

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS
INTELIGENTES EN LA WEB DE LAS COSAS



Santiago Guerrero Narváz
Dalila Jhoana Riobamba

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán
2016

ESCENARIO DE INTERACCION SEMÁNTICA DE OBJETOS
INTELIGENTES EN LA WEB DE LAS COSAS



Santiago Guerrero Narváez
Dalila Jhoana Riobamba

Monografía

Director: PhD(c). Miguel Ángel Niño Zambrano
Asesor: PhD. Gustavo Adolfo Ramírez González

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán
2016

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

TABLA DE CONTENIDO

1	INTRODUCCIÓN	1
1.1	CONTEXTO GENERAL	1
1.2	DECLARACIÓN DEL PROBLEMA	2
1.3	ESCENARIO DE MOTIVACIÓN	2
1.3.1	Objetivo general	3
1.3.2	Objetivos específicos	3
1.4	CONTRIBUCIONES	4
2	MARCO TEORICO.....	5
2.1	IoT.....	5
2.1.1	Arquitectura IoT	5
2.1.2	Middleware.....	6
2.1.3	Dispositivos IoT.....	8
2.2	WoT.....	9
2.2.1	Objeto inteligente	9
2.2.2	Contexto.....	11
2.2.3	<i>Context aware</i>	13
2.2.4	Protocolos de comunicación	13
2.2.5	<i>Open Group</i>	15
2.3	WEB SEMÁNTICA	18
2.3.1	Metadatos.....	18
2.3.2	Servicios y aplicaciones de la web semántica.....	19
2.3.3	Indexación semántica	20
2.3.4	Ontologías	20
2.4	MISWoT	21
2.5	TRABAJOS RELACIONADOS	24
3	ARQUITECTURA PARA EL ESCENARIO DE INTERACCIÓN SEMÁNTICA EN LA WoT	27
3.1	CONCEPTOS ESPECÍFICOS.	27
3.2	VISTA ESTÁTICA.....	31
3.2.1	Capa de objetos.....	31
3.2.2	Capa <i>middleware</i>	31
3.2.3	Capa de objeto semántico.....	33
3.2.4	Capa de servicio.....	33

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

3.2.5	Capa de interacción	34
3.2.6	Capa de aplicaciones.....	34
3.3	VISTA DINÁMICA.....	35
4	IMPLEMENTACIÓN DEL ESCENARIO.....	39
4.1	FASE DE INICIO.....	39
4.1.1	Requisitos.....	39
4.1.2	Casos de uso críticos.....	39
4.1.3	Elementos tecnológicos.....	40
4.2	FASE DE ELABORACIÓN.....	43
4.2.1	Módulos.....	44
4.2.2	Formatos expandidos.....	45
4.3	FASE DE CONSTRUCCIÓN.....	50
4.3.1	Diagramas de secuencia.....	50
4.3.2	Interfaces al usuario.....	57
4.3.3	Iteraciones.....	59
4.4	FASE DE TRANSICIÓN.....	64
5	EVALUACIÓN DEL ESCENARIO.....	65
5.1	REFERENTES DE EVALUACIÓN.....	65
5.1.1	Latencia de interacción.....	66
5.1.2	QoS en interacción.....	66
5.2	ANÁLISIS DE LATENCIA EN LOS EXPERIMENTOS.....	67
5.2.1	Análisis de los casos de uso.....	67
5.3	CONCLUSIONES.....	73
6	CUMPLIMIENTO DE OBJETIVOS.....	75
6.1	LINEAMIENTOS DE CONFORMACIÓN E INTERPRETACIÓN DE LOS INDICADORES.....	75
6.2	DESCRIPCIÓN Y ALCANCE DEL CUMPLIMIENTO DE LOS OBJETIVOS.....	75
7	CONCLSIONES Y TRABAJO FUTURO.....	81
7.1	CONCLUSIONES.....	81
7.2	TRABAJO FUTURO.....	82
8	BIBLIOGRAFÍA.....	83

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

ÍNDICE DE TABLAS

Tabla 1 Características herramientas middleware IoT.	7
Tabla 2. Clasificación de los objetos del IoT.	10
Tabla 3. Tabla comparativa de los protocolos IoT.	15
Tabla 4. Caso de uso visualizar objeto.	46
Tabla 5. Caso de uso modificar servicio básico.	47
Tabla 6. Caso de uso crear servicio de interacción.	48
Tabla 7. Caso de uso eliminar servicio de interacción.	48
Tabla 8. Caso de uso modificar estado servicio de interacción.	49
Tabla 9. Caso de uso modificar estado servicio de interacción.	49
Tabla 10: Resultados primera fase de evaluación de funcionalidad.	61
Tabla 11: Plan de pruebas de la segunda fase de pruebas del escenario.	62
Tabla 12: Cálculo de QoS de interacción relacionado a la latencia de interacción.	67
Tabla 13: Latencia promedio de visualizar objeto.	68
Tabla 14: Latencia promedio en modificar estado de servicio básico.	69
Tabla 15: Latencia promedio en crear servicio de interacción.	70
Tabla 16: Latencia promedio en modificar estado servicio de interacción.	71
Tabla 17: Latencia promedio en eliminar servicio de interacción.	72
Tabla 18: Comparación de latencias en los casos de uso.	73
Tabla 19 Cumplimiento del primer objetivo específico.	77
Tabla 20. Cumplimiento del segundo objetivo específico.	79
Tabla 21. Cumplimiento del tercer objetivo específico.	80

ÍNDICE DE FIGURAS

Figura 1. Arquitectura en capas de Internet de las Cosas.....	5
Figura 2 Elemento objeto.....	16
Figura 3 Elemento Infoltem.....	17
Figura 4. Elemento omiEnvelope.	18
Figura 5. Marco conceptual del MISWoT.....	21
Figura 6. Arquitectura del MISWoT.....	22
Figura 7. Módulo de las propiedades del SO	23
Figura 8. MetaData.xml.....	30
Figura 9 Vista estática de la arquitectura.....	32
Figura 10 primera vista dinámica de la arquitectura.	36
Figura 11. Segunda vista dinámica de la arquitectura.....	37
Figura 12 Tercera vista dinámica de la arquitectura.	37
Figura 13 Cuarta vista dinámica de la arquitectura	38
Figura 14. Casos de uso críticos.....	40
Figura 15. Diagrama de protocolos.....	43
Figura 16. Elementos arquitectónicos según módulos.....	44
Figura 17. Diagrama de secuencia caso de uso visualizar objeto.....	50
Figura 18. Diagrama de secuencia caso de uso modificar estado servicio básico	52
Figura 19. Diagrama de secuencia caso de uso conectar objeto.....	53
Figura 20. Diagrama de secuencia caso de uso modificar estado servicio de interacción.....	54
Figura 21. Diagrama de secuencia caso de uso crear servicio de interacción.....	55
Figura 22. Diagrama de secuencia caso de uso eliminar servicio de interacción	56
Figura 23 Interfaces de Clipio.....	57
Figura 24. Interfaces de inicializar objeto	58
Figura 25. Interfaces de panel de usuario.....	59
Figura 26. Tendencia latencia promedio en visualizar objeto.....	69
Figura 27. Tendencia latencia promedio en modificar estado de servicio básico.	69
Figura 28. Tendencia latencia promedio en crear servicio de interacción.	70
Figura 29 Tendencia latencia promedio en modificar estado servicio de interacción.	71
Figura 30. Tendencia latencia promedio en eliminar servicio de interacción.	72

1 INTRODUCCIÓN

1.1 CONTEXTO GENERAL

Todos los objetos¹ con los que interactuamos pueden ser dotados de herramientas software y hardware para generar información sobre su estado, su entorno y sus relaciones, esta información puede ser entendida por el hombre de diferentes formas. Los sentidos humanos eran utilizados para interpretar de forma directa el estado de algún objeto que nos interesara y las primeras herramientas nos permitieron fabricar otros objetos que necesitásemos. Después, la llegada de alquimistas y los primeros pasos en las ciencias de materiales nos revelaron nuevos secretos sobre nuestro entorno y como utilizarlo para nuestro beneficio. Con el pasar del tiempo los objetos proliferaron en la vida cotidiana, y de mano a mano a los avances tecnológicos y científicos, se puede conocer más a fondo las variables de nuestro entorno. Ahora, nuestra vida cotidiana funciona alrededor de muchos y diversos objetos, que gracias a la tecnología, pueden aumentar sus capacidades, haciendo que cada objeto pueda comunicar sus estados y pueda modificar su entorno. Si se considera que otra rama de nuestro desarrollo ha sido el internet y la flecha apunta a que todos deben estar conectados, resulta natural pensar que los objetos también deben estar conectados, y no solo conectados con personas, sino conectados entre sí. A esta apuesta se la ha llamado internet de las cosas “*Internet of Things*” – IoT.

IoT es un término que actualmente está tomando mucha fuerza en el horizonte científico e industrial, ya que este supone que los objetos del mundo están dejando de tener un papel inerte y solitario, y están convirtiéndose en partes fundamentales y activas en la vida de las personas y las organizaciones. Un ejemplo, es el objeto más famoso del mundo actual, el dispositivo móvil. Sin embargo la IoT se encuentra en construcción y aún existen vacíos por resolver [1] [2].

Uno de los retos de la IoT, es lograr que los objetos interactúen entre sí de manera inteligente [3], esto se refiere a que debe minimizarse en lo posible la intervención humana para la ejecución de servicios entre los objetos conectados, para lograrlo, es necesario dotar de inteligencia a los objetos de nuestro mundo, convirtiéndolos en objetos inteligentes [4]; además es necesario aumentar las capacidades de los distintos sistemas que soportan la interacción entre los objetos. De esta forma se hace necesario que la IoT sume nuevas aristas y sea soportada sobre diferentes tecnologías para lograr una verdadera inteligencia en la interacción entre objetos.

¹ En el proyecto se utilizará el concepto de objeto como sinónimo de cosa o su traducción al inglés “thing”.

1.2 DECLARACIÓN DEL PROBLEMA

Lograr que los objetos puedan interactuar de forma inteligente implica resolver primero los problemas de heterogeneidad y conectividad [1] [2], debido a que existen diferentes tipos de hardware y software en la IoT que ha sido implementado utilizando protocolos de comunicación propios y poco estandarizados. Esto impide la comunicación entre los objetos de diferentes fabricantes o entre objetos que no soporten actualizaciones a nuevos protocolos. Para esto, la solución más aceptada ha sido el desarrollo de *middlewares* o servidores mediadores que realizan una representación digital de los objetos. Con el uso del *middleware* se superó en su gran mayoría la brecha de la heterogeneidad y se alcanzó a obtener la información unificada de gran cantidad de objetos inteligentes. Por otra parte la conectividad, además de definir como conectar los objetos, debe entender cómo lograr cooperación con los mismos en tiempo real; para solucionar lo anterior, un enfoque ha sido el desarrollo de técnicas semánticas, haciendo abstracciones entre hardware, aplicaciones y usuarios de la web de las cosas “*Web of Things*” - WoT, permitiendo construir servicios de información inteligentes y que suplan las necesidades de información de los usuarios; a esto último se le ha denominado la web semántica de las cosas [5].

Actualmente en la universidad del Cauca se encuentra en curso un proyecto de investigación denominado “interacción semántica de objetos en la WoT” [6], que tiene como principal objetivo “modelar la interacción semántica entre objetos inteligentes en la WoT, a través de un enfoque de indexación y búsqueda semántica”, en el cual se ha propuesto el: “Modelo de Interacción Semántica entre objetos inteligentes de la WoT” - MISWoT, involucrando el trabajo con índices semánticos, ontologías y manejo de servidores de objetos.

El presente trabajo aporta al anterior proyecto a través de la pregunta de investigación ***¿Cómo implementar mecanismos de interacción semántica de objetos inteligentes para la WoT, que permitan identificar, crear y ejecutar nuevos servicios como producto de su interacción?*** El trabajo realiza la implementación de un escenario de prueba para el MISWoT propuesto por Niño-Zambrano [6] definiendo las características hardware y software de los objetos inteligentes, implementando el manejo de ontologías de objetos inteligentes, definiendo e implementando el mecanismo de comunicación entre objetos inteligentes e implementando la creación de nuevos servicios derivados de la interacción entre objetos inteligentes.

1.3 ESCENARIO DE MOTIVACIÓN

Al estar inmersos en un mundo cambiante en que los objetos han tomado posesión de las rutinas diarias, hasta el punto que definen, relacionan e interactúan entre sí y con las personas; es imposible no centrar la atención en los diferentes objetos que se han convertido en parte activa para mejorar la calidad de vida, controlando espacios y, entre otras cosas, aumentando el bienestar individual y grupal.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Al ir en aumento el número de objetos conectados, se ve la necesidad de ahondar en problemas específicos que contribuyan a la creación de nuevos servicios que faciliten el quehacer de las personas, de acuerdo a esto, los objetos deben actuar de una manera inteligente para aportar soluciones y no aumentar los problemas que pueden presentarse a los diferentes usuarios, ya que se manejan gran cantidad de datos que están presentes en sensores y actuadores haciendo falta modelos y/o mecanismos para aprovechar dichos datos. Para resolver algunos de los problemas presentados en las diferentes áreas de la IoT en la universidad del Cauca se propone el MISWoT, como un modelo que mejora las formas de interacción entre objetos en la WoT con la ayuda de herramientas semánticas.

La motivación de este trabajo se centra en probar el modelo propuesto por el MISWoT haciendo una implementación física con el fin de realimentar el trabajo teórico y resolver los obstáculos que se encuentran. Además se detalla la capa de interacción definiendo las herramientas, protocolos y elementos que deben existir y que no están definidos en el MISWoT. Para lograr una implementación física del MISWoT se propone un escenario de interacción semántica de objetos inteligentes en la WoT, siguiendo un objetivo general y sus respectivos objetivos específicos.

1.3.1 Objetivo general

Implementar un escenario base de interacción semántica para la WoT, aplicando el modelo de interacción semántica en la WoT (MISWoT) propuesto por Niño-Zambrano [6], mediante la integración de herramientas y tecnologías existentes en la WoT, creando nuevos servicios derivados de dicha interacción a los usuarios.

1.3.2 Objetivos específicos

1. Desarrollar un conjunto de módulos software que permitan ejecutar mecanismos base de interacción semántica entre objetos de la WoT, siguiendo los conceptos y estructuras definidas en el MISWoT. Para esto se debe:
 - Determinar las características técnicas que debe tener un objeto inteligente, de tal forma que permita determinar, enviar y recibir información semántica entre los mismos objetos de la WoT.
 - Definir un mecanismo adecuado de gestión de la ontología de objeto semántico propuesta en el MISWoT, la cual se debe implementar para cada objeto inteligente.
 - Describir un escenario de comunicación entre objetos inteligentes de la WoT, de tal forma que permita la creación de nuevos servicios derivados de su interacción.
2. Construir un escenario de interacción semántica de objetos inteligentes de la WoT creando un conjunto de objetos inteligentes y sus servicios, para un caso de estudio particular que será seleccionado de acuerdo con los recursos existentes.
3. Evaluar el escenario de interacción semántica de objetos inteligentes de la WoT construido en el caso de estudio particular, para establecer la efectividad de la interacción y los servicios creados entre los objetos inteligentes.

1.4 CONTRIBUCIONES

- **Fortalecimiento teórico en las áreas del internet de las cosas, la web semántica y objetos inteligentes.**

A partir del estudio y el análisis de diferentes investigaciones a través de fichas bibliográficas, en adición a la presente propuesta; se fortalece la base teórica perteneciente a la institución académica en lo referente al IoT, WoT, los objetos inteligentes y la creación de nuevos servicios a partir de la interacción de objetos inteligentes soportados en la web semántica. Esto permite establecer un soporte para futuras investigaciones en dichas áreas. Se detalla el producto de las fichas bibliográficas consultadas en el anexo A.

- **Modelo de interacción semántica de objetos inteligentes en la web de las cosas para el escenario de Interacción semántica.**

Se realiza una adaptación y complementación del modelo MISWoT en el módulo de “interacción” fortaleciendo así este desarrollo y sus futuras aplicaciones. Este módulo se puede ver con más detalle en el capítulo 3.

- **Escenario de interacción semántica de objetos inteligentes en la web de las cosas.**

Desplegando los módulos hardware necesarios y utilizando aplicaciones de dispositivos embebidos para la gestión de los objetos inteligentes, una aplicación móvil para realizar la interfaz con el usuario, una aplicación web para la evaluación y control del escenario y herramientas web para la gestión del contexto, se implementa la primera versión del escenario de interacción semántica de objetos inteligentes para la web de las cosas.

- **Monografía del Trabajo de Grado**

Corresponde al presente documento, donde se describe el proceso utilizado en el desarrollo del proyecto, los problemas que se presentaron, las respectivas soluciones, los principales aportes, las conclusiones y recomendaciones para el desarrollo de futuras investigaciones. Este documento se divide en 8 capítulos diferentes, dentro de los cuales el primero es la introducción y una contextualización general del proyecto, el segundo capítulo define el marco teórico que reúne conceptos que se utilizaron para definir el trabajo, el tercer capítulo contiene la arquitectura del escenario, donde se definen los distintos elementos, herramientas, protocolos, etc. que soporta el escenario; el cuarto capítulo trata la construcción e implementación del escenario, en el quinto capítulo se puede encontrar las distintas pruebas realizadas al escenario, el sexto capítulo explica cómo se cumplieron los objetivos del trabajo propuesto, el séptimo capítulo muestra las conclusiones y los trabajos futuros y el octavo capítulo lista los referentes bibliográficos.

2 MARCO TEORICO

En este capítulo se puede encontrar los principales conceptos teóricos sobre los cuales se basa el presente trabajo, además se presentan algunas comparaciones entre diferentes tecnologías y las decisiones que se tomaron frente a las distintas alternativas. Para mayor detalle sobre el marco teórico y los conceptos que no se presentan en los siguientes apartados consultar el Anexo B.

2.1 IoT

Se basa en el actual internet definido como una red de redes, es decir una red que interconecta redes de computadoras con el fin de compartir recursos [7]. Acorde a lo anterior IoT, puede describir como la conexión inteligente de los objetos cotidianos como dispositivos móviles, televisores, sensores y actuadores, a internet, permitiendo nuevas formas de comunicación entre los objetos y las personas [3]. Es importante entender la arquitectura de la IoT para poder aprovecharla adecuadamente, además identificar las capas y sus funciones con el fin de diferenciar los posibles problemas y las herramientas que pueden ser aplicadas para resolverlos.

2.1.1 Arquitectura IoT

Hay muchas propuestas arquitectónicas de la IoT, pero todas en general proponen una arquitectura de capas, por lo cual se presenta en la Figura 1 una arquitectura enfocada en capas, la cual muestra una mayor generalización de los componentes del IoT, esta tiene dos divisiones distintas con una capa intermedia que comunica las otras. Las dos capas inferiores contribuyen a la captura de datos, mientras que las dos capas en la parte superior son responsables de utilización de los datos en las aplicaciones [8]

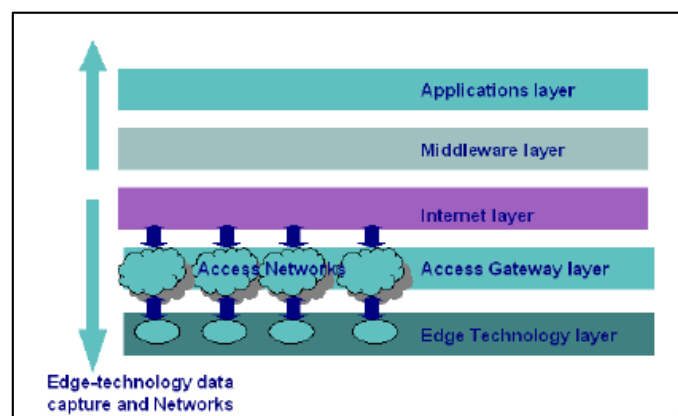


Figura 1. Arquitectura en capas de Internet de las Cosas. Fuente: [8]

- **Capa hardware (Edge technology layer).** Constituida por redes de sensores, sistemas embebidos, etiquetas identificación por radio frecuencia, etc. Esta capa representa a los datos primarios de los sensores y actuadores desplegados.

- **Capa de acceso (*Access gateway layer*).** Es la primera etapa del tratamiento de datos, se ocupa de enrutamiento de mensajes, publicación y suscripción y también lleva a cabo la comunicación entre plataformas.
- **Capa de aplicación (*Application layer*).** Entrega las aplicaciones a diferentes usuarios en la IoT.
- **Capa de middleware (*Middleware layer*).** Opera en modo bidireccional, actúa como una interfaz entre la capa de hardware y la capa de aplicación. Es responsable de las funciones críticas tales como administración de dispositivos y gestión de la información, también se ocupa de cuestiones como el filtrado de datos, la agregación de datos, el análisis semántico, control de acceso, el descubrimiento de información y el servicio de nombre de objetos.

2.1.2 Middleware

El *middleware* [9] se soporta sobre una plataforma software que permite integrar los objetos inteligentes en la red de información. Su principal objetivo consiste en abordar la interoperabilidad entre dispositivos heterogéneos que se encuentran en diversos dominios de aplicaciones. Otros objetivos del *middleware* con respecto a los objetos inteligentes son: adaptación, descubrimiento, administración, escalabilidad, gestión de grandes cantidades de datos y aspectos de seguridad. Los servidores *middleware* presentan las siguientes características:

- **Interoperabilidad.** Comparte información y utiliza la misma a través de diversos dominios de las aplicaciones que utilizan diversas interfaces de comunicación. Existen tres categorías. Primero, red; define los protocolos para el intercambio de información entre los diversos objetos a través de diferentes redes de comunicación, sin tener en cuenta el contenido de la información. Segundo, semántica; define los protocolos para el intercambio de información entre las diversas cosas a través de diferentes redes de comunicación, sin tener en cuenta el contenido de la información. Y tercero, sintácticas; hace referencia al formato y la estructura de la codificación de la información intercambiada entre los objetos.
- **Detección de contextos.** Recoge datos e identifica los factores que tienen un impacto significativo en la respuesta, además de esto es preciso procesar el contexto y realiza la toma de decisiones en base a ello.
- **Detección del dispositivo.** Permite a cualquier dispositivo de la red, detectar todos sus dispositivos vecinos y hacer notar su presencia a cada vecino en la red.
- **Seguridad y privacidad.** Son responsables de la confidencialidad y autenticidad.
- **Administración de volúmenes de datos.** Los datos pueden ser: identificación, posición, ambientales, históricos y descriptivos.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

A continuación se presenta la Tabla 1 donde se comparan las principales opciones de *middleware* enfocados en la IoT:

Tabla comparativa de herramientas middleware para IoT					
Nombre	Tipo Servidor	Protocolo web*	API	Soporte	
				Comunidad	Tutorial
Oracle	Nube	R	Java	Bueno	Bueno
Swarm	Nube	R	Python	Malo	Bueno
Axeda	Nube	R, C, M	Java, C	Malo	Bueno
OpenRemote	Nube	R	Java	Bueno	Bueno
Etherios	Nube	H	Android, Java, Python	Bueno	Medio
ioBridge	Nube	R	Arduino	Bueno	Bueno
Zatar	Nube	R, C	Java	Medio	Medio
Ayla	Nube	R	Base Linux	Malo	Malo
Echelon	Nube	R	C, C++ , Python	Bueno	Bueno
EVRYTHNG	Nube	R, C, M	Java, Jacascript	Malo	Medio
Exosite	Nube	H, C	Arduino, Python, C/C++, Java	Medio	Bueno
Xively	Nube	R, M	Arduino, Python, C, Java	Bueno	Bueno
Carriots	Nube	R, M	Arduino, Python	Bueno	Bueno
GroveStreams	Nube	R	Arduino, Python, Java	Bueno	Bueno
Lab of Thing	Local	R	Java, Arduino	Bueno	Bueno
Paraimpu	Nube	R	Arduino	Medio	Medio
SensorCloud	Nube	R	Python, Java, C#	Malo	Medio
ThingSepak	Nube	R	Arduino, C, Python	Bueno	Bueno
XOBXOB	Nube	R	Arduino	Medio	Medio
Lab of Thing	Local	NO	Arduino	Bueno	Bueno
2Lemetry	Nube	M	Arduino, Python, C	Medio	Medio
M2MLabs	Local	R	Arduino, Python	Malo	Bueno
OpenHAB	Local	H, M	No especifica	Medio	Medio
Node-RED	Local	H, M	Node.JS, Arduino	Bueno	Bueno
The Thing System	Local	Propio	Arduino	Bueno	Medio
DeviceHive	Local	R	C++, Python, C, Arduino	Malo	Medio
Mosquitto	Broker	M	Python, Java	Meio	Medio
Paho	Broker	M	Python, C++, Java, JavaScript	Bueno	Bueno
ThingMQ	Broker	R, C, M	Python, C++, Java, JavaScript	Medio	Medio

*R: REST, C:CoAP, M:MQTT, H:HTTP

Tabla 1 Características herramientas middleware IoT. Fuente: propia

La Tabla 1 compara diferentes herramientas middleware ofrecidas en la web, encontradas en distintos foros y sitios web que comparten información sobre temas de IoT. Las herramientas que se presentan son de acceso gratuito, algunas cuentan con un modo de prueba limitado. Se relaciona la herramienta con el tipo de servidor en la cual está soportado, encontrando tres principales: servidor alojado en la nube (nube), servidor que debe alojarse de forma local (local) y servidor de tipo *bróker* (*bróker*). Se relaciona la

herramienta con los tipos de protocolos que soporta, diferenciando los siguientes: REST (R), CoAP (C), MQTT (M) y HTTP (H). Se relaciona la herramienta con las API's y librerías ofrecidas por la plataforma o por la comunidad en distintos lenguajes de programación. Y por último, se evalúa los soportes ofrecidos por la comunidad y por la plataforma (tutoriales, manuales, ejemplos) en tres niveles: bueno, medio y malo.

A partir de la Tabla 1 se realizó la comparación entre las herramientas que pueden ser utilizados para el presente proyecto como servidor *middleware* y se concluyó en seleccionar *Xively*, esto debido a que este posee buen respaldo en la comunidad, entrega API's para los lenguajes de programación utilizados en el desarrollo del escenario y existe un desarrollo previo en el modelo MISWoT utilizando *Xively* como *middleware*. Además la interfaz de *Xively* es muy cómoda y fácil de utilizar, presentando varias funcionalidades útiles, como curvas sobre los datos entregados por los objetos en tiempo real, anotación de conceptos por medio de etiquetas y división del objeto y sus recursos en canales.

2.1.3 Dispositivos IoT

Para el presente proyecto se hace necesario seleccionar una herramienta que permita manipular de una forma profunda el comportamiento del objeto, por eso quedan descartados objetos de código fuente cerrado o que no provean API's u otras herramientas que permitan su desarrollo. Se opta por herramientas de hardware libre, específicamente placas de desarrollo. Existen muchas opciones en el mercado de placas de desarrollo que presentan diferentes capacidades, entre ellas se puede destacar las siguientes:

- *Arduino* [10], es una placa de desarrollo electrónica de código abierto basada en hardware y software fácil de usar. Está pensada para que la pueda utilizar cualquier persona que haga proyectos interactivos.
- *Raspberrypi* [11], un ordenador de placa reducida de bajo coste desarrollado en Reino Unido, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.
- *BeagleBone* [12] es una placa de hardware libre de bajo consumo producida por *Texas Instruments* en asociación con *DigiKey* y *Newark*. Fue diseñado con el desarrollo de software de código abierto.

Después de comparar las distintas posibilidades en cuestión se opta por seleccionar la placa *Intel Galileo*, principalmente porque la universidad del Cauca tiene la posibilidad de proporcionar estas placas para el desarrollo del presente proyecto.

- *Intel Galileo* [13], Es el primer producto de una línea de tarjetas de desarrollo *Arduino* basado en la arquitectura *Intel x86*, lanzada por la empresa Intel en el año 2013 con fines educativos. Cuenta con un gran procesamiento de bajo consumo y coste soportado en el procesador *Intel Quark Soc X1000*. Está diseñado para ser compatible con los escudos y el lenguaje de programación de *Arduino*, aunque es compatible, también, con sistemas operativos como Linux y Windows. Cuenta con los pines analógico - digital (al igual que las placas *Arduino*) permitiendo realizar lecturas de elementos como sensores. También tiene una sección dedicada a realizar la regulación

a 5 voltios de fuentes hasta de 12 voltios, facilitando el manejo de alimentación. Finalmente mediante la interfaz SD es posible almacenar los distintos sistemas operativos que puede soportar la placa.

2.2 WoT

La WoT [14] hace referencia a un entorno donde los objetos cotidianos son identificables, legibles, reconocibles, direccionables y controlables a través de internet. El ámbito contextual de la WoT supera al de web actual, permitiendo que los objetos físicos se puedan manejar a través de motores web.

En la actualidad se carece de una buena accesibilidad a los objetos a través de interfaces comunes, siendo esta característica esencial para la construcción de aplicaciones que exploten las capacidades en un contexto. Además, la heterogeneidad de los sistemas de computación ubicua plantea un problema importante: no hay ninguna especificación clara de las características comunes de los procesos para controlarlos o consultarlos. Los objetos físicos pueden ser dinámicos en el espacio y el tiempo, lo que implica nuevos enfoques para gestionar sus estados.

Los avances sobre el acceso virtual y el control de un gran número de objetos físicos en una plataforma común, está ganando un gran interés, y su investigación se ha visto impulsada en los últimos años. Se cuenta con la especificación de los tipos de objetos que deben integrarse a través de la web, clasificándolos utilizando una estructura ontológica que identifica sus diversas propiedades. Este es un primer paso para la creación de un modelo común de gestión.

2.2.1 Objeto inteligente

Los objetos inteligentes, son todos los entes físicos o virtuales, que pueden compartir cualquier tipo de información sobre sí mismos. Estos son la base fundamental dentro del paradigma del IoT; y representan, básicamente, las necesidades de información de los usuarios.

A los objetos IoT, se les puede atribuir el desempeño de una función o rol [15], dependiendo de sus capacidades informáticas tales como: conectividad de red, energía disponible, escenarios de tiempo y espacio, entre otras. Estas características se pueden tener en cuenta al momento de clasificar los sensores de acuerdo a las capacidades que adquieren. Dichas características se pueden estructurar como:

- Básicas, un objeto puede pertenecer al mundo real o al mundo virtual.
- Generales, un objeto puede utilizar un servicio como interfaz para comunicarse con otro objeto. El objeto contiene sensores y actuadores para interactuar con el medio ambiente.
- Sociales, un objeto puede interactuar con otro objeto y con personas generando una información semántica de dicha relación.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- Autonomía de funciones, un objeto realiza tareas autónomas y tiene capacidad de razonamiento, capacidad de interactuar y aprender de otros objetos y el medio ambiente.
- Auto replicación y control de funciones, red de información con capacidades de detección. Establece relaciones entre los humanos y los sistemas físicos.

Los objetos tienen características especiales como: diferentes tipos de datos o señales [16] capacidad de almacenamiento, procesamiento, comunicación, autonomía energética, localización, origen, estado, utilización y una identificación única [17]. Es necesario definir una taxonomía sobre las características y propiedades de los objetos actuales, esto con el fin de fundar las bases para la conceptualización de los mismos. Así el trabajo [4], permite caracterizar los objetos basándose en tres dimensiones: comunicación, procesamiento y almacenamiento; además de una identidad. Las cuatro características proponen una clasificación de objetos de la IoT de acuerdo a sus capacidades intrínsecas

Capacidades		Capacidades intrínsecas				Ejemplos
		I	P	C	A	
Núcleo		✓				RFID o códigos de barras.
Primitiva	Difuso	✓	✓			Lavadora, horno microondas.
	Conexión	✓		✓		Altavoces, auriculares.
	Lectura	✓			✓	CD, DVD.
Compleja	Social	✓	✓	✓		Control remoto, teléfonos fijos.
	Adheridos	✓		✓	✓	USB stick, RFID tags.
	Aparato	✓	✓		✓	Calculadora, juegos de mano.
Objetos inteligentes		✓	✓	✓	✓	PDA's, PC's.

Tabla 2. Clasificación de los objetos del IoT. Fuente: Propia

En la Tabla 2 se distinguen cuatro características: identificación (I), procesamiento (P), comunicación (C) y almacenamiento (A).

A partir de la Tabla 2, se puede definir los objetos inteligentes [18], como objetos de la IoT que se puede identificar claramente y exhiben capacidades de procesamiento, comunicación y almacenamiento. Estos objetos cuentan con una representación digital, la cual permite interactuar con las aplicaciones y los usuarios en la web. Debido a sus características son los elegidos para crear de manera más intuitiva servicios web semánticos y exponer su funcionalidad en la WoT. Los otros tipos de objetos necesitan desarrollar interfaces superiores que los administre.

2.2.2 Contexto

El contexto según [19] [20] , hace referencia a cualquier información que se puede utilizar para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que se considera relevante para la interacción entre un usuario y una aplicación, incluyendo el usuario y las propias aplicaciones.

En el marco de la IoT, el contexto es un concepto subjetivo que se define por la entidad que lo percibe. Puede ser descrito generalmente como el subconjunto de estados físicos y conceptuales de interés para una entidad en particular. Se ha identificado cuatro capacidades contextuales genéricas: detección, adaptación, descubrimiento de recursos e incremento.

Según el procesamiento que se realice de la información, el contexto puede clasificarse en dos niveles importantes [19]. Contexto primario, consta de elementos como ubicación, identidad, actividad y tiempo, que son fundamentales para caracterizar una entidad en particular. Contexto secundario, el cual se genera a partir de la fusión de datos y la transferencia del contexto primario por medio de cualquier función. Por ejemplo, la información de ubicación de un sensor GPS corresponde a un dato de contexto primario, mientras que la información resultado de realizar el cálculo de la distancia entre mi ubicación y cualquier punto, corresponde al contexto secundario.

A continuación se presentan las fases del que pueden seguir aplicaciones enfocadas en la construcción de contexto [19]

- **Adquisición.** En esta fase, los datos deben ser adquiridos de diversas fuentes. Algunos factores que deben tenerse en cuenta para el desarrollo de soluciones son:
 - Responsabilidad, la adquisición de los datos puede ser lograda mediante el método *push*, donde el componente software hace consultas periódicas al hardware del sensor; o mediante el método *pull*, donde el sensor entrega los datos al componente software de forma periódica o instantánea, sin necesidad de realizar ningún tipo de consulta.
 - Frecuencia, cuando se genera el contexto se puede realizar en dos tipos de eventos. Eventos instantáneos, los cuales ocurren solo en un instante de tiempo, como abrir o cerrar una puerta o encender o apagar la luz. O eventos de intervalo, en este caso, los eventos que tienen una duración de un periodo de tiempo.
 - Fuente, el contexto puede ser adquirido de tres formas principales. Primero, de forma directa; donde el contexto se adquiere directamente del sensor. Segundo, las aplicaciones pueden recuperar los datos mediante las soluciones middleware. Tercero, el contexto puede ser adquirido en servidores de contexto, tales como las bases de datos o los servidores web.
 - Tipos de sensor, se pueden definir tres tipos de sensores. Primero, físicos, son los más comunes, estos generan los datos por sí mismos. Segundo, virtuales; encargados de recuperar los datos de muchas fuentes publicarlos como datos

- propios. Y por último, se encuentran los lógicos; consisten en una combinación entre sensores físicos y virtuales.
- Proceso de adquisición, existen tres formas de adquirir el contexto. Primero, la detección; los datos se detectan a través de los sensores directamente. Segundo, la derivación; los datos se generan mediante operaciones (simples o complejas) de los datos de los sensores. Y tercero, manualmente; los usuarios proporcionan información de contexto mediante ajustes predefinidos, como preferencias.
 - **Modelado.** Corresponde a la segunda fase, donde se define como se representarán los datos del contexto. No existe estándar para especificar qué tipo de información irá en el modelado del contexto. El modelado basado en ontologías se considera la técnica más popular para modelar y gestionar el contexto, ya que las ontologías ofrecen un lenguaje expresivo para representar las relaciones y el contexto, además existe una gran comunidad que la soporta.
 - **Razonamiento.** Tercera fase del ciclo del contexto. Es un método es utilizado para deducir nuevos conocimientos a partir de los contextos existentes. El razonamiento comprende 3 fases. Primero, pre-procesamiento; limpia los datos de los sensores. Segundo, fusión de datos; método de combinación de datos de los sensores encargada de generar información más precisa. Y tercero, inferencia del contexto; genera información a partir de contextos de bajo nivel.
 - **Técnicas de razonamiento.** Una sola técnica de razonamiento no puede lograr resultados perfectos, es necesario combinar múltiples modelos de forma que aumenten las probabilidades de éxito. Se establecen las siguientes técnicas como las principales para desarrollar el razonamiento del contexto.
 - Aprendizaje supervisado, se construyen ejemplos de entrenamiento, se los etiqueta según los resultados esperados y se deriva una función que puede generar los resultados esperados. Un árbol de decisión es una técnica de aprendizaje supervisado.
 - Aprendizaje no supervisado, esta categoría puede encontrar estructuras ocultas en los datos no etiquetados.
 - Reglas, son estructuras en formato "IF-THEN-ELSE", que permiten generar información de contexto de alto nivel utilizando contexto de bajo nivel.
 - Lógica difusa, permite un razonamiento de valores aproximados. En teoría lógica tradicional, los valores de verdad aceptables son 0 o 1, para lógica difusa se aceptan valores parciales, además permite el uso de lenguaje natural (por ejemplo, temperatura: un poco caliente, bastante frío) permitiendo que escenarios del mundo real sean representados de forma más natural.
 - Basado en ontologías, se basa en descripción de la lógica como una representación formal del conocimiento. El razonamiento ontológico está apoyado por dos idiomas de la web semántica, RDF, OWL, entre otros.

- Lógica probabilística, permite la toma de decisiones sobre la base de las probabilidades de los hechos relacionados con el problema.

2.2.3 Context aware

Un sistema es *context aware* si se utiliza el contexto para proporcionar información y/o servicios de interés para el usuario, donde la relevancia depende de la tarea del usuario. En esta definición se hace un especial énfasis en el usuario, sin limitar la parte consiente (*aware*) solo a la interfaz de la aplicación [19] [21] [22].

También el *context aware* [19] se define como: la capacidad de un programa o dispositivo para detectar diversos estados de su entorno y en sí mismo. Esta característica permite descubrir otros recursos dentro del mismo contexto y explotarlos mientras permanezcan en el mismo contexto. La información brindada por parte del contexto permite: detectar, reaccionar e interactuar con el entorno.

Para la utilización del *context aware* se requiere la disposición de los contextos de otras entidades. La comprensión del contexto permite a los diseñadores de las aplicaciones elegir cuál contexto se va a utilizar.

2.2.4 Protocolos de comunicación

A continuación se realiza una muestra de los protocolos de comunicación WoT más relevantes para el presente caso de estudio. Luego se realiza una comparación entre estos y finalmente se definen los protocolos más adecuados para ser ejercitados.

A. Simple object access protocol – SOAP

SOAP [23] es un protocolo para el intercambio de información estructurada en un entorno descentralizado y distribuido. Utiliza tecnologías XML para definir un marco de mensajería extensible que proporciona una construcción de mensajes que se puedan intercambiar a través de múltiples protocolos de transporte. El marco ha sido diseñado para ser independiente de cualquier modelo de programación particular

La especificación de SOAP establece que los mensajes sean codificados en XML, presentando ventajas como: representar datos de forma independiente a las plataformas o lenguajes que soporten la aplicación, mejorando la interoperabilidad.

El mensaje SOAP se compone de dos elementos, el encabezado y el cuerpo. El encabezado proporciona un mecanismo de extensión y de enrutamiento de los mensajes. El cuerpo es el elemento obligatorio, donde se encuentra la carga útil del mensaje.

B. State representational transfer – REST

Se define como una arquitectura de servicio para desarrollo apoyada totalmente en el estándar HTTP. REST transfiere el estado del servidor al cliente, asignando una representación que es transferida al cliente por una petición, la cual puede variar con diferentes operaciones. Se utilizan los métodos propios de HTTP como GET (transfiere la representación de un recurso del cliente al servidor), POST (cambia el estado de un

recurso), *PUT* (cambia la representación del recurso directamente) y *DELETE* (borra el recurso).

C. *Constrained Application Protocol – CoAP*

CoAP [24] Es un protocolo especializado para el uso de nodos inalámbricos restringidos y limitados de baja potencia, que pueden comunicarse de forma interactiva a través de internet. CoAP surge con la necesidad de suplir un protocolo web genérico que permita la comunicación en redes con necesidades especiales debido al entorno restringido, especialmente: la energía y la comunicación M2M. Este protocolo está diseñado para aplicaciones *machine to machine*. CoAP proporciona un modelo de interacción cliente/servidor similar al de HTTP, esto para facilitar el intercambio de información entre los dos protocolos.

D. *Extensible messaging and presence protocol – XMPP*

XMPP [25] Es un protocolo de la capa de aplicación diseñado originalmente por Jeremie Miller en 1998 para la mensajería instantánea. El protocolo consiste en un sistema de mensajería presencial, el cual usa datagramas en XML para implementar las funcionalidades necesarias. Fue creado con la intención de unificar los servicios de mensajería instantánea, dado que para su época, no existía un protocolo que pudiese satisfacer dicha necesidad, lo que llevaba a situaciones incómodas, como el uso de múltiples clientes (uno por cada servidor/servicio) proceso altamente ineficiente.

E. *MQ telemetry transport – MQTT*

MQTT [26] es un protocolo cliente/servidor diseñado para intercambiar mensajes entre pequeños dispositivos con un reducido ancho de banda, basado en el protocolo de transporte TCP y *Web Services - WS*.

La arquitectura utilizada por MQTT sigue una topología de estrella, con un nodo central que hace de servidor *bróker*, encargado de gestionar la red y transmitir los mensajes entre los extremos. La comunicación puede ser cifrada en el *bróker* para el manejo de seguridad.

Para poder realizar la comunicación es preciso que los clientes realicen una suscripción al servidor creando un canal o tópico, el cual se representa mediante una cadena de caracteres y tiene una estructura jerárquica que separa sus elementos por una barra inclinada (/). Por otro lado, el usuario receptor debe suscribirse a este canal y esperar la publicación de los mensajes.

El protocolo MQTT es idóneo para aplicaciones IoT, en las cuales se envían cantidades pequeñas de información y por tanto no se necesita un gran ancho de banda.

F. Comparación entre los protocolos

A continuación se presenta una tabla comparativa entre los protocolos para realizar el intercambio de información entre objetos. El protocolo SOAP no se encuentra dentro de esta comparación debido a que este será utilizado por el proyecto en el intercambio de información entre distintos módulos, este tema entra con mayor detalle en el capítulo donde se define la arquitectura.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

	CoAP	MQTT	XMPP
Abierto	X	X	X
Transporte	Corre sobre UDP	Basado en TCP	Corre sobre TCP
Mensajería	Cliente servidor	Publish/subscribe	Publish / Subscribe
Negociación de contenido	Si	No	No
Topología	Malla	Estrella	Malla
Seguridad integrada	Medio – opcional	Medio – opcional	Alta - obligatoria
Recursos computacionales	10Ks/RAM Flash	10Ks/RAM Flash	10Ks/RAM Flash
Bajo consumo y pérdidas (1000's)	Excelente	Bueno	Justa

Tabla 3. Tabla comparativa de los protocolos IoT. Fuente: Propia

En la Tabla 3 se puede encontrar las siguientes características comparadas:

- Abierto, hace referencia a la estandarización del protocolo.
- Transporte, protocolo de transporte utilizado.
- Mensajería, forma de comunicación entre los nodos del protocolo.
- Negociación del contenido, posibilidad de realizar la comunicación de forma directa o si es necesario conocer con anterioridad el tipo de dato a recibir.
- Topología, presenta la disposición de la red.
- Seguridad integrada, seguridad de la transmisión de datos a través.
- Recursos computacionales, requisitos mínimos para el funcionamiento del protocolo en un dispositivo.

A partir del análisis de los diferentes protocolos de la Tabla 3, utilizados en la IoT se concluye que para el proyecto es factible utilizar MQTT como el protocolo mediador entre los objetos del escenario. Aunque los protocolos presentaban igualdad de ventajas para la implementación de la interacción entre los objetos del escenario, se ha seleccionado MQTT, debido principalmente a la facilidad y familiaridad del equipo en el desarrollo con este protocolo. A esta conclusión se llegó después de realizar un ejercicio simple de desarrollo, se intentó establecer una conexión simple (utilizando los tres protocolos) entre dos computadores y enviar una cadena de texto y se pudo ver que el protocolo MQTT fue el más fácil de manejar y el que mejor se comportaba para los propósitos del presente proyecto.

2.2.5 Open Group

Open Group es un consorcio global que establece estándares y publica una amplia gama de documentación técnica sobre tecnologías de información. Busca capturar, entender y atender las necesidades actuales y emergentes, además de establecer políticas y mejores prácticas sobre acciones, facilitar la interoperabilidad, desarrollo de consenso, y evolucionar e integrar las especificaciones y tecnologías de código abierto, entre otros.

Open Group ha compartido el estándar “*Open Data Format*” - ODF, el cual pretende representar la información de los objetos de forma estandarizada, por medio de formatos XML, se busca que la información pueda ser entendida e intercambiada de manera

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

universal entre los distintos sistemas que gestionan datos de la IoT, superando los problemas de interoperabilidad sintáctica.

El estándar propuesto por *Open Group* se utilizó en el proyecto para generar los elementos de mensajería, explicados más a profundidad en el capítulo 3. La mensajería que se intercambia entre los objetos del escenario se basa en el estándar ODF, de donde se obtienen los elementos básicos los cuales describirán los atributos y la forma de organización de los documentos que soporten los mensajes. Los elementos utilizados se describen a continuación.

A. Elemento *Object*

El elemento *object* representa un objeto físico o virtual dentro del paradigma del IoT. Esta etiqueta tiene por lo menos un elemento llamado *id* que identifica el objeto. El identificador del objeto debe ser global y único, o al menos único para la aplicación específica, el dominio o la red de las organizaciones involucradas. Los otros elementos son opcionales y su declaración dependerá de la estructura de mensajería de aplicaciones. El elemento *description* da una explicación destinada a los usuarios humanos sobre las características del objeto. La Figura 2 muestra la organización del elemento *object*.

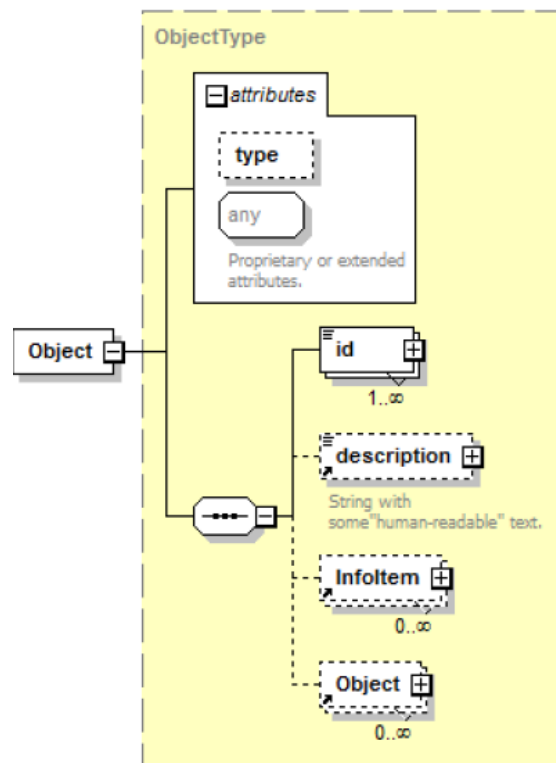


Figura 2 Elemento objeto. Fuente: [27]

B. Elemento *Infoltem*

Representa cualquier atributo que pueda tener el objeto. Dentro de este elemento es posible definir otros elementos como: *name*, utilizado para definir uno o varios nombres que pueda tener el elemento *Infoltem*. *Description*, da una explicación destinada a los usuarios humanos sobre el elemento *Infoltem*. *MetaData*, proporciona información sobre los

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

metadatos del objeto. Los metadatos deben ser referenciados por medio de un *InfoItem*, debido a que el objeto puede tener varios atributos que definan de formas distintas los metadatos. La Figura 3 muestra la organización del elemento *InfoItem*.

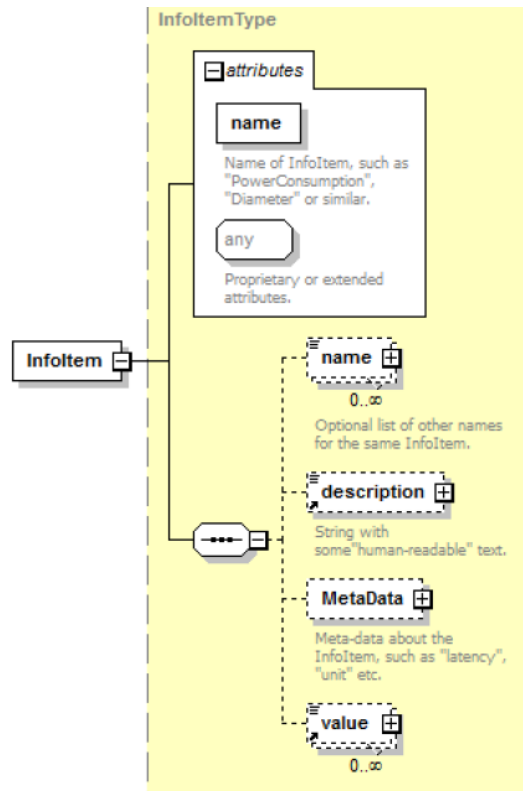


Figura 3 Elemento *InfoItem* Fuente: [27]

C. Elemento *value*

Representa una cantidad, número, carácter, entre otros formatos, que desata el estado de un objeto en un determinado en un tiempo. Por lo general es la etiqueta más dinámica, y soporta distintos tipos de formatos de tiempo.

D. Elemento *omiEnvelope*, *response*, *result* y *return*

Estos tres elementos permiten representar mensajes y requerimientos sobre los estados internos del sistema. Cada etiqueta *omiEnvelope* sólo puede tener uno de los elementos secundarios. Para el presente caso de estudio, esta estructura será utilizada para responder sobre el estado de un trámite dentro del objeto utilizando para ello el elemento *response*. La Figura 4 muestra la organización del elemento *omiEnvelope*.

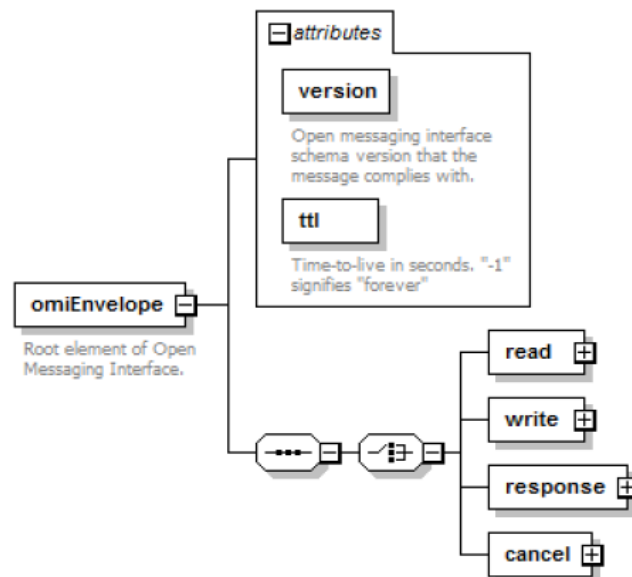


Figura 4. Elemento omiEnvelope. Fuente: [27]

El elemento response debe tener por lo menos un elemento *result*, con el cual representará el mensaje retornado por el sistema, sin embargo puede asociar otros elementos *result*, para responder diferentes peticiones utilizando el mismo mensaje. El elemento *result* contiene al elemento *return* en donde se representará específicamente el código de respuesta al requerimiento (basado en los códigos de respuesta http).

2.3 WEB SEMÁNTICA

La web semántica nace en respuesta a la demanda en nuevos servicios y funcionalidades y al rápido avance y crecimiento de la web. Actualmente en la web existen varias facilidades para que el usuario humano pueda encontrar información, pero no es posible que una máquina, como un agente autónomo, pueda realizar búsquedas similares con el mismo éxito, esto, debido a que la información de la web no está bien estructurada, es decir, no se define formalmente la información. Esta falta de formalidad supone diversos problemas para las herramientas de la web actual [4].

En [28] se ha denominado web semántica a la web donde las herramientas como: aplicaciones y servicios, serán inteligentes. Estas tendrán la capacidad de procesar información de forma más profunda, “entendiendo” el contenido de los documentos y las páginas web, comparando información entre varias fuentes y realizando inferencias y deducciones lógicas para poder recuperar información más cercana a las necesidades del usuario.

2.3.1 Metadatos

La manera que se ha propuesto para codificar los significados de la información que contiene la web, es por medio de etiquetas las cuales especifiquen objetivamente el contenido, dándole a la información una interpretación más correcta. Sin embargo, existe cierta incertidumbre sobre qué etiquetas son relevantes y en qué formatos deben estar

especificadas, debido a que las etiquetas son entregadas por los usuarios humanos. Para esto se ha propuesto reunir y normalizar las etiquetas en un solo documento o recurso llamado metadatos.

Los metadatos corresponden a un conjunto de propiedades y atributos necesarios para etiquetar, catalogar, describir y clasificar información de una fuente o recurso [29] presentan información del contenido, especificaciones formales (tamaño, formato, idioma), derechos de autor, autenticación y finalmente el contexto. Estos se han convertido en una base fuerte sobre la cual se construye la web semántica, haciendo posible que las aplicaciones de la web semántica, puedan obtener el significado de los recursos que recuperan.

Para lograr aplicaciones y servicios que contengan más profundidad semántica, es necesario organizar los valores y propiedades de metadatos en repositorios especiales, estos deben permitir que los significados puedan relacionarse dependiendo del dominio de conocimiento (jerarquías, etc.) posibilitando que las aplicaciones realicen operaciones de razonamiento lógico entre los metadatos.

2.3.2 Servicios y aplicaciones de la web semántica

A continuación se presentan algunas aplicaciones y servicios que pueden ser implementados utilizando herramientas de la web semántica.

- Recuperación de la información, a diferencia de los buscadores actuales, que realizan la búsqueda basándose en palabras clave; los buscadores semánticos realizan la búsqueda basados en el significado de las palabras, entregando al usuario documentos o páginas que se relacionen con los significados de la búsqueda.
- Publicación de la información, los usuarios de la web semántica visualizarán simplemente la información que se relacione con sus necesidades, filtrando la información de los documentos que no sean relevantes.
- Interfaces inteligentes, las herramientas semánticas permiten la creación de aplicaciones con interfaces inteligentes, como las basadas en lenguaje natural.
- Sistemas de inferencia, es posible que se presente información útil al usuario a partir de inferencias y deducciones lógicas.
- Intercambio de información, aumentar la compatibilidad entre sistemas heterogéneos, normalizando la información de diferentes aplicaciones con formatos y organizaciones independientes.

La web semántica permite también la creación de servicios, los cuales pretenden resolver las necesidades de las aplicaciones mediante la integración de diversas soluciones web. En primera instancia, los servicios web semánticos deben ser descubiertos, descritos e invocados por las aplicaciones, esto se logra gracias a la estandarización de protocolos como WSDL, UDDI y SOAP [30]. Por otra parte, estos servicios deben permitir que las aplicaciones puedan entender su alcance y posibilidades, con el fin de que puedan decidir su utilidad para lograr los objetivos de procesamiento e integración automática [31].

2.3.3 Indexación semántica

La indexación es el mecanismo para representar el contenido de un documento mediante un conjunto de términos o conceptos. Un documento es cualquier fuente de información, en este caso digital. Se entiende como concepto a una unidad de pensamiento, el cual se puede expresar como una combinación de otros conceptos.

Para crear un índice, se debe hacer un proceso de representación de las fuentes documentales mediante un conjunto de términos o conceptos, de tal forma que se pueda registrar ordenadamente los temas de los que trata dicho documento con el fin de permitir una clasificación y consulta rápida.

La indexación se hace semántica cuando a las herramientas y métodos de indexación se le suman otras herramientas que permiten aumentar el significado de los conceptos, permitiendo consultas con un mayor grado de relevancia para los usuarios. Este tipo de indexación se enfoca en la asociación de los conceptos con el fin de buscar mayor precisión en los significados.

Según [32] las características de un índice semántico son:

- a. Multidimensionalidad, los elementos de indexación son valores de atributos que pueden estar basados en complejas descripciones.
- b. Adaptabilidad, el índice es altamente adaptable a las necesidades de cada proyecto. Los conceptos de indexación pueden ser añadidos o eliminados como se desee, lo cual los hace muy densos y precisos con respecto al interés de un grupo de personas.
- c. Disponibilidad, dado que el índice es en realidad un conjunto de descripciones parciales de los objetos indexados, mucha información se puede extraer directamente del índice.

Para realizar un índice semántico es preciso utilizar un tesoro o una ontología, para saber cuál se debe escoger es preciso tener en cuenta que un tesoro agrupa un conjunto de palabras, lo que nos sirve para trabajar con un idioma en particular; mientras que la ontología presenta un nivel semántico más alto, describiendo mejor el objeto, sus prioridades y relaciones. Si se desea, se puede utilizar las dos herramientas para adaptar mejor los dominios.

2.3.4 Ontologías

La definición más aceptada sobre las ontologías fue dada por Thomas Gruber, y reza: “Una Ontología es una especificación formal y explícita de una conceptualización compartida” [33] donde el término “especificación formal” sugiere que los elementos deben estar expresados siguiendo formalidades idénticas, permitiendo que la ontología pueda ser reutilizada y leída por cualquier máquina, cualquier plataforma o lenguaje. El término “especificación explícita” refiere que todos los elementos no deben dar nada por supuesto o por obvio. La palabra “conceptualización” aclara que las ontologías desarrollan modelos abstractos del mundo real. La palabra “compartida” es el elemento de la afirmación más importante, ya que una ontología, lo es, cuando sus conceptualizaciones sobre el mundo real y su formalización del conocimiento, es aceptado y acogido por todos los usuarios y posibles usuarios de dicha ontología.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Básicamente las ontologías funcionan como cualquier modelo conceptual existente [34], sin embargo estas se diferencian de otros modelos en tres aspectos principales. Primero, las ontologías buscan la integración de información entre aplicaciones. Segundo, permiten que los significados sean descritos sin ambigüedades. Y tercero, permiten realizar razonamiento utilizando cálculos en tiempos de ejecución.

Algunos de los lenguajes mayormente utilizados para las ontologías son: XOL [35], RDF [36], OIL [37], DAML [38] y OWL [39], los cuales están basados en el lenguaje XML para el intercambio y estandarización de la información.

2.4 MISWoT

El trabajo [6] aporta como uno de sus productos un modelo de interacción semántica en la WoT, este modelo identifica los elementos básicos que se deben tener en cuenta al momento de realizar interacción entre los objetos de la WoT. En la Figura 5 se pueden apreciar dichos elementos.

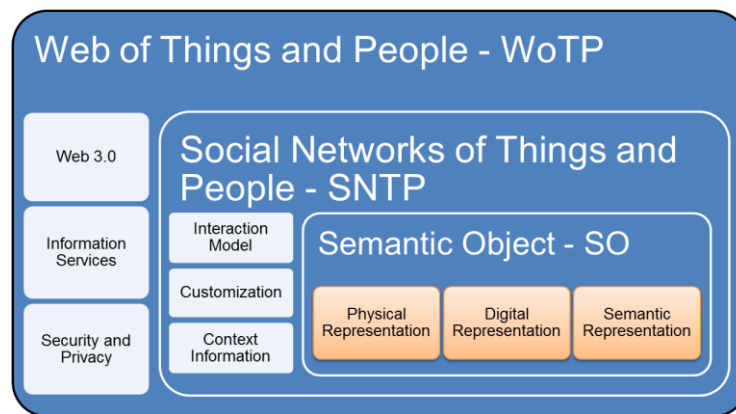


Figura 5. Marco conceptual del MISWoT. Fuente: [6]

El modelo define tres capas importantes: capa objeto semántico “*Semantic Object – SO*”, capa redes sociales de objetos y personas “*Social Networks of Things and People - SNTP*” y capa web de las cosas y las personas “*Web of Things and People - WoTP*”

La WoTP se ha desarrollado para permitir a las personas comunicarse con los objetos a través de las tecnologías de la Web 2.0 y se están haciendo esfuerzos para incluir las tecnologías de la Web 3.0, estas tecnologías se pueden aprovechar para desarrollar aún más la WoT estableciendo un entorno de trabajo parecido al de la web de las redes sociales de personas pero de forma híbrida, es decir, que los objetos puedan utilizar herramientas y mecanismos de la web para comunicarse entre objetos y comunicarse entre personas y objetos. Adicionalmente esta capa debe definir correctamente los servicios de información y características de seguridad y privacidad, sin embargo estos aspectos están por fuera del MISWoT debido al alcance de los objetivos.

La SNTP propone reutilizar la infraestructura de redes sociales ya instaladas con el fin de hacer transparente la aparición de los objetos como entes autónomos que interactúan entre ellos y con las personas. La capacidad de interacción y la autonomía de estos objetos

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

estarán limitadas a la posibilidad de incorporar tecnologías de inteligencia artificial y agentes inteligentes a los objetos. En este entorno híbrido es muy importante la personalización (*customization*), la cual se maneja con perfiles de usuario, estos perfiles se definen tanto a personas como a objetos, sin olvidar que estos objetos deben caracterizarse con respecto a sus capacidades de inteligencia. También, se define un modelo de interacción (*interaction model*) que establezca los mecanismos de interoperabilidad y las capacidades de la red social para lograr sus objetivos, así como la información del contexto en el que pretenda crearse un servicio que usen los objetos que rodean a las personas.

El SO aparece como un integrador de los diferentes miembros de las redes sociales de objetos y personas en un solo elemento, así una persona y un objeto se pueden representar a través de este solo concepto, el cual integra tres partes distintas pero complementarias: representación física, representación digital y representación semántica.

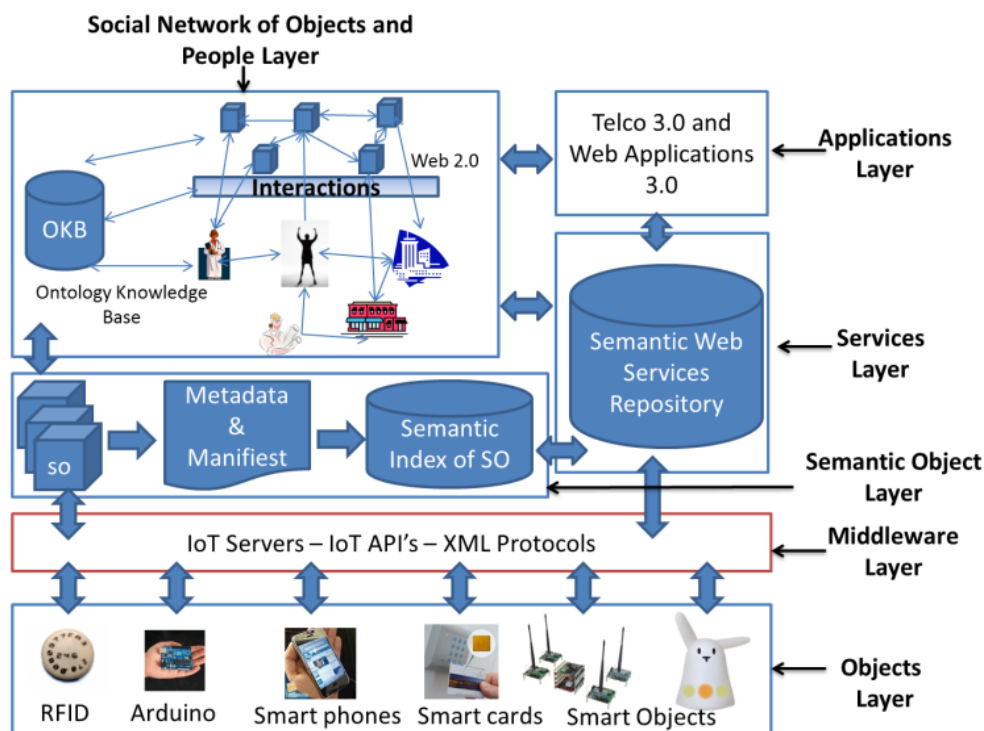


Figura 6. Arquitectura del MISWoT. Fuente: [6]

El trabajo también propone una arquitectura a implementar se puede ver en Figura 6, en la cual se hace uso de *middlewares* y servidores IoT (*middleware layer*), como mediadores para resolver el problema de heterogeneidad de hardware y protocolos existentes en la capa de objetos (*object layer*). Estos servidores permiten acceder a los metadatos y datos en formatos estandarizados, por ejemplo: JSON por medio de APIs REST, que se despliegan en las aplicaciones web.

La capa semántica de objetos (*semantic object layer*) utiliza los servidores IoT para crear representaciones semánticas de los objetos, con una estructura basada en una ontología llamada ontología de objeto semántico. Al mismo tiempo, se construye un índice semántico

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

de los objetos explorados, basándose en ontologías de dominio específico para crear un contexto por ontología particular, clasificando el uso de los diferentes sensores en éstos contextos. Los índices semánticos se convierten en un repositorio de índices semánticos a los cuales se puede acceder a través de servicios web semánticos (*services layer*) disponibles para las aplicaciones y para la red social de objetos y personas.

Finalmente, en la capa de red social de objetos y personas (*social network of object and people layer*) se implementan los conceptos descritos en el marco conceptual, tomando como base la ontología de objeto semántico instanciada en cada objeto y el repositorio de índices semánticos que contextualiza el uso de dichos objetos en una red de interacciones, creando nuevos servicios a los usuarios, esta red de interacciones se puede soportar en un repositorio de conocimiento para establecer comportamientos de los objetos y servicios compartidos en dicha red.

Para el objeto semántico se hace una propuesta de un conjunto de conceptos básicos modelados en una ontología que es instanciada para cada objeto.

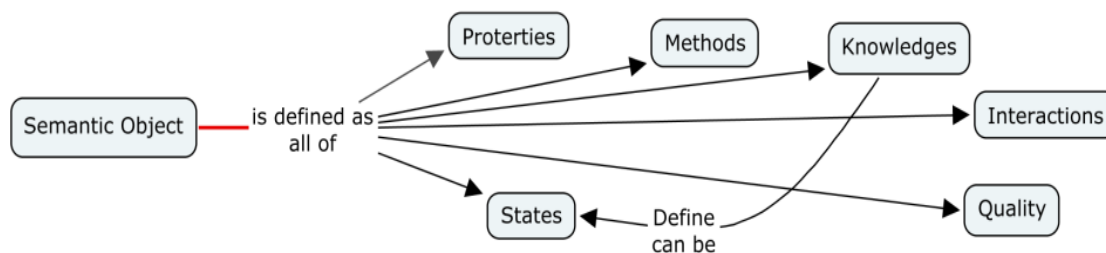


Figura 7. Módulo de las propiedades del SO. Fuente: [6].

En la Figura 7 se presenta sólo el nivel superior del SO, con cada uno de los conceptos clave, los cuales despliegan más conceptos contenidos en la propuesta y definidos así:

- Propiedades, corresponden a los atributos que describen al objeto o dispositivo de la IoT. Principalmente se almacenan metadatos que se obtienen de los servidores IoT.
- Métodos, definen unas funciones que permiten enviar datos y recibir datos a otros SO, los datos son de propiedades, estados y comandos.
- Conocimientos, se maneja información del contexto, perfil de usuario y servicios disponibles, los cuales pueden enlazar otras ontologías o una red de conceptos elaborados al respecto.
- Interacciones, define las relaciones establecidas con otros SO (interacciones estructurales) a través de una estructura de red. También define interacciones dinámicas en las cuales se modelan acciones sobre un entorno tomando como base la sentencia de Evento – Condición – Acción (ECA).
- Calidad, establece objetivos e indicadores que permiten monitorear el estado del servicio, del mismo sensor y otros indicadores, que el usuario considere relevante monitorear.
- Estados, permite gestionar los valores de las propiedades medidas por el objeto IoT en un tiempo determinado.

2.5 TRABAJOS RELACIONADOS

Los proyectos [40] [41] presentan escenarios que manejan objetos con un servidor middleware para determinar acciones a seguir de acuerdo a los datos que los objetos arrojen en un contexto específico, como aplicaciones en educación y la atención de pacientes por personal de enfermería. Los proyectos concluyen que al utilizar herramientas del IoT, se mejoran y facilitan distintas actividades humanas; sin embargo, estos proyectos no utilizan objetos totalmente inteligentes ni tampoco herramientas semánticas, relegando muchas configuraciones del sistema al usuario final, a otros actores humanos u aplicaciones terceras, adicionalmente éstos proyectos se enfocan primordialmente en la interacción de objeto – persona y no entre objetos, como lo plantea el presente proyecto.

Los proyectos [21] [22] presentan herramientas y sus respectivos modelos que manejan el contexto, realizando el descubrimiento y la configuración de objetos dentro de un entorno, reduciendo así las tareas que el usuario debe realizar para conectarse con los servicios que proveen los objetos. El aporte se puede definir en el descubrimiento y configuración automática de los objetos que se conectaban a una red. El proyecto no utiliza técnicas semánticas y se enfoca principalmente en manejo de protocolos. Como el proyecto no aborda la interacción entre objetos, las aplicaciones deben definir su propio mecanismo de trabajo.

El proyecto [42] estipula que para crear un entorno inteligente de interacciones semánticas de objetos inteligentes, se debe tener en cuenta tres mecanismos básicos: el descubrimiento de servicios, mensajería común para todos los participantes y notificación de eventos. Para esto, utilizan ontologías para modelar el razonamiento de la información compartida entre los objetos pertenecientes al escenario propuesto, “jardines inteligentes”. Se logra concluir que las tecnologías semánticas pueden ser aplicadas en el dominio del IoT para generar interacciones más ricas. Este proyecto se enfoca en el mismo tema del modelo MISWoT que se pretende implementar, con la diferencia que el modelo MISWoT realiza indexación semántica para el descubrimiento y contextualización de los servicios, además de la ejecución de acciones en la interacción. El artículo no presenta en detalle el experimento realizado sobre las interacciones en el escenario seleccionado, sin embargo si presenta una interacción entre el usuario y los objetos inteligentes a través de una interfaz que usa un dispositivo móvil que está filmando en el momento.

En el proyecto [43], se plantea la vigilancia a distancia sobre estados de objetos provistos de sensores. Para ello se implementó una arquitectura que propone la descentralización de la inteligencia, la cual debe estar repartida en la capa física o de sensores, la capa de coordinación y la capa de supervisión. Para nuestro proyecto la inteligencia del objeto se encuentra descentralizada, esto facilita el desarrollo del objeto y la utilización de diversos objetos con distintos propósitos sin tener que depender de las configuraciones y capacidades del sistema.

El trabajo [44] plantea una interfaz estándar para controlar un sistema que integra aplicaciones con dispositivos *context aware*. HomeOS tiene como objetivo principal

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

proporcionar un sistema de comunicación estándar entre todos los dispositivos inteligentes en un hogar, ser un controlador de eventos y permitir generar nuevas aplicaciones y servicios para administrar las funciones de los dispositivos del hogar. Los principales componentes del sistema se dividen en tres capas: capa de dispositivo, que define las características de los dispositivos y sus interfaces; la capa de núcleo, que maneja el tráfico de información entre los dispositivos y las aplicaciones finales; y la capa de aplicación, donde se encuentran las aplicaciones realizadas por los usuarios. HomoOS representa el trabajo más cercano al propuesto en esta monografía, sin embargo HomeOS realiza el almacena los dispositivos en un servidor centralizado lo que restringe el contexto a un montaje específico y a medida, por otra parte HomeOS parte de aplicaciones con parámetros predefinidos (Ejemplo, apagar temperatura, prender luz) limitando las posibilidades que se puedan generar entre la interacción de los dispositivos.

También existen soluciones comerciales como [45] [46] [47] estas proveen servicios de domótica permitiéndole al usuario el control a distancia de los dispositivos del hogar mediante aplicaciones móviles o web. Cuentan con dispositivos del hogar como: equipos de audio y video, iluminación, alarmas, cámaras, cortinas, sistemas de riego, entre otros, sin embargo estos dispositivos están restringidos a marcas específicas acotando la solución solo a un tipo de dispositivos.

3 ARQUITECTURA PARA EL ESCENARIO DE INTERACCIÓN SEMÁNTICA EN LA WoT

La arquitectura presentada toma como base de la arquitectura propuesta por el proyecto de Niño-Zambrano [6]. La presente representa de forma más específica los elementos y herramientas utilizados en el escenario de interacción semántica desarrollado. Por lo tanto la arquitectura presenta capas bien definidas, es orientada a servicios y guiada mediante un modelo semántico de interacción en la WoT.

La arquitectura está basada en el uso de objeto inteligentes debido a que:

- Distribuyen el procesamiento de la información en los bordes de la red (“*fog computing*” [48]) reduciendo requerimientos en la comunicación y el tiempo de respuesta.
- Reutilizables en diversos contextos y servicios, dado que en definición no están acoplados a una entidad única, entorno o propósito particular y sus relaciones dependerán de los objetivos del usuario en un momento dado.
- Permiten crear redes de objetos con el fin de cooperar en la creación de nuevos servicios, participando en objetivos diferentes para lo que originalmente fueron creados.

3.1 CONCEPTOS ESPECÍFICOS.

Se definen los conceptos sobre los que se fundamenta la propuesta arquitectónica

- **Mundo físico.** Esta investigación comparte el concepto dado por [49] en la que definen el mundo real como “... *el entorno físico que es instrumentado a través de máquinas identificadas a través de etiquetas, sensores, actuadores y elementos de procesamiento organizados en islas de dominios específicos, con el fin de monitorear e interactuar con las entidades físicas en las que estamos interesados*”. A los elementos del mundo físico se les conoce como representaciones físicas.
- **Mundo digital.** Se refiere a la información que se almacena digitalmente acerca del mundo físico mediado por tecnologías de la IoT. A la información sobre un elemento en el mundo digital se le conoce como representación digital.
- **Recursos.** Representaciones digitales de los instrumentos que poseen los objetos para interactuar con el mundo físico. Se relacionan a entidades del mundo físico con el fin de capturar información, procesarla y comunicarla, además tienen la capacidad de cambiar algún estado de dicha entidad. Un recurso puede ser:
 - Sensor, examina el entorno del mundo físico y capta la información que entra al sistema.
 - Controlador, procesa la información extraída del mundo real y gestiona los recursos del objeto.
 - Actuador, efectúa una acción sobre el mundo físico, con el fin de cambiar el estado de la entidad.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- **Entidades.** Son representaciones de personas, lugares y cosas del mundo real. Una entidad necesariamente está relacionada con uno o más objetos.
- **Entidades de Interés.** Entidades que son objeto de búsqueda de información o interacción por parte de los usuarios.
- **Propiedades.** Corresponde las características de las entidades que pueden describir el estado en un momento dado.
- **Propiedades de interés.** Propiedades que son objeto de búsqueda de información o interacción por parte de los usuarios.
- **Objeto semántico.** Un objeto semántico es en base un objeto inteligente que tiene una representación digital (identificador único y mecanismo para recuperarlo), está asociado a una entidad de interés específica y utiliza la ontología de objeto semántico – OOS para almacenar el conocimiento de su infraestructura (metadatos).
- **Servicio básico.** Son los servicios expuestos al usuario que presentan los objetos por defecto, están asociados al funcionamiento original con el cual el objeto fue creado. El objeto debe tener por lo menos un servicio básico asociado. El estado (prendido o apagado) de un servicio básico está referenciado en los metadatos del objeto.
- **Servicio de interacción.** Es generado por la interacción entre dos objetos. Está asociado a una entidad de interés, a la propiedad que desencadena una acción y a la propiedad que se desea modificar. Cada servicio de interacción está asociado a un ECA y es ejecutado por el objeto que contiene el recurso actuador que puede modificar la propiedad deseada.
- **Usuarios:** Personas o aplicaciones que interactúan con los objetos que están definidos en una entidad de interés específica. Los usuarios tienen una experiencia natural y transparente con el escenario.
- **Escenario de interacción semántica:** Entorno físico mediado con herramientas semánticas donde los objetos ofrecen sus recursos, funcionalidades e interactúan de forma dinámica con otros objetos y los usuarios. En este espacio se pueden identificar las entidades de interés, de las cuales se quiere conocer su estado midiendo las propiedades de interés. La información sobre cada entidad está almacenada en una etiqueta NFC, la cual contiene el nombre de la entidad y su localización. También se puede encontrar etiquetas NFC que están asociadas con la identificación del objeto, los cuales permiten al usuario obtener información sobre un objeto en particular sin asociarlo a una entidad.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- **Coordinador.** Aplicación utilizada por el usuario como interfaz para comunicarse con el escenario. Recoge la información de los distintos elementos del escenario por medio de servicios web especializados en IoT, y también por medio de etiquetas NFC dispersas en el escenario.
- **Archivos mensajería.** Utilizados para soportar el intercambio de información que se realiza en el servidor *bróker* MQTT. Estos archivos son documentos XML basados en el estándar ODF propuesto por *Open Group*.
 - SimpleValue.xml, entrega el valor instantáneo de una propiedad de interés captado por un recurso perteneciente a un objeto.
 - SimpleResponse.xml, entrega un código de respuesta (200 o 404) después que el objeto termina una operación interna. Tiene el fin de informar sobre el estado de una petición realizada por el usuario.
 - SetECASState.xml, soporta la solicitud que realiza el usuario a un objeto para el cambio del estado (prendido o apagado) de un servicio de interacción.
 - SetBasicState.xml, soporta la solicitud que realiza el usuario a un objeto para el cambio del estado (prendido o apagado) de un servicio básico.
 - MetadataQuery.xml, soporta la consulta realizada por el usuario sobre los metadatos de un objeto.
 - Metadata.xml, contiene toda la información sobre los metadatos de un objeto. Utilizado para intercambiar la información básica de un objeto con otro objeto o con el usuario.
 - ECADelete.xml, Soporta la solicitud que realiza el usuario a un objeto para eliminar un servicio de interacción.
 - ECA.xml, Contiene toda la estructura sobre el servicio de interacción. Utilizado para crear la sentencia ECA en cada objeto participante en el servicio de interacción.

La Figura 8 muestra el contenido del archivo Metadata.xml, el cual corresponde a la información de los metadatos de un objeto, este archivo se divide en dos partes importantes, la primera señalada con “<Infoltem name='id_objeto'>” que contiene la información general del objeto, como nombre, estado, longitud, latitud y comentario; y la segunda señalada con “<Infoltem name='id_recurso'>” la cual contiene la información particular de cada recurso perteneciente a dicho objeto, esta se repite según los recursos que contenga el objeto. La figura representa una generalidad del archivo, los elementos en blanco son variables que contienen la información a comunicar, en contraste los otros elementos son estáticos. Los otros archivos de mensajería pueden ser consultados con más detalle en el anexo C.

```
<?xml version="1.0" encoding="UTF-8"?>
<Objects>
  <Object>
    <id>id_objeto</id>
    <InfoItem name=id_objeto>
      <MetaData>
        <InfoItem name="object_name">
          <value type="string">object_name</value>
        </InfoItem>
        <InfoItem name="service_state">
          <value type="string">service_state</value>
        </InfoItem>
        <InfoItem name="longitud">
          <value type="string">longitud</value>
        </InfoItem>
        <InfoItem name="latitud">
          <value type="string">latitud</value>
        </InfoItem>
        <InfoItem name="description">
          <value type="string">comentario</value>
        </InfoItem>
      </MetaData>
    </InfoItem>
    <InfoItem name=id_recurso>
      <MetaData>
        <InfoItem name="resource_name">
          <value type="string">resource_name</value>
        </InfoItem>
        <InfoItem name="format">
          <value type="string">format</value>
        </InfoItem>
        <InfoItem name="unit">
          <value type="string">unidad</value>
        </InfoItem>
        <InfoItem name="resource_type">
          <value type="string">resource_type</value>
        </InfoItem>
      </MetaData>
    </InfoItem>
  </Object>
</Objects>
```

Figura 8. MetaData.xml. Fuente: Propia.

- **ECA (Evento - Condición - Acción).** Define la interacción entre dos objetos en una misma entidad de interés permitiendo generar nuevos servicios (servicios de interacción). Intercambia información sobre las facultades y atributos de cada objeto, como información sobre sus recursos y tipos de datos que manejan. Se compone de tres partes
 - Evento, propiedad de interés captada por un recurso. El objeto asociado al recurso seleccionado para el evento es denominado objeto evento.
 - Condición, sentencia condicional (mayor, menor o igual) que compara el estado del recurso seleccionado en el evento (propiedad de interés) con un valor ingresado por el usuario. Ambas variables se sincronizan acorde a los metadatos recuperados del objeto evento.

- Acción, sentencia condicional (mayor, menor o igual) que compara el estado de un recurso de tipo actuador con un valor ingresado por el usuario. El objeto asociado al recurso seleccionado es denominado objeto acción. Ambas variables se sincronizan acorde a los metadatos recuperados del objeto acción.

El ECA tiene asociado un nombre, un estado (prendido o apagado) y una entidad de interés, así los objetos pueden tener varios ECA's que ejecuten un mismo recurso, sin que esto signifique un problema de conflicto. Sin embargo para la fase de experimentación existen algunos casos en los que se presenta conflicto. Queda a disposición de la coherencia del usuario el proponer sentencias que desaten un conflicto.

3.2 VISTA ESTÁTICA

A continuación se presenta Figura 9 que distingue las distintas capas de la arquitectura propuesta para el diseño del escenario. La arquitectura está basada en el MISWoT resolviendo la capa de interacción y detallando las herramientas utilizadas en las distintas capas.

3.2.1 Capa de objetos

Se encuentra la representación física de los objetos y se definen los recursos asociados. Esta capa se conecta con la capa *middleware*, donde los objetos actualizan sus datos periódicamente. Cada objeto contiene asociada una etiqueta NFC la cual contiene la identificación del objeto en el sistema. Se diferencian 3 objetos.

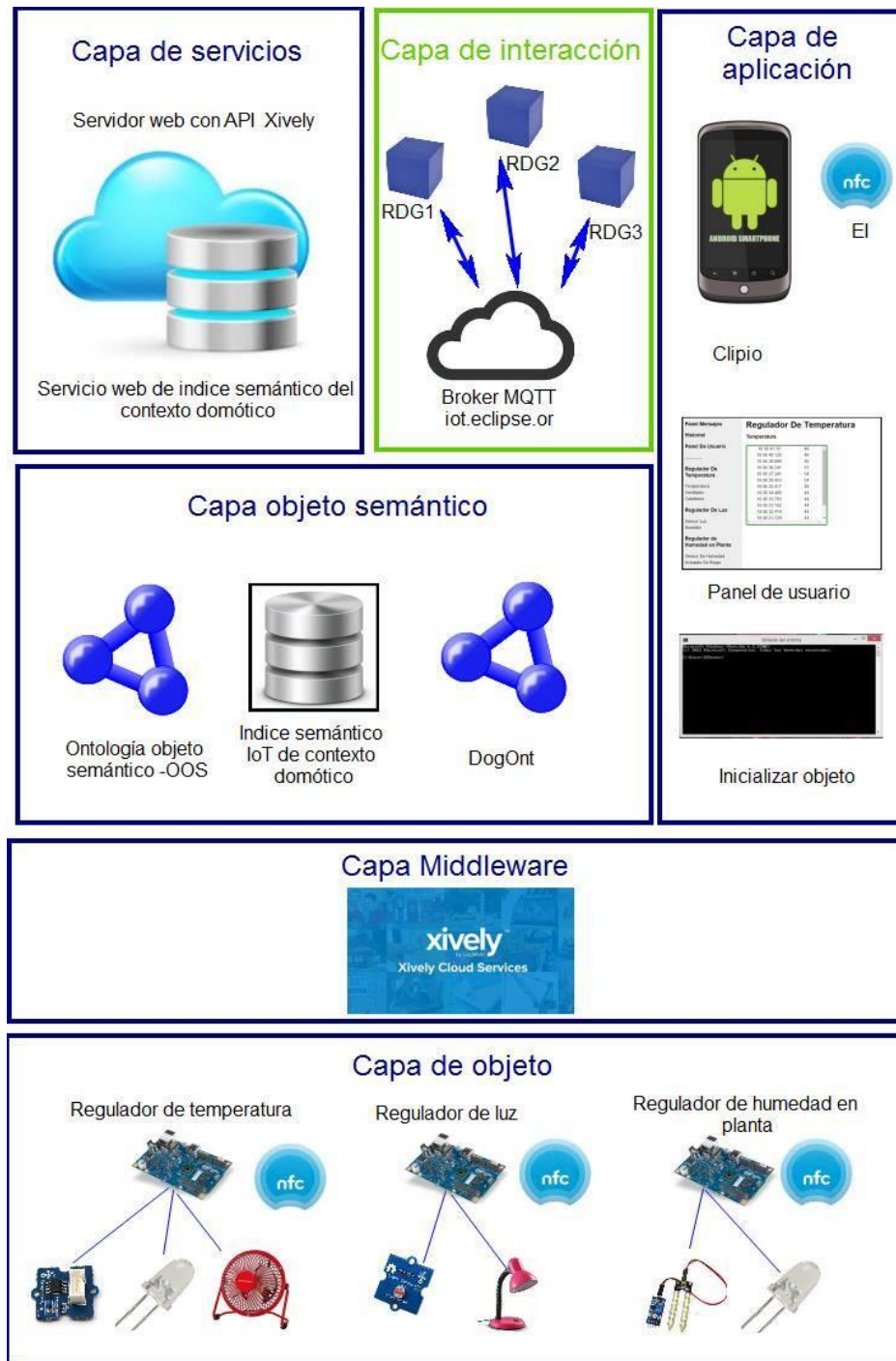
- Regulador de temperatura, está asociado con el recurso sensor de temperatura, actuador ventilador y actuador calefactor.
- Regulador de luz, está asociado con el recurso sensor de luz, y actuador bombillo
- Regulador de humedad en planta, está asociado con el recurso sensor de humedad y actuador de riego.

En el capítulo 4 serán detallados más a fondo cada uno de los objetos antes mencionados.

3.2.2 Capa *middleware*

La arquitectura delega el problema de heterogeneidad y conectividad a esta capa, por lo tanto el servidor debe representar digitalmente el objeto de forma parcial (no todos los metadatos necesarios para la representación digital del objeto podrán ser recuperados del servidor *middleware*) y asignarle una identificación única, además debe permitir el acceso a los datos y metadatos del objeto utilizando esquemas estandarizados como XML o JSON y también debe exponer API's para el desarrollo de aplicaciones tipo cliente. Finalmente la capa *middleware* almacenará los conceptos para representar semánticamente al objeto y sus recursos asociados.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS



RDG# = Representación digital Galileo #
 Ei: Entidad de interés.

Figura 9 Vista estática de la arquitectura. Fuente: propia

Esta capa será consultada por la capa de servicios para recuperar la información de los objetos como los datos metadatos y conceptos. El servidor *middleware* definido en esta capa es *Xively* [50], donde se definen los objetos mediante el elemento *Feed* y los recursos mediante canales. *Xively* ofrece una interfaz donde se pueden consultar directamente los

datos en tiempo real, de cada uno de los canales creados y también se puede gestionar los metadatos, además permite crear etiquetas relacionadas a cada objeto y a cada recurso para anotar los conceptos que permiten contextualizar al objeto.

Xively se ha especificado sobre las recomendaciones del MISWoT, ya que es de acceso gratuito y existe una gran comunidad que soporta el desarrollo en distintas API's, por otra parte existe un previo desarrollo por el grupo de trabajo en base a la plataforma *Xively*.

3.2.3 Capa de objeto semántico

Encargada de alojar el motor semántico. Se diferencian 3 elementos:

- Ontología objeto semántico (OOS), elemento que aloja, por medio de una ontología, los metadatos del objeto. Este elemento recupera los metadatos del objeto que se encuentran en la capa de servicio y luego expone estos metadatos para que el objeto pueda consultarlo y construir el archivo *MetaData.xml*. El objeto consulta a OOS utilizando sentencias SPARQL, estas sentencias y la ontología pueden ser consultados en el anexo H.
- *DogOnt* [51], ontología de dominio utilizada por el índice para crear el contexto a partir de los conceptos y relaciones. Esta ontología contiene los conceptos referentes a un sistema domótico, y en general, conceptualiza y recoge los elementos pertenecientes a una casa o vivienda. Soporta los elementos de una casa inteligente actualmente disponibles y también posibles futuros, clasificándolos según sus características de controlabilidad, ubicación, funcionalidad y vistas arquitectónicas. Se puede encontrar información más detallada en el anexo B.
- Índice semántico (domótica IoT), elemento que contextualiza los objetos según los conceptos que tengan asociados. Para esto el índice realiza un barrido de todos los conceptos y sus relaciones de la ontología *DogOnt* y los compara con la información recuperada sobre cada objeto del servidor *middleware*, genera listas en niveles de relevancia según los conceptos y expone el resultado en la capa de servicio mediante métodos bien definidos. Utilizando la ontología de dominio, el índice semántico soporta el contexto de casa inteligente entregando listas de objetos según conceptos que puedan caber en este contexto.

3.2.4 Capa de servicio

Encargado de almacenar la información del índice semántico, encapsularla y exponerla a través de servicios web, haciendo posible que las aplicaciones y los objetos puedan consumir los metadatos, datos y demás información de contexto de manera interoperable y transparente. Expone una serie de métodos enfocados en la recuperación de información en la IoT basados en SOAP, se define los siguientes grupos de métodos estandarizados para su consumo:

- Servicios básicos del índice, métodos basados en una consulta en lenguaje natural los cuales retornan información ordenada por relevancia sobre objetos que pueden satisfacer las necesidades de información detectadas en la consulta.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- Los servicios de geolocalización, retornan información sobre localización de los recursos IoT basada en la API Geonemes.
- Servicios de indexación particular, el servidor expone métodos para realizar la indexación de un objeto en particular ingresando su identificación.

La capa de servicio recupera información sobre los objetos en el servidor Xively, luego la información del objeto es utilizada por el índice semántico generando listas de objetos contextualizados, los métodos expuestos por el servidor son utilizados por el objeto semántico para recuperar sus metadatos y son utilizados por la capa de aplicación para recuperar objetos según un concepto determinado.

3.2.5 Capa de interacción

Soporta las herramientas que permiten comunicar a los objetos con las aplicaciones de los usuarios y con otros objetos, esta capa está basada en un servidor *broker* con protocolo MQTT, por el cual los objetos y los usuarios pueden comunicar sus estados, requerimientos y consultas. Todas las tareas de mensajería de los objetos y de los elementos que interactúan con el *bróker*, están basadas en el estándar ODF que se basa en documentos XML. Las interacciones realizadas sobre esta capa son:

- Objeto a objeto. El objeto evento comparte el estado de uno de sus recursos al objeto acción por medio del archivo SimpleValue.xml.
- Objeto a usuario. El objeto responde al usuario sobre el resultado de una operación interna por medio del archivo SimpleResponse.xml. El objeto responde el requerimiento de metadatos del usuario a través del archivo MetaData.xml.
- Usuario a objeto. El usuario realiza una consulta sobre los metadatos del objeto por medio del archivo MetaDataQuery.xml. El usuario comunica la estructura de un nuevo servicio de interacción al objeto evento y acción por medio del archivo ECA.xml. El usuario solicita el cambio de estado de un servicio básico por medio del archivo SetBasicState.xml. El usuario solicita el cambio de estado de un servicio de interacción por medio del archivo SetECASState.xml. El usuario solicita la eliminación de un servicio de interacción al objeto evento y acción por medio del archivo ECADelete.xml.

El protocolo MQTT se ha especificado sobre las recomendaciones del MISWoT, principalmente por su facilidad al momento del desarrollo y su gran aceptación dentro de la comunidad IoT, además es un protocolo que consume pocos recursos de ancho de banda y permite su desarrollo sobre varios lenguajes de programación.

3.2.6 Capa de aplicaciones

Permite al sistema integrar a los usuarios con los objetos. Esta capa contiene dos elementos con los cuales el usuario interactúa y sirven como una ventana para administrar los objetos.

- Clipio, es una aplicación móvil la cual sirve como interfaz al usuario para administrar los objetos del escenario. Clipio se conecta a la capa de servicio, consulta la información que el índice semántico expone sobre los objetos y presenta una lista de los objetos basándose en la relevancia conceptual y posteriormente en la distancia física que existe

entre el objeto y la entidad de interés y la consulta en lenguaje natural realizada por el usuario (esta consulta incluye el nombre de la entidad de interés). Clipio es capaz de administrar los objetos comunicando sus solicitudes mediante el servidor MQTT, de esta forma puede crear y eliminar servicios de interacción, modificar los estados (prendido y apagado) de los servicios básicos y de interacción y consultar metadatos sobre los objetos.

- Etiqueta NFC correspondiente a una entidad. Esta etiqueta contiene el nombre de la entidad con la cual está relacionada, también contiene su posición global determinada en longitud y latitud. Esta etiqueta es consultada por Clipio, el cual extrae la información de la entidad cuando pasa sobre la etiqueta.
- Panel de usuario. Se utiliza como interfaz al usuario para presentar información sobre el escenario. Presenta los estados instantáneos de cada recurso de los objetos, así como información general sobre el funcionamiento de cada objeto. Además, presenta la información en tiempo real correspondiente a las actividades de interacción entre los objetos. Toda esta información es almacenada en un modelo de base de datos relacional, y es posible consultar la actividad histórica de interacción del escenario, filtrándola por fecha y por tipo de interacción. La información almacenada en el modelo de esta aplicación es utilizada previamente para realizar la evaluación del escenario.
- Inicializar objeto. Utilizado como interfaz al usuario para sincronizar la representación física con la representación digital del objeto, y complementar la representación digital parcial del objeto. Para esto el objeto requiere al usuario ingresar la identificación del objeto, luego el objeto recupera la información utilizando los métodos del índice semántico expuestos en la capa de servicio, y finalmente requiere al usuario los metadatos que no se han recuperado guardando esta información en el objeto. Esta interfaz solo interactúa una sola vez por cada objeto con el usuario y es cuando el objeto se está dando de alta en el sistema.

3.3 VISTA DINÁMICA

A continuación se presenta la vista dinámica de la arquitectura, esta se divide en 4 partes diferenciadas por la interacción que realizan con el usuario. Estas partes se definen en 4 graficas diferentes y la explicación de los procesos que sostienen.

El proceso inicia con la incorporación del *middleware Xively*, el cual provee un interfaz que permite generar una representación digital de los objetos y sus recursos. *Xively* establece una identificación única al objeto mediante el elemento *FeedId* y permite identificar los recursos del objeto mediante canales, creando así un perfil a cada objeto. La interfaz de *Xively* permite anotar los objetos y sus recursos utilizando etiquetas, las cuales contienen los conceptos que relacionan al objeto con los contextos. El índice semántico consulta las referencias de los objetos en documentos JSON suministrados por *Xively* a través de la capa de servicio, e incorpora el resultado de la indexación en una interfaz con métodos bien definidos, exponiendo estos métodos en la capa de servicio. De forma paralela, el índice semántico se encarga de tomar la ontología de dominio (DogOnt) para hacer un barrido de conceptos y a partir de ellos organizar los objetos en listas de relevancia según los

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

conceptos por los cuales sea consultado el índice semántico. En la Figura 1 se muestra una representación gráfica de los procesos antes mencionados.

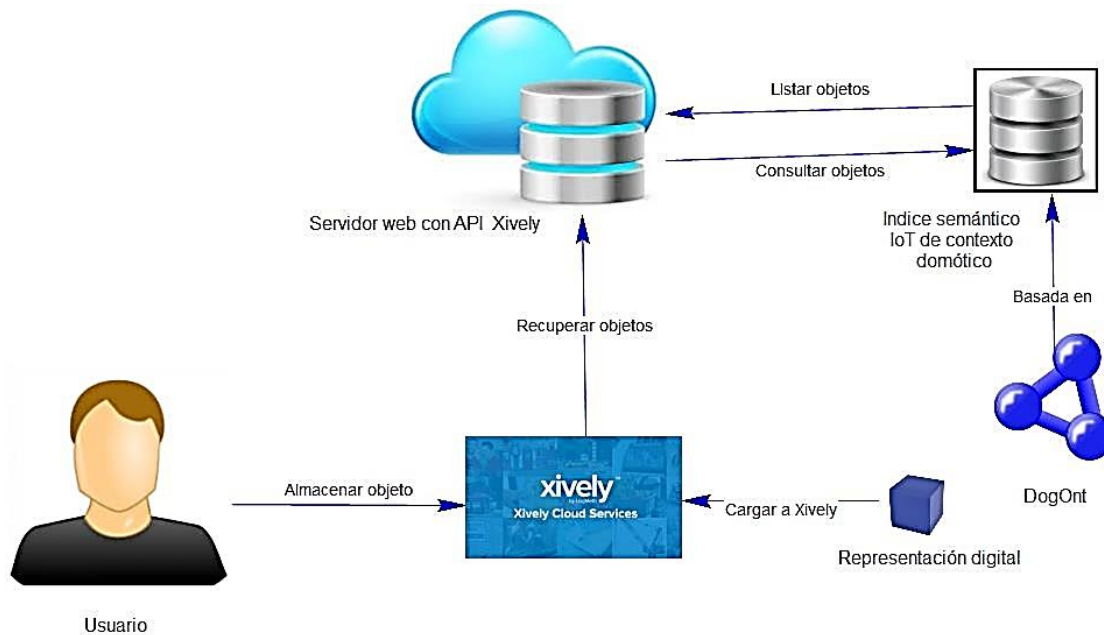


Figura 10 primera vista dinámica de la arquitectura. Fuente: propia

El escenario está provisto de etiquetas NFC, las cuales se dividen en dos tipos. Primero, las etiquetas que referencian a un objeto; estos contienen el identificador de un objeto (esta identificación corresponde con el *FeedId* entregado por Xively) y están visualmente relacionados a un objeto (se utilizan etiquetas adhesivas, las cuales estarán pegadas a los objetos). Segundo, las etiquetas que referencian a una entidad; estas contienen el nombre de la entidad y su ubicación global (longitud y latitud) y al igual que las anteriores están visualmente relacionadas a la entidad.

Por otra parte los objetos son inicializados manualmente por el usuario, y para ello el usuario debe realizar la conexión de cada objeto con sus respectivos recursos, luego se deben prender las placas y sincronizarlas con la representación digital, esto se hace ingresando la identificación del objeto en la aplicación inicializar objeto, la cual consulta la información del objeto en la capa de servicio. Finalmente se requiere al usuario ingresar la información sobre los metadatos de cada objeto que el servidor no retorna. De forma paralela la OOS guarda la información de los metadatos de cada objeto que posteriormente pueden ser consultados. De esta forma el objeto queda listo para ser utilizado dentro del escenario. En Figura 11 se muestra una representación gráfica de los procesos antes mencionados.

En la capa de interacción los objetos evento comunican los estados de sus recursos a los objetos acción utilizando el *bróker* MQTT como intermediario, los objetos acción ejecutan el ECA correspondiente a un servicio de interacción y accionan sus recursos actuadores de acuerdo a la programación del ECA.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

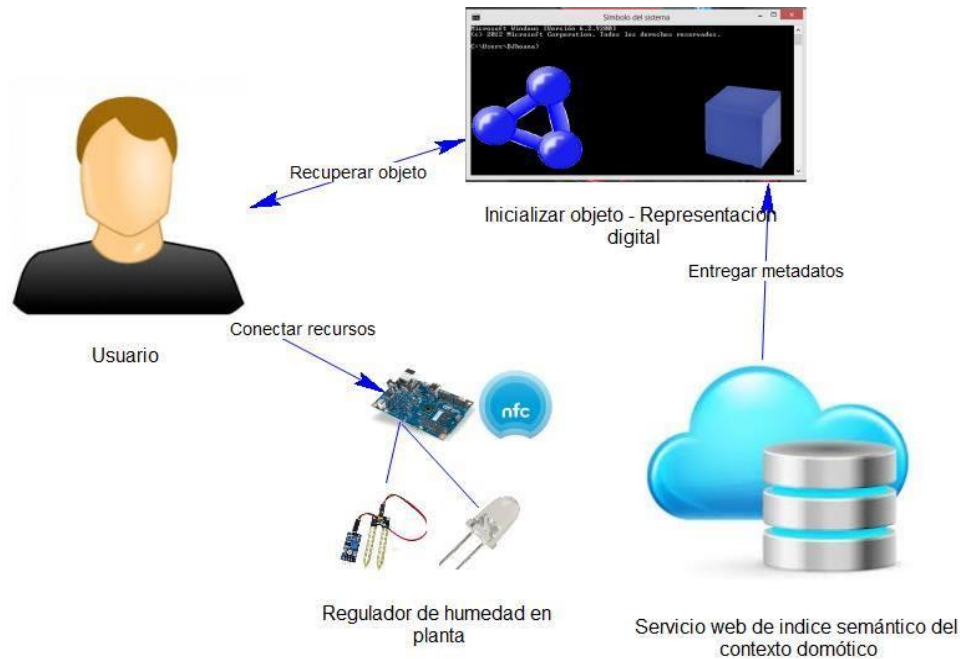


Figura 11. Segunda vista dinámica de la arquitectura. Fuente: propia

En la capa de aplicación, Clipio se asocia una etiqueta que referencia la entidad de interés, luego se administran los objetos relacionados a esa entidad comunicando sus requerimientos utilizando el *bróker*. Clipio puede consultar los metadatos y cambiar el estado del servicio básico de un objeto en particular, si este se asocia con la etiqueta correspondiente al objeto. Por otra parte los objetos consultan y publican la información sobre los estados de sus recursos utilizando el *bróker* MQTT. En Figura 12 se muestra una representación gráfica de los procesos antes mencionados.

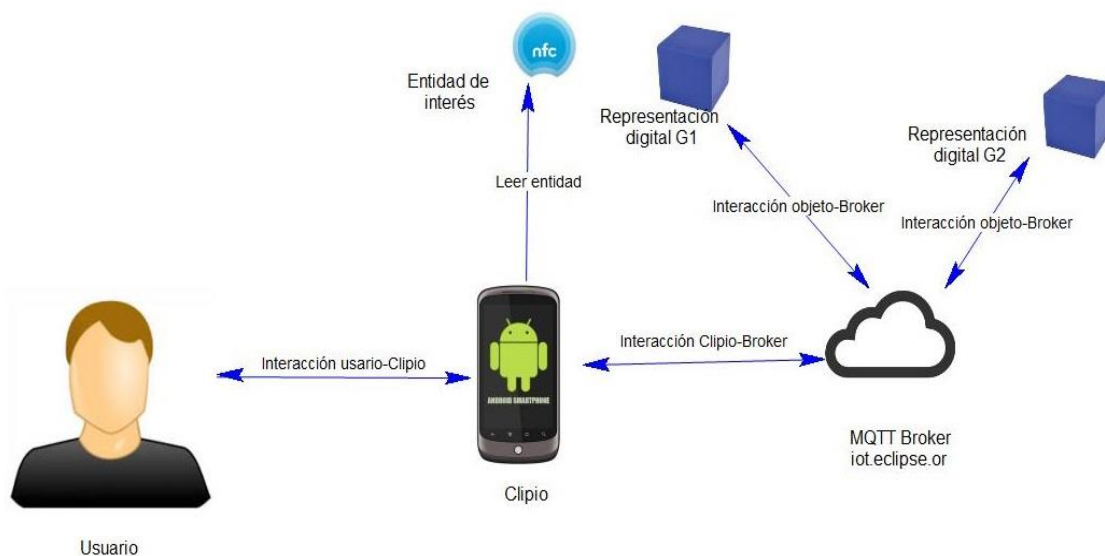


Figura 12 Tercera vista dinámica de la arquitectura. Fuente: propia

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

El panel de usuario se conecta al *bróker* MQTT como un elemento pasivo, el cual consulta constantemente los estados de todos los objetos y sus recursos y de todo su trabajo de mensajería provocado por los servicios de interacción. Todos los mensajes que viajan entre los elementos del escenario están soportados por el estándar ODF. En Figura 13 se muestra una representación gráfica de los procesos antes mencionados.

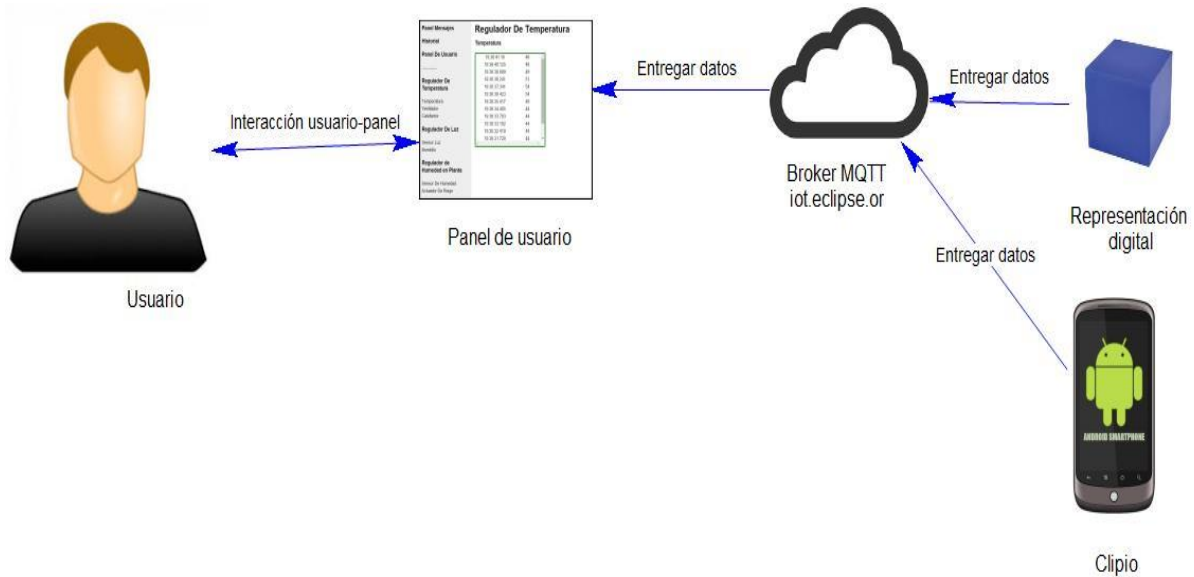


Figura 13 Cuarta vista dinámica de la arquitectura. Fuente: propia

4 IMPLEMENTACIÓN DEL ESCENARIO

En este punto del proyecto se materializa la arquitectura en una implementación con el fin de analizar las fortalezas, debilidades y su impacto en la construcción de este tipo de escenarios de interacción en la WoT. Para el desarrollo de las fases se hace uso de la metodología de desarrollo de software UP Ágil que define 4 fases. Para el presente documento se presentan las fases de inicio, de elaboración, de construcción y de transición.

4.1 FASE DE INICIO

Para fines de organización de la presente documentación se describirán los parámetros de evaluación en el capítulo 5. A continuación se presentan los requisitos definidos, los casos de uso críticos seleccionados y las decisiones sobre los elementos tecnológicos específicos que se utilizaron para el desarrollo del escenario.

4.1.1 Requisitos

En este apartado se presentan los requisitos básicos que el escenario debe cumplir para poder implementar correctamente el escenario propuesto.

- Desarrollar o utilizar herramientas que permitan almacenar, recuperar y visualizar la información de los tres objetos propuestos en la arquitectura. Esta información debe estar contenida en el servidor *middleware Xively*.
- Desarrollar o utilizar herramientas que permitan almacenar y recuperar el contexto de cada objeto inteligente.
- Desarrollar o utilizar herramientas que permitan definir requisitos de usuario, creación de servicios, consulta y ejecución de los servicios de interacción. Para esto es necesario definir herramientas que permitan modificar el estado de los servicios básicos de cada objeto, creación de un perfil de usuario y la gestión de los servicios de interacción por medio de crear, ver, modificar y eliminar los servicios de interacción.

4.1.2 Casos de uso críticos

En Figura 14 se presentan los casos de uso crítico diferenciados por el requisito al cual atienden. Se cuenta con los casos de uso para almacenar, recuperar y visualizar el objeto, segundo los casos de uso que permiten almacenar y recuperar el contexto de cada objeto y finalmente se presentan los casos de uso que permiten definir un usuario, modificar el estado de un objeto y gestionar los servicios de interacción.

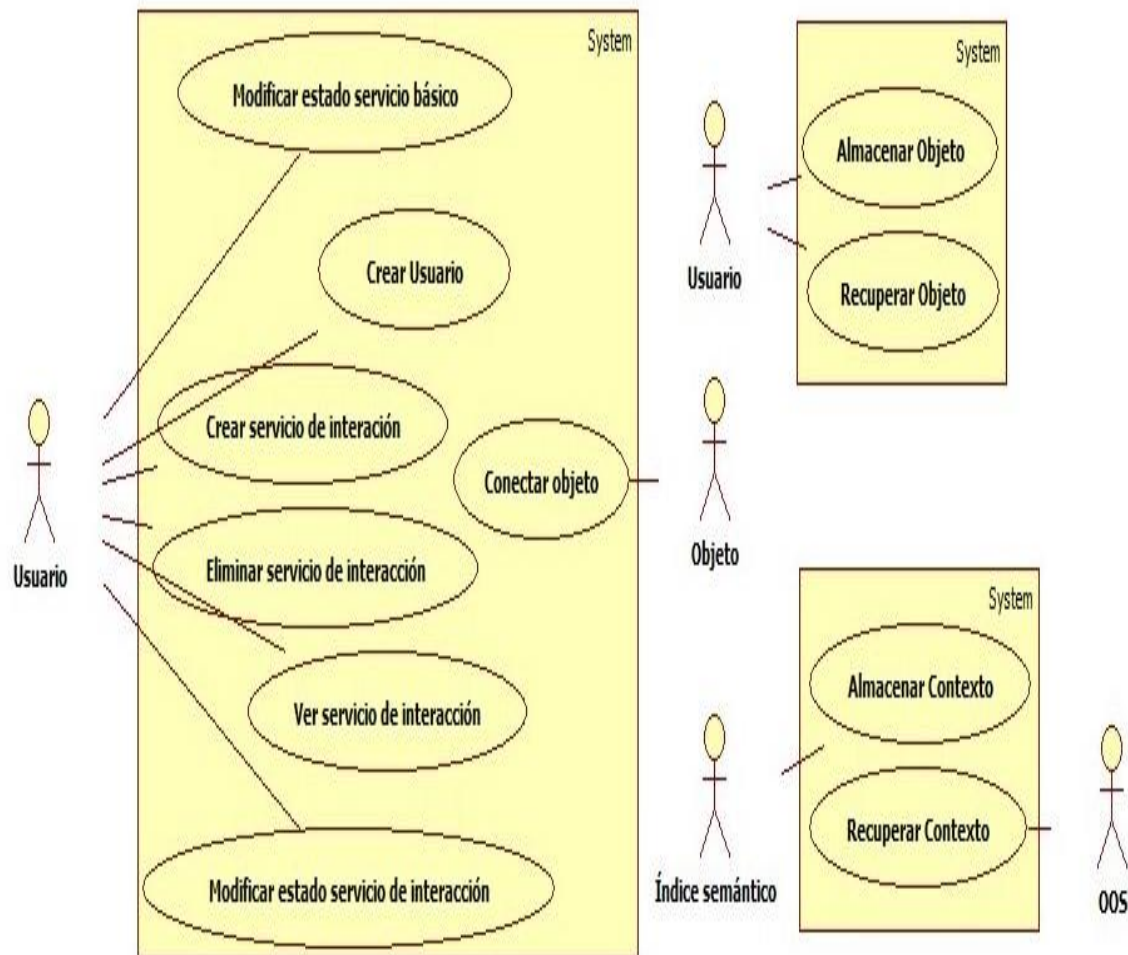


Figura 14. Casos de uso críticos. Fuente: propia.

4.1.3 Elementos tecnológicos

El escenario seleccionado es una casa inteligente que pertenece al grupo de ambientes domóticos, donde se pueden implementar muchos objetos con diversos propósitos, se puede identificar variadas propiedades, automatizar procesos y los distintos espacios dan lugar a plantear gran variedad de dispositivos. Para el presente caso de estudio se maneja únicamente el espacio sala, igualmente el mismo espacio se lo caracteriza como una entidad de interés y se asocia una etiqueta NFC la cual contiene el nombre “sala” y la localización (escenario ubicado en la oficina 402 del edificio de la facultad de ingeniería electrónica y de telecomunicaciones. Longitud: -76.5980193, latitud: 2.4470888), además cada objeto dentro del escenario tiene asociada una etiqueta NFC la cual almacena su identificación.

Cada objeto implementado en este caso de estudio está soportado por la placa Galileo de Intel, contiene la distribución Yocto de Linux y herramientas de programación de Python que soportan el manejo de XML (por medio de la librería *elementtree*), MQTT (por medio de la librería de *Paho* para Python) y SOAP (por medio de la librería *suds*). Inicialmente se

consideró utilizar Windows como sistema operativo, debido a que existía mayor experiencia por parte del grupo de trabajo con este tipo de desarrollos, sin embargo se observó que la distribución Windows ofrecida para Galileo es poco madura (para la fecha de entrega de este trabajo) y con escasa comunidad que la soporte, además presenta varios problemas al momento de desarrollar y ejecutar las aplicaciones. Por otra parte se ha seleccionado Python como lenguaje de programación para las placas Galileo gracias a que es el lenguaje recomendado por la comunidad de Intel para este tipo de desarrollos, además Python es un lenguaje multiplataforma, lo que permite que el desarrollo pueda escalar fácilmente a otros dispositivos con distintas distribuciones de Linux.

Los objetos cuentan con la librería ObjetoInteligente², la cual le permite realizar toda la lógica interna de funcionamiento de cada objeto en el escenario. La placa Galileo se ha seleccionado debido a la disponibilidad de placas que existe en la universidad del Cauca, además cumplen todos los requerimientos para crear un objeto inteligente, según la clasificación de Mathew [14]. Los objetos propuestos para el presente caso de estudio son:

- **Regulador de temperatura.** Establece la temperatura mediante mecanismos que permiten detectarla, reducirla y aumentarla. El objeto está desplegado en el espacio sala y puede ser relacionado a la entidad con este nombre. Permite medir la propiedad temperatura utilizando el módulo³ *Arduino Grove-Temperature V1.2*, el cual consta de un termistor que detecta los cambios de la temperatura ambiente y una resistencia sobre la cual varía el voltaje acorde a los cambios en las propiedades del termistor, el rango de detección del termistor es de -40°C a 125°C con una precisión de $\pm 1.5^{\circ}\text{C}$. Permite disminuir el nivel de la propiedad temperatura utilizando el actuador ventilador y aumentar el nivel de la misma mediante el actuador calefactor. Como servicio básico este objeto ofrece la regulación de la temperatura en 10°C .
- **Regulador de luz.** Establece el nivel de luz mediante mecanismos que permiten detectar cuando la propiedad se encuentre por debajo de un umbral deseado y aumentarla. El objeto está desplegado en el espacio sala y puede ser relacionado a la entidad de con este nombre. Permite medir la propiedad luz utilizando el módulo *Arduino Grove-Light*, el cual consta de un foto-resistor que detecta los cambios de luz y una resistencia sobre la cual varía el voltaje acorde a los cambios en las propiedades del foto-resistor. Permite aumentar el nivel de la propiedad luz utilizando el actuador bombillo, el aumento del nivel de luz se considera binario (prender o apagar bombillo). Como servicio básico este objeto ofrece prender el bombillo cuando el nivel de luz sobrepase el 10% o apagarlo de lo contrario.
- **Regulador de humedad en una planta:** Establece el nivel de humedad en una planta mediante mecanismos que permiten detectar cuando la propiedad se encuentra por debajo de un umbral deseado y aumentarla. El objeto está desplegado en el espacio sala y puede ser relacionado a la entidad con este nombre. Las mediciones de este

² Esta librería fue desarrollada por el equipo de trabajo y es un elemento entregable del proyecto.

³ Este módulo es el recurso de tipo sensor que hace parte del objeto

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

recurso serán emuladas por un arreglo de resistencias variables (a causa de no tener el sensor o módulo que permite realizar este tipo de mediciones). El aumento de los niveles de humedad se realiza utilizando el actuador riego, el cual consta de un motor que abre las compuertas para dejar pasar cierta cantidad de agua hacia la planta, el aumento del nivel de humedad se considera binario (prender o apagar riego). Como servicio básico este objeto ofrece prender el riego cuando el nivel de humedad esté por debajo del 10% o apagarlo de lo contrario.

Para encontrar información más detallada sobre la implementación de los objetos se debe acceder el anexo B.

El elemento coordinador (Clipio) fue desarrollado como aplicación móvil para versiones 5.0 o superiores de Android. El dispositivo debe tener como requisitos básicos de hardware: interfaz para realizar lectura de etiquetas NFC y conexión a internet. La aplicación consta con librerías Java que soportan el manejo de XML (por medio de la librería *jdom*), MQTT (por medio de la librería de *Paho* para Android) y SOAP (por medio de la librería *Ksoap2*). Para las pruebas se utiliza el dispositivo móvil Nexus 5 de LG, el cual satisface todos los requerimientos para montar la aplicación Clipio.

El elemento panel de usuario fue desarrollado como aplicación web basado en Javascript, Php, Ccs y para el modelo MySql. Actualmente la aplicación no se encuentra publicada sobre ningún dominio, por lo tanto es necesario que se ejecute en la maquina local utilizando la distribución Xampp de Apache.

El elemento inicializar objeto fue desarrollado como una aplicación de escritorio en Python. Para ingresar a la aplicación se debe hacer uso del cliente SSH Putty, con el cual es posible manejar el terminal Linux de cada placa Galileo conectada en una red local y ejecutar la aplicación Python correspondiente a la inicialización del objeto.

Para el servidor *bróker* MQTT se seleccionó *Paho* (iot.eclipse.org) soportado por Eclipse, este *bróker* entrega librerías para desarrollar clientes en Python, JavaScript y Android, además de una extensa documentación y comunidad. Este servidor es capaz de soportar aplicaciones basadas en TCP y WS, facilitando la interoperabilidad entre las aplicaciones. Se realizó la prueba entre distintos servidores MQTT como los que ofrece: *MQThing* y *Xively*⁴, entregando resultados similares en su comportamiento, sin embargo se seleccionó *Paho* debido a que *MQThing* y *Xively* exigen una decodificación de las cadenas de caracteres, lo cual suma un proceso de desarrollo en los elementos suscriptores.

⁴ Además del servicio middleware, ofrece puertos que soportan MQTT.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

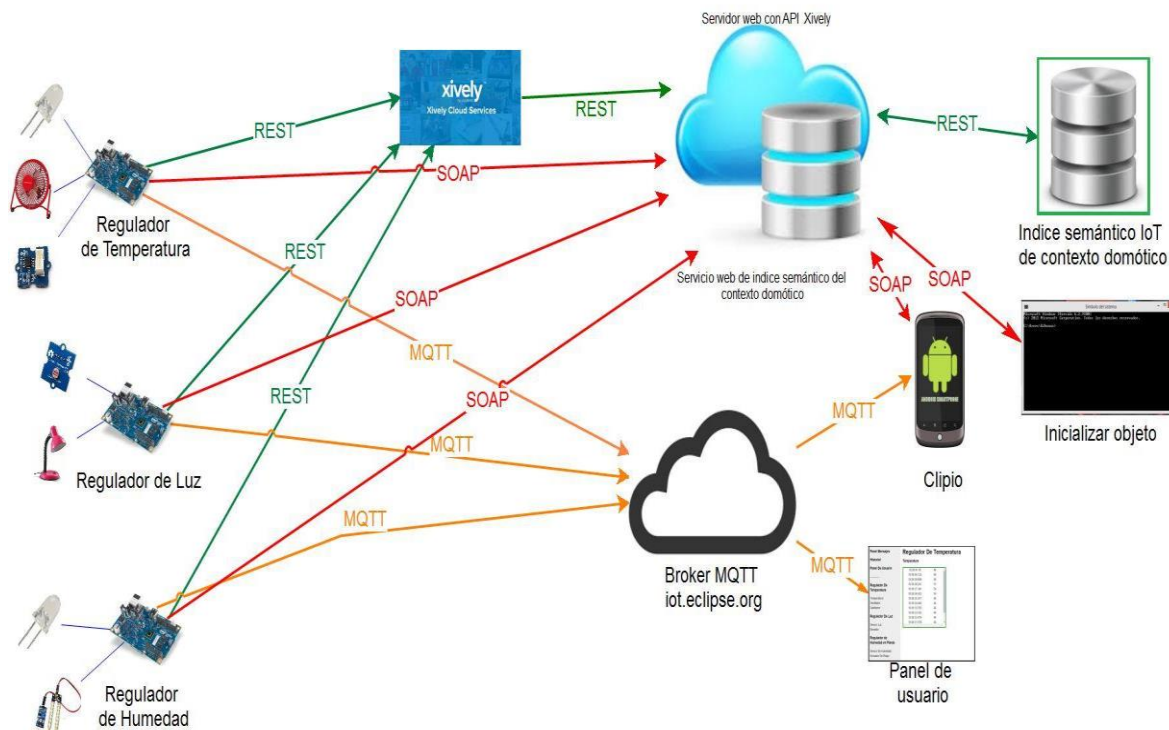


Figura 15. Diagrama de protocolos Fuente: propia

Como se presenta ver en la *Figura 15*, los protocolos involucrados en la comunicación del escenario son: MQTT, SOAP y REST. La comunicación entre los objetos y sus recursos no se especifica, debido a que para el presente caso de estudio se conectan directamente en la placa Galileo, sin embargo se pueden considerar Bluetooth o Xbee para realizar esta conexión. Cada objeto publica en Xively sus estados mediante el método PUT soportado en REST, se comunican con Clipio y con otros objetos mediante MQTT y se comunican con el índice semántico mediante los métodos SOAP expuestos en la capa de servicio. De igual forma Clipio consulta al índice mediante los métodos SOAP expuestos la misma capa. El índice semántico se conecta a la capa de servicio utilizando métodos REST y la capa de servicio a su vez recupera la información en archivos JSON de Xively utilizando REST. Finalmente la aplicación inicializar objeto consume información a través de SOAP en el servidor web.

4.2 FASE DE ELABORACIÓN

Para el desarrollo del escenario de interacción semántica se definen tres módulos importantes, módulo de recuperación de objetos inteligentes – MROI, módulo de contexto de objetos inteligentes – MCOI y módulo de interacción de objetos inteligentes – MIOI, cada uno de estos módulos responde respectivamente a los requerimientos básicos planteados en la fase de inicio.

A continuación se presenta una gráfica que agrupa a los distintos elementos de la arquitectura según los módulos, luego se presentan los formatos expandidos de los casos de uso críticos desarrollados asociados a los distintos módulos. Los casos de uso reales

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

pueden ser consultados en el anexo D. Los diagramas conceptuales y los diagramas de clases son remplazados por diagramas de secuencia y se presentan en la siguiente fase, esto debido a que se quiere mostrar con mayor detalle la naturaleza de interacción del escenario.

4.2.1 Módulos

En la Figura 16 se muestra un arreglo de los distintos elementos de la arquitectura antes propuesta y como están organizados según los módulos, además se muestran las relaciones que realizan los elementos según los casos de uso críticos y los procesos contenidos en cada uno.

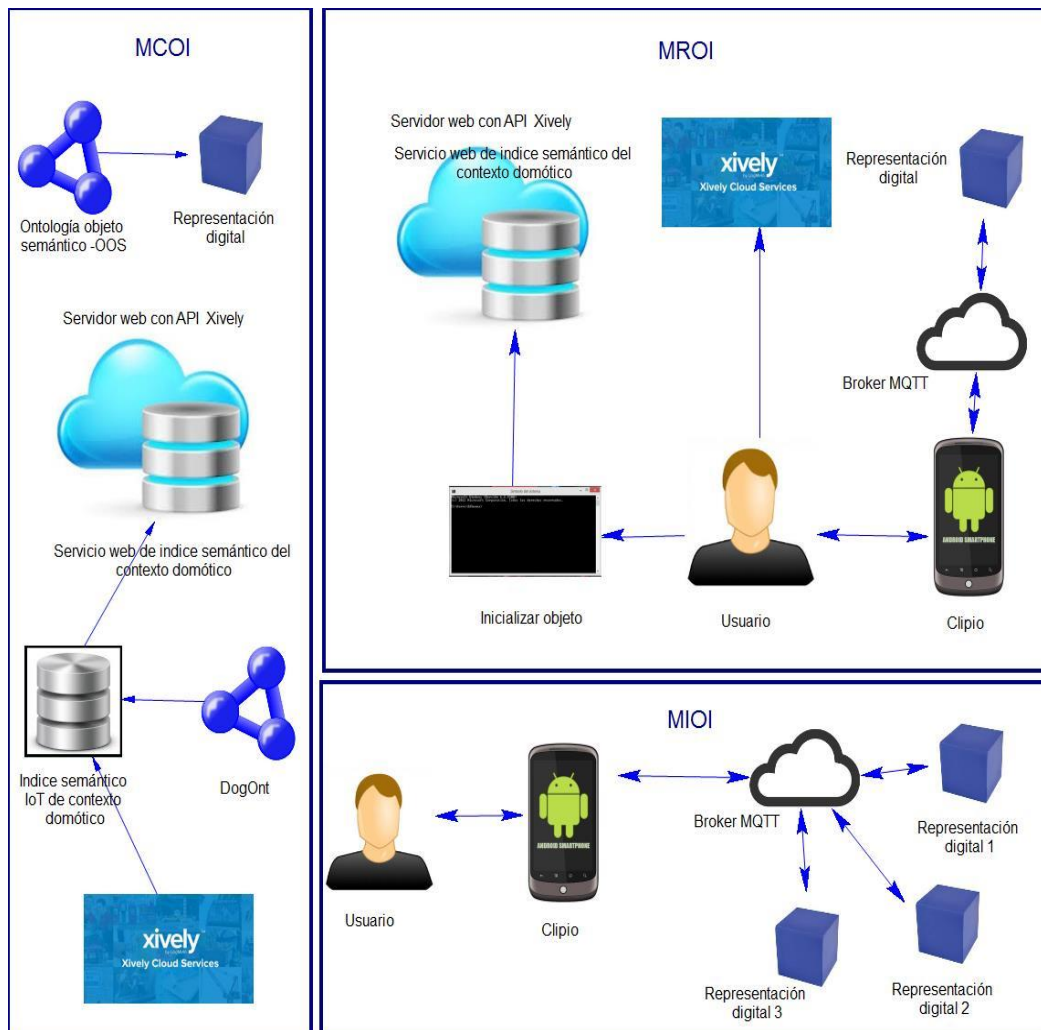


Figura 16. Elementos arquitectónicos según módulos. Fuente: propia

En la Figura 16 se puede diferenciar los siguientes módulos:

- **MROI.** Este módulo permite almacenar, recuperar y visualizar la información de los diferentes objetos en el servidor *middleware* Xively. En este módulo se definen los casos de uso: almacenar objeto, recuperar objeto y visualizar objeto.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- Recuperar objeto, como se demuestra en la Figura 16. Elementos arquitectónicos según módulos. Fuente: propia el usuario se comunica directamente con la interfaz de *Xively* donde ingresa información sobre los objetos
 - Almacenar objeto, en la figura el usuario se comunica con la interfaz de inicializar objeto para recuperar la representación digital almacenada en el servidor web del objeto
 - Visualizar objeto, en la figura el usuario utiliza *Clipio* para visualizar la información contenida en el objeto, la cual es entregada por medio del *bróker* MQTT.
- **MCOI.** Este módulo permite recuperar el contexto con el índice semántico y almacena el contexto en la OOS utilizando los metadatos del objeto. En este módulo se definen los casos de uso: almacenar contexto y recuperar contexto.
 - Almacenar contexto, en la figura el objeto consulta directamente a la OOS para recuperar la información de sus metadatos. Este caso de uso construye el archivo *MetaData.xml* utilizando consultas SPARQL realizadas directamente a la OOS, estas consultas se pueden detallar en el anexo H.
 - Recuperar contexto, en la figura el índice semántico consulta los conceptos de *DogOnt* y consulta la información de los objetos en *Xively* para generar listas de objetos y exponerlas en el servidor web como métodos SOAP.
 - **MIOI.** Este módulo muestra la interacción entre objetos, aquí se definen requisitos de usuario, creación de servicios, consulta y ejecución de los nuevos servicios. En este módulo se definen los casos de uso: Modificar estado servicio básico, crear usuario, crear servicio de interacción, ver servicio de interacción, eliminar servicio de interacción, modificar estado servicio de interacción y conectar objeto.
 - Todos los casos de uso (menos conectar objeto) son realizados por el mismo medio donde el usuario puede administrar los objetos comunicando sus requerimientos por el *bróker* MQTT.
 - Conectar objeto, en la figura los objetos comunican sus estados a otros objetos utilizando el *bróker* MQTT.

Los detalles sobre los procesos de cada caso de uso se presentan con mayor claridad en los formatos expandidos. Además se puede consultar la descripción de cada proceso en el anexo E.

4.2.2 Formatos expandidos

A continuación se presentan los formatos expandidos para explicar más detalladamente los casos de uso visualizar objeto, modificar estado servicio básico, crear servicio de interacción, modificar estado deservicio de interacción, eliminar servicio de interacción. Los formatos expandidos de los otros casos de uso se encuentran en el anexo D.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Nombre	Visualizar objeto	
Actores	Usuario	
Propósito	Permite visualizar los metadatos de los objetos.	
Resumen	Se presentan los detalles sobre los metadatos de un objeto. El usuario identifica el objeto mediante la etiqueta NFC asociada al objeto.	
Tipo	Primario.	
Precondición	<ul style="list-style-type: none"> - El objeto debe tener la información completa de los metadatos (caso de uso, recuperar objeto) - El objeto debe estar asociado con una etiqueta NFC que contenga su identificación. - El usuario debe ubicarse en la interfaz principal de Clipio donde se pueden leer etiquetas NFC. - El objeto debe estar suscrito al canal MQTT con la estructura "id_objeto/coordinador", donde "id_objeto" cambia según el objeto. - Clipio debe estar suscrito al canal MQTT con la estructura "id_objeto/MetaData", donde "id_objeto" cambia según el objeto. 	
Curso normal de los eventos		
Acción del actor usuario	Respuesta del sistema	
Lee la etiqueta NFC correspondiente al objeto.		
	Consulta los metadatos correspondientes a la identificación entregada, publicando petición en id_objeto/coordinador.	
	Publica metadatos en id_objeto/MetaData	
	Organiza y presenta los metadatos.	

Tabla 4. Caso de uso visualizar objeto. Fuente: propia.

Nombre	Modificar estado servicio básico	
Actores	Usuario.	
Propósito	Modifica mediante Clipio el estado del servicio básico que posee un objeto.	
Resumen	El estado del servicio básico se modifica directamente sobre los metadatos del objeto, por lo tanto el servicio básico del objeto puede estar prendido o apagado.	
Tipo	Primario.	
Precondición	<ul style="list-style-type: none"> - El objeto debe tener representación física y digital (caso de uso, almacenar objeto). - El objeto debe tener los metadatos completos (caso de uso, recuperar objeto) - El objeto debe estar asociado con una etiqueta NFC que contenga su identificación. - El usuario debe ubicarse en la interfaz donde se despliegan los metadatos del objeto a modificar (caso de uso, visualizar objeto) - El objeto debe estar suscrito al canal MQTT con la estructura "id_objeto/coordinador", donde "id_objeto" cambia según el objeto. - Clipio debe estar suscrito al canal MQTT con la estructura "id_objeto/EstadoBásico", donde "id_objeto" cambia según el objeto. 	
Curso normal de los eventos		
Acción del actor usuario	Respuesta del sistema	

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Presiona la interfaz toggle button para iniciar el requerimiento de cambio de estado.	
	Modifica el estado del servicio básico del objeto, publicando solicitud en id_objeto/coordinador.
	Publica código de respuesta en id_objeto/EstadoBasic
	Organiza y presenta la información, incluyendo el nuevo estado del servicio básico.

Tabla 5. Caso de uso modificar servicio básico. Fuente: propia

Crear servicio de interacción	
Nombre	
Actores	Usuario
Propósito	Permite crear un servicio para generar interacción entre dos objetos.
Resumen	Busca los metadatos de los objetos y crea un nuevo servicio de interacción a partir de una sentencia ECA.
Tipo	Primario.
Precondición	<ul style="list-style-type: none"> - Los objetos participantes deben tener representación física y digital (caso de uso, almacenar objeto). - Los objetos participantes deben tener los metadatos completos (caso de uso, recuperar objeto) - El usuario debe obtener el nombre de la entidad de interés por medio de la etiqueta NFC asociada a la entidad. - El usuario debe ubicarse en la interfaz principal de Clipio y accionar la utilidad de crear. - El objeto debe estar suscrito al canal MQTT con la estructura "id_objeto/coordinador", donde "id_objeto" cambia según el objeto. - Clipio debe estar suscrito al canal MQTT con la estructura "id_objeto/CrearECA" e "id_objeto/MetaData", donde "id_objeto" cambia según el objeto.
Curso normal de los eventos	
Acción del actor usuario	Respuesta del sistema
Ingresa la búsqueda de los objeto (evento y acción) en lenguaje natural y define el radio de búsqueda a partir de la posición de la entidad de interés.	
	Realiza la búsqueda de los objetos adicionando el nombre de la entidad de interés.
	Lista y presenta al usuario los recursos asociados a los objetos encontrados.
Selecciona los recursos de los objetos para implementar el servicio de interacción.	
	Consulta los metadatos de los objetos asociados a los recursos seleccionados. (caso de uso, visualizar objeto)
	Presenta la interfaz para seleccionar las condiciones y variables del servicio de interacción.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Ingresar las condiciones y variables.	
	Crea la sentencia ECA en los objetos correspondientes a los recursos seleccionados, publicando solicitud en id_objeto/coordinador
	Publica código de respuesta en id_objeto/CrearECA
	Notifica al usuario.

Tabla 6. Caso de uso crear servicio de interacción. Fuente: propia

Eliminar servicio de interacción	
Nombre	
Actores	Usuario
Propósito	Permite al usuario eliminar un servicio de interacción.
Resumen	Se eliminan el servicio de interacción almacenado en los dos objetos que participan en la interacción.
Tipo	Primario.
Precondición	<ul style="list-style-type: none"> - El servicio de interacción debe existir (caso de uso: crear servicio de interacción) - Los objetos participantes deben tener representación física y digital (caso de uso, almacenar objeto). - Los objetos participantes deben tener los metadatos completos (caso de uso, recuperar objeto) - El usuario debe ubicarse en la interfaz que muestra los detalles de los servicios de interacción (caso de uso, ver servicio de interacción). - El objeto debe estar suscrito al canal MQTT con la estructura "id_objeto/coordinador", donde "id_objeto" cambia según el objeto. - Clipio debe estar suscrito al canal MQTT con la estructura "id_objeto/EliminarECA", donde "id_objeto" cambia según el objeto.
Curso normal de los eventos	
Acción del actor usuario	Respuesta del sistema
Presiona la interfaz botón eliminar para iniciar el requerimiento.	
	Elimina el servicio de interacción en los dos objetos involucrados publicando solicitud en id_objeto/coordinador.
	Publica código de respuesta en id_objeto/EliminarECA
	Notifica al usuario.

Tabla 7. Caso de uso eliminar servicio de interacción. Fuente: propia

Modificar estado servicio de interacción	
Nombre	
Actores	Usuario
Propósito	Permite al usuario modificar el estado de un servicio de interacción.
Resumen	Se modifica el estado de un servicio de interacción almacenado en los dos objetos que participan en la interacción.
Tipo	Primario.
Precondición	<ul style="list-style-type: none"> - El servicio de interacción debe existir (caso de uso: crear servicio de interacción)

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

	<ul style="list-style-type: none"> - Los objetos participantes deben tener representación física y digital (caso de uso, almacenar objeto). - Los objetos participantes deben tener los metadatos completos (caso de uso, recuperar objeto) - El usuario debe ubicarse en la interfaz que muestra los detalles de los servicios de interacción (caso de uso, ver servicio de interacción). - El objeto debe estar suscrito al canal MQTT con la estructura "id_objeto/coordinador", donde "id_objeto" cambia según el objeto. - Clipio debe estar suscrito al canal MQTT con la estructura "id_objeto/EstadoECA", donde "id_objeto" cambia según el objeto.
Curso normal de los eventos	
Acción del actor usuario	Respuesta del sistema
Presiona la interfaz toggle button para iniciar el requerimiento.	
	Modifica el estado del servicio de interacción en los dos objetos involucrados, publicando solicitud en id_objeto/coordinador.
	Publica código de respuesta en id_objeto/EstadoECA
	Notifica al usuario.

Tabla 8. Caso de uso modificar estado servicio de interacción. Fuente: propia

Nombre	Conectar objeto
Actores	Objeto
Propósito	Consumir la información sobre el estado de un determinado recurso del objeto evento determinado en un servicio de interacción.
Resumen	El objeto se conecta al canal del objeto evento por medio del bróker MQTT para consumir la información que el objeto evento está publicando constantemente sobre el estado de un determinado recurso.
Tipo	Secundario. Se realiza de forma automática al usuario.
Precondición	<ul style="list-style-type: none"> - El servicio de interacción debe existir (caso de uso: crear servicio de interacción) - Los objetos participantes deben tener representación física y digital (caso de uso, almacenar objeto). - Los objetos participantes deben tener los metadatos completos (caso de uso, recuperar objeto) - El objeto evento debe estar publicando el estado de su recurso de interés en el canal MQTT con la estructura "id_objeto_evento/id_recurso_evento", donde "id_objeto_evento" e "id_recurso_evento" cambian según el objeto evento que esté asociado al servicio de interacción.
Curso normal de los eventos	
Acción del actor objeto	Respuesta del sistema
Realiza la suscripción al canal del objeto evento "id_objeto_evento/id_recurso_evento"	
	Publica el estado de un recurso cada segundo

Tabla 9. Caso de uso modificar estado servicio de interacción. Fuente: propia.

4.3 FASE DE CONSTRUCCIÓN

En esta fase se implementan las herramientas que expondrán la funcionalidad a los usuarios, para esto primero se muestran los diagramas de secuencia que explican la naturaleza interactiva de los distintos elementos en cada caso de uso definido en la fase de inicio, luego se definen las interfaces resultantes de cada aplicación que sirve de interfaz al usuario y finalmente se detallan los productos de cada una de las cuatro iteraciones propuesta en esta fase

4.3.1 Diagramas de secuencia

A continuación se presentan únicamente los diagramas de secuencia relacionados a los formatos expandidos explicados en la fase de elaboración, los otros diagramas de secuencia pueden ser consultados en el anexo D. Se debe tener en cuenta que se deben cumplir las mismas precondiciones mencionadas en los formatos expandidos para que se pueda inicializar los procesos del respectivo proceso detallado en el diagrama de secuencia.

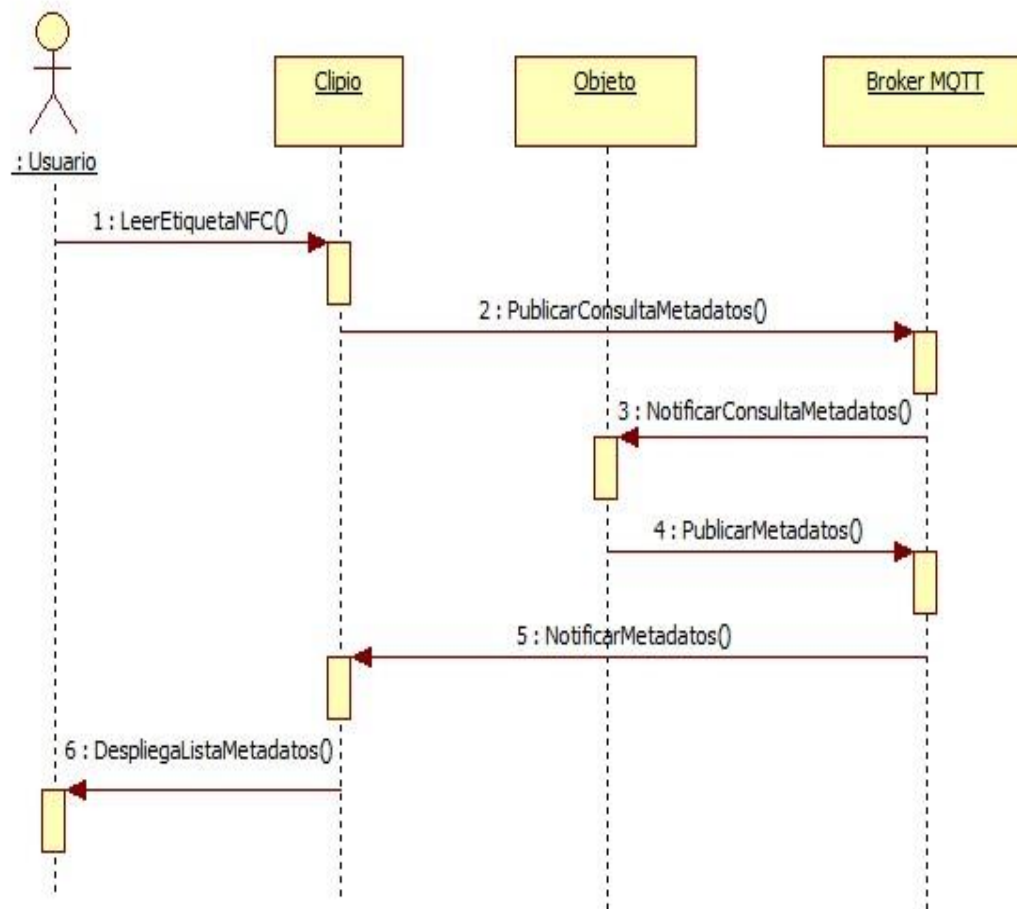


Figura 17. Diagrama de secuencia caso de uso visualizar objeto. Fuente: propia

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

La Figura 17 muestra la secuencia de relaciones que se genera entre los distintos elementos de la arquitectura al momento de visualizar un objeto. Para este diagrama se debe tener en cuenta que el escenario está provisto de etiquetas NFC asociadas a cada objeto del escenario, y estas son capaces de entregar la identificación del objeto. Clipio comunica la consulta de los metadatos publicando el archivo MetadataQuery.xml y el objeto responde con la información de sus metadatos utilizando el archivo Metadata.xml

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

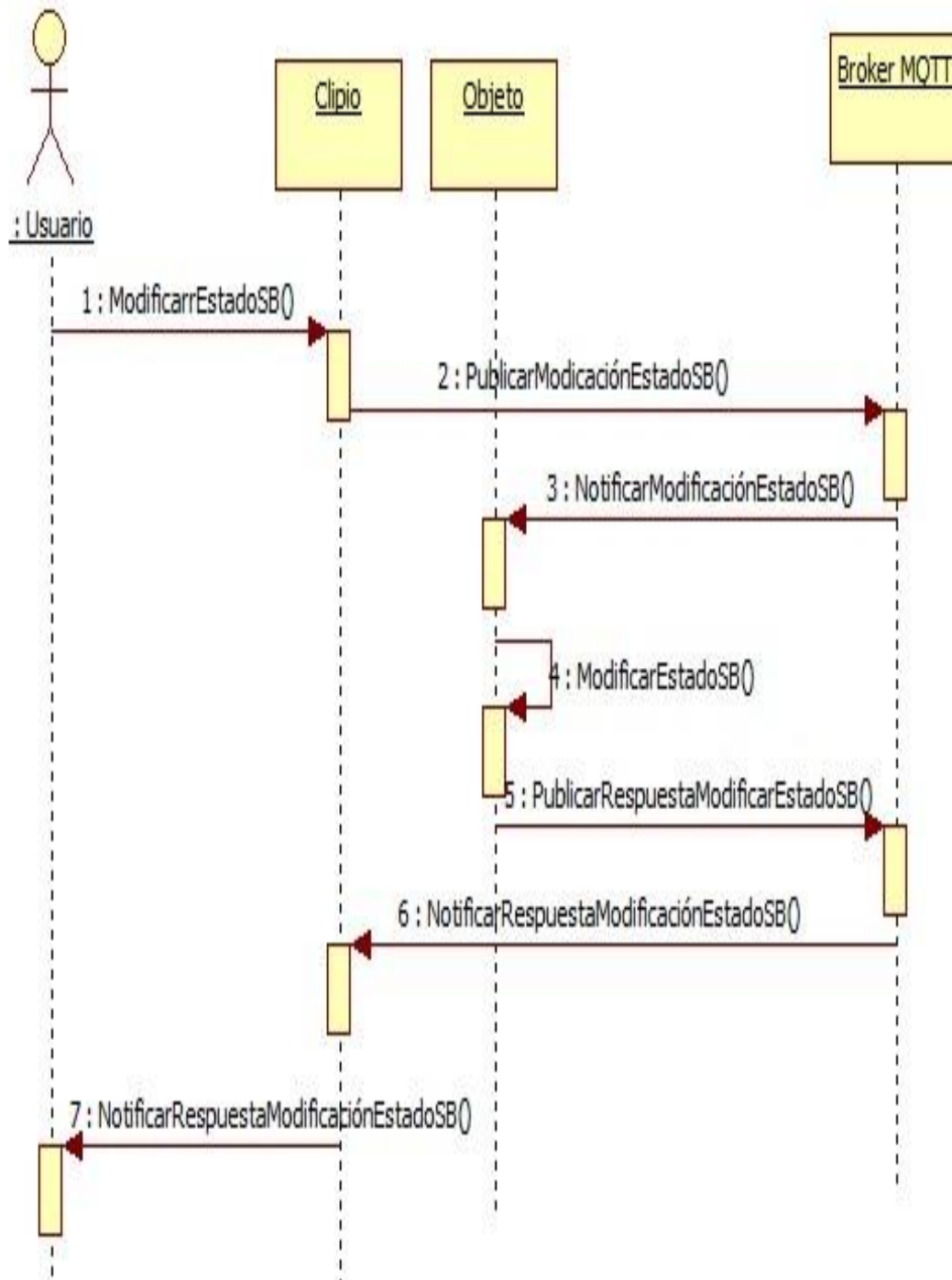


Figura 18. Diagrama de secuencia caso de uso modificar estado servicio básico. Fuente: propia

La Figura 18 muestra la secuencia de relaciones que se genera entre los distintos elementos de la arquitectura al momento de modificar el estado del servicio básico de un objeto. Clipio realiza el requerimiento de cambio de estado del servicio básico publicando

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

el archivo SetBasicState.xml, el objeto realiza la modificación internamente y publica el archivo SimpleResponse.xml con el resultado de su proceso interno.

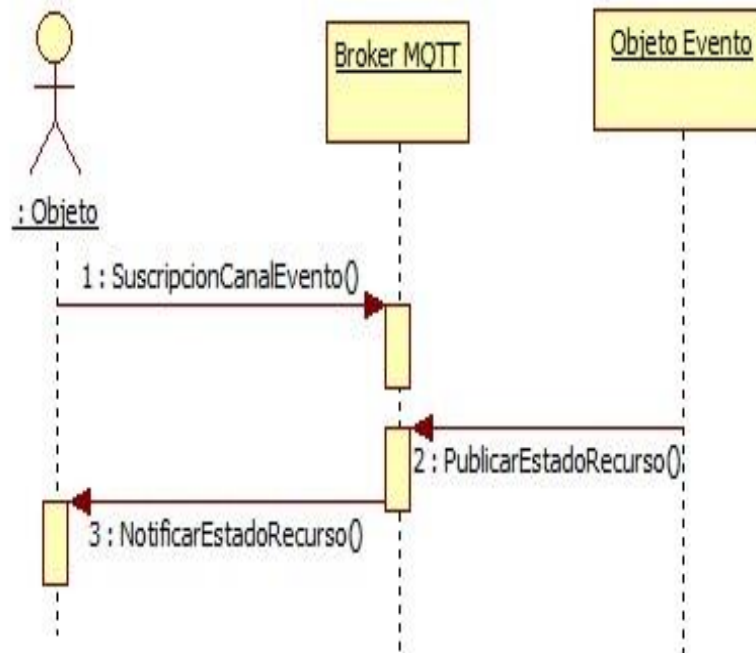


Figura 19. Diagrama de secuencia caso de uso conectar objeto. Fuente: propia

La Figura 19 muestra la secuencia que se genera al momento de realizar una conexión del objeto acción con el objeto evento, con el fin de que el objeto acción pueda consumir la información que publica el objeto evento sobre el estado, para ejecutar la sentencia ECA correspondiente a un servicio de interacción. El objeto evento publica el archivo SimpleValue.xml donde está contenido el último estado que informa un determinado recurso.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

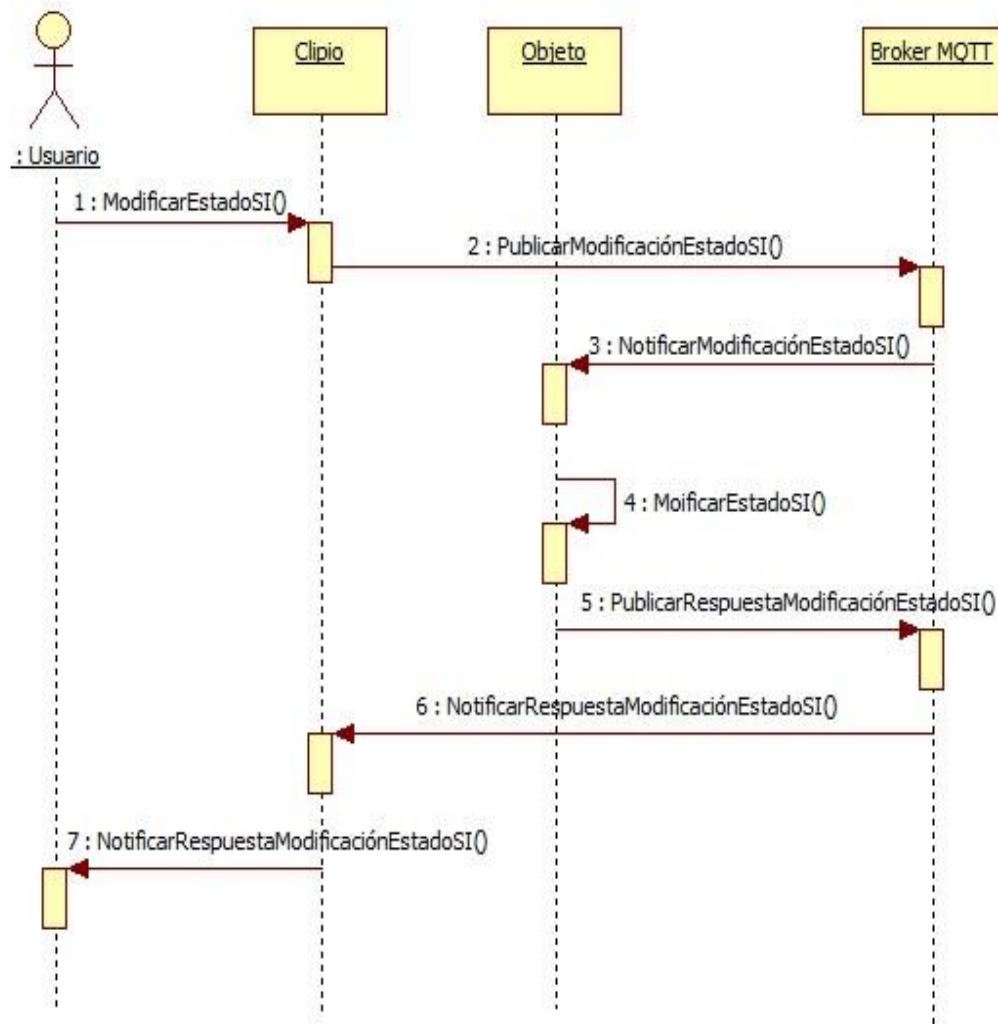


Figura 20. Diagrama de secuencia caso de uso modificar estado servicio de interacción. Fuente: propia

La Figura 20 muestra la secuencia que se genera al momento de modificar el estado de un servicio de interacción. El estado puede ser prendido o apagado. Clipio realiza el requerimiento de cambio de estado del servicio de interacción publicando el archivo SetECASState.xml, el objeto realiza la modificación internamente y publica el archivo SimpleResponse.xml con el resultado de su proceso. Se debe tener en cuenta que los procesos descritos se repiten de forma paralela en los dos objeto que están participando de la interacción, el objeto evento y en el objeto acción.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

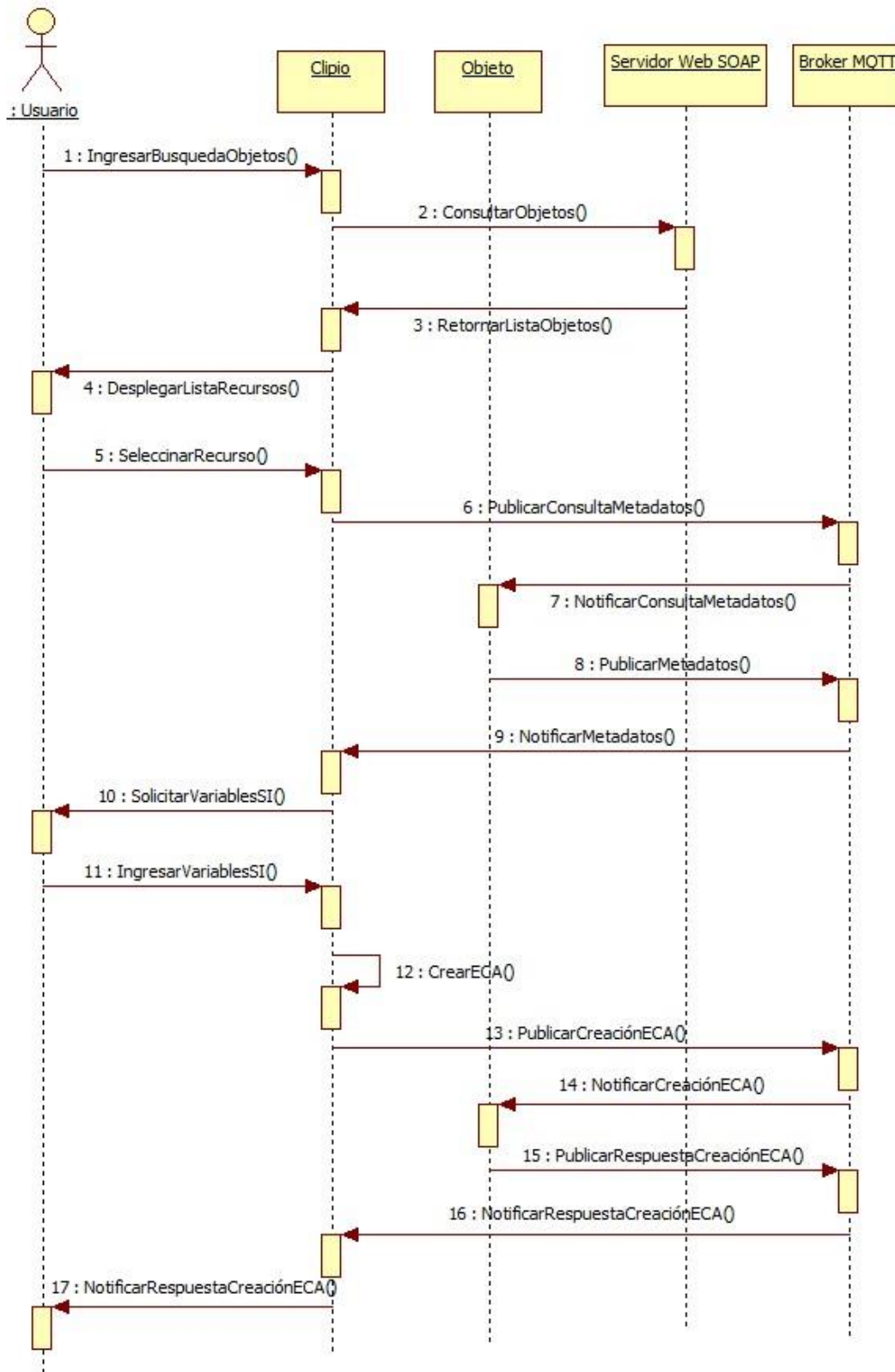


Figura 21. Diagrama de secuencia caso de uso crear servicio de interacción. Fuente: propia.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

La Figura 21 muestra la secuencia que se genera al momento de crear un servicio de interacción. Clipio realiza el requerimiento de crear un servicio de interacción publicando el archivo ECA.xml, el objeto realiza la creación internamente y publica el archivo SimpleResponse.xml con el resultado de su proceso. Se debe tener en cuenta que los procesos descritos se repiten de forma paralela en los dos objeto que estén participando de la interacción, el objeto evento y en el objeto acción.

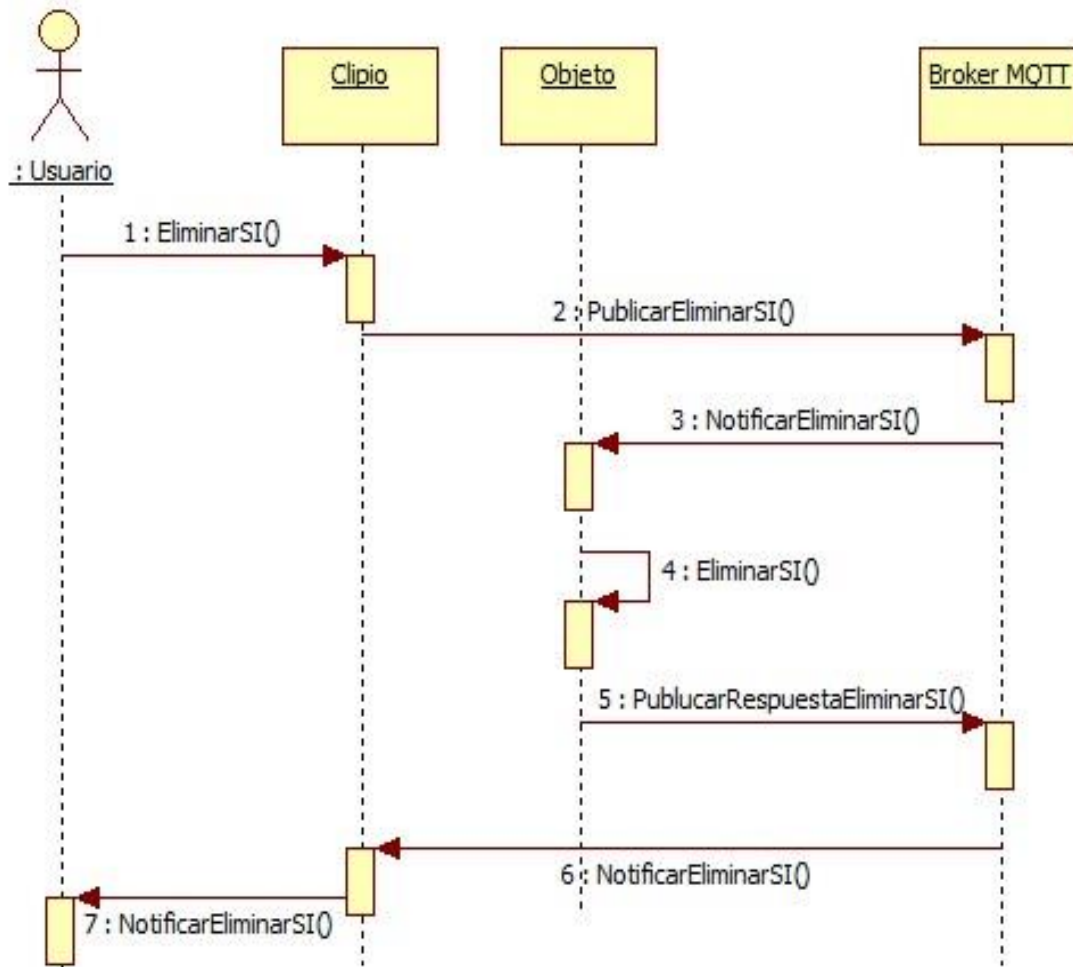


Figura 22. Diagrama de secuencia caso de uso eliminar servicio de interacción. Fuente: propia.

La Figura 22 muestra la secuencia que se genera al momento de eliminar un servicio de interacción. Clipio realiza el requerimiento de eliminar un servicio de interacción publicando el archivo ECADelete.xml, el objeto realiza la eliminación internamente y publica el archivo SimpleResponse.xml con el resultado de su proceso. Se debe tener en cuenta que los procesos descritos se repiten de forma paralela en los dos objeto que estén participando de la interacción, el objeto evento y en el objeto acción.

4.3.2 Interfaces al usuario

A continuación se exponen las principales interfaces que las aplicaciones desarrolladas para el escenario muestran al usuario.

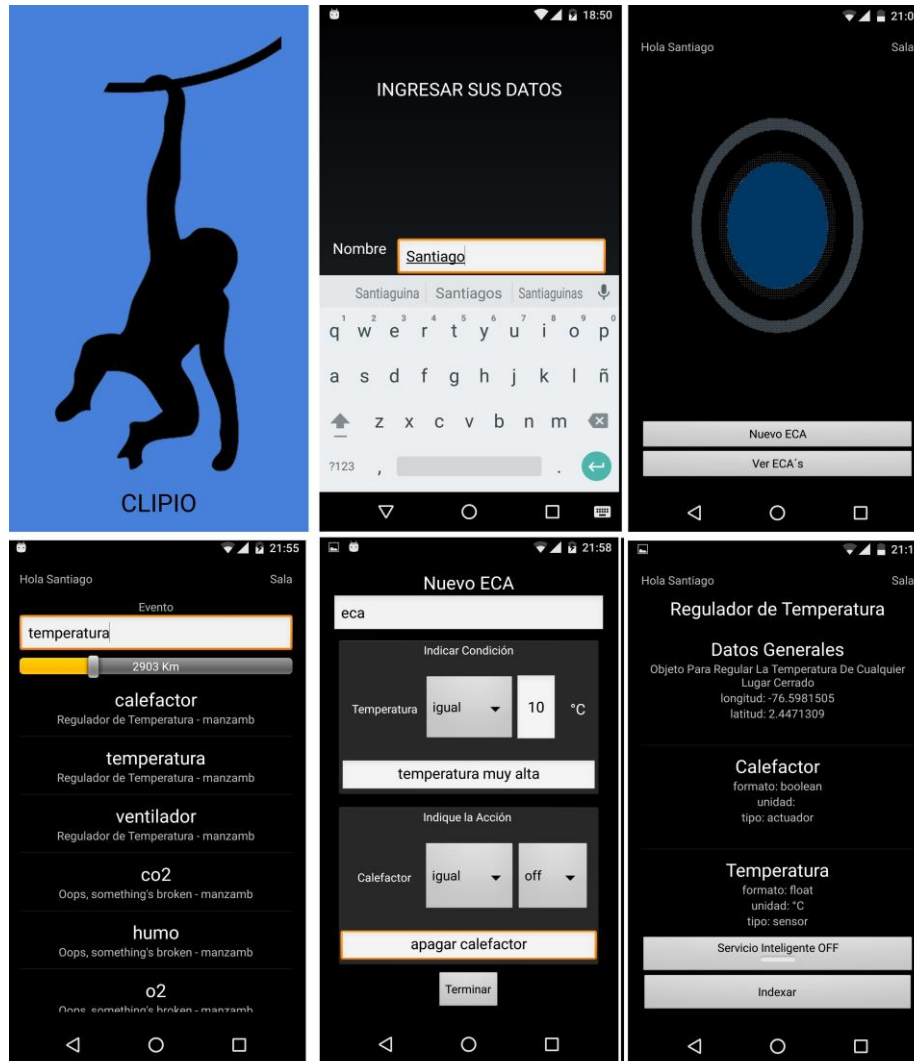
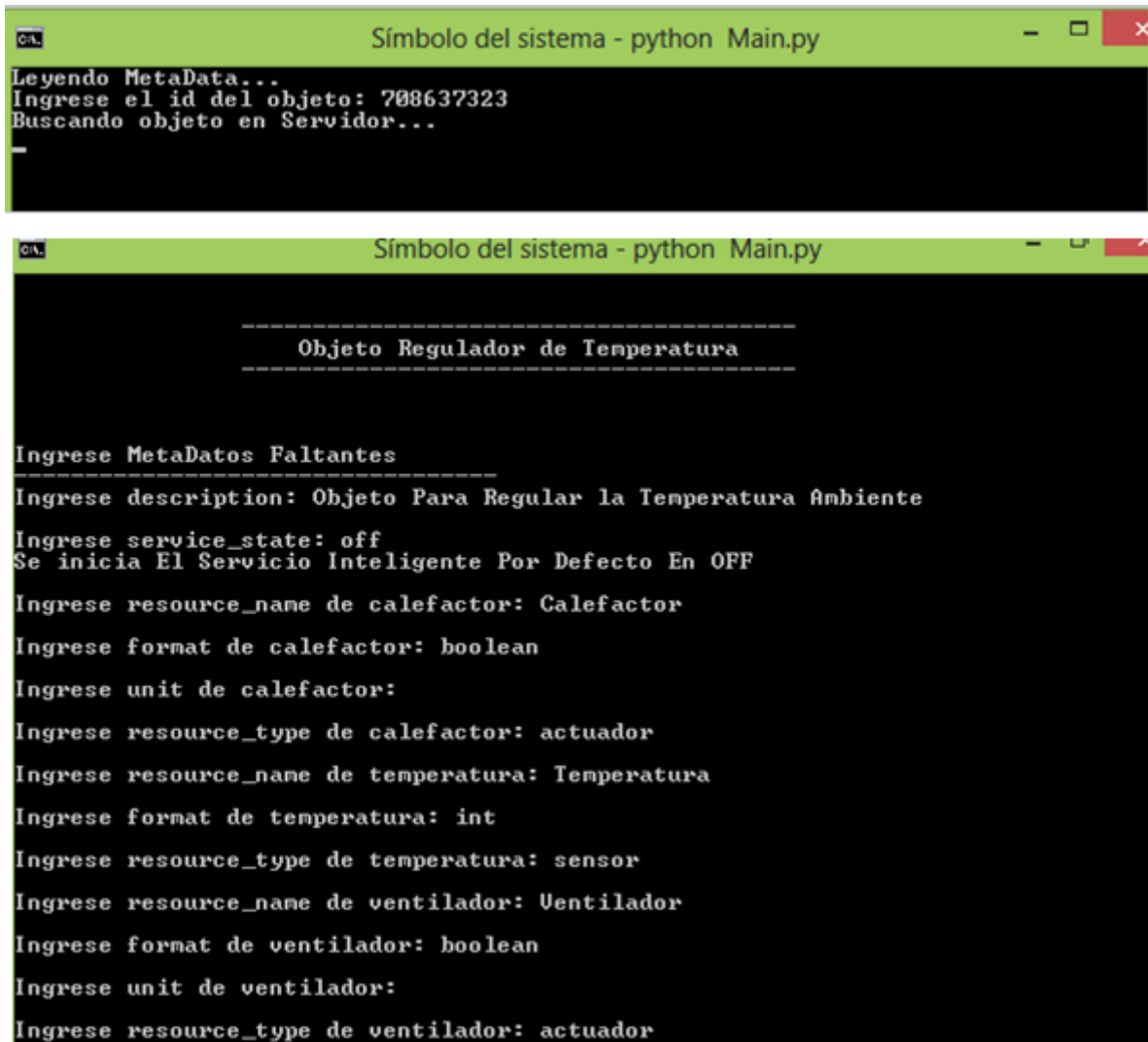


Figura 23 Interfaces de Clipio. Fuente: Propia.

La Figura 23 muestra las 6 interfaces más importantes con las cuales el usuario interactúa con Clipio. De izquierda a derecha y de arriba abajo; primero se inicia con la interfaz de presentación, donde se despliega el logo de la aplicación y su nombre, segundo la interfaz donde se crea el usuario, tercero la interfaz principal, donde el usuario puede seleccionar las distintas acciones y leer etiquetas NFC, cuarto la interfaz donde se realiza la búsqueda de objetos con los cuales deseo generar un nuevo servicio de interacción, quinto la interfaz para generar el servicio de interacción y sexto la interfaz donde se despliega la información de un objeto en particular.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS



```
Símbolo del sistema - python Main.py
Leyendo MetaData...
Ingrese el id del objeto: 708637323
Buscando objeto en Servidor...
-

-----
Objeto Regulador de Temperatura
-----

Ingrese MetaDatos Faltantes
-----
Ingrese description: Objeto Para Regular la Temperatura Ambiente
Ingrese service_state: off
Se inicia El Servicio Inteligente Por Defecto En OFF
Ingrese resource_name de calefactor: Calefactor
Ingrese format de calefactor: boolean
Ingrese unit de calefactor:
Ingrese resource_type de calefactor: actuador
Ingrese resource_name de temperatura: Temperatura
Ingrese format de temperatura: int
Ingrese resource_type de temperatura: sensor
Ingrese resource_name de ventilador: Ventilador
Ingrese format de ventilador: boolean
Ingrese unit de ventilador:
Ingrese resource_type de ventilador: actuador
```

Figura 24. Interfaces de inicializar objeto. Fuente: propia.

La Figura 24 muestra las dos interfaces que contiene la aplicación de inicializar objeto. La primera interfaz nos muestra el momento en que la aplicación pide la identificación del objeto que deseo inicializar y de forma seguida realiza la búsqueda en el servidor. La segunda interfaz muestra el momento en el que la aplicación requiere la información que el servidor no pudo recuperar sobre los metadatos el objeto.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

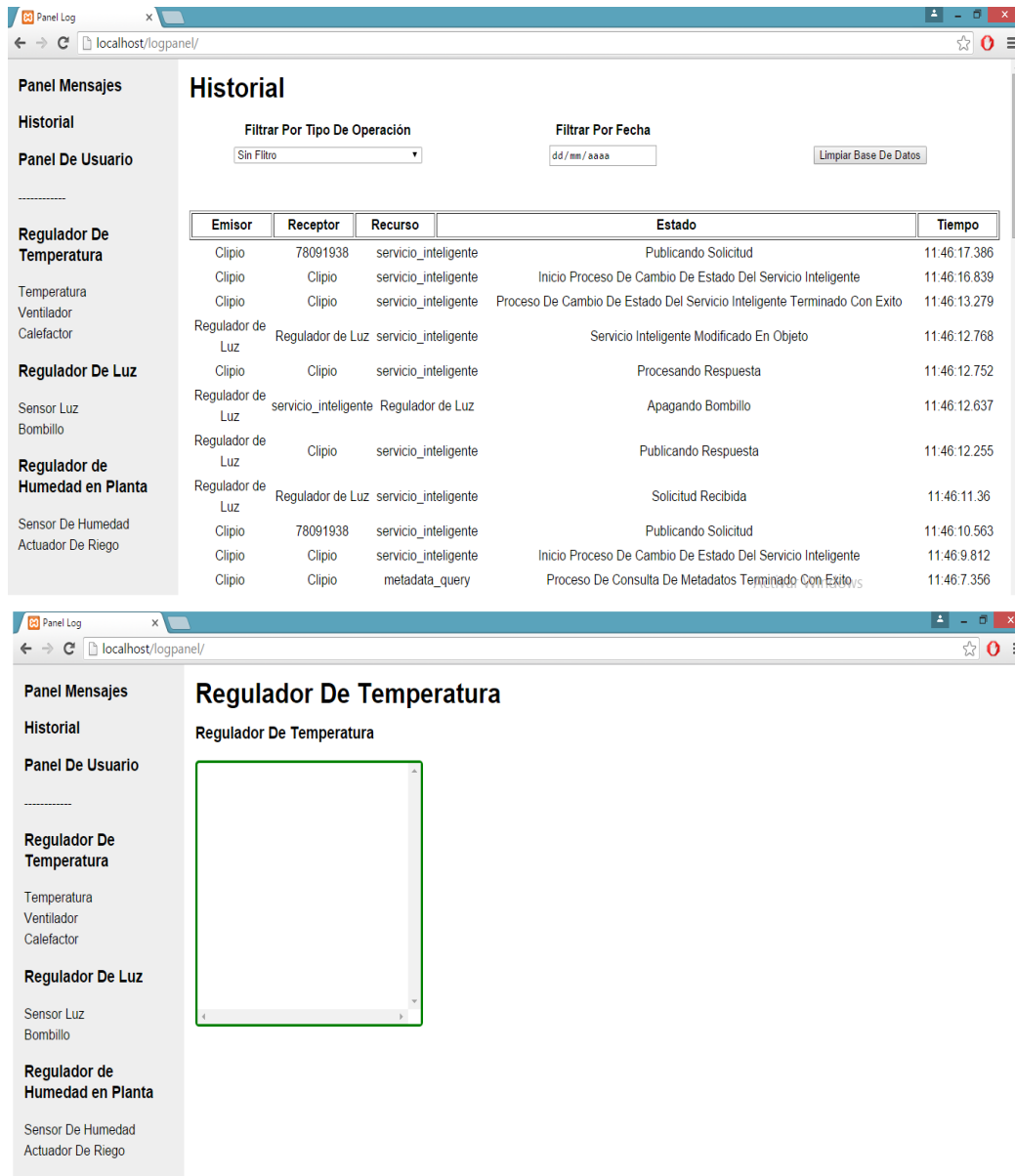


Figura 25. Interfaces de panel de usuario. Fuente: propia.

La Figura 25 muestra las interfaces más representativas de la aplicación panel de usuario. La primera interfaz muestra cómo se despliega la información de las distintas interacciones que realizan los objetos y el coordinador Clipio, la aplicación detalla un emisor, un receptor, un tipo de mensaje (recurso), el mensaje (estado) y el tiempo instantáneo que se cumple el proceso. La segunda interfaz muestra los datos que envía cada objeto en particular, en la sección de navegación se puede ingresar a cada recurso del objeto para ver el estado instantáneo del recurso.

4.3.3 Iteraciones

A continuación se mencionan los productos obtenidos en cada una de las iteraciones propuestas para el proyecto.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

A. Iteración 1

Para esta iteración se realizaron tareas enfocadas en cumplir los casos de uso almacenar objeto, recuperar objeto y visualizar objeto pertenecientes al módulo MROI.

- Se implementaron las representaciones físicas de cada objeto con sus recursos sobre las placas Galileo y se asociaron las etiquetas NFC correspondientes a cada objeto.
- Se incorporó el *middleware Xively*, ingresando la información sobre los metadatos de cada objeto y también se anotaron los conceptos para que puedan ser relacionados con un contexto.
- Se implementó la aplicación inicializar objeto y seguido a esto se inició con la recuperación de cada uno de los objetos.
- Se inició el desarrollo de la aplicación Clipio con la primera funcionalidad, la cual le permite leer etiquetas NFC, realizar la conexión con el servidor *bróker* MQTT y desplegar los metadatos de un objeto en forma de lista.

Para esta iteración no se desarrollaron pruebas alfa de funcionalidad, debido a que las aplicaciones no se encontraban en una fase estable y se requirió mayor tiempo de desarrollo.

B. Iteración 2

Para esta iteración se realizaron tareas enfocadas en cumplir los casos de uso almacenar contexto y recuperar contexto pertenecientes al módulo MCOI. Para iniciar con las pruebas de funcionalidad también se implementaron algunos casos de uso correspondientes al MIOI sobre Clipio.

- Se asoció una etiqueta NFC a la entidad sala.
- Se actualizó la ontología de dominio DogOnt, agregando el lenguaje español a los conceptos de la ontología, luego esta se incorporó en el índice semántico.
- Se desarrolló la aplicación panel de usuario, con el fin de facilitar las pruebas y el seguimiento Clipio y los objetos.
- Se continuó con el desarrollo de las aplicaciones Clipio e inicializar objeto hasta dejarlas en una fase estable. Para Clipio se implementaron los casos de uso modificar estado servicio básico, crear servicio de interacción, modificar estado de servicio de interacción, eliminar servicio de interacción y conectar objeto.

Para este punto las aplicaciones estaban en fases más estables y fue posible iniciar con la primera prueba alfa de funcionalidad. Para esta prueba se ejecutaron cada una de las funcionalidades construidas y se fue tomando nota de los tiempos de latencia. Para la prueba se evaluaron los casos de uso: visualizar objeto correspondiente al MORI y modificar estado servicio básico, crear servicio de interacción, modificar estado servicio interacción y eliminar servicio de interacción, correspondientes al MIOI. Cada uno de los casos de uso contiene diferentes procesos que se detallan en el anexo E. Los principales resultados se resumen en la Tabla 10:

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Caso de Uso	Tiempo de Respuesta (segundos)	Realimentación
Visualizar objeto	2,5	Presentar al usuario la información con una interfaz amigable.
Modificar estado servicio básico	11 (prender) 6,58 (apagar)	Presentar al usuario la información (tiempos y elementos que intervienen) del estado de la comunicación con una interfaz amigable. Tiempo muy alto.
Crear servicio de interacción	20	La búsqueda de los objetos por lenguaje natural requiere utilizar muchas palabras y expresiones para que se retorne un recurso deseado con éxito. Tiempo muy alto
Modificar estado servicio interacción	16,505 (prender) 6,928 (apagar)	El tiempo de respuesta es muy alto.
Eliminar servicio de interacción	5	El tiempo de respuesta es considerable.

Tabla 10: Resultados primera fase de evaluación de funcionalidad. Fuente: propia

Por otra parte se evaluó el comportamiento del escenario en general, obteniendo las siguientes retroalimentaciones:

- El escenario se congestionó cuando se crearon 37 servicios de interacción diferentes y dejó de funcionar.
- No se sincroniza (entre Clipio y los objetos) correctamente los estados de los servicios básicos y de interacción.
- La aplicación panel de usuario no almacena correctamente los tiempos del caso de uso modificar estado servicio de interacción.

C. Iteración 3

Para esta iteración se realizaron tareas enfocadas en cumplir los casos de uso faltantes para el MIOI. Además se desarrollaron algunos ajustes basándose en la retroalimentación de la primera prueba de funcionalidad realizada en la iteración anterior.

- d. Se desarrollaron los casos de uso en Clipio correspondientes a crear usuario y ver servicio de interacción, de esta forma queda más completa y estable la aplicación Clipio
- e. Se identificaron los procesos pesados y redundantes y se los ejecutó dentro de hilos de programación, esto para evitar los retrasos presentados debido a la espera de la finalización de otros procesos, que necesariamente no correspondían a prerrequisitos.
- f. Se modificó el panel de usuario desplegando mensajes más entendibles y se relacionó la entidad de interés con los distintos estados con el fin de disminuir la amigabilidad.
- g. Se arreglaron los problemas de sincronización de Clipio con los objetos.

Para la segunda prueba alfa de funcionalidad el objetivo fue revisar si los arreglos desarrollados han permitido mejorar la funcionalidad. A continuación se muestran los servicios de interacción creados para la prueba y se los desglosan según la sentencia ECA que cumplen. Se relaciona el recurso seleccionado en el evento y las variables ingresadas en la condición con la sentencia realizada por el recurso actuador cuando la condición se cumple.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Evento Recurso	Condición	Acción	
		Recurso	Estado
T	Temperatura > 50 °C	R	ON
T	Temperatura < 50 °C	B	ON
T	Temperatura > 10 °C	R	ON
T	Temperatura < 10 °C	R	OFF
T	Temperatura > 50 °C	R	ON
H	Humedad > 50 %	V	ON
H	Humedad < 50 %	C	ON
H	Humedad > 60 %	V	ON
H	Humedad > 60 %	V	ON
L	Luz < 30 %	C	ON
R	Riego = ON	V	ON
R	Riego = ON	C	ON
V	Ventilador = ON	R	ON

Tabla 11: Plan de pruebas de la segunda fase de pruebas del escenario. Fuente: propia

La Tabla 11, permite distinguir los siguientes elementos

- T, recurso sensor de temperatura, parte del objeto regulador de temperatura.
- L, recurso sensor de luz, parte del objeto regulador de luz.
- H, recurso sensor de humedad, parte del objeto regulador de humedad en planta.
- R, recurso actuador de riego, parte del objeto regulador de humedad en planta.
- B, recurso actuador bombillo, parte del objeto regulador de luz
- V, recurso actuador ventilador, parte del objeto regulador de temperatura.
- C, recurso actuador calefactor, parte del objeto regulador de temperatura.

Como resultado se observó que no existe variación de tiempos para la creación de servicios de interacción entre los distintos objetos. También se observó que si un servicio de interacción tiene como evento un recurso de tipo actuador, éste debe esperar que se cumpla la condición que cambie el estado de este recurso para poder ejecutar la acción del servicio de interacción en mención. En conclusión, se puede definir una nueva naturaleza de los servicios de interacción, donde es posible concatenarlos para que generen una serie de acciones.

Los demás elementos como la amigabilidad al usuario, la recuperación de los objetos específicos y la información de tiempos mejoraron, sin embargo fue necesario revisar mejor el funcionamiento del sistema para mejorar los tiempos de respuesta.

D. Iteración 4

Esta iteración se centró en mejorar la funcionalidad a base de la retroalimentación de la segunda prueba alfa de funcionalidad entregada en la anterior iteración. Para esto se realizaron las siguientes tareas.

- Se revisa el modelo de hilos de programación desarrollados con el fin de mejorar los tiempos de respuesta.
- Se mejora los tiempos en la lectura de los metadatos y de la OOS para reducir los tiempos de respuesta.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Para la tercera prueba alfa de funcionalidad se analizaron todas las posibles combinaciones para llevar a cabo la interacción entre objetos y se crearon sus respectivos servicios de interacción (ver anexo F).

Para la primera parte de la prueba se dejan prendidos los servicios básicos correspondientes al objeto que contiene el recurso participante como evento en el servicio de interacción, y se realizan las posibles combinaciones tomando como acción los recursos de los otros dos objetos. Para la siguiente fase se crean servicios de interacción concatenados en cascada, conectando el cambio de estado de un recurso actuador de un servicio de interacción con el evento de otro servicio de interacción.

De las pruebas se observó:

- Cuando el servicio básico se encuentra prendido y se toma como evento el recurso sensor de temperatura del objeto regulador de temperatura, no se refleja un comportamiento conflictivo con los servicios de interacción, tanto así que se puede apagar el servicio básico y los servicios de interacción continúan funcionando correctamente.
- Cuando los recursos actuadores asumen el rol de evento, es necesario que exista un cambio en dicho recurso (un segundo servicio de interacción que modifique el estado del actuador) para que se ejercite el servicio de interacción, pero ende es necesario un proceso adicional para poder ejercitar un servicio de interacción con un recurso actuador como evento.
- Algunas servicios de interacción utilizan los mismos recursos pero en diferentes direcciones de la interacción (intercambiar el recurso evento por el recurso acción, generando un segundo servicio de interacción). Se observó que el comportamiento era igual para los dos servicios y se decidió no explorar estas opciones ya que la prueba total duraba entre 7 a 8 horas por una batería completa.
- Se prendió los servicios de interacción en cascada paralelamente con los servicios básicos, a lo cual el sistema empezó a presentar conflictos de la siguiente manera: se observó inicialmente un comportamiento de encendido y apagado irregular de los recursos por un tiempo, y al final conservó el estado condicionado por el servicio básico. En conclusión, cuando se deja activo el servicio básico y los servicios de interacción tratan de cambiar el estado de los recursos actuadores, el comportamiento tiende a dar prioridad al servicio básico.
- Cuando se prenden dos o más servicios de interacción que modifican el mismo recurso actuador de forma contradictoria, éste responde de forma intermitente e irregular por un tiempo, pero después se comporta de acuerdo al primer servicio de interacción prendido.
- Los errores y anomalías observados en el funcionamiento del escenario se deben principalmente a la congestión de la red, al compartir la red con diferentes usuarios, la tasa de pérdidas de comunicación es considerable.
- Otra fuente de error fue de carácter humano, cuando el usuario olvidaba apagar servicios de interacción o servicios básicos y prende nuevos servicios, es posible

encontrar comportamientos conflictivos. Se observó que es preciso mejorar la interfaz de Clipio que permite modificar el estado de los servicios de interacción y los básicos.

- Se observó que el sistema presenta inestabilidades en ciertos momentos y algunos de los errores y anomalías no son informados correctamente al usuario para que realice las respectivas correcciones.

4.4 FASE DE TRANSICIÓN

Para esta fase se cuenta con un producto el cual es entregado con el código fuente del proyecto, contiene principalmente el código de cada uno de los objetos implementados en el escenario, así como el código de la aplicación Clipio y el panel de usuario, los manuales de las aplicaciones Clipio e inicializar objeto se encuentran en el anexo G, el manual de usuario de panel de usuario se descarta debido a que esta interfaz no tiene muchas funcionalidades y los procesos que el usuario final debe realizar en esta aplicación son pasivos y muy pocos. Las pruebas realizadas a este prototipo están contenidas en el capítulo 5. Para esta fase no se realizaron pruebas beta debido a que las aplicaciones desarrolladas se considera que no tienen la madurez necesaria para ser presentada a un usuario no técnico, por lo tanto se descartan este tipo de pruebas. Un nuevo grupo de pruebas se proponen, se realizan y se analizan en el siguiente capítulo.

5 EVALUACIÓN DEL ESCENARIO

Para evaluar el escenario lo primero que se hizo fue establecer que indicadores de proyectos similares se podía utilizar para realizar las mediciones y posteriormente hacer la evaluación del mismo.

No se hace la evaluación de todos los casos de uso presentados en el proyecto debido a que el proyecto está interesado más en la interacción semántica entre objetos, por lo tanto se evita realizar la evaluación con los siguientes casos de uso:

- Almacenar objeto (MROI), este caso de uso se realiza directamente en la interfaz del servidor *middleware Xively* y el fin de este proyecto no es de medir indicadores de herramientas externas.
- Recuperar objeto (MROI), por una parte los procesos son ejecutados manualmente por el usuario, por lo que los indicadores definirían aspectos relacionados con la interacción humano a máquina y no interacción entre objetos. Por otro lado, se utiliza el índice semántico como herramienta para recuperar la información del objeto, sin embargo todos los indicadores relacionados con el índice están resueltos por MISWoT.
- Almacenar contexto y recuperar contexto (MCOI), aunque se utiliza el índice semántico como fuente de descubrimiento y contextualización, los indicadores utilizados en la recuperación de la información están diseñados para medir la relevancia de la información buscada y no la interacción semántica. Este módulo se evaluó en el proyecto macro correspondiente al MISWoT.
- Ver servicio de interacción (MIOI), los procesos de este caso de uso solo repercuten en Clipio, haciendo que los indicadores puedan evaluar solo aspectos internos del dispositivo móvil y la aplicación Clipio y no la interacción entre dos elementos distintos del escenario.
- Conectar objeto (MIOI), los procesos de este caso de uso solo repercuten en el objeto que realiza la acción de la sentencia ECA, haciendo que los indicadores puedan evaluar solo aspectos internos del objeto y no la interacción con otros.

A continuación se presentan los indicadores tomados y la evaluación realizada.

5.1 REFERENTES DE EVALUACIÓN

Aunque una de las medidas más utilizadas para evaluar escenarios y demás implementaciones son los indicadores de la calidad del servicio, se decide no manejarlos ya que el objetivo de medición es la interacción entre los objetos inteligentes y no la calidad de los distintos aspectos que puede tener el escenario. Sin embargo el escenario retoma las siguientes medidas de calidad del servicio “*Quality of Service*” – QoS:

- **Duplicación de tareas** [52]. En tareas críticas como el envío de mensajes MQTT al servidor, se replican los procesos dedicados a publicar los mensajes por lo menos tres veces, esto con el fin de asegurar la llegada del paquete al *bróker* y su replicación en

los respectivos suscriptores, y de esta forma aumentar la tolerancia a fallos del escenario.

- **Enfoque en hilos de programación** [53]. Los elementos del escenario han sido programados pensando en sistemas de multi-hilos, buscando organizar y sincronizar diferentes procesos redundantes y pesados en hilos separados, para mejorar los tiempos de respuesta del escenario.
- **Interactividad, retardo y criticalidad** [54]. La interactividad corresponde a las consultas que realiza el usuario al índice semántico para detectar los objetos y sus recursos. El retardo es un indicador que se mide en los experimentos bajo el nombre de latencia de interacción, con él se puede establecer una calidad de servicio de interacción. Finalmente la criticalidad se tuvo en cuenta al momento de enviar los estados de cada recurso al servidor MQTT, debido a que estos procesos son críticos en el escenario.

Para la evaluación del escenario se decidió adaptar el indicador de latencia a los mensajes enviados y recibidos por los dispositivos cuando están interactuando. Adicionalmente hay que tener en cuenta que el escenario se implementó utilizando un punto de acceso a internet que maneja velocidades de 9.74 Mbps para carga y 17.9 Mbps para carga, además la red es compartida con aproximadamente otros 70 usuarios.

5.1.1 Latencia de interacción

La latencia se puede entender como el tiempo transcurrido desde que un elemento hace una consulta o envía un mensaje, hasta que los resultados son retornados.

En el escenario propuesto es conveniente discriminar los tiempos que se suman en los diferentes procesos, por ello este indicador es discriminado en esos tiempos tal como se presenta en la siguiente ecuación. Así es posible establecer en donde se pueden producir la mayor cantidad de latencia en el funcionamiento del sistema.

$$\text{Latencia Interacción} = Tr - Ts$$

Dónde:

- T_s (tiempo de solicitud), tiempo en el cual se inicia una solicitud. Ejemplo, cuando un objeto envía una solicitud de consulta de sus metadatos al índice.
- T_r (Tiempo de retorno), tiempo en el cual se recibe alguna respuesta. Ejemplo cuando el objeto recibe la información sobre sus metadatos completa o cuando recibe un mensaje de fallo del sistema.

5.1.2 QoS en interacción

Teniendo en cuenta los tiempos de latencia de interacción, se puede medir una calidad de servicio de interacción. Los resultados de las operaciones sobre el escenario deben ser en tiempos no mayores a 1 segundo, para que sean clasificados como tiempo real de acuerdo

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

con el proyecto [55]. Se toma este tiempo como referencia para los experimentos y se define la Tabla 12 para su interpretación cualitativa con respecto al QoS en interacción:

Rango latencia interacción	QoS interacción	Descripción
Menor a 0,799 ds.	Excelente	El mejor para la percepción de los usuarios
0,8 – 0,99 ds.	Bueno	Aceptable para la percepción de los usuarios.
Mayor a 1 s.	Deficiente	Inaceptable y se pierde interés por parte del usuario.

Tabla 12: Cálculo de QoS de interacción relacionado a la latencia de interacción. ds: Décimas de segundo, s:segundo. Fuente: propia

5.2 ANÁLISIS DE LATENCIA EN LOS EXPERIMENTOS

Los tiempos de respuesta mejoraron considerablemente conforme se evolucionaba en las pruebas alfa de funcionalidad, por lo tanto se procedió a realizar un nuevo grupo de pruebas para detectar los retardos (latencias) con tiempos precisos y con un análisis adecuado. Los pasos realizados para el análisis fueron:

- Las mediciones de los tiempos de latencia se aplicaron únicamente a los casos de uso y no a sus procesos contenidos.
- Los tiempos corresponden a la latencia exacta desde que sale el mensaje de inicio hasta que llega la confirmación con el mensaje de finalización. Los tiempos percibidos pueden ser un poco mayores por demora en la recomposición de las interfaces.
- Se recogió una tabla que guarda todo el trabajo de mensajería entre los objetos del escenario y Clipio, se los organizo según al caso de uso al que pertenecen y finalmente se realiza un proceso de análisis de tiempos de latencia.
- Se realizaron 12 pruebas durante 12 días diferentes, una por día.
- Las pruebas se basaron en la pila de pruebas propuesta en la iteración 4 que se encuentra en el anexo F, estas pruebas fueron diseñadas teniendo en cuenta que el escenario es sensible a conflictos, que se pueden presentar cuando se accionan dos o más servicios de interacción que ejecutan un mismo recurso actuador con acciones opuestas (Ejemplo: servicio interacción 1, tiene como acción, prender calefactor y servicio de interacción 2 tiene como acción, apagar calefactor).
- No todas las pruebas se realizaron ejecutando los mismos servicios de interacción, ya que cada prueba buscaba probar una funcionalidad determinada, sin embargo se promediaron los tiempos resolviendo los problemas de las diferencias en cantidad de registros recopilados.
- Dado que en algunos momentos las mediciones individuales tenían tiempos extremos con respecto a la función normal promedio, se eliminaron los extremos para evitar distorsiones. Se almacenaron los tiempos eliminados por aparte con el fin de determinar qué es lo que sucedía en esos puntos.

5.2.1 Análisis de los casos de uso

Se recopilaron un total de 57824 mensajes en cada uno de los casos de uso ejecutados en el escenario. Los tiempos se midieron en décimas de segundos, los resultados se agrupan según el caso de uso a evaluar.

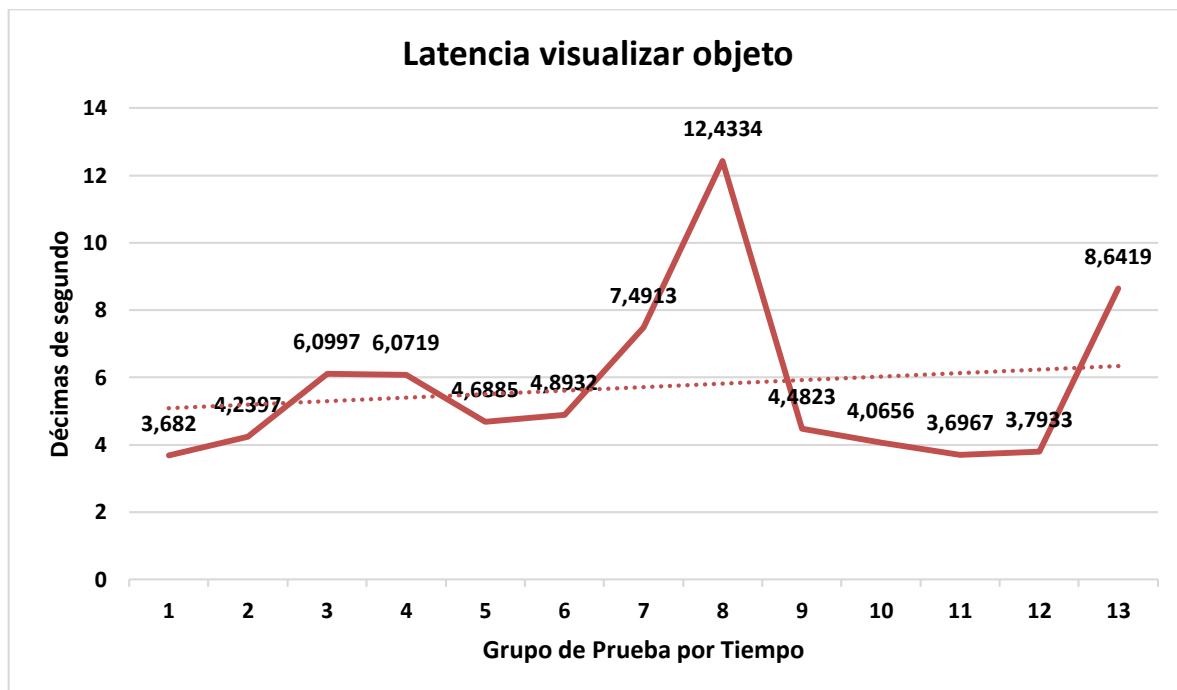
ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- **Latencia en visualizar objeto**

Los tiempos reportados en la Tabla 13 presentan una latencia promedio de 5,71 décimas de segundo calificándolo como con una calidad de servicio excelente de acuerdo a la Tabla 12. Sin embargo se puede notar en la Figura 26 dos picos importantes, esto se debe tanto a errores humanos como a fallas en elementos externos del sistema, como desconexiones momentáneas de internet. Sin embargo la tendencia promedio muestra que existe una estabilidad pero cierta inclinación en aumentar los tiempos, esto puede deberse al estrés que sufren los componentes físicos del escenario al ser utilizados constantemente.

Fecha	Grupo de prueba	Promedio (ds)
21/11/2015	1	3,682
23/11/2015	2	4,2397
24/11/2015	3	6,0997
25/11/2015	4	6,0719
26/11/2015	5	4,6885
27/11/2015	6	4,8932
28/11/2015	7	7,4913
11/12/2015	8	12,4334
12/12/2015	9	4,4823
13/11/2015	10	4,0656
14/11/2015	11	3,6967
15/11/2015	12	3,7933
Total		5,713807692

Tabla 13: Latencia promedio de visualizar objeto. Fuente: propia



ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Figura 26. Tendencia latencia promedio en visualizar objeto. Fuente: propia

- **Latencia en modificar estado de servicio básico**

Los tiempos reportados en la Tabla 14 presentan una latencia promedio de 4,98 décimas de segundo calificándolo como una calidad excelente de acuerdo a la Tabla 12. A diferencia del caso de uso visualizar objeto, en la Figura 27 se puede ver que la latencia se mantiene casi constante, indicando una tendencia estable y asegurando que el usuario siempre podrá, en la mayoría de los casos, percibir un comportamiento adecuado al modificar el estado de un servicio básico. Sin embargo también se puede notar un pico importante, estos picos serán analizados con mayor detenimiento al finalizar el presente apartado

Fecha	Grupo de prueba	Promedio (ds)
21/11/2015	1	3,5877
23/11/2015	2	4,4095
24/11/2015	3	4,34
25/11/2015	4	3,874
26/11/2015	5	7,6147
27/11/2015	6	3,4811
28/11/2015	7	4,3111
11/12/2015	8	11,0786
12/12/2015	9	4,3405
13/11/2015	10	4,333
14/11/2015	11	3,8576
15/11/2015	12	3,8491
Total		4,984707692

Tabla 14: Latencia promedio en modificar estado de servicio básico. Fuente: propia

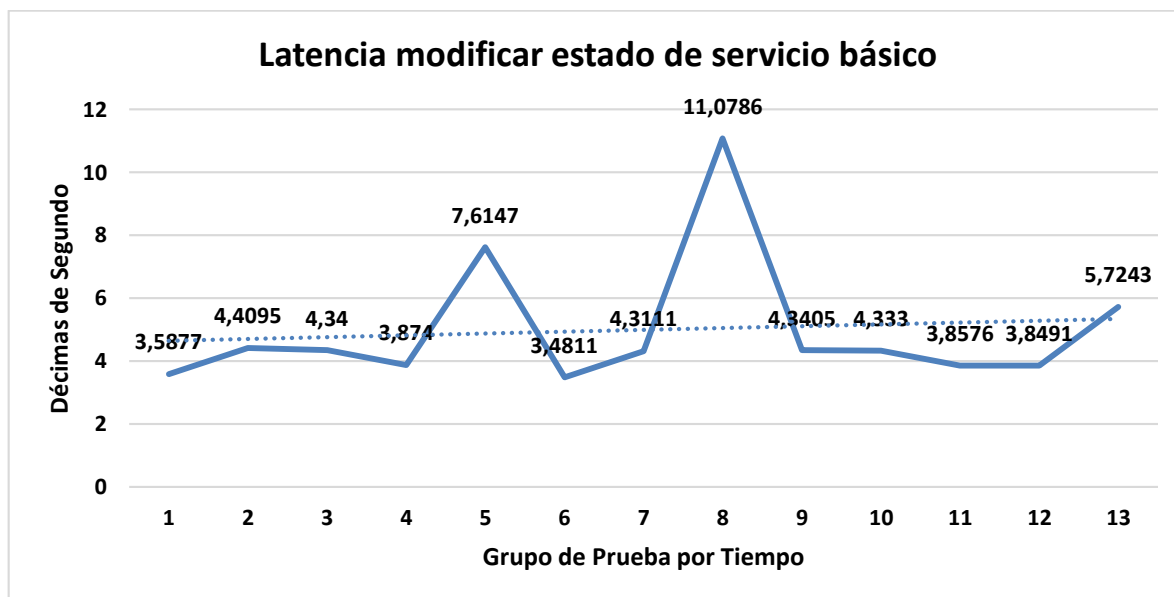


Figura 27. Tendencia latencia promedio en modificar estado de servicio básico. Fuente: propia

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- **Latencia en crear servicio de interacción**

Los tiempos reportados en la Tabla 15 presentan una latencia promedio de 7,68 décimas de segundo, calificándolo como una calidad de servicio excelente de acuerdo a la Tabla 12. En la Figura 28 no se nota una clara tendencia al incremento del tiempo de latencia, sin embargo todos los tiempos fueron considerables, encontrando otra vez un pico importante. Esto nos dice que es necesario afinar más los procesos de este caso de uso para mejorar los tiempos. Por otra parte este caso de uso se considera el que mayor peso de procesamiento tiene, por lo que un resultado cercano a lo bueno y con una clara tendencia estable, resulta relativamente una calificación aceptable.

Fecha	Grupo de prueba	Promedio (ds)
21/11/2015	1	4,3353
23/11/2015	2	5,527
24/11/2015	3	5,3641
25/11/2015	4	9,4598
26/11/2015	5	7,2823
27/11/2015	6	8,7503
28/11/2015	7	17,4728
11/12/2015	8	7,418
12/12/2015	9	6,9603
13/11/2015	10	5,8541
14/11/2015	11	6,0753
15/11/2015	12	7,6819
Total		7,681766667

Tabla 15: Latencia promedio en crear servicio de interacción. Fuente: propia

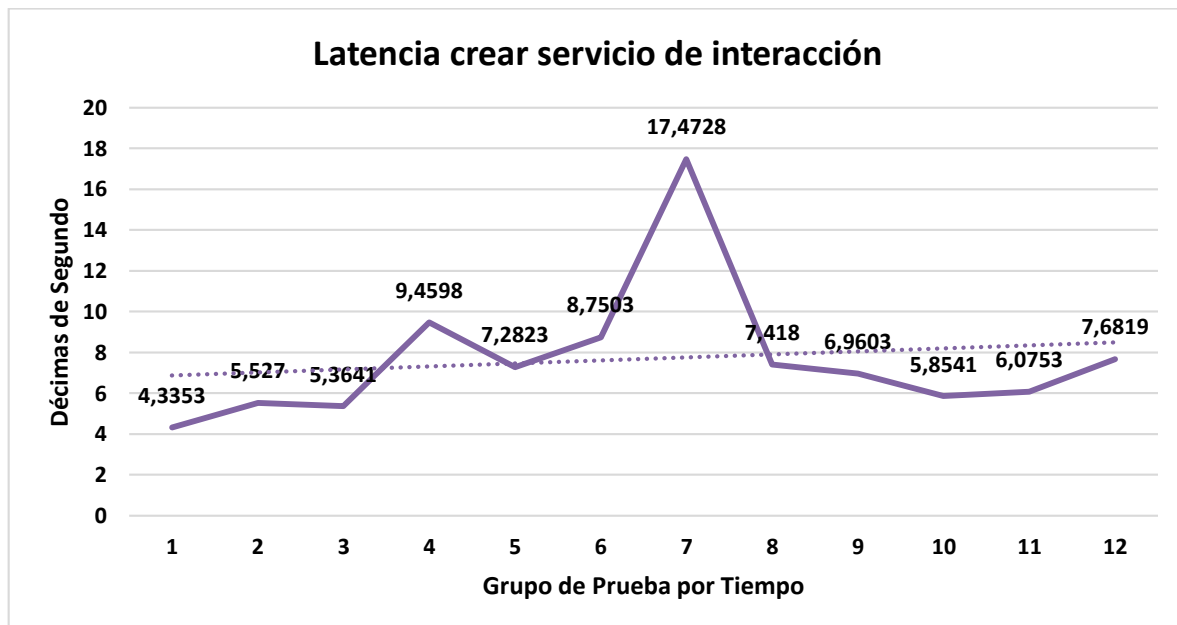


Figura 28. Tendencia latencia promedio en crear servicio de interacción. Fuente: propia

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- **Latencia en modificar estado servicio de interacción**

Los tiempos reportados en la Tabla 16 presentan una latencia promedio de 5,01 décimas de segundo calificándolo como una calidad de servicio excelente, de acuerdo a la Tabla 12. En la Figura 29 se puede observar que la tendencia en la latencia se mantiene durante todo el experimento. Cabe resaltar que en este caso de uso no aparecen picos que deban considerarse.

Fecha	Grupo de prueba	Promedio (ds)
21/11/2015	1	4,0408
23/11/2015	2	5,2597
25/11/2015	3	4,5004
26/11/2015	4	4,7313
27/11/2015	5	4,6947
28/11/2015	6	3,5054
11/12/2015	7	5,3239
12/12/2015	8	5,7637
13/11/2015	9	6,0104
14/11/2015	10	5,1773
15/11/2015	11	5,4822
16/11/2015	12	5,6743
Total		5,013675

Tabla 16: Latencia promedio en modificar estado servicio de interacción. Fuente: propia

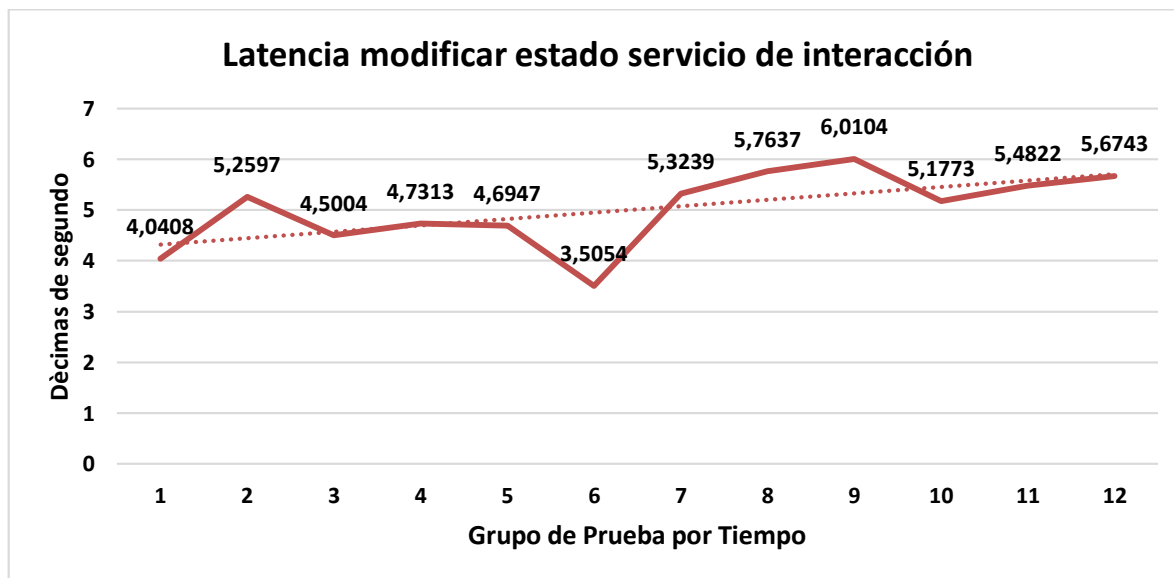


Figura 29 Tendencia latencia promedio en modificar estado servicio de interacción. Fuente: propia

- **Latencia en eliminar servicio de interacción**

Los tiempos reportados en la Tabla 17 presentan una latencia promedio de 3,82 décimas de segundo calificándolo como una calidad de servicio excelente, de acuerdo a la Tabla 12.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

En la Figura 30 se puede ver que la tendencia en la latencia se mantiene durante todo el experimento y con un promedio muy bueno, esto puede deberse a que la eliminación del servicio de interacción es un proceso menos costoso en cuanto a utilización de recursos de los elementos que intervienen en los procesos. Sin embargo se encontró nuevamente un pico considerable

Fecha	Grupo de prueba	Promedio (ds)
21/11/2015	1	2,4015
23/11/2015	2	2,585
25/11/2015	3	7,5902
26/11/2015	4	3,602
27/11/2015	5	3,412
28/11/2015	6	3,829
11/12/2015	7	3,6467
12/12/2015	8	3,7912
13/11/2015	9	4,3247
14/11/2015	10	3,331
15/11/2015	11	3,6658
16/11/2015	12	3,7672
Total		3,828845455

Tabla 17: Latencia promedio en eliminar servicio de interacción. Fuente: propia

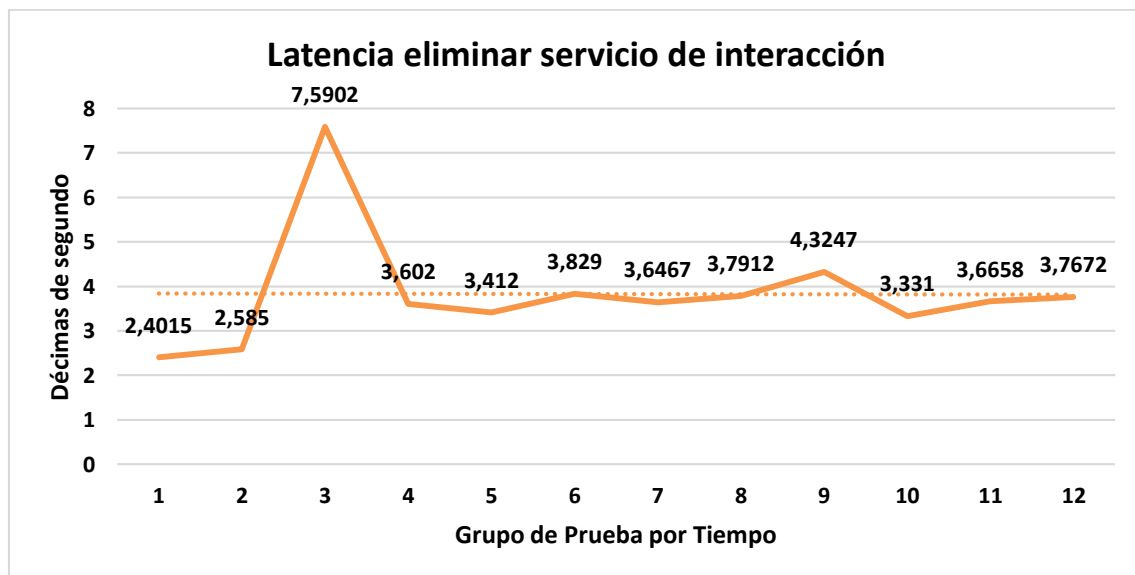


Figura 30. Tendencia latencia promedio en eliminar servicio de interacción. Fuente: propia

- **Comparación latencia los casos de uso**

Los tiempos reportados en la Tabla 18 presentan una latencia promedio de 5,44 décimas de segundo calificándolo como una calidad excelente, de acuerdo a la Tabla 12. La

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

tendencia de la mayoría de los casos de uso es a un comportamiento de calidad del servicio entre excelente y bueno, sin embargo el proceso de creación de servicio de interacción es el que presenta el mayor tiempo de latencia en interacción, tal como se comento puede deberse al algoritmo de lectura y escritura de metadatos y a que el mismo, debe ser replicado en todos los dispositivos. Por otra parte todos los casos se presentan picos considerables de aumento en el tiempo, esto puede deberse a errores humanos al momento de implementar la prueba, es posible que se accionen muchos servicios de interacción sin tener el cuidado de apagarlos posteriormente, generando conflictos y retrasando la respuesta del sistema. Aunque en estas pruebas se ha llevado el sistema a sobrecargas, no se espera que se maneje muchos servicios de interacción prendidos al mismo tiempo.

Caso de uso	Promedio
Eliminar servicio de interacción	3,828845455
Crear servicio de interacción	7,681766667
Visualizar objeto	5,713807692
Modificar estado servicio básico	4,984707692
Modificar estado servicio de interacción	5,013675
Total	5,444560501

Tabla 18: Comparación de latencias en los casos de uso. Fuente: propia

5.3 CONCLUSIONES

Después de analizar los diferentes resultados se puede concluir lo siguiente:

- El escenario está limitado a las relaciones semánticas que establezca el usuario y a partir de la lógica de las mismas se puede tener un comportamiento cooperativo de los objetos inteligentes, lo que permite automatizar procesos de manera inteligente o si están mal definidas un caos en la ejecución de sus tareas, por tanto es importante que el usuario este consciente de los efectos de las relaciones que defina.
- Las herramientas desarrolladas y el montaje del escenario requiere la intervención experta, lo cual lo hace poco accesible a las personas en general, pero se espera poder mejorar este aspecto con futuras implementaciones ya que la plataforma se ha dejado lo más genérica posible.
- Por ahora el escenario no utiliza las ontologías para generar razonamientos complejos o tomar decisiones autónomas, las usa para identificar contexto, relaciones y metadatos en la fase de descubrimiento y creación del objeto inteligente, pero se espera en una próxima versión del proyecto incluir esta característica para que la intervención del usuario sea cada vez menos presente y sus relaciones con los objetos más intuitiva.

6 CUMPLIMIENTO DE OBJETIVOS

Para el cumplimiento de los objetivos se presenta un modelo de indicadores que permite evaluar de manera objetiva el cumplimiento de los mismos. A continuación se exponen los lineamientos de conformación e interpretación de los indicadores propuestos.

6.1 LINEAMIENTOS DE CONFORMACIÓN E INTERPRETACIÓN DE LOS INDICADORES

Con el fin de expresar los resultados finales de cada uno de los objetivos se presenta a continuación una explicación sencilla de los tipos de indicadores utilizados en la evaluación de los resultados y la forma correcta de interpretarlos.

Los indicadores de desempeño que se evalúan, básicamente adoptan la forma de un cociente, en el cual, el denominador es un valor numérico que ayuda a efectuar la comparación con el logro obtenido así:

$$\text{Indicador} = \left(\frac{\text{Numerador}}{\text{Denominador}} \right) * \text{FactorEscala}$$

De esta forma se definen los siguientes modelos de indicadores que se deben personalizar y aplicar a los actores, productos, funciones, etc. dependiendo del contexto del objetivo evaluado:

- **Indicador de Eficiencia.** Permite identificar la relación que existe entre las metas alcanzadas, el tiempo y los recursos consumidos con respecto a un estándar. Representa el buen uso de los recursos.

$$\text{Eficiencia} = \left(\frac{\text{Metas alcanzadas}}{\text{Recursos Consumidos}} \right) * 100$$

- **Indicador de Calidad.** Están orientados a medir la satisfacción de los beneficiarios. La calificación está discriminada en: 1: mala (0%), 2: regular (50%), 3: buena (75%) y 4: excelente (100%)

Con el modelo de indicadores aquí presentado, se desarrolló un conjunto de indicadores que permiten evaluar adecuadamente el nivel de cumplimiento de cada uno de los objetivos. A continuación se presenta la evaluación realizada.

6.2 DESCRIPCIÓN Y ALCANCE DEL CUMPLIMIENTO DE LOS OBJETIVOS

A continuación se especifican, de arriba hacia abajo, los objetivos comprometidos en el proyecto, los productos esperados derivados de cada objetivo, los resultados obtenidos, los indicadores que evalúan el objetivo, los medios de verificación de los resultados y

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

finalmente, unas observaciones que permiten aclarar los resultados en cada objetivo. Se desarrolla una tabla por cada objetivo específico

Objetivo	1
Descripción	Desarrollar un conjunto de módulos software que permitan ejecutar mecanismos base de interacción semántica entre objetos de la WoT, siguiendo los conceptos y estructuras definidas en el MISWoT.
Productos	<ul style="list-style-type: none"> a. Determinar las características técnicas que debe tener un objeto inteligente. b. Definir un mecanismo adecuado de gestión de la OOS. c. Describir un escenario de comunicación entre objetos inteligentes de la WoT, de tal forma que permita la creación de nuevos servicios derivados de su interacción.
Resultados	<ul style="list-style-type: none"> a. En el marco teórico del presente documento se definen las características que según [14] un objeto inteligente debe tener. Además se definieron y se conceptualizaron características adicionales y las relaciones que un objeto inteligente debe tener, esto puede ser consultado en el capítulo 4. b. El caso de uso almacenar objeto perteneciente al MCOI del capítulo 4 definido en el presente documento, muestra como la OOS es gestionada, realizando sobre ella consultas SPARQL (las consultas pueden ser consultadas con más detalle en el anexo H) para generar los metadatos del objeto. Además la capa de objeto semántico de la arquitectura definida en el mismo capítulo muestra las relaciones que tiene la OOS con los diferentes elementos del escenario. c. La arquitectura definida en el capítulo 4 en función de su estructura estática y dinámica y el MIOI definido en el capítulo de implementación del escenario, describen la parte del escenario que permite realizar comunicación entre los objetos para crear servicios de interacción. Los módulos MCOI y el MROI definidos en el mismo capítulo, soportan las funciones principales y los prerequisites que deben presentar los elementos del escenario para permitir la creación de nuevos servicios de interacción.
Indicadores	$Eficiencia = \frac{ProductosObtenidos}{ProductosAObtener} = \frac{3}{3} * 100 = 100\%$ <ul style="list-style-type: none"> a. Las características definidas para soportar un objeto inteligente funcionaron de forma óptima y transparente en el escenario. $Calidad1 = 4 = 100\%$ b. La OOS no es gestionada totalmente en el escenario propuesto ya que no es utilizada para generar razonamientos complejos o tomar decisiones autónomas. El escenario la utiliza para identificar los metadatos en la fase de descubrimiento y creación del objeto. $Calidad2 = 3 = 75\%$ c. Utilizando la arquitectura es posible crear un escenario para que los objetos se comuniquen y creen servicios de interacción. $Calidad3 = 4 = 100\%$

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

Estrategias, problemas y/o observaciones	<p>Se creó un marco teórico para condensar los conocimientos sobre los distintos conceptos y referentes que se deben tener en cuenta para crear los elementos que componen al escenario. En base a estos conceptos y al MISWoT se logra proponer una arquitectura que soporta el escenario de interacción semántica entre objetos inteligentes, teniendo como núcleos la capa de objetos, donde se definen los conceptos, características, relaciones y funcionalidades que deben tener los objetos; y la capa de objeto semántico, que define los elementos, sus características y funcionalidades básicas necesarias para soportar el motor semántico de cada objeto. También se definen las otras capas, no menos importantes, que funcionan como intermediarios que manipulan la información que es entregada y requerida por los objetos, permitiéndoles interactuar por medio de nuevos servicios (servicios de interacción). En base a la arquitectura se definieron 3 módulos (MCOI, MROI y MIOI) que contienen la lógica de los mecanismos que el escenario efectúa para realizar la interacción entre los objetos.</p> <p>Aunque el proyecto se centra principalmente en la interacción entre objetos, en el transcurso del proyecto se vio relevante la incorporación del usuario humano como elemento importante en el escenario, por lo tanto la arquitectura cuenta con una capa de aplicación en la cual se desarrollaron aplicaciones que sirven como interfaz para que el usuario pueda administrar el escenario.</p> <p>Los módulos definen únicamente un solo caso de uso que ejecuta directamente la OOS enfocado en el almacenamiento de información, por lo tanto no se utiliza toda la potencialidad de una ontología como el generar razonamientos complejos o tomar decisiones autónomas, pero se espera que próximas versiones incluyan otras características a forma de casos de uso, para que así la intervención del usuario sea cada vez menos presente y las relaciones entre objetos más intuitiva.</p>
---	---

Tabla 19 Cumplimiento del primer objetivo específico. Fuente: propia

Objetivo	2
Descripción	Construir un escenario de interacción semántica de objetos inteligentes de la WoT creando un conjunto de objetos inteligentes y sus servicios, para un caso de estudio particular que será seleccionado de acuerdo con los recursos existentes.
Productos	<ul style="list-style-type: none"> a. Seleccionar el caso de estudio de acuerdo a los recursos existentes. b. Crear objetos inteligentes y sus servicios (servicios básicos). c. Construir escenario de interacción semántica de objetos inteligentes
Resultados	<ul style="list-style-type: none"> a. Apoyado por los recursos hardware prestados por la Universidad del Cauca y por las herramientas propuestas por el MISWoT, el proyecto desarrolla el caso de estudio dentro de un ambiente inteligente doméstica enfatizando en el lugar o ubicación sala, el cual es simulado por la oficina 422 de la facultad de ingeniería electrónica y de telecomunicaciones. b. Se tienen 3 objetos bien definidos: regulador de temperatura, regulador de luz y regulador de humedad en planta. El capítulo 4 describe las características de cada objeto y sus recursos mientras que el capítulo 3 muestra sus relaciones con otros elementos del escenario. Cada objeto

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

	<p>cuenta con un servicio básico que hace uso de sus recursos para darle un funcionamiento normal al objeto (Ejemplo, la función básica del regulador de temperatura es mantener una temperatura constante), además cuenta con la funcionalidad de modificar el estado del servicio básico (prendido o apagado) por medio de Clipio.</p> <p>c. El capítulo 3 define el modelado que soporta la lógica que los elementos del escenario utilizan para crear, modificar estado, ver y eliminar servicios de interacción, además muestra otros casos de uso que se determinaron como necesarios para administrar los objetos del escenario. Por otra parte el código fuente de los objetos, el elemento coordinador (Clipio) y el panel de usuario es entregado y puede ser utilizado para replicar o escalar el escenario.</p>
Indicadores	$Eficiencia = \frac{ProductosObtenidos}{ProductosAObtener} = \frac{3}{3} * 100 = 100\%$ <p>a. El caso de estudio seleccionado respondió de forma exitosa al momento de implementar el escenario y realizar los experimentos. Los recursos brindados para el proyecto fueron suficientes y abastecieron las necesidades hardware y software para la implementación del escenario.</p> <p style="text-align: center;">$Calidad1 = 4 = 100\%$</p> <p>b. Al inicio del proyecto se propone un cuarto objeto, regulador de calidad de aire, sin embargo por falta de tiempo para el desarrollo y por cuestiones de facilitar la evaluación del escenario, se sustrajo de la propuesta dicho objeto. Los tres objetos que se manejaron, y su recursos, fueron implementados exitosamente, y el funcionamiento durante el ejercicio del proyecto fue aceptable. Los servicios básicos de cada objeto fueron implementados exitosamente y cumplen a cabalidad cada una de sus funciones.</p> <p style="text-align: center;">$Calidad2 = 3 = 75\%$</p> <p>c. Se crearon los elementos del escenario como los objetos, Clipio y el panel de usuario con los requerimientos que el escenario necesitaba para funcionar correctamente. Las funcionalidades de los elementos se ejecutan correctamente, sin embargo se hace necesario implementar refinamientos en las aplicaciones, enfocados a crear interfaces para el usuario más amables y con mayores notificaciones.</p> <p style="text-align: center;">$Calidad3 = 4 = 100\%$</p>
Estrategias, problemas y/o observaciones	<p>Para la fase de evaluación del escenario se implementaron potenciómetros que permitían simular los posibles valores entregados por los recursos sensores, además los actuadores se remplazaron con leds, esto con el fin de facilitar y agilizar el cambio de estado de los recursos sensores. Sin embargo al final se remplazaron los potenciómetros y actuadores led por los recursos reales que se proponen en el escenario.</p> <p>Para los servicios básicos se propone la modificación del estado por medio de Clipio, facilitando la manipulación de los objetos dentro del escenario. Este caso de uso se hizo crítico al momento de encontrar la forma de ejecutar los</p>

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

	<p>servicios de interacción, evitando que el escenario sufriera problemas de conflictos en los recursos, sin embargo no se propuso la funcionalidad de establecer valores deseados al servicio de básico (Ejemplo, objeto regulador de temperatura, regular temperatura en 20°C u otro valor) y en contraste los objetos manejan un valor estático.</p> <p>Aunque las funcionalidades críticas del escenario quedaron cubiertas y bien desarrolladas, se nota falta de mecanismos que permitan al usuario común manipular efectivamente las aplicaciones. Además, hace falta definir una interfaz más amigable para manejar la inicialización de un objeto, interviniendo en el caso de uso recuperar objeto de MROI.</p>
--	---

Tabla 20. Cumplimiento del segundo objetivo específico. Fuente: propia

Objetivo	3
Descripción	Evaluar el escenario de interacción semántica de objetos inteligentes de la WoT construido en el caso de estudio particular, para establecer la efectividad de la interacción y los servicios creados entre los objetos inteligentes.
Productos	<ul style="list-style-type: none"> a. Definir plan de pruebas acordes a los indicadores que mejor definan al escenario. b. Ejecución de las pruebas definidas en el plan de pruebas. c. Analizar los resultados obtenidas en la ejecución del plan de evaluación.
Resultados	<ul style="list-style-type: none"> a. El capítulo 5 presenta una definición de los indicadores que mejor se ajustan al escenario propuesto, con estos se realiza un plan de pruebas y se ajustan las calificaciones acordes a estamentos preestablecidos en proyectos similares. b. Las pruebas se ejecutaron y se almacenó los datos necesarios para el posterior análisis. Se entrega una pila de pruebas (puede ser consultada en el anexo F) basada en la gestión de servicios de interacción, relacionando la información recopilada con la fecha de ejecución del sector de la pila. Además el capítulo 5 detalla los resultados de las pruebas en gráficas y realiza un análisis sobre los tiempos obtenidos c. El capítulo 4 presenta el análisis de cada una de las pruebas alfa realizadas para el desarrollo del escenario, además presenta la evolución y las modificaciones que sufrieron los distintos elementos del escenario conforme los resultados fueron descubiertos. Por otra parte el capítulo 5 muestra un análisis más detallado sobre la fase más estable del escenario, calificando los resultados de los tiempos de latencia en cada caso de uso que interviene en procesos de interacción.
Indicadores	$Eficiencia = \frac{ProductosObtenidos}{ProductosAObtener} = \frac{3}{3} * 100 = 100\%$ <ul style="list-style-type: none"> a. Los indicadores propuestos se ajustaron de una buena forma a las necesidades de evaluación del escenario, además las formas de calificación resultaron efectivas al momento de definir las falencias y fortalezas de los elementos del escenario. $Calidad1 = 4 = 100\%$

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

	<p>b. La pila de pruebas ejecutada resultó una herramienta de mucha utilidad al momento de organizar la evaluación del escenario, sin embargo se considera que la evaluación debe complementarse modificando cualquier variable posible en el escenario, como: utilizar distintos puntos de acceso a internet con distintas velocidades y usuarios conectados, utilizar otras herramientas hardware similares a las placas Galileo para soportar la representación física de los objetos, ejecutar las aplicaciones en distintos terminales con capacidades diferentes, entre otras.</p> <p style="text-align: center;"><i>Calidad2 = 3 = 75%</i></p> <p>c. El análisis de los resultados permite determinar los puntos débiles del escenario así como los errores no detectados en el proceso de la implementación. Por otro lado se logró determinar los aspectos faltantes y los trabajos futuros que el presente proyecto presenta.</p> <p style="text-align: center;"><i>Calidad3 = 4 = 100%</i></p>
<p>Estrategias, problemas y/o observaciones</p>	<p>Es necesaria la constante evaluación y modificación de los elementos del escenario con el fin de alcanzar mejores resultados en los indicadores propuestos, sin embargo por cuestiones de tiempo se determinó rangos de fechas limitadas para la ejecución de la pila de pruebas.</p> <p>En un punto del proyecto se detectó el problema que ocurre cuando dos servicios de interacción diferentes intentan ejecutar un mismo recurso de forma contraria (Ejemplo: servicio de interacción 1, prender calefactor y servicio de interacción 2, apagar calefactor), a este fenómeno se lo llamó ECA's conflictivos y se pudo diezmar el problema integrando el caso de uso modificar estado de servicio de interacción, donde se daba la posibilidad de apagar o prender un servicio de interacción. Sin embargo los ECA's conflictivos son un problema que no ha podido ser resuelto en su totalidad y la ejecución de los servicios que generan conflictos se deja a manos de la coherencia del usuario.</p> <p>Al momento de realizar las pruebas del escenario se notó que hacía falta en la interfaz de Clipio que muestra la lista de servicios de interacción creados, un elemento que facilite la identificación de una forma eficiente del estado del servicio de interacción. Al principio esto generó demoras en la ejecución de la prueba, sin embargo esta utilidad se desarrolló para agilizar y facilitar la evaluación.</p>

Tabla 21. Cumplimiento del tercer objetivo específico. Fuente: propia

7 CONCLUSIONES Y TRABAJO FUTURO

Este capítulo describe inicialmente las principales conclusiones del trabajo realizado a las que se llegó durante su desarrollo y posteriormente se propone los trabajos futuros.

7.1 CONCLUSIONES

- Se ha creado un marco de referencia para escenarios de interacción semántica de objetos inteligentes en la WoT, que identifica las tecnologías, aportes y brechas existentes en la interacción entre objetos. Se aportaron análisis en las tecnologías *middleware* de la IoT, arquitecturas IoT, WoT y modelamiento inteligente de objetos de la IoT que pueden ser reutilizados en otros estudios. El avance en el área es tan dinámico y veloz de tal forma que se debe actualizar éstos conceptos en periodos de tiempo muy cortos.
- Se ha propuesto un modelo arquitectónico para escenarios de interacción semántica de objetos inteligentes en la WoT, tomando como enfoque de solución, la creación de índices semánticos como se propone en MISWoT, dando la posibilidad de descubrir, buscar e interactuar los objetos inteligentes en un contexto particular. La posibilidad de reutilizar los datos de los objetos en diferentes contextos abre un campo amplio de aplicaciones para la WoT.
- El modelo arquitectónico propuesto, establece mecanismos para gestionar la información de metadatos y datos de los objetos inteligentes de la WoT soportados en ontologías de dominio y estándares internacionales para su correcto apropiamiento generalizado. El uso de estándares permitieron avanzar más rápido en las soluciones y aseguran el uso posterior de las mismas por la comunidad investigativa.
- Se implementó un escenario de interacción semántica en la WoT, con dispositivos físicos reales en los cuales se desplegó el modelo propuesto y se incorporaron todos los elementos definidos en la arquitectura. Los resultados preliminares permiten afianzar que la propuesta de un escenario de interacción en la WoT basado en índices semánticos, servicios web y de objetos semánticos, son viables para generar escenarios de interacción en la WoT. Sin embargo es necesario realizar más implementaciones para mejorar las herramientas y los conceptos aplicados.
- Para evaluar el escenario de interacción semántica en la WoT, se propuso un conjunto de indicadores preliminares para evaluar la interacción, el indicador de latencia de interacción y el indicador de calidad del servicio de interacción. Estos indicadores se fundamentan en las propuestas de calidad del servicio enfocadas explícitamente en retardos y latencias. Permiten establecer un funcionamiento adecuado a la percepción de disponibilidad de los servicios creados en tiempo real y de la posible aceptación por parte del usuario. Estos indicadores aparecen por la poca literatura al respecto de la evaluación de la interacción semántica en la WoT, sin embargo es necesario definir unos indicadores que midan otros aspectos de la interacción misma.
- El diseño de los experimentos, su ejecución y el análisis realizado sugieren que el escenario implementado es viable como solución para crear relaciones semánticas entre objetos y generar nuevos servicios ajustados a las necesidades de los usuarios de la WoT. En el capítulo de implementación del escenario se comprueba como el

desarrollo fue avanzando según la retroalimentación de las pruebas realizadas, y como el parámetro de latencia en cada uno de los casos de uso se fueron reduciendo hasta alcanzar tiempos aceptables.

- El proyecto incluye la OOS en cada objeto para darle un perfil semántico y generar una representación semántica del objeto, sin embargo esta no es utilizada en toda la potencialidad que puede dar una ontología, como el generar razonamientos complejos o tomar decisiones autónomas. Se espera que próximas versiones incluyan otras características a forma de casos de uso, para que así la intervención del usuario sea cada vez menos presente y las relaciones entre objetos más intuitiva.
- La realización de este trabajo permite validar la propuesta realizada por [6], demostrando que el modelo de interacción semántica en la web de las cosas plantea los requisitos que se deben cumplir para permitir la cooperación de objetos inteligentes en la creación de nuevos servicios producto de dicha interacción, aporta también la implementación de la comunicación de objetos semánticos mediante MQTT.

7.2 TRABAJO FUTURO

- Es posible ajustar las OOS que almacenan la semántica del objeto con el fin de realizar operaciones de razonamiento autónomo de los objetos y brindarle al escenario capacidades de autonomía haciendo que el usuario intervenga en menor medida.
- Mejorar la interfaz gráfica de Clipio con el fin de añadirle funcionalidades deseables para los usuarios como gráficas, tendencias y avisos en sus redes sociales existentes. Además afinar las interfaces con el fin de ser una herramienta más amable para el usuario.
- Definir una nueva interfaz que permita realizar la inicialización de los objetos por parte del usuario. Esta interfaz debe enfocarse en el uso de herramientas que mejoren la experiencia del usuario al momento de programar un nuevo objeto en su escenario.
- Incluir diferentes dispositivos en la capa de objetos, esto con el fin de realizar una evaluación más prolífica y observar nuevos comportamientos y necesidades en los escenarios. Para eso se debe mejorar principalmente la interfaz de inicializar objeto, dando la posibilidad de seleccionar distintos dispositivos con configuraciones totalmente diferentes. Además se puede incluir otras entidades y realizar un escenario mucho más real e interactivo, donde se pueda vislumbrar de mejor manera la capacidad dinámica de la propuesta.
- Ampliar las experimentaciones con varios objetos y varias entidades con el fin de poder medir otros indicadores del escenario.
- Es necesario establecer nuevas capas que permitan realizar una mejor autenticación y personalización para el usuario, de esta forma se propone realizar un proyecto donde se enfoque en la creación de nuevos servicios a partir de aplicaciones que soporten múltiples usuarios, además estos usuarios deben ser construidos con jerarquías para evitar el conflicto de servicios entre usuarios.
- Este escenario trabaja sobre protocolos de internet y utiliza herramientas de la web, además comparte información que posiblemente sea delicada para el usuario, por lo tanto se hace necesario proponer nuevos proyectos donde se analicen factores de seguridad en la información.

8 BIBLIOGRAFÍA

- [1] T. Teixeira, S. Hachem, V. Issarny y N. & Georgantas, «Service oriented middleware for the internet of things: A perspective,» de *Towards a Service-Based Internet*, Springer Berlin Heidelberg., 2011, pp. 220-229.
- [2] L. Atzori, A. Iera y G. Morabito, «The internet of things: A survey,» *Computer networks*, vol. 54, nº 15, pp. 2787-2805, 2010.
- [3] Piyare, S. Rajeev y R. Lee, Towards internet of things (iots): Integration of wireless sensor network to cloud services for data collection and sharing., arXiv preprint arXiv, 2013.
- [4] S. Mathew, Y. Atif, Q. Z. Sheng y Z. Maamar, «Web of things: Description, discovery and integration. In Internet of Things (iThings/CPSCoM),» de *International Conference on and 4th International Conference on Cyber, Physical and Social Computing (pp. 9-15)*. IEEE., 2011.
- [5] F. Scioscia y M. Ruta, «Building a Semantic Web of Things: issues and perspectives in information compression,» de *IEEE International Conference*, 2009.
- [6] N.-Z. M. A., «Interacción Semántica de Objetos en la Web de las Cosas,» de *XI Coloquio Doctoral de CLADEA*, Rio de Janeiro, 2013.
- [7] M. L. Barry, G. C. Vinton y D. C. David, «A Brief History of the Internet,» 23 Junio 1999. [En línea]. Available: <http://arxiv.org/html/cs/9901011>. [Último acceso: 7 1 2015].
- [8] D. Bandyopadhyay y J. Sen, «Internet of things: Applications and challenges in technology and standardization,» *Wireless Personal Communications*, vol. 1, nº 58, pp. 49-69, 2011.
- [9] S. Bandyopadhyay, M. Sengupta, S. Maiti y S. Dutta, « Role of middleware for internet of things: A study,» *International Journal of Computer Science & Engineering Survey (IJCSSES)*, vol. 2, nº 3, pp. 94-105, 2011.
- [10] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 17 1 2016].
- [11] Raspberrypi, «Raspberrypi,» [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: 17 1 2016].
- [12] BeagleBone, «BeagleBone,» [En línea]. Available: <http://beagleboard.org/bone>. [Último acceso: 17 1 2016].

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- [13] Intel, «Intel Galileo Board,» [En línea]. Available: <http://ark.intel.com/products/78919/Intel-Galileo-Board>. [Último acceso: 7 Enero 2016].
- [14] S. S. Mathew, Y. Atif, Q. Z. Sheng y Z. Maamar, «Web of things: Description, discovery and integration,» de *International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, 2011.
- [15] Y. Liu y G. Zhou, «Key technologies and applications of internet of things,» de *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference*, IEEE, 2012, pp. 197-200.
- [16] J. Cooper y A. & James, «Challenges for database management in the internet of things,» *IETE Technical Review*, vol. 5, nº 26, pp. 320-329, 2009.
- [17] D. C. Mansilla, M. V. Barbas y M. T. L. Merayo, «Infraestructuras Inteligentes en el Internet del Futuro,» de *Actas del I Encuentro de Investigadores en Infraestructuras Inteligentes (EI3 2011)*, Guadalajara, Servicio de Publicaciones, 2011, pp. 97-100.
- [18] G. Kortuem, F. Kawsar, D. Fitton y V. Sundramoorthy, «Smart objects as building blocks for the internet of things,» *Internet Computing IEEE*, vol. 14, nº 1, pp. 44-51, 2010.
- [19] C. Perera, A. Zaslavsky, P. Christen y D. Georgakopoulos, «Context aware computing for the internet of things: A survey,» *Communications Surveys & Tutorials, IEEE*, vol. 16, nº 1, pp. 414-454, 2014.
- [20] J. Pascoe, «Adding generic contextual capabilities to wearable computers,» de *Second International Symposium*, IEEE, 1998, pp. 92-99.
- [21] C. Perera, P. P. Jayaraman, A. Zaslavsky, P. Christen y D. Georgakopoulos, «Context-Aware Dynamic Discovery and Configuration of ‘Things’ in Smart Environments,» *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 215-241, 2014.
- [22] C. Perera, P. P. Jayaraman, A. Zaslavsky, D. Georgakopoulos y P. Christen, «Sensor discovery and configuration framework for the internet of things paradigm,» *Internet of Things (WF-IoT), 2014 IEEE World Forum*, pp. 94-99, 2014.
- [23] M. M. N. Hadley, J. Moreau, H. Nielsen y M. Gudgin, «SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation,» 27 Abril 2007. [En línea]. Available: <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>. [Último acceso: 7 Enero 2016].
- [24] Z. Shelby, K. Hartke y C. Bormann, «The Constrained Application Protocol (CoAP),» Junio 2014. [En línea]. Available: <https://tools.ietf.org/html/rfc7252>. [Último acceso: 17 1 2016].

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- [25] P. Saint-Andre, «Extensible Messaging and Presence Protocol (XMPP),» Octubre 2004. [En línea]. Available: <https://tools.ietf.org/html/rfc3920>. [Último acceso: 17 1 2016].
- [26] A. Banks y R. Gupta, «MQTT Version 3.1. 1.,» 29 Octubre 2014. [En línea]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>. [Último acceso: 7 Enero 2016].
- [27] T. o. droup, «Open Messaging Interface (O-MI), an Open Group Internet of Things (IoT) Standard,» Octubre 2014. [En línea]. Available: <https://www2.opengroup.org/ogsys/catalog/C14B>. [Último acceso: 7 Enero 2016].
- [28] T. Berners-Lee, J. Hendler y O. Lassila, «The semantic web,» *Scientific american*, vol. 284, nº 5, pp. 28-37, 2001.
- [29] S. Suárez, R. Martínez y P. Redondo, «Biblioteca semántica de webquest,» 2004. [En línea]. Available: LuceneTutorial.com. [Último acceso: 7 1 2015].
- [30] M. Gudgin, M. Hadley, N. Mendelsohn, J. J. Moreau, H. F. Nielsen, A. Karmarkar y Y. & Lafon, *SOAP Version 1.2*, W3C recommendation 24, 2003.
- [31] J. e. Cardoso, *Semantic Web Services: Theory, Tools and Applications: Theory, Tools and Applications*, IGI Global, 2007.
- [32] M. Suárez Barón y K. Salinas Valencia, «An approach to semantic indexing and information retrieval,» *Revista Facultad de Ingeniería Universidad de Antioquia*, vol. 48, pp. 174-187, 2009.
- [33] T. R. Gruber, «Toward principles for the design of ontologies used for knowledge sharing,» *International journal of human-computer studies*, vol. 43, nº 5, pp. 907-928, 1995.
- [34] D. Oberle, «Semantic management of middleware,» *Proceedings of the 1st international doctoral symposium on Middleware*, pp. 299-303, 2004.
- [35] P. D. Karp, V. K. Chaudhri y J. Thomere, «XOL: An XML-based ontology exchange language,» 1999. [En línea]. Available: <http://xml.coverpages.org/xol-03.html>. [Último acceso: 7 Enero 2016].
- [36] D. Brickley y R. V. Guha, «RDF vocabulary description language 1.0: RDF schema,» 2004. [En línea]. Available: <http://www.w3.org/TR/PR-rdf-schema>. [Último acceso: 7 Enero 2016].

- [37] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann y M. Klein, «OIL in a nutshell,» *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 1-16, 2000.
- [38] F. P.-S. P. F. van Harmelen y I. Horrocks, «Reference description of the DAML+OIL (March 2001) ontology markup language,» Mayo 2001. [En línea]. Available: <http://www.daml.org/2001/03/reference.html>. [Último acceso: 7 Enero 2016].
- [39] D. C. M. Dean, F. v. Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider y L. Stein, «OWL Web Ontology Language 1.0 Reference, W3C Recommendation,» 10 Febrero 2004. [En línea]. Available: <http://www.w3.org/TR/owl-ref/>. [Último acceso: 7 Enero 2016].
- [40] J. Gómez, J. F. Huete, O. Hoyos, L. Perez y D. Grigori, «Interaction System based on Internet of Things as Support for Education,» *Procedia Computer Science*, vol. 21, pp. 132-139, 2013.
- [41] F. Kawsar, G. Kortuem y B. Altakrouri, «Supporting interaction with the internet of things across objects, time and space,» de *Internet of Things (IOT)*, IEEE, 2010, pp. 1-8.
- [42] P. E. Estrada–Martinez y J. A. Garcia–Macias, «Semantic interactions in the Internet of Things,» *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 13, nº 3, pp. 167-175, 2013.
- [43] R. Piyare y S. R. Lee, «TOWARDS INTERNET OF THINGS (IOTS) INTEGRATION OF WIRELESS SENSOR NETWORK TO CLOUD SERVICES FOR DATA COLLECTION AND SHARING,» *International Journal of Computer Networks & Communications (IJCNC)*, vol. 5, nº 5, pp. 59-72, 2013 .
- [44] N. Rosen, R. Sattar, R. W. Lindeman, R. Simha y B. Narahari, «HomeOS : Context-Aware Home Connectivity,» de *International Conference on Wireless Networks*, 2004.
- [45] J. Sixte y N. Peralta, «SMARTEC: domótica y seguridad,» Smartec, [En línea]. Available: <http://www.smartec.com.ar/>. [Último acceso: 24 1 2016].
- [46] Ozom, «Ozom,» Ozom, [En línea]. Available: http://help.ozom.me/help_center. [Último acceso: 24 1 2016].
- [47] Control4, «Control4,» Control4, [En línea]. Available: <http://www.control4.com/>. [Último acceso: 24 1 2016].
- [48] B. F., M. R., Z. J. y A. S., «Fog computing and its role in the internet of things,» de *presented at the Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, Helsinki, Finland, 2012.

ESCENARIO DE INTERACCIÓN SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WEB DE LAS COSAS

- [49] A. G. M. Hauswirth, S. Krco, N. Stojanovic, M. Bauer y R. Nielsen, «An architectural blueprint for a Real-World Internet,» *Springer Berlin Heidelberg*, pp. 67-80, 2011.
- [50] LogMeIn, «Xively,» LogMeIn, 2003. [En línea]. Available: <http://xively.com/>. [Último acceso: 24 1 2016].
- [51] D. Bonino y F. Corno, «Dogont-ontology modeling for intelligent domotic environments,» *Springer Berlin Heidelberg*, pp. 790-803, 2008.
- [52] J. Gubbi, R. Buyya, S. Marusic y M. Palaniswami, «Internet of Things (IoT): A vision, architectural elements, and future directions,» *Future Generation Computer Systems*, vol. 29, nº 7, pp. 1645-1660, 2013.
- [53] G. Vithya y B. Vinayagasundaram, «QOS by Priority Routing in Internet of Things,» *Research Journal of Applied Sciences, Engineering and Technology*, vol. 8, nº 21, pp. 2154-2160, 2014.
- [54] M. A. Nef, L. Perlepes, S. Karagiorgou, G. I. Stamoulis y P. K. Kikiras, «Enabling qos in the internet of things,» de *Proceedings of the Fifth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2012)*, France, Chamonix/Mont Blanc, 2012, pp. 33-38.
- [55] D. Barata, G. Louzada, A. Carreiro y A. Damasceno, «System of Acquisition, Transmission, Storage and Visualization of Pulse Oximeter and ECG Data Using Android and MQTT,» *Procedia Technology*, vol. 9, p. 1265–1272, 2013.