel para Kappa Cohen

	A	B	C	D	E
1	CONSULTA 1		JUEZ 2		
2			SI	NO	TOTAL
3	JUEZ 1	SI	SI/SI	SI/NO	T2
4		NO	NO/SI	NO/NO	T4
5		TOTAL	T1	T3	N

6					NA
7					NE
8					K. COHEN
9					PA
10					PE
11					K. COHEN

En la Tabla 58 y la Tabla 59 se presentan dos formas de calcular la métrica Kappa Cohen, esto se hizo con el animo de corroborar los resultados.

**Tabla 58: Primera forma de calcular K. Cohen en Excel**

Calculo 1	Celda	FORMULA EXCEL
T1	C5	=SUMA(C3:C4)
T2	E3	=SUMA(C3:D3)
T3	D5	=SUMA(D3:D4)
T4	E4	=SUMA(C4:D4)
N	E5	=SUMA(C5:D5)
NA	E6	=SUMA(C3;D4)
NE	E7	=((C5*E3)/E5)+(D5*E4/E5)
K. COHEN	E8	=(E6-E7)/(E5-E7)

**Tabla 59: Segunda forma de calcular K. Cohen en Excel**

Calculo 2	Celda	FORMULA EXCEL
PA	E9	=(E6/E5)
PE	E10	=(C5*E3+D5*E4)/(E5*E5)
K. COHEN	E11	=(E9-E10)/(1-E10)

De la Tabla 60 a la Tabla 63 se calcula la métrica Kappa Cohen para cada una de las consultas.

**Tabla 60: Kappa Cohen consulta 1**

CONSULTA 1		JUEZ 2		TOTAL
	SI	NO		
JUEZ 1	SI	4	2	6
	NO	1	18	19
	TOTAL	5	20	25
				22
				16,4
				0,651162791
				0,88
				0,656

				0,651162791
--	--	--	--	-------------

**Tabla 61: Kappa Cohen consulta 2**

CONSULTA 2		JUEZ 2		TOTAL
JUEZ 1		SI	NO	
	SI	3	2	5
	NO	0	20	20
	TOTAL	3	22	25
				23
				18,2
				0,705882353
				0,92
				0,728
				0,705882353

**Tabla 62: Kappa Cohen consulta 3**

CONSULTA 3		JUEZ 2		TOTAL
JUEZ 1		SI	NO	
	SI	3	0	3
	NO	2	20	22
	TOTAL	5	20	25
				23
				18,2
				0,705882353
				0,92
				0,728
				0,705882353

**Tabla 63: Kappa Cohen consulta 4**

CONSULTA 4		JUEZ 2		TOTAL
JUEZ 1		SI	NO	
	SI	1	1	2
	NO	2	21	23
	TOTAL	3	22	25
				22
				20,48
				0,336283186
				0,88
				0,8192
				0,336283186

#### 4.10.2 Interpretación de Resultados de Kappa Cohen

- En las tres primeras consultas se presentó una concordancia (Sustancial) esto indica que las preguntas son consistentes por que le permite a los usuarios identificar los dispositivo que se encuentran en el dominio de búsqueda.
- En la última consulta se evidencia el grado de concordancia (Discreta) ya que la consulta fue muy específica y esto le dificulto a los usuarios tomar una decisión clara sobre la relevancia.

#### 4.11 Métricas Kappa Fleiss

Esta métrica de concordancia tiene un enfoque más específico y el cual tiene en cuenta todas las (Clases) de respuestas que se presentan en la prueba prueba y se cuenta es el numero de usuarios que determinaron que un documento era relevante.

##### 4.11.1 Condiciones de la prueba

- Se identifican 3 clases de resultados (1=Relevante), (0=NO relevante), (Vacío=NO reporto).
- Muestra de 25 dispositivos (n).
- Número de jueces 13 (m).
- Numero de consulta 4.

En la Tabla 64 se muestra el modelo en Excel para calcular la métrica KappaFleiss, la diferencia entre los dos son las clases que presenta cada uno, el modelo de la izquierda contempla dos clases y el de la derecha 3 clases:

**Tabla 64: Modelo Excel para Fleiss**

	A	B	C	D
1	CST#	Clases 2		
2	#Doc	SI	NO	
3	1			M
4	2			N
5	3			PA
6	4			PE
7	5			K. FLEEISS
8	6			
9	7			
10	8			
11	9			
12	10			
13	11			
14	12			
15	13			
16	14			
17	15			
18	16			
19	17			
20	18			
21	19			
22	20			
23	21			

	A	B	C	D	E
1	CST#	Clases 3			
2	#Doc	SI	NO	Vacío	
3	1				M
4	2				N
5	3				PA
6	4				PE
7	5				K. FLEEISS
8	6				
9	7				
10	8				
11	9				
12	10				
13	11				
14	12				
15	13				
16	14				
17	15				
18	16				
19	17				
20	18				
21	19				
22	20				
23	21				

24	22			
25	23			
26	24			
27	25			
28		Q1	Q2	

24	22				
25	23				
26	24				
27	25				
28		Q1	Q2	Q3	

En Tabla 65 y Tabla 66 se muestran las formulas Excel que se utilizaron para colcular la métrica Kappa Fleiss para 2 clases y 3 clases. El calculo de las celdas (SI, NO, Vacío) es común a las dos tablas anteriores. Para calcular estas celdas se requiere de la hoja "Muestra" que expone en la Tabla 55.

**Tabla 65: Formulas Excel para calcular K. Fleiss para 2 clases**

Calculo para 2 clases	Celda	FORMULA EXCEL
SI	B3	=CONTAR.SI(Muestras!C2:Muestras!O2;"1")
NO	C3	=CONTAR.SI(Muestras!C2:Muestras!O2;"0")
Vacío	D3	=CONTAR.SI(Muestras!C2:Muestras!O2;"")
M	D3	Número de Jueces
N	D4	Número de Documentos
PA	D5	=(SUMA.CUADRADOS(B3:C27) - (D3*D4))/(D3*D4*(D3-1))
PE	D6	=SUMA.CUADRADOS(B28:C28)
K. FLEEISS	D7	=(D5-D6)/(1-D6)
Q1	B28	=SUMA(B3:B27)/(D3*D4)
Q2	C28	=SUMA(C3:C27)/(D3*D4)

**Tabla 66: Formulas Excel para calcular K. Fleiss para 3 clases**

Calculo para 3 clases	Celda	FORMULA EXCEL
M	D3	Número de Jueces
N	D4	Número de Documentos
PA	D5	=(SUMA.CUADRADOS(B3:D27) - (E3*E4))/(E3*E4*(E3-1))
PE	D6	=SUMA.CUADRADOS(B28:D28)
K. FLEEISS	E7	=(E5-E6)/(1-E6)
Q1	B28	=SUMA(B3:B27)/(E3*E4)
Q2	C28	=SUMA(C3:C27)/(E3*E4)
Q3	E28	=SUMA(D3:D27)/(E3*E4)

En la Tabla 67 a la Tabla 68 se calcula la métrica Kappa Fleiss para cada una de las consultas.

**Tabla 67: Kappa Fleiss consulta 1 y 2**

CST1	Clases 3				CST2	Clases 2		
	SI	NO	Vacío			SI	NO	
1	10	3	0	13	1	11	2	13
2	9	4	0	25	2	10	3	25
3	9	4	0	0,8271	3	10	3	0,8738

4	9	4	0	0,7108	4	8	5	0,7229
5	4	9	0	0,40236	5	8	5	0,5447
6	2	11	0		6	1	12	
7	1	12	0		7	1	12	
8	2	11	0		8	1	12	
9	0	13	0		9	0	13	
10	4	9	0		10	0	13	
11	0	13	0		11	1	12	
12	0	13	0		12	1	12	
13	0	13	0		13	0	13	
14	0	13	0		14	0	13	
15	0	13	0		15	0	13	
16	0	13	0		16	0	13	
17	0	12	1		17	0	13	
18	0	13	0		18	0	13	
19	0	13	0		19	1	12	
20	0	12	1		20	0	13	
21	1	12	0		21	1	12	
22	1	11	1		22	0	13	
23	0	12	1		23	0	13	
24	0	13	0		24	0	13	
25	0	13	0		25	0	13	
	0,16	0,8276	0,0123			0,1661	0,8338	

Tabla 68: Kappa Fleiss consulta 3 y 4

CST3	Clases 3				CST4	Clases 3			
	#Doc	SI	NO			Vacío	#Doc	SI	
1	10	2	1	13	1	7	2	4	13
2	9	3	1	25	2	0	8	5	25
3	9	3	1	0,7610	3	5	4	4	0,5097
4	9	3	1	0,6289	4	5	3	5	0,4824
5	8	4	1	0,3559	5	1	9	3	0,0526
6	1	11	1		6	1	9	3	
7	0	12	1		7	1	9	3	
8	0	12	1		8	0	10	3	
9	1	11	1		9	0	9	4	
10	0	12	1		10	6	3	4	
11	0	12	1		11	0	9	4	
12	1	11	1		12	1	8	4	
13	0	12	1		13	0	9	4	
14	0	12	1		14	0	8	5	
15	0	12	1		15	0	10	3	
16	0	12	1		16	0	10	3	
17	0	12	1		17	0	10	3	
18	0	12	1		18	0	10	3	
19	0	12	1		19	0	10	3	
20	0	12	1		20	0	10	3	
21	0	12	1		21	0	10	3	

22	0	12	1		22	0	10	3	
23	0	12	1		23	0	8	5	
24	0	12	1		24	0	8	5	
25	0	12	1		25	0	7	6	
	0,1476	0,7753	0,0769			0,1476	0,7753	0,0769	

#### 4.11.2 Interpretación de Resultados de Resultados de Kappa Fleiss

- En las dos primeras consultas se presentó una concordancia (Discreto) esto indica que los dispositivos son fáciles de determinar su relevancias y que con un número relativamente bueno de jueces el resultado es consistente.
- En las dos últimas consultas se evidencia el grado de concordancia con tendencia (Insignificante) ya que esta métrica da especial interés a las clases y en estas últimas consultas la clase adicional de (Vacío) evidencio un la indecisión de respuesta por parte de los usuarios a determinar si el dispositivo era relevante, al aumentar los posibles resultados disminuye la concordancia entre los usuarios.

## ANEXO E: CREACIÓN DE SERVICIOS WEB SEMÁNTICOS

### 1. Instalación de las herramientas:

Para una adecuada comprensión de la importancia del uso de herramientas en la creación de SWS se presentara una breve guía de instalación y una descripción de los aspectos en que cada herramienta da apoyo a la creación de SWS.

#### 1.1 Herramienta WSMT se basa en el lenguaje WSML:

El uso de la herramienta WSMT facilita la creación de documentos en lenguaje WSML apoyando la comprobación de la estructura de los documentos según el modelo WSMO.

- Descargar el entorno WSMT el cual está integrado en el entorno eclipse.
- Descargar el JDK de java no hay que configurar variables de entorno.
- Descomprimir el archivo en la carpeta, luego de esto se procede a ejecutar el entorno desde el acceso directo.

#### 1.2 Herramienta WSMX servidor de SWS:

La herramienta WSMT se comunica con WSMX para publicar y luego realizar selección e invocación de SWS. Nota: La versión WSMXv1.0 no se comunica correctamente con el WSMT.

- Descargar el entorno WSMX el cual se iniciara a través de línea de comandos.
- Descomprimir el archivo en la carpeta, luego de esto se procede a ejecutar el entorno desde el acceso directo.
- Descargar el JDK se configura variable "Path" asignándole la ruta de la carpeta "\bin" del JDK y la ruta de la carpeta "\bin" del WSMX.
- Se inicia línea de comandos como administrador si se encuentra en window 7, haciendo clic en Inicio en Buscar Programas y Archivos escribir "cmd" y luego se hace clic derecho y ejecutar como administrador de windows.
- Con línea de comandos se va a la ruta "\wsmx-0.5-dit" y de da el comando "startc.bat".
- Estando iniciado el servidor ya se pueden enviar datos desde WSMT.

### 2. Modelo de Servicios

Es un modelo que establece las interfaces a desarrollar con respecto a los servicios web semánticos de los objetos y los servicios Web del índice mismo.

#### 2.1 Los servicios básicos del índice son:

9. **EnviaEntidad (string Entidad):** Esta función recibe un conjunto de caracteres a los cuales les hace un proceso de ampliación de consulta y posteriormente busca en el índice las coincidencias con respecto a toda la información que ha sido encontrada en el repositorio de información en la etapa anterior de indexación. Retorna un conjunto de resultados

ordenados por relevancia y con enlaces a los servicios web básicos que traen la información del repositorio Servidor IoT.

10. **RetornaConceptos (string Conceptos)**: Esta función recibe un término y lo compara en la ontología con el fin de retornar los conceptos más relacionados al mismo, desde el punto de vista medioambiental.
11. **RetornarMapaLugar (string Lugar)**: Esta función recibe un lugar en términos de latitud y longitud y devuelve la información relacionada a los sensores que tienen influencia en dicho lugar y sus relaciones a conceptos medioambientales.
12. **RetornarMapaVariableAmbiental (string variableAmbiental)**: Esta función recibe una variable medioambiental y devuelve la información relacionada a los sensores que miden dicha variable y su ubicación geográfica.

## 2.2 Los servicios web semánticos de los sensores son:

1. **RetornarMetadatos (IdSensor)**: Este servicio permite retornar en formato XML la información relacionada al sensor particular. El enriquecido con la ontología.
2. **RetornarDatos([Fecha Inicio], [Fecha Fin])**: Esta función permite obtener información de los datos que mide el sensor. Si no aporta las fechas se devuelve el dato actual y son fechas los datos producidos entre las mismas.

## 2.3 Lo servicios del Caso se estudió de Información medioambiental:

1. **CercaníaSensor (lugar)**: Corresponde al cálculo que se debe hacer de las medidas de posicionamiento geográfico obtenidas con respecto al área de influencia del biotipo (lugar) y los sensores que se encuentran en dicha área de influencia. El área de influencia por ahora se maneja como el área circular alrededor del lugar y definida en el concepto de área de impacto del biotipo que se identifique.
2. **ObtenerInfoMedioAmbiental (lugar)**: Corresponde a un cuadro elaborado de conceptos y mediciones que se pueden obtener de los sensores del área de impacto y la información en la ontología.
3. **HistóricosSensor (IdSensor, periodo de tiempo)**: A partir del periodo de tiempo retorna la información de los datos medidos y almacenados en el servidor IoT. Utiliza la función RetornarDatos de los servicios de sensores, pero debe implementar un mecanismo adecuado de presentación de la información y relación con la información medioambiental con el fin de que sea fácilmente entendible para el usuario.
4. **TendenciaVariableAmbiental (variable medioambiental)**: La tendencia debe implementar un sistema de análisis correlacional básico con el fin de proveer información de tendencias de variables. Se utilizará lo más básico pero aquí se pueden colocar análisis más elaborados que se dejan para futuros estudios.
5. **VariablesDiferentesBiotipo (lugar)**: Este método debe comparar los valores de referencia almacenados en el Biotipo de ciertos lugares y posteriormente con las mediciones de los sensores establecer si se encuentran en los rangos establecidos o se han salido de sus valores esperados. Para esto hay que buscar la información de referencia y poblar la ontología con la misma en un lugar seleccionados para demostrar sus posibilidades de uso.
6. **CausandoDañoAmbiental (lugar)**: Este método devuelve la información de los posibles agentes contaminantes que están afectando cierto lugar. Para esto debe tomar los sensores de dicho lugar y posteriormente utilizar la función de VariablesDiferentesBiotipo,

las variables identificadas se relacionan a sus agentes contaminantes y se retorna una respuesta en una estructura de conceptos provista por la ontología en la cual se identifican dichos contaminantes.

7. ***Daños Ambientales (lugar, AgentesContaminantes)***: A partir de un conjunto de agentes contaminantes y su magnitud se desarrolla un modelo matemático sencillo para relacionarlo a un daño medioambiental. La respuesta son los posibles daños causados el ecosistema por estos agentes.

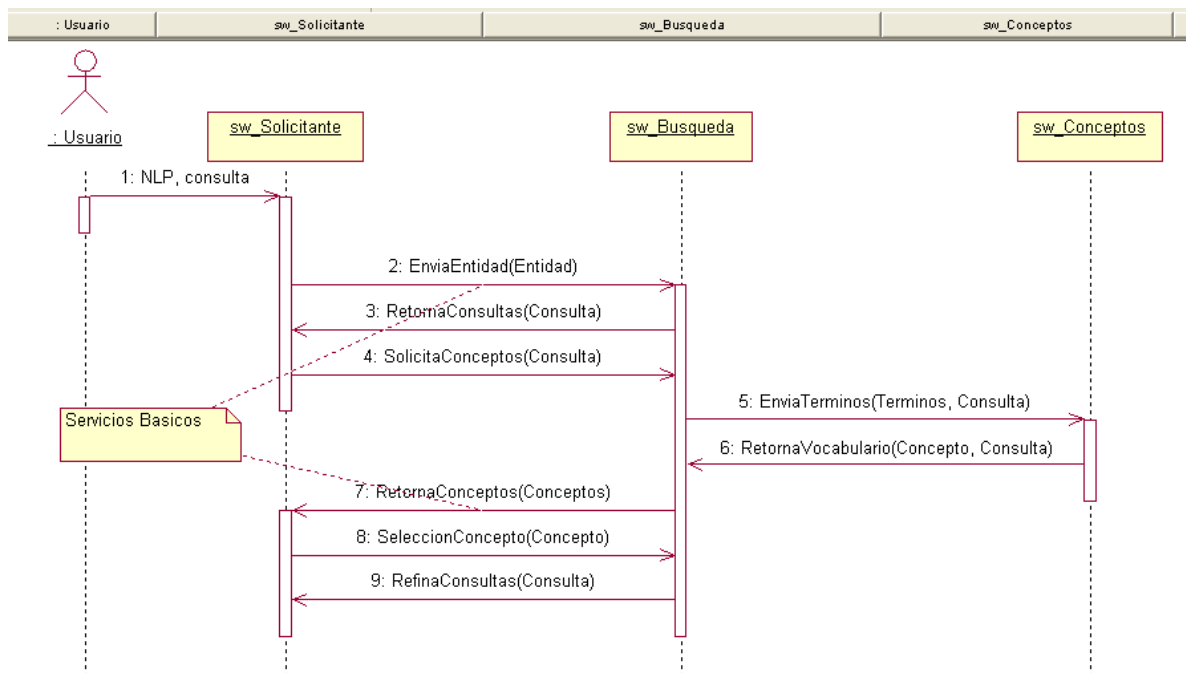
### 3. Implementación de los Servicios Básicos del Índice

La implantación de los servicios básicos del índice EnvíaEntidad() comprende hacer una consulta al índice y el servicio RetornaConceptos() comprende concretamente el refinamiento de la consulta se plantean unidos ya que siempre se requiere de hacer una consulta para luego refinar o buscar mapa o por variable ambiental, metadatos, fecha, cercanía, información ambiental etc.

Siguiendo el meta lenguaje WSMO se planteó el siguiente procedimiento para el desarrollo de los SWS:

#### 3.1 Creación del modelo de secuencia de los servicios:

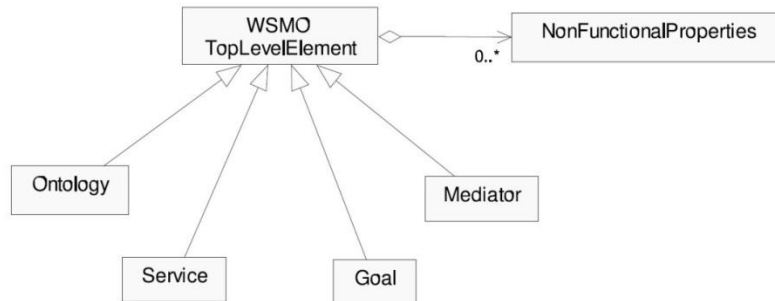
Figura 40. Modelo de secuencia de servicios



#### 3.2 El modelo semántico en WSMO de servicios web semánticos

El modelo conceptual(Bouros 2005) del meta lenguaje WSMO es materializado completamente por la herramienta WSMT.

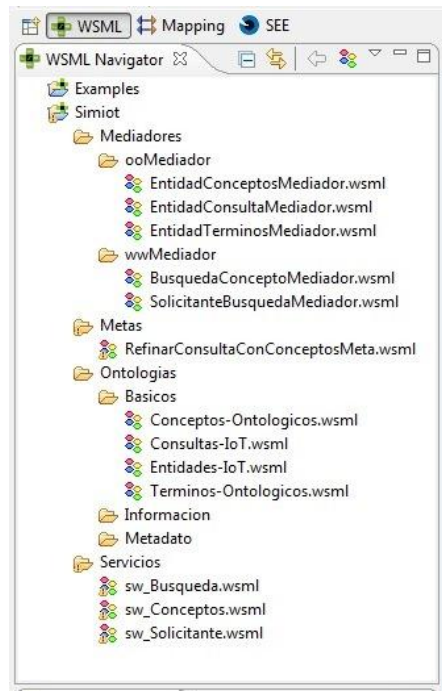
Figura 41. Modelo semántico en WSMO



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

Una vez implementado en la herramienta WSMT:

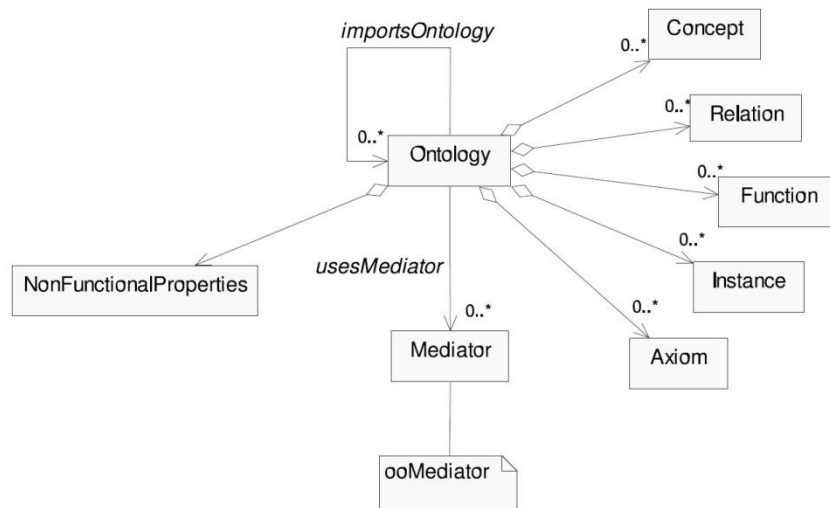
Figura 42. SIMIOT en la herramienta WSMT



### 3.3 Creación de las Ontologías:

El componente principal de las ontologías el son los conceptos por este motivo se crean una ontología por concepto. WSMO establece un modelo para la creación de las ontologías:

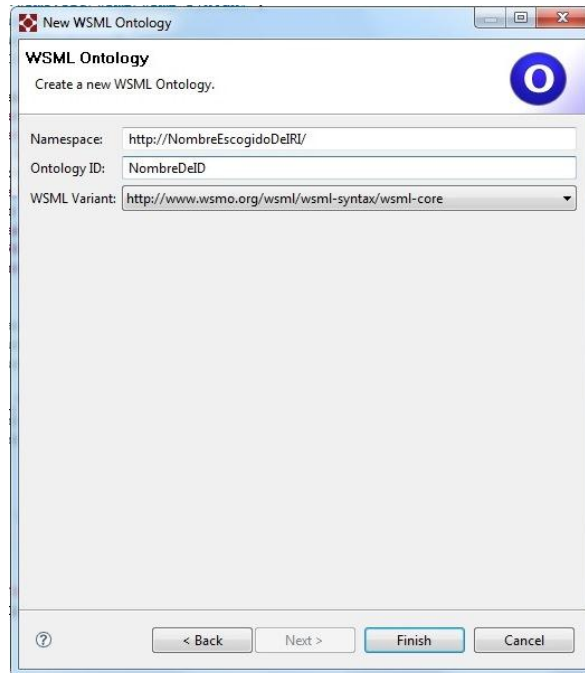
Figura 43. Modelo de Ontologías en WSMO



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

Para crear una ontología en WSMT se hace clic derecho sobre un "Folder" >New>Ontology se crea un archivo como ".wsml". Una vez realizado esto se establece un nombre de archivo.

**Figura 44. Crear nueva Ontologías en WSMT**



El “Namespace” es un IRI o dirección URL el cual debe llevar el (/) al final para que el WSMT concatene la IRI con el nombre de las ontologías para y así direccionar la ontología. El “WSMML Varians” son las variantes de WSMML, el servidor WSMX puede trabajar hasta con la variante “Fight”. Esta es una forma simple de definir el IRI.

Formas más elaboradas de los nombres de archivos o “File name” del archivo “wsml” puede ir en minúsculas y si son varias palabras separado por (-). Por ejemplo (contaminación-ambiental).

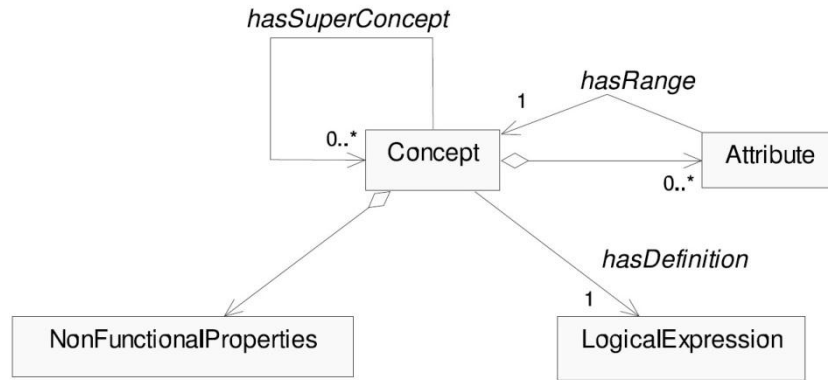
Una forma más adecuada para de definir IRI que ayude a determinar a simple vista las características del paquete que nos sirva de guía en el trabajo con las IRIs. Por ejemplo (<http://www.simiot.ca/ontologias#ContaminacionAmbiental>):

- Sección IRI para el paquete principal o Namespace (<http://www.simiot.ca/>)
- Sección IRI para el paquete de elementos (ontologias#)
- Sección IRI para el nombre del elemento o Ontology ID (ContaminacionAmbiental)

### **3.4 Creación de los Conceptos:**

Los conceptos son una representación de los mensajes que serán enviados entre los servicios, entonces se identifica un concepto para cada tipo de mensaje. Una vez creada la ontología se crean los conceptos, si en el modelo del servicio se representó como EnvíaEntidad(Entidad) el nombre del concepto sería “Entidad” pues es el tipo de mensaje que se va a enviar al servicio que está esperando la descripción de la entidad buscada. WSMO establece en el modelo para la creación de conceptos:

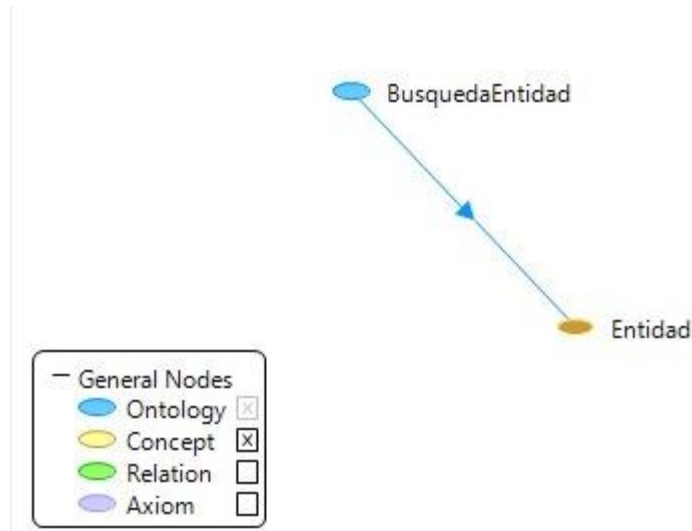
**Figura 45. Modelo de Conceptos en WSMO**



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

Para crear un concepto se hace clic derecho en la ontología>Add>Concept y luego se define los atributos no funcionales si se desea describir más detalladamente el concepto:

**Figura 46. Representación grafica de Concepto en WSMT**



Formas más elaboradas de los nombre al "Identifier" del concepto, que puede ir en minúsculas y si son varias palabras separado por (-). Por ejemplo (contaminantes-aire).

Luego en la definición de los atributos no funcionales del concepto definir el atributo "title" y en valor colocar el nombre bien escrito con espacios, tildes y mayúsculas. Por ejemplo (Contaminantes del Aire).

### **3.5 Creación de Atributos:**

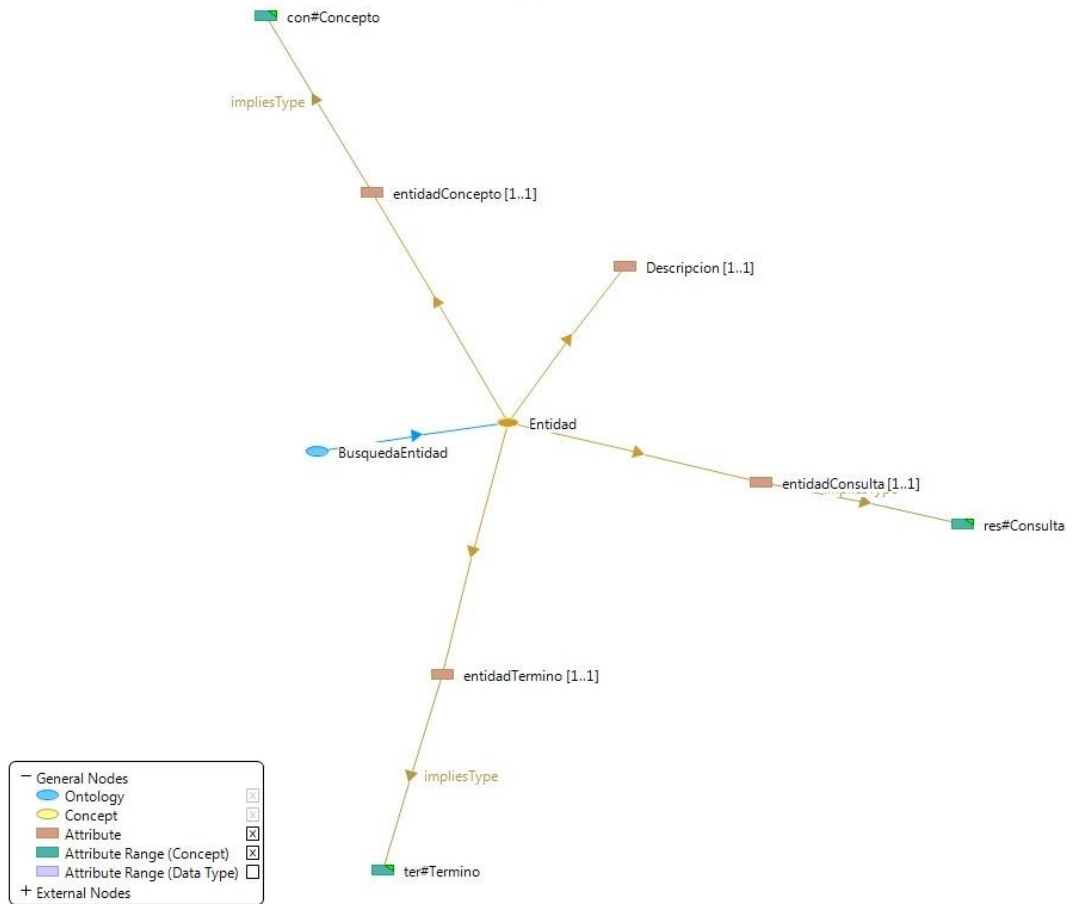
Los Atributos hacen parte de los Conceptos según el modelo de WSMO. Una primera función de los atributos es representar las precondiciones para el servicio solicitante que inicia, creamos atributos en el concepto "Entidad" de la siguiente forma:

- Atributo "EntidadConsulta" de tipo "Consulta".
- Atributo "EntidadTermino" de tipo "Termino".
- Atributo "EntidadConcepto" de tipo "Concepto".

Una segunda función de los atributos es representar los mensajes que serán enviados entre servicios en el caso nuestro se le dio esta tarea al atributo "Descripcion" el cual representa una cadena de texto en lenguaje natural que describe los entidades en que los usuarios están interesados consultar. "Descripcion" no presenta tipo de concepto porque no es un atributo de un concepto externo al concepto "Entidad".

En WSMT se presentan de la siguiente forma:

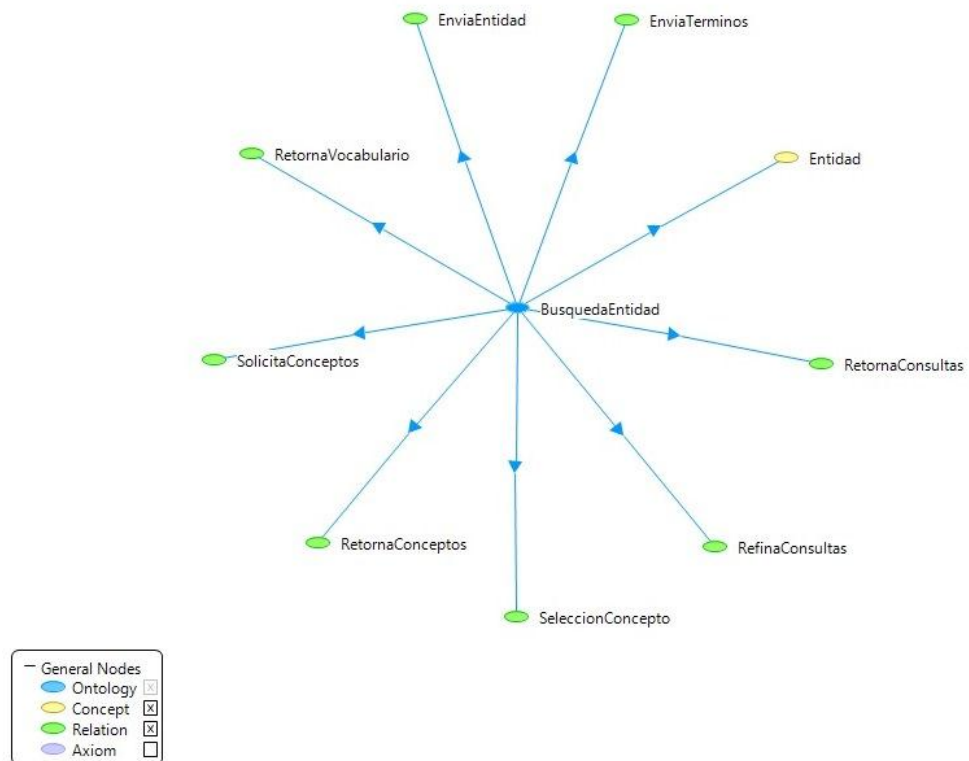
**Figura 47. Representación grafica de Atributos en WSMT**



### 3.6 Creación de Relaciones

Las Relaciones hacen parte de la ontología según el modelo WSMO. Las Relaciones son las conexiones por las cuales se envían los mensajes entre servicios, en nuestro caso fueron definidos en el modelo conceptual de servicios, los nombres de las relaciones deben coincidir con las definidas en el modelo de servicios. En WSMT las relaciones para la ontología "Entidad" se representan de la siguiente forma:

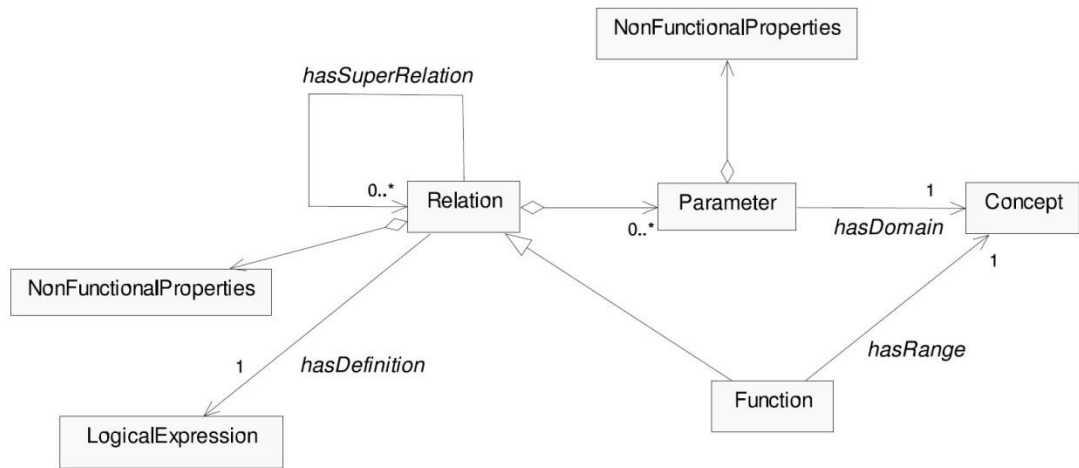
**Figura 48. Representación grafica de Relaciones en WSMT**



### 3.7 Creación de la Parámetros:

Para cada Relación definimos los parámetros que serán enviados entre servicios. Los parámetros hacen parte de las relaciones según el modelo WSMO y establecen una relación entre "Relation" y el "Concept":

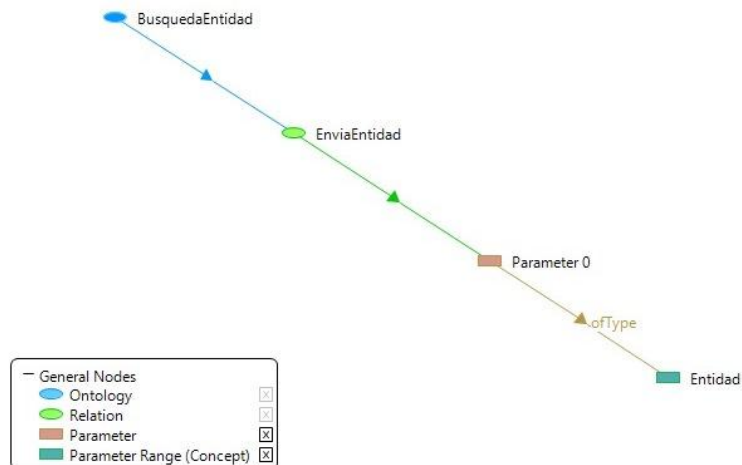
**Figura 49. Modelo de Parametros en WSMO**



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

Los parámetros en el modelo WSMO se crean como un tipo de concepto, “no como un parámetro con algún nombre“, el objetivo de esto es poder asociar el Atributo del Concepto que representa el mensaje a enviar con la Relación que envía el mensaje. En el caso de la relación EnviarEntidad(Entidad) el parámetro es Entidad, en WSMT se visualiza de la siguiente manera:

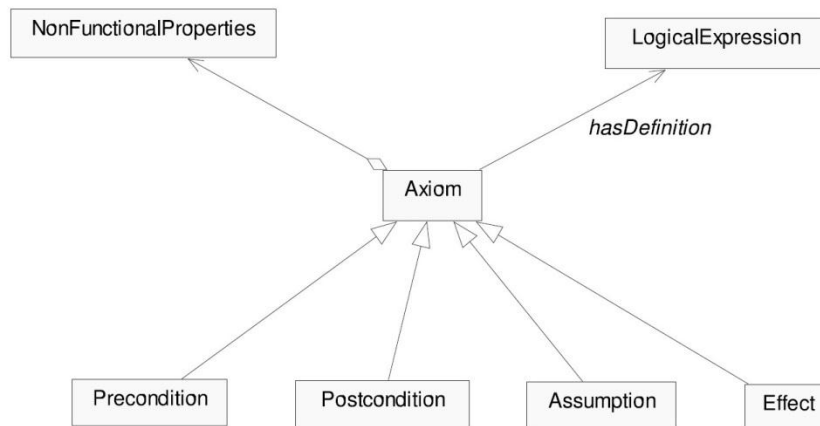
**Figura 50. Representación grafica de Parametros en WSMT**



### 3.8 Creación de Axiomas:

Los Axiomas (Fensel, Facca et al. 2011) hacen parte de las ontologías según el modelo WSMO. Tienen el objetivo de establecer relaciones explícitas entre conceptos, relaciones, servicios etc. Los axiomas crean nuevas anotaciones semánticas las cuales apoyan el razonamiento. El modelo WSMO define a los axiomas de la siguiente manera:

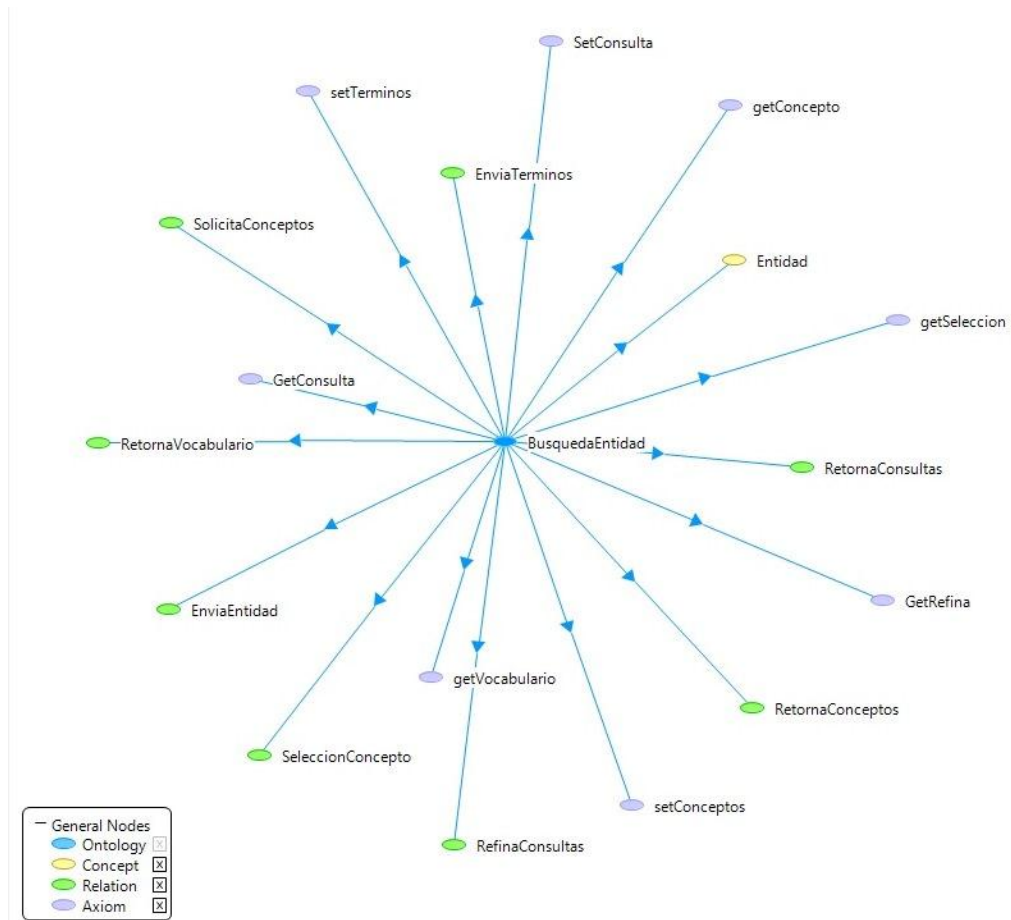
**Figura 51. Modelo de Axiomas en WSMO**



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

En nuestro caso se crea un axioma por relación definida con el objetivo es relacionar los parámetros de las relaciones y los atributos de los conceptos encargados de representar los mensajes que serán intercambiados entre servicios. Existe un axioma por relación definida en WSMT como se muestra en la siguiente figura:

**Figura 52. Representación grafica de Axiomas en WSMT**



La construcción de los axiomas es autocompletada por WSMT utilizando la tecla “Barra espacio” facilitando la definición y evitando errores. Los axiomas se pueden editar desde el editor visual como desde el editor de texto. Algunos ejemplos para la creación de aximas son relacionados a continuación.

### **3.8.1 Axioma de la relación EnviaEntidad(Entidad):**

La relación representa el envío de una descripción en lenguaje natural de una entidad por parte del servicio solicitante al servicio búsqueda. La relación “EnviaEntidad” solo contiene un parametro de tipo “Entidad” que se relaciona por medio del axioma con el atributo “Descripcion” que pertenece al concepto “Entidad”. Todo esto se realiza utilizando la variable de instancia “?y” asignar la variable de instancia “?x”. Variable instancia significa que es el nombre que recibirá el nombre al momento de retornar la consulta:

```

axiom SetConsulta
  definedBy
    EnviaEntidad(?x) :- ?y memberOf EnviaEntidad and ?y[ Descripcion hasValue ?x].

```

### 3.8.2 Axioma de la relación RetornaConsulta(Consulta):

La relación “RetornaConsulta” contiene un parametro de tipo “Consulta” que se relaciona por medio del axioma con los atributos “Titulo, Enlace, ID\_Consulta” que pertenece al concepto “Consulta” y con la variable “?y” asigna “?x”.

```

axiom GetConsulta
  definedBy
    RetornaConsultas(?x) :- ?y memberOf res#Consulta
      and ?y[ res#Titulo hasValue ?x, res#Enlace hasValue ?x, res#ID_Consulta hasValue ?x ].

```

### 3.8.3 Axioma de la relación RetornaVocabulario(Concepto, Consulta):

La relación “RetornaVocabulario” contiene dos parámetros. El primero de tipo “Concepto” que se relaciona por medio del axioma con los atributos “PalabraConcepto” que pertenece al concepto “Concepto” y con la variable “?k” asigna “?x”. El segundo primero de tipo “Consulta” que se relaciona por medio del axioma con los atributos “ID\_Consulta” que pertenece al concepto “Consulta” y con la variable “?m” asigna “?y”.

```

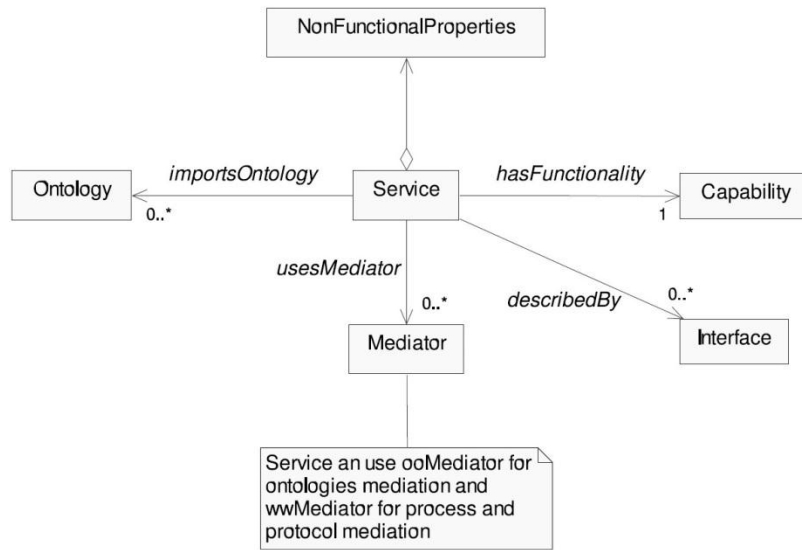
axiom getVocabulario
  definedBy
    RetornaVocabulario(?x, ?y)
      and ?k[ con#PalabraConcepto hasValue ?x] memberOf con#Concepto
      and ?m[ res#ID_Consulta hasValue ?y] memberOf res#Consulta.

```

## 3.9 Creación de Servicios:

Para crear una ontología en WSMT se hace clic derecho sobre un “Folder” >New>Web Service, se crea un archivo como “.wsml”. El modelo de servicios para WSMO importa las ontologías y se relaciona directamente con las capacidades y por ultimo puede o no necesitar mediadores. La representación del Servicio es:

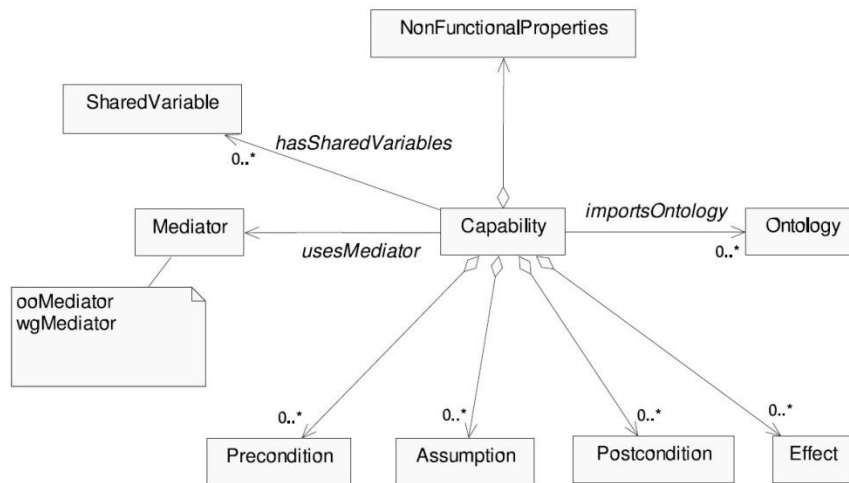
**Figura 53. Modelo de Servicios en WSMO**



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

Como para crear servicios es necesario crear “Capability” el modelo según WSMO:

**Figura 54. Modelo de Capacidades en WSMO**



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

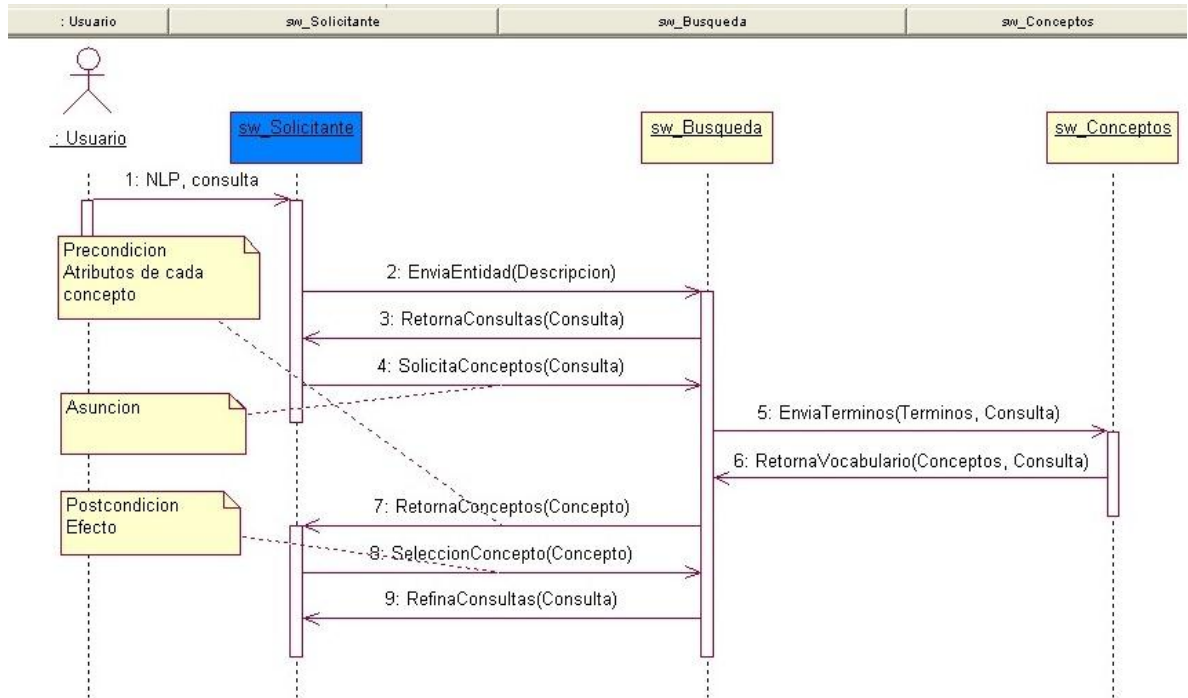
La definición de los elementos de las capacidades es una labor que tiene un especial cuidado, debido a que la precondiciones y pos condiciones de los servicios son elementos que se tienen en

cuenta a la hora de seleccionar los servicios automáticamente, siendo este aspecto uno de los pilares más característicos de los servicios web semánticos SWS.

Para nuestro caso definimos los siguientes modelos de servicios:

### 3.10 Servicio sw\_Solicitante

Figura 55. Modelo de Capacidades de servicio solicitante



#### 3.10.1 Expresión lógica para Pre condiciones:

Como se puede observar en el modelo de servicios la precondición del servicio solicitante está compuesta por los atributos “entidadConsulta, entidadTermino, entidadConcepto” y también por la relación “RetornarConceptos(Concepto)”. Para los primeros a través de la variable instancia “?Entidad” asignamos los atributos:

- Atributo “entidadConsulta” a la variable instancia “?cadena1”
- Atributo “entidadTermino” a la variable instancia “?cadena2”
- Atributo “entidadConcepto” a la variable instancia “?cadena3”

En segundo lugar la precondición es la relación RetornaConcepto(Concepto) definimos la variable instancia “?Concepto” concepto y de qué tipo de concepto es, en nuestro caso es de tipo “Concepto”.

```

definedBy
    ?Entidad memberOf _"http://BusquedaEntidad/Entidad"
and ?Entidad[
    _"http://BusquedaEntidad/entidadTermino" hasValue ?cadena1,
    _"http://BusquedaEntidad/entidadConcepto" hasValue ?cadena2,
    "http://BusquedaEntidad/entidadConsulta" hasValue ?cadena3]
and RetornarConcepto(?Concepto)
and ?Concepto memberOf _"http://BusquedaConcepto/Concepto".

```

### 3.10.2 Expresión lógica para Asunciones:

Esta expresión lógica ilustra el caso en que hay más de una relación haciendo porte de las asunciones, en este caso son la relación "RetornaConceptos(Consulta), BusquedaConcepto(Conceptos)".

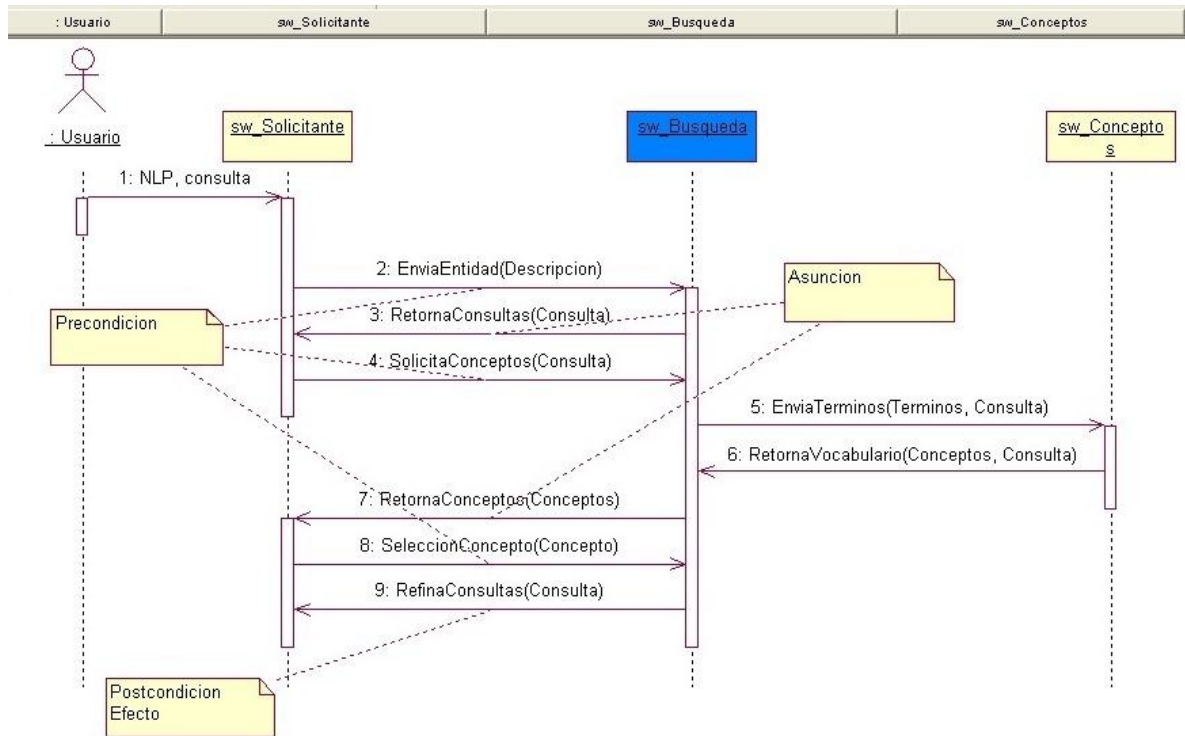
```

definedBy
    RetornaConsultas(?Consulta)
and RetornaConceptos(?Conceptos)
and ?Consulta memberOf _"http://ResultadosConsulta/Consulta"
and ?Conceptos memberOf _"http://BusquedaConcepto/Concepto".

```

### 3.11 Servicio sw\_Busqueda

Figura 56. Modelo de Capacidades de servicio busqueda



### 3.11.1 Expresión lógica para Pre condiciones:

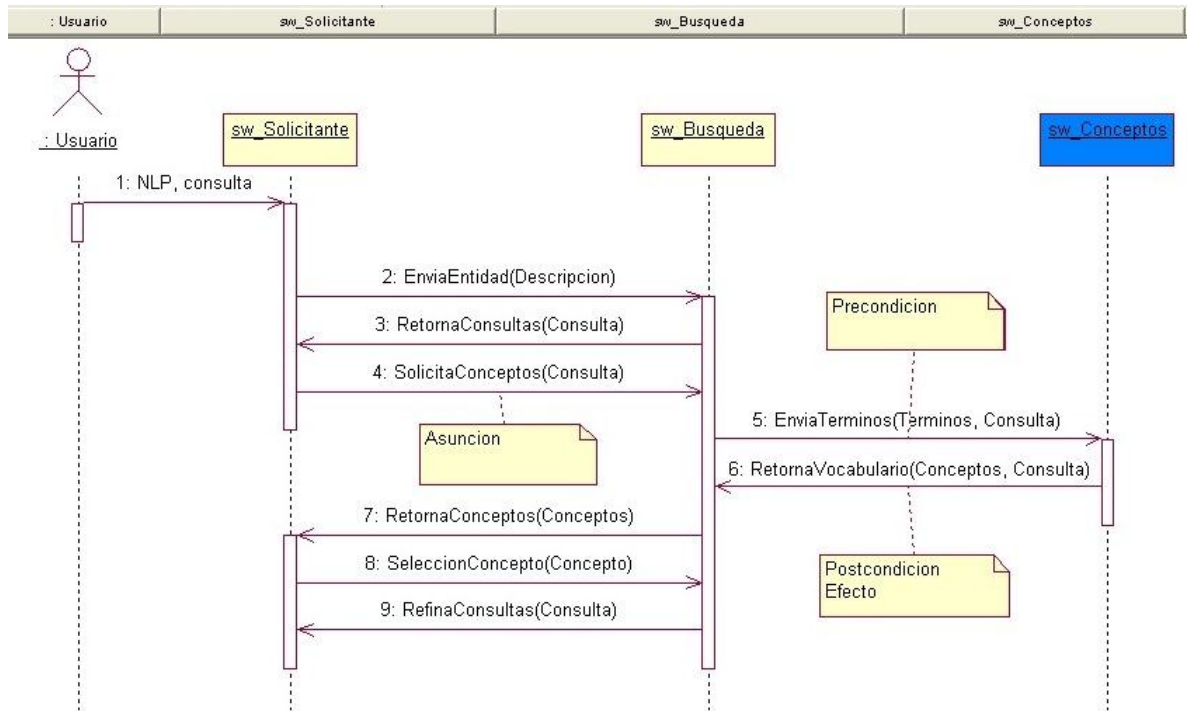
Esta expresión lógica ilustra el caso en que hay más de una relación haciendo parte de las precondiciones del "sw\_Busqueda", en este caso son la relación "EnviaEntidad(Entidad), SolicitaConceptos(Consulta), SeleccionConcepto(Concepto)". Nota: No es necesario colocar el mismo nombre del parametro "EnviaEntidad(?Descripcion)".

```

definedBy
  EnviaEntidad(?Descripcion) |
  and SolicitaConceptos (?Consulta)
  and SeleccionConcepto (?Concepto)
  and ?Descripcion memberOf _"http://BusquedaEntidad/Entidad"
  and ?Consulta memberOf _"http://ResultadosConsulta/Consulta"
  and ?Concepto memberOf _"http://BusquedaConcepto/Concepto".
  
```

### 3.12 Servicio sw\_Conceptos

Figura 57. Modelo de Capacidades de servicio concepto



### 3.12.1 Expresión lógica para Pos condiciones

Esta expresión lógica ilustra el caso en que una relación tiene más de un parámetro. La pos condición del “sw\_Conceto”, es la relación “RetornaVocabulario(Conceptos, Consulta)”. Lo que se tiene que hacer para cada parámetro definir el tipo de concepto al que pertenece.

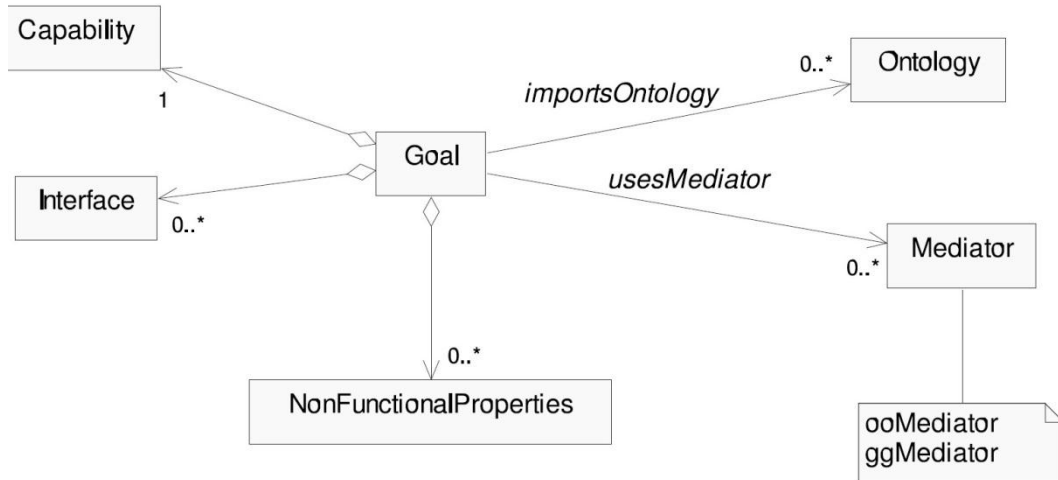
```

definedBy
    RetornaVocabulario (?Conceptos, ?Consulta)
and ?Conceptos memberOf _"http://BusquedaConcepto/Concepto"
and ?Consulta memberOf _"http://ResultadosConsulta/Consulta".
    
```

### 3.12.2 Creación de Metas

Para crear una ontología en WSMT se hace clic derecho sobre un “Folder” >New>Goal, se crea un archivo como “.wsml”. El “Goal” es la meta que tiene la combinación de los servicios que se comunican. El modelo WSMO define las metas de la siguiente manera:

Figura 58. Modelo de Metas para WSMO



Tomado de: Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches, (Bouros 2005)

### 3.12.3 Expresión lógica para Pos condiciones:

La expresión lógica de los goals define postcondiciones para cada atributo de concepto por medio de una instancia explicita, no variable:

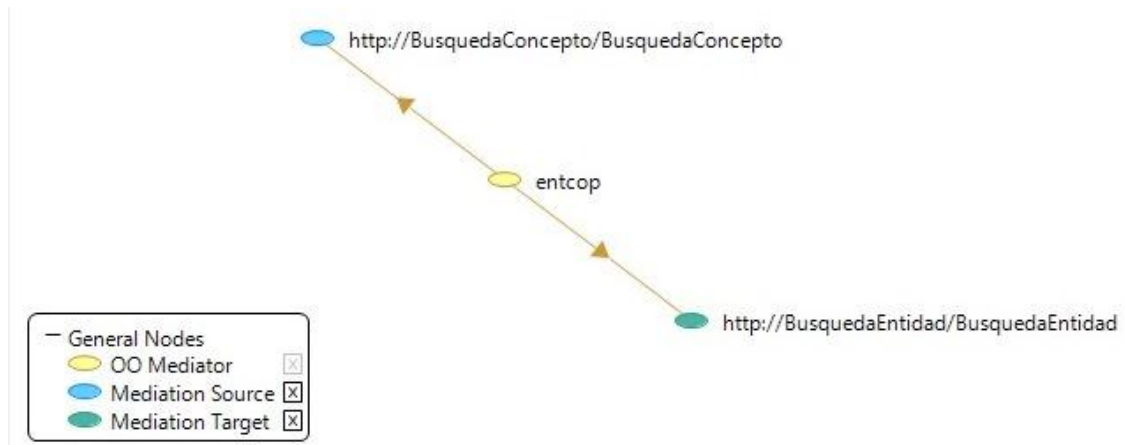
- Con la variable instancia “?Refina1” asigna el atributo “Descripcion” del concepto “Entidad” se asigna la instancia “Sensores de temperatura en bucaramanga”.
- Con la variable instancia “?Refina1” asigna el “Enlace” del concepto “Entidad” se asigna la instancia “http://www.xively.com”.
- Con la variable instancia “?Refina1” asigna el “Titulo” del concepto “Entidad” se asigna la instancia “Temperatura bucaramanga”.
- Con la variable instancia “?Refina1” asigna el “PalabrasTermino” del concepto “Entidad” se asigna la instancia “temperatura, bucaramanga”.

```

?Refina1[
  _"http://BusquedaEntidad/Descripcion"
  hasValue "Sensores de temperatura en bucaramanga"]
memberOf _"http://BusquedaEntidad/Entidad"
and ?Refina2[
  _"http://ResultadosConsulta/Enlace" hasValue "url",
  _"http://ResultadosConsulta/Titulo" hasValue "nombre"]
memberOf _"http://ResultadosConsulta/Consulta"
and ?Refina3[
  _"http://BusquedaTermino/PalabraTermino"
  hasValue "termino"] memberOf _"http://BusquedaTermino/Termino".
    
```



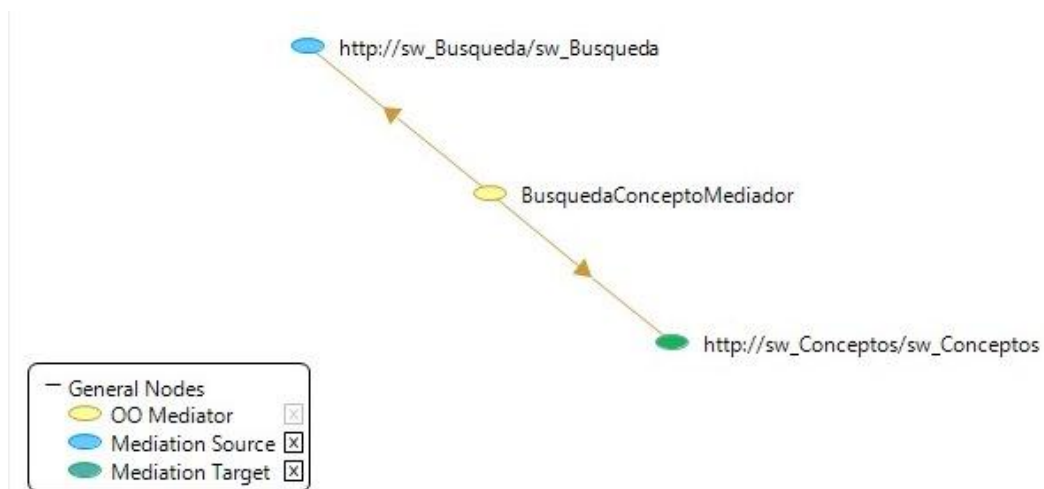
**Figura 60. Representación grafica para mediador de ontologias**



### 3.13.2 Definición de ww\_Mediator BusquedaConceptoMediatos:

Este mediador establece secuencias entre el servicio “sw\_Busqueda” como recurso (source) y el servicio “sw\_Concepto” como recurso (target). En WSMT se representa de la siguiente forma:

**Figura 61. Representación grafica para mediador de servicios**



Una vez creado el servicio se realiza la búsqueda desde la herramienta WSMT la cual da soporte para este fin por medio del WSMT-Reasoner implementado en la herramienta, finalmente los servicios se colocan en producción con la herramienta WSMX que es el servidor encargado de responder a las solicitudes de los clientes que solicitan servicios semánticamente y así se descubren e invocan los servicios automáticamente.