



MODELADO DE UNA ARQUITECTURA DE LA INTERNET DE LAS COSAS PARA GESTIONAR LA EMISIÓN DE CARBONO ATMOSFÉRICO EN LAS CIUDADES

Autor

Mauricio Orlando Bermúdez Amaya

Tutor

Octavio Salcedo Parra

PhD En Ingeniería Informática

PhD En Estudios Políticos

Universidad Distrital Francisco José De Caldas

Maestría en Ciencias de la Información y las Comunicaciones

Énfasis en Ingeniería de Software

Bogotá, Colombia

junio de 2022

CONTENIDO

RESUMEN.....	5
ABSTRACT	5
INTRODUCCION.....	7
PALABRAS CLAVE	8
1. PROBLEMA DE INVESTIGACION	9
1.1 PLANTEAMIENTO DEL PROBLEMA.....	9
1.2 PREGUNTA DE INVESTIGACIÓN	11
1.3 OBJETIVOS	12
1.4 RESULTADOS ESPERADOS	12
1.5 HIPÓTESIS DE INVESTIGACIÓN.....	13
1.5.1 Hipótesis de Trabajo.....	13
1.5.2 Hipótesis Nula	14
2. MARCO REFERENCIAL	15
2.1 MODELOS ARQUITECTONICOS FUNCIONALES DE LA INTERNET DE LAS COSAS IoT	15
2.1.1 Red de objetos conectados mediante API de servicios.....	17
2.1.2 Red estratificada con capa de servicios.....	19
2.1.3 Redes aplicadas a ciudad inteligente.....	20
2.1.4 Red de evolución a largo término avanzada	23
2.1.5 Red de tres capas orientada a servicios	24
2.1.6 Red de sensores inalámbricos WSN	26

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

2.1.7	Red de sensores con transporte de cola de mensajes por telemetría MQTT-SN	27
2.1.8	Red de fibra e inalámbrica FiWi.....	30
2.1.9	Red inalámbrica definida por software.....	32
2.1.10	Discusión de Arquitecturas Funcionales	35
2.1.11	Conclusión de las de Arquitecturas Funcionales.....	37
2.2	MODELOS ONTOLÓGICOS DE BASE MÁQUINA A MÁQUINA M2.....	37
2.2.1	Ontología de Red de Sensores Semánticos	39
2.2.2	Ontología Machine to Machine Measurement M3.....	41
2.2.3	Ontología oneM2M.....	43
2.2.4	Ontología IoT-Lite.....	47
2.2.5	Discusión de las Ontologías	49
2.2.6	Conclusión acerca de las Ontologías.....	50
2.3	Modelo arquitectónico funcional M2M–IoT propuesto.....	52
2.3.1	Definición de Ontología	52
2.3.2	Arquitectura funcional de base ontológica	52
2.3.3	Ontología de unidades de Medida OM.....	54
2.3.4	Ontología de Referencia de Dispositivos Inteligentes SAREF	59
2.3.5	Ontología de Referencia de Dispositivos Inteligentes para Energía SAREF4ENER.....	76
2.3.6	Ontología Máquina a Máquina Monádica oneM2M.....	105
3.	DISEÑO DE INVESTIGACIÓN	166
3.1	MARCO DE TRABAJO.....	166
3.1.1	Instancia de Inicio.....	167
3.1.2	Instancia de elaboración.....	170
3.1.3	Instancia de construcción	171

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

3.1.4	Instancia de transición	172
4.	Diseño experimental preliminar.....	174
5.	Preparación de datos de entrada	177
6.	Modelo de traducción	182
7.	Verificación y validación.....	184
8.	Diseño experimental final.....	188
9.	SIMULACIÓN	195
10.	ANÁLISIS DE RESULTADOS	200
11.	CONCLUSIONES, TRABAJOS FUTUROS Y RECOMENDACIONES	203
11.1	Conclusiones	203
11.2	TRABAJOS FUTUROS.....	209
11.3	Recomendaciones	212
12.	CONTRASTACIÓN DE OBJETIVOS, HIPÓTESIS Y APORTES ORIGINALES..	214
12.1	Contrastación de objetivos.....	214
12.2	Contrastación de hipótesis.....	217
12.3	Aportes originales	217
13.	REFERENCIAS.....	220
	ANEXOS.....	230
	ANEXO 1:	230
	ANEXO 2:	239
	ANEXO 3:	245

RESUMEN

La presente investigación atañe a la configuración de una estructura lógica referente a la interconexión digital de objetos cotidianos –y que integran dispositivos informáticos para el tratamiento de datos–, la cual posibilite el control de las emanaciones de gas carbónico en las grandes urbes de nuestro país, cuyo incremento por causal antropogénico ha incidido en el calentamiento global actual.

En consecuencia, se ha seleccionado el sector de dispositivos electrónicos de consumo para la correspondiente gestión de la emisión de dióxido de carbono CO₂ a la atmósfera, definiendo la estructura y el funcionamiento de un sistema de la Internet de las Cosas en cuanto a una capa de servicios que funja como interfaz entre la red y sus aplicaciones, permitiendo el control de dicho exhalamiento a partir del umbral y autorregulación de los artefactos en mención.

Se obtuvo como resultado de esta iniciativa, un modelo de arquitectura funcional flexible el cual, sustentado en un corpus semántico combinado de ontologías, propone una manera de controlar el acrecentamiento de la concentración de CO₂ en la atmósfera, al procurar que los dispositivos habituales que consumen energía eléctrica para su funcionamiento –y que dicha energía se haya producido mediante la quema de combustibles fósiles–, mantengan dicho gasto regulado al estar tales objetos comunicados a través de la Internet sobre interfaces específicas.

ABSTRACT

The present investigation concerns the configuration of a logical structure referring to the digital interconnection of everyday objects –and that integrate computing devices for data processing–, which enables the control of carbon dioxide emissions in the large cities of

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

our country , whose increase due to anthropogenic causes has affected current global warming.

Consequently, the sector of consumer electronic devices has been selected for the corresponding management of the emission of carbon dioxide CO₂ into the atmosphere, defining the structure and operation of an Internet of Things system in terms of a layer of services that act as an interface between the network and its applications, allowing the control of said exhalation from the threshold and self-regulation of the mentioned devices.

Got as a result of this initiative, a flexible functional architecture model was obtained which, based on a combined semantic corpus of ontologies, proposes a way to control the increase in the concentration of CO₂ in the atmosphere, by ensuring that the usual devices that consume electrical energy for its operation –and that said energy has been produced by burning fossil fuels–, keep such spending regulated as such objects are communicated over the Internet on specific interfaces.

INTRODUCCION

La Internet de las Cosas (en inglés Internet of Things IoT) se ha convertido en un recurso oportuno para la conectividad permanente de la generalidad de los objetos existentes en la naturaleza y las civilizaciones. En dichas sociedades, las personas hacemos uso de todo tipo de artefactos para satisfacer nuestras necesidades de mejora en la calidad de vida, los cuales en su mayoría requieren de un suministro constante de energía para su funcionamiento. Una gran parte de esa energía en mención se produce mediante la quema de combustibles fósiles, situación que está ocasionando el calentamiento global con sus efectos ya experimentados y cuya eventualidad se requiere mitigar y/o controlar.

La IoT es una alternativa adecuada y conveniente para moderar el cambio climático mencionado entre límites tolerables para el planeta, al conectar dispositivos electrónicos de bajo consumo mediante la tecnología Máquina a Máquina (en inglés Machine to Machine M2M), gestionando su respectivo gasto energético en línea sin menoscabo de su desempeño, lo cual permite reducir la emisión de gases de efecto invernadero al utilizar la cantidad de energía eléctrica específica para sus funciones y cuya generación se dio con antelación en las centrales respectivas al combustionar materiales orgánicos y/o sus derivados.

En las ciudades actuales, uno de los gases de efecto invernadero que en mayor cantidad se arroja al ambiente es el Dióxido de Carbono (fórmula química CO₂), siendo dificultoso para la naturaleza su reciclaje dada la gran proporción del mismo en la atmósfera. No obstante, la emisión de CO₂ en el entorno urbano causada por la actividad industrial, comercial y de servicios, no se presenta en el consumo de la energía eléctrica per se, ya que al cerrar el circuito (i.e. utilizar la máquina) para que fluya dicha energía, esta ya ha sido producida con anticipación; por lo que el gas carbónico generado ya fue emitido y se encuentra circulante en el ecosistema.

Lo anterior conduce a una iniciativa de concebir un recurso o medio que permita –mas que mitigar– gestionar la emisión de dióxido de carbono en las ciudades sin menoscabo de su desarrollo sostenible y sustentable, siendo dentro del conjunto de alternativas, la que corresponde al modelado de una arquitectura funcional flexible de la Internet de las

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Cosas, una opción pertinente, coherente y viable; al proponer interoperar semánticamente la miríada de dispositivos M2M en el mundo para que se autorregulen, independientemente de la tecnología de comunicación e información inherente y/o subyacente.

Dicha arquitectura planteada requiere de la generalización formal de los conceptos de un universo de discurso determinado, a partir de sus propiedades y relaciones inmanentes. Tal universo se compone de los dominios correspondientes a la Internet de las Cosas IoT, los dispositivos M2M, los protocolos de comunicación, la energía y las magnitudes fundamentales, los cuales conforman una ontología cuyo modelo integral, sienta las bases para que las máquinas en mención se conecten y gestionen el gasto energético de sus procesos en tiempo real y por ende, el gestionar la emisión de CO2 en las ciudades.

PALABRAS CLAVE

Internet de las Cosas IoT, arquitectura IoT, Dióxido de Carbono CO2, emisión de CO2, ontología, modelado ontológico, gestión de CO2.

1. PROBLEMA DE INVESTIGACION

1.1 PLANTEAMIENTO DEL PROBLEMA

Desde finales del siglo XX, al interior de la civilización actual ha adquirido relevancia el concepto de "transición energética" (UPME, 2015, 8), cuya definición atañe a "un conjunto significativo de cambios en los patrones de uso de la energía en una sociedad, afectando los recursos, los portadores, los equipos y los servicios energéticos" (O'Connor, 2010, p. 8). Dicho cambio se destaca por la transformación de la producción de energía mediante la quema de combustibles fósiles y carbón, hacia las fuentes renovables como generación de potencia eléctrica primordial (óp. cit.).

Ante el problema latente del cambio climático a nivel mundial que enfrenta la humanidad, la ansiedad de las naciones con economías de altos ingresos ha suscitado la realización de dicha transición, dada la actual aceleración en la variación del estado del sistema meteorológico del planeta causado por el calentamiento del mismo, producto de la intensificación del efecto invernadero en razón al aumento de los gases respectivos - especialmente CO₂-, dado su origen antropogénico lamentablemente (ibídem).

Dicho cambio energético mencionado está ocurriendo hoy por hoy en la actividad económica de transporte de pasajeros y carga, en razón a que el arquetipo de desplazamiento particular carece de sustentabilidad por el enorme gasto energético inherente, a más de la infraestructura vial demandada. Ello sin tener en cuenta el tráfico urbano in crescendo, los viajes cada vez más prolongados y la contaminación atmosférica, lo que genera detrimento en la economía y el bienestar de las comunidades (ib., pág. 9).

En el sector fabril (manufactura, construcción, minería, etc.), el aprehender el CO₂ de sus procesos productivos antes de emitirse a la atmósfera y almacenarlo geológicamente, ha sido una opción para reducir al mínimo tales exhalaciones, incluyendo las de sus propias instalaciones termoeléctricas al igual que en la industria del cemento, sustancias químicas, el acero, etc.; dado que -no en balde- La Agencia Internacional de Energía (en inglés International Energy Agency IEA), ha estimado que el gas carbónico susceptible de retenerse hasta mediados de la actual centuria llegaría al guarismo de 45%, supeditado

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

ello -claro está-, al VPN y el TIR de las inversiones en grandes superficies, como también a la evolución de la tecnología per se (ibídem, pág. 25).

En cuanto a la actividad económica de servicios públicos (energía eléctrica, gas combustible, etc.), se observa un entorpecimiento persistente a la sustitución y desarrollo planificado de las redes de transmisión -en razón a su obsolescencia y demanda-, situación suscitada por los inherentes derechos de servidumbre, los precios elevados y la resistencia a edificar sus instalaciones. Dicha postura conduce indefectiblemente a modificar el modo de producir potencia eléctrica, pasando de una generación reconcentrada a una repartida, contigua a los lugares de consumo y atomizada, permitiendo el concurso amplio en el mercado energético a nivel de operamiento y acometimiento, al igual que la manera de generar energía renovable (solar, eólica, etc.) con responsabilidad y democracia (ibíd., pág. 8).

Dados los factores de producción energética, impacto ambiental y servicios públicos, se deduce que las emisiones de CO₂ en las ciudades dependen de la combinación de los tipos de generación de energía utilizada por el proveedor, en especial la eléctrica, por lo que la incidencia en el efecto invernadero a partir de la tecnología de consumo (dispositivos electrónicos) es alta. Se sabe que la mayoría de los países industrializados (con excepción de Francia y Canadá) obtienen gran parte de su electricidad (entre el 60% y 80%) a partir de la combustión de los combustibles fósiles (EIA, 2013) (CDIAC, 2011).

De acuerdo con lo anterior, Vermesan y Friess establecen que el mundo se está encaminando hacia la idea de la "Smart City", al observarse que la concentración urbanística junto con el desplazamiento y la electrificación en cuanto a sistema, se conectan y consolidan de manera holística no solamente en lo tocante a infraestructura, sino también en lo concerniente a sustentabilidad, optimización de procesos y bienestar humano para la diversidad de sus componentes, como lo son la gestión de aguas y desperdicios, la provisión de carburantes, potencia eléctrica, movilidad, comunicaciones, gobernanza, etcétera. Dicho concepto de urbe sostenible se verá perfeccionada en tiempo cercano e in crescendo, con la implementación de la Internet de las Cosas (Internet of Things IoT) (2013, Citado en UPME, 2015, pág. 23).

Por lo tanto, el sector energético se verá transformado por las determinaciones adoptadas en cuanto a innovación, mudando desde un mercado al por mayor de generación y distribución de potencia, a un conjunto de operaciones comerciales al detal; id est, un tránsito desde el mono u oligopolio a un entorno económico de libre competencia de bienes y servicios cuya sostenibilidad ambiental, equidad y suministro de dicho recurso - acorde al trilema del Consejo Mundial de Energía (World Energy Council WEC) (citado por UPME, p. 81)-, facilitará la implementación de una política de mitigación de GEI (bonos de carbono), acompañada del criterio ya en boga de eficiencia en la producción y consumo de energía mediante gestores (Customer Energy Manager CEM en inglés) (ibíd., pág. 25).

Para efectos de lo anterior en cuanto a gasto energético a nivel urbano, una de las diversas maneras de gestionar la emisión de carbono atmosférico atañe a poder contar con un interfuncionamiento, tanto a nivel de las tareas y servicios que prestan los diversos dispositivos, como también de la independización de la tecnología subyacente a los mismos; i.e., una interoperabilidad semántica y sintáctica (oneM2M, 2019, págs. 9-10), que facilite el obtener una comprensión compartida del dominio de interés (esto es, los agentes del sistema) y en simultánea, proporcionar un modelo formal de dicho dominio e interpretable por todas y cada máquina (ibídem, p. 8).

1.2 PREGUNTA DE INVESTIGACIÓN

La siguiente sería la pregunta de investigación a enunciar...

¿De qué manera diseñar un modelo de arquitectura funcional flexible que facilite la interoperabilidad semántica entre los diversos sistemas de dispositivos inteligentes, fungiendo como iniciativa de gestión para la emisión de gas carbónico atmosférico en las ciudades?

1.3 OBJETIVOS

El propósito general atañe a una iniciativa de modelado de una arquitectura funcional flexible para la Internet de las Cosas IoT y de base Máquina a Máquina M2M, que facilite la gestión de las emisiones de dióxido de carbono CO₂ en las ciudades, específicamente en el sector de dispositivos electrónicos de consumo -los cuales presentan un gasto de energía producida mediante la quema de combustibles fósiles-, teniendo en cuenta los siguientes objetivos específicos:

- Identificar dentro los diversos tipos de arquitectura de la Internet de las cosas existentes, el más apropiado para el modelo a proponer.
- Determinar al interior de los diversos modelos ontológicos de base Máquina a Máquina existentes, el más adecuado y conveniente para la arquitectura funcional a plantear.
- Reconocer de manera detallada las ontologías que conforman el modelo propuesto, las cuales han de corresponder tanto al universo de discurso Máquina a Máquina M2M como a la Internet de las Cosas IoT y a su vez, con arreglo al dominio de los dispositivos inteligentes y la optimización energética.
- Definir el modelo arquitectónico funcional flexible IoT a proponer, a partir de la combinación de ontologías M2M que facilitan la gestión de CO₂ atmosférico emitido por el uso de dispositivos inteligentes.
- Establecer el diseño experimental del modelo arquitectónico funcional flexible propuesto, que facilite su construcción y la comprensión de su comportamiento.
- Validar el modelo de arquitectura funcional propuesta mediante una simulación del mismo, acorde a las especificaciones oficiales de la W3C que permitan certificarlo.

1.4 RESULTADOS ESPERADOS

A partir del desarrollo de la propuesta para un modelo de arquitectura funcional flexible que permita gestionar la emisión de CO₂ en las ciudades, se espera como resultado controlar el acrecentamiento de la concentración de CO₂ en la atmósfera, al procurar que los dispositivos habituales que consumen energía eléctrica para su funcionamiento -y que dicha energía se haya producido mediante la quema de combustibles fósiles-, mantengan dicho gasto regulado al estar tales objetos comunicados a través de la Internet sobre interfaces específicas. Los resultados aludidos se listan como sigue:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Identificación de una arquitectura funcional de la Internet de las cosas IoT como la más apropiada para utilizarla como marco de trabajo en el desarrollo de la interoperabilidad sintáctica y semántica de los dispositivos inteligentes
- Reconocimiento de un modelo ontológico de base Máquina a Máquina M2M como el más adecuado y conveniente, el cual facilite el desarrollo de la interoperabilidad mencionada.
- Desarrollo de un modelo arquitectónico funcional de base M2M aplicado a la IoT que, a partir del corpus semántico de la ontología combinada, pueda aplicarse al moderamiento de las emisiones de carbono al ambiente mediante la optimización de la eficiencia energética de los dispositivos inteligentes conectados.
- Determinación de los recursos al interior de la ontología propuesta que faciliten las funciones de medición, control, obtención y establecimiento de datos de potencia y energía producidos/consumidos por parte de los dispositivos inteligentes.
- Comprobación de la idoneidad del modelo arquitectónico propuesto mediante una simulación que permita evaluar su consistencia y robustez.

1.5 HIPÓTESIS DE INVESTIGACIÓN

A partir del planteamiento del problema, los objetivos y los resultados esperados, se lleva a cabo la formulación de la hipótesis, cuyo pronóstico o proposición se describe de tal modo que se pueda demostrar de manera lógica, facilitando inferir consecuencias acorde a los conceptos que se detallan en el marco referencial; es decir, acorde al razonamiento deductivo propio del proceso de investigación cuantitativa (Hernández, et. al., 2006).

1.5.1 Hipótesis de Trabajo

La hipótesis de trabajo (Hi), la cual es descriptiva de un valor pronosticado de manera afirmativa (ibídem), se enuncia a continuación:

"El modelo arquitectónico funcional flexible propuesto ES VALIDADO por la simulación del sistema, sustentado en el corpus semántico combinado de ontologías para la gestión de la emisión de CO₂ atmosférico por parte de dispositivos inteligentes".

1.5.2 Hipótesis Nula

La Hipótesis Nula (H_0), la cual se enuncia contradiciendo a la hipótesis de trabajo (ibíd.), se formula como sigue:

"El modelo arquitectónico funcional flexible propuesto NO ES VALIDADO por la simulación del sistema, sustentado en el corpus semántico combinado de ontologías para la gestión de la emisión de CO₂ atmosférico por parte de dispositivos inteligentes".

2. MARCO REFERENCIAL

2.1 MODELOS ARQUITECTONICOS FUNCIONALES DE LA INTERNET DE LAS COSAS IOT

Se considera a la Internet de las Cosas (Internet of Things IoT) como una red de redes, dado que permite la conexión de sistemas de comunicación de datos individuales, con otros que poseen seguridad, análisis y gestión en diversos dominios como transporte, energía, educación, negocios, hogar, etc. (Evans, 2011, 4).

El IoT hace que Internet sea sensorial, al detectar y capturar señales (temperatura, presión, luz, etc.) mediante nano-dispositivos electrónicos llamados sensores (Evans, 2011, 6), los cuales transmiten a servidores mediante protocolos establecidos, para que dichas máquinas las conviertan en datos, las recopilen, las analicen y por último, las distribuyan a escala masiva para que las personas procesen dicha información y generen conocimiento (ibíd., 7).

La IoT se encuentra abocada a superar diversos obstáculos en su evolución como son:

- Autosuficiencia en energía por parte de los sensores.
- La implementación del IPv6 para asignar direcciones únicas a los sensores tanto nuevos como existentes.
- El acuerdo sobre los estándares especialmente en cuanto a seguridad, privacidad, arquitectura y comunicaciones (Evans, 2011, 10).

Para salvaguardar el primer impedimento, las grandes multinacionales de la electrónica están investigando a nivel de micro y nano componentes que optimicen los requerimientos energéticos de los sensores. En cuanto a la segunda dificultad, la implementación de la nueva versión de IP es cuestión de tiempo para su expansión y consolidación, dado que no son los dispositivos los que requieren la dirección única, sino las interfaces de conexión.

En cuanto al último obstáculo para la evolución de la IoT, las empresas, gobiernos, organismos de normalización y las universidades están investigando en sinergia para alcanzar la interoperabilidad según un conjunto común de estándares (ibídem), los cuales

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

a nivel de arquitecturas funcionales fundamentales se destacan en los siguientes modelos:

- Objetos conectados mediante API de servicios
- Estratificación con capa de servicios
- Redes aplicadas a ciudad inteligente
- Autónoma
- Ubicua
- Capa de aplicación superpuesta
- Orientada a servicios
- Evolución a largo término avanzada LTE-A
- Tres capas orientada a servicios
- Sensores inalámbricos WSN
- Sensores con transporte de cola de mensajes por telemetría MQTT-SN
- Fibra e inalámbrica FiWi
- Inalámbrica definida por software

Los artículos de investigación de la tabla 1 son afines al presente trabajo y fungen como marco teórico del mismo:

No.	Autor	Artículo de investigación	Arquitectura de red
1	Inge Grønbaek	Architecture for the Internet of Things (IoT): API and interconnect	Objetos conectados mediante API de servicios
2	Duan R., Chen X., Xing T.	QoS Architecture for IOT	Estratificada con capa de servicios
3	Jin J., Gubbi J., et. al.	Network Architecture and QoS Issues in the Internet of Things for a Smart City	Aplicadas a ciudad inteligente (Autónoma, Ubicua, etc.)
4	Wali P., Das D.	A Novel Access Scheme for IoT Communications in LTE-Advanced Network	Evolución a largo término LTE Avanzada
5	Li L., Li S., Zhao S.	QoS-Aware Scheduling of Services-Oriented Internet of Things	Tres capas orientada a servicios
6	Ezdiani S., Acharyya I., et. al.	An IoT Environment for WSN Adaptive QoS	Sensores inalámbricos WSN
7	Govindan K., Prakash A.	End-to-end Service Assurance in IoT MQTT-SN	Sensores con transporte de cola de mensajes por telemetría MQTT-SN

8	Liu J., Guo H., et. al.	New Perspectives on Future Smart FiWi Networks: Scalability, Reliability and Energy Efficiency	Fibra e inalámbrica FiWi
9	El-Mougy A., et. al.	Software-Defined Wireless Network Architectures for the Internet-of-Things	Inalámbrica definida por software

Tabla 1. Trabajos relacionados con la arquitectura funcional de la IoT (Elaborada por el autor)

2.1.1 Red de objetos conectados mediante API de servicios

En lo referente a la arquitectura de la IoT, ya desde el año 2008 se comenzaron a definir posibles estructuras de la misma como la sugerida por Grønbæk, la cual comprende un par de dominios de red lógicamente distintos, pero cercanamente acoplados, es decir, los dominios (ámbitos) de la troncal y los dispositivos de red. Ambos entornos pueden estar sirviendo Objetos Conectados de manera directa (Grønbæk, 2008, p. 4).

Una arquitectura de IoT podría considerar que los Objetos Conectados CO (dispositivos electrónicos de consumo) se enlazan directamente a (ibídem):

La troncal de Internet: (en la capa de red e IP), en la cual se encuentran la red IP, servidores de encuentro o reunión (Rendezvous Server RVS), manejador de resoluciones (Resolution Handler RH como extensión de RVS), servicio de nomenclatura de objetos (Object Naming Service ONS, integrado a un RVS como parte de una red global de códigos de productos electrónicos, Electronic Product Codes, EPC Global Network) y por último, servidores de nombres de dominio (Domain Name Server DNS). Así mismo, se hallan en la la red IP de la troncal servicios de hospedaje (hosting), protocolo de arranque (bootstrap), inicio de sesión (logging), clasificación (rating) y facturación (billing).

Redes de dispositivos: artefactos electrónicos conectados a una LAN o red interna cableada, la cual se vincula a Internet mediante una puerta de enlace o pasarela (Gate Way GW) que cuenta con un servidor de etiquetas de identificación (Host Identity Tag HIT). Dicho HIT Gateway está soportado en el servidor de protocolo de identificación (Host Identity Protocol HIP) y permite direccionamiento global de de Objetos Conectados CO, mientras se mantiene el uso de direcciones IPv4.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

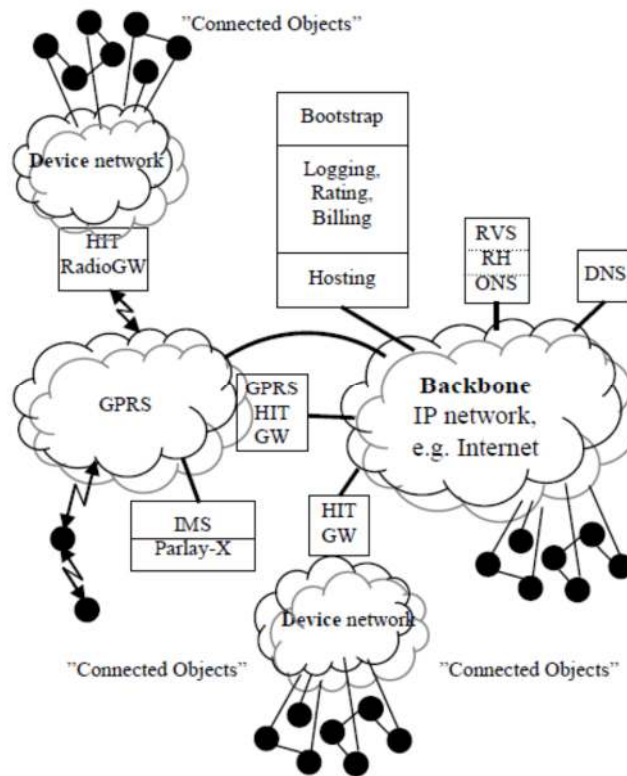


Figura 1. Componentes relacionados en una arquitectura IoT (Grønbaek, 2008, p. 4)

Cuando la red de dispositivos cuenta con una HIT GW inalámbrica (radio), se debe hacer uso del sistema general de paquetes de radio GPRS (General Packet Radio System) para enlazar a Internet, el cual incluye servicios del subsistema IP multimedia (IP Multimedia Subsystem IMS), redes de siguiente generación (Next Generation Networking NGN), APIs para redes telefónicas Parlay-X y por último, servidores GPRS HIT GW (ibídem).

Es factible que Objetos Conectados CO inalámbricos o pertenecientes al sistema global de comunicaciones móviles GSM (Global System for Mobile communications), no estén vinculados a una red interna y se enlacen directamente al servicio de GPRS y mediante este último, se conectasen a la troncal de Internet (ibídem).

2.1.2 Red estratificada con capa de servicios

Duan, Chen y Xing proponen una arquitectura de calidad de servicio basado en la estructura en capas del IoT. Esta arquitectura establece un agente de QoS en las capas más bajas que transmiten los requisitos de calidad de servicio, tratando de garantizar la consistencia, así como utilizar de forma eficaz los mecanismos de QoS existentes en cada capa (2011, p. 1).

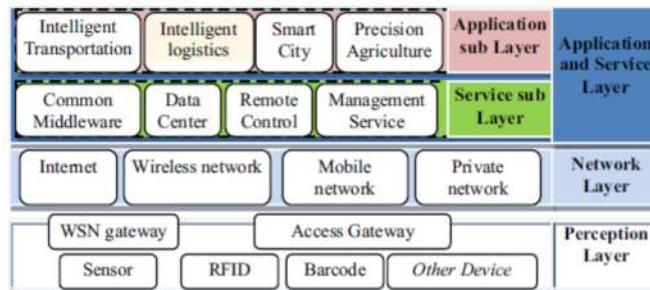


Figura 2. Arquitectura de la Internet de las Cosas (Duan, et. al., 2011, p. 1)

Las capas de la figura anterior se detallan a continuación:

La capa de percepción corresponde a los objetos conectados CO (dispositivos como sensores, RFID, lectores de código de barras, electrónicos de consumo, etc). Dichos componentes hacen parte de la capa física del modelo OSI.

La capa de red atañe a las puertas de enlace de los sensores inalámbricos de red (Wireless Sensor Network WSN) y de acceso, como también al direccionamiento IP de Internet y redes locales, móviles y wi-fi, lo cual hace parte de la capa de enlace y de red del modelo OSI; i.e. de la troncal (backbone).

La capa de aplicación y servicio consiste en las soluciones de software para usuario final con todos los servicios de hospedaje, repositorios e intermediación, aspecto que equivale a las capas de transporte, sesión, presentación y aplicación del modelo OSI.

En esta arquitectura, los requerimientos de calidad del servicio QoS tienen que ver con tres tareas: indagación, control y monitoreo. Ejemplos respectivos son: el estado de un

ítem en logística inteligente, control remoto en tráfico inteligente, registro de eventos del medio ambiente. Así mismo, se implementan un par de agentes de QoS en las capas de red y de percepción, los cuales son regulados mediante un gestor de instalación de calidad de servicio.

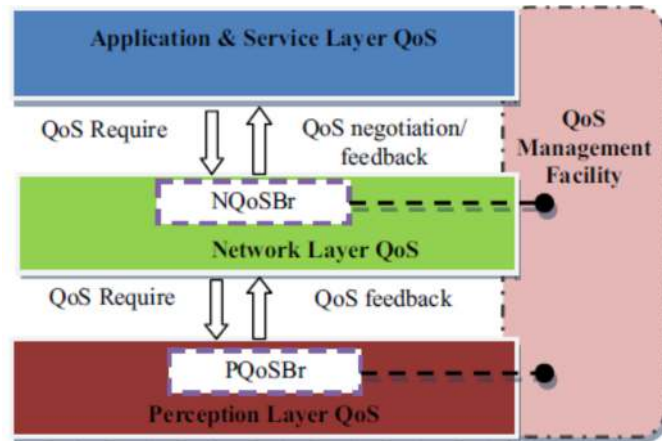


Figura 3. Calidad del Servicio QoS en la arquitectura de la IoT (Duan, et. al., 2011)

2.1.3 Redes aplicadas a ciudad inteligente

Jin, Gubbi, Luo, y Palaniswami (2012), han definido cuatro arquitecturas de red diferentes para la IoT, que abarcan diversas aplicaciones de ciudad inteligente con sus correspondientes requerimientos de red en cuanto a calidad de servicio QoS las cuales se visualizan a continuación.

- Arquitectura de Red Autónoma:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

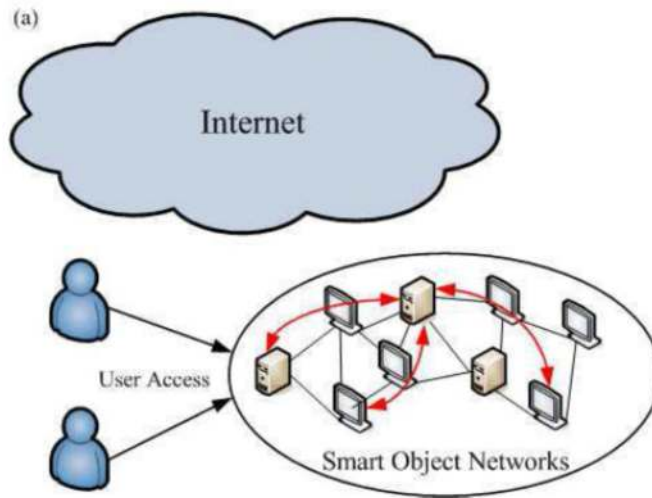


Figura 4. Arquitectura de Red Autónoma (Jin, et. al., 2012, p. 2)

- Arquitectura de Red Ubicua:

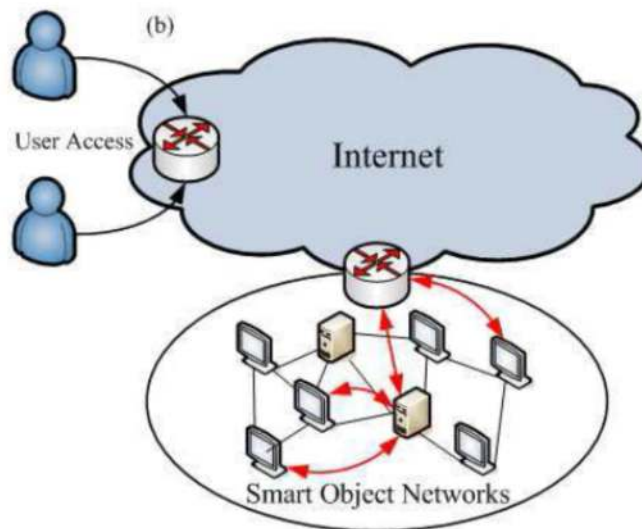


Figura 5. Arquitectura de Red Ubicua (Jin, et. al., 2012, p. 2)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Arquitectura de red con capa de aplicación superpuesta:

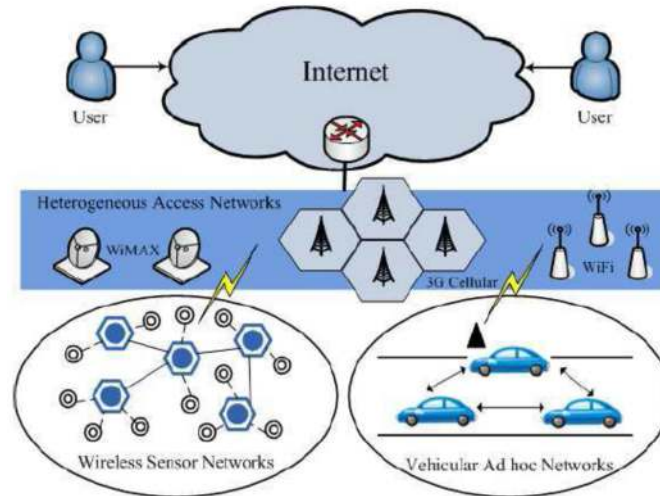


Figura 6. Arquitectura de Red con Capa de Aplicación Superpuesta (Jin, et. al., 2012, p. 3)

- Arquitectura de Red Orientada a Servicios:

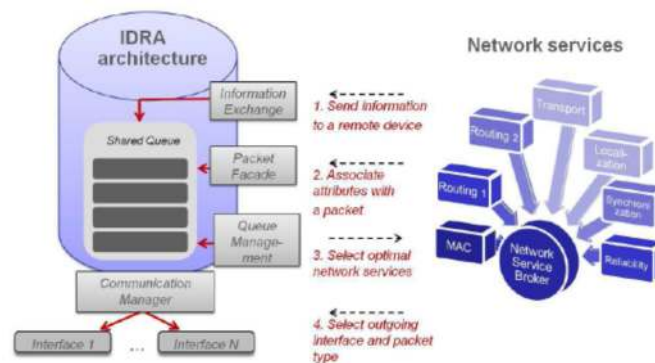


Figura 7. Arquitectura de Red Orientada a Servicios (Jin, et. al., 2012, p. 4)

2.1.4 Red de evolución a largo término avanzada

Wali y Das formularon una arquitectura para la IoT soportada en una Red Avanzada de Evolución a Largo Término (Long Term Evolution Advanced LTE-A), mediante un algoritmo de programación que utiliza la arquitectura heterogénea de la red (eNodeB macro y mini estaciones base) para reducir la tasa de colisión y el retraso del acceso causado por la densidad de los dispositivos IoT, mediante la aleatorización de su acceso sin afectar a la calidad de servicio (QoS) ofrecido a los equipos de usuario móviles (User Equipment UE), ya que LTE-A se está consolidando como una tecnología prometedora para servir al Internet de las Cosas (IoT), junto con las necesidades de los UE, debido a la alta velocidad (tasa) de datos, baja latencia y los servicios prioritarios (Wali, 2014, p. 1).

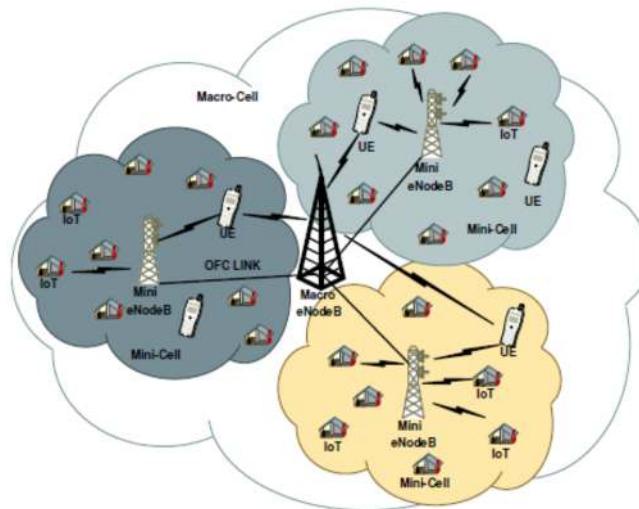


Figura 8. Marco de Trabajo de una Red LTE-A para acceso a la IoT (Wali, 2014, p. 3)

Los autores identificaron los principales retos de la incorporación de dispositivos de la IoT en la LTE-A. Tales desafíos son: el gran número de dispositivos IoT que intentan acceder a la red en tiempo corto, el efecto en la comunicación los equipos móviles de usuario existentes y por último, la necesidad de ancho de banda adicional para acomodar dichos equipos (ibídem).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Los resultados que hallaron Wali y Das, en la simulación para una red heterogénea dentro de una sencilla macro célula con 1 eNodoB macro y 4 mini estaciones base (mini células) indicaron que el marco de trabajo y la programación propuesta redujeron la tasa de colisión para los dispositivos de IoT en un 80% y el retraso de acceso en más de 50% durante el procedimiento de LTE al Canal Aleatorio de Acceso RACH (Random Access CHanel), así como el mejoramiento en la Calidad de Servicio QoS a los Equipos de Usuario móviles UE, en comparación con el esquema de restricción de acceso (Access Barring AB) (ibídem).

2.1.5 Red de tres capas orientada a servicios

En las arquitecturas expuestas se ha observado que la Calidad del Servicio es un tema de gran importancia para la implementación de las mismas, dado el gran número de diferentes dispositivos y de redes heterogéneas que hacen que sea difícil satisfacer los diversos requerimientos de QoS. Li y Zhao corroboran esto al proponer un modelo de QoS de planificación de tres capas de la IoT orientada a servicios (2014, p. 1).

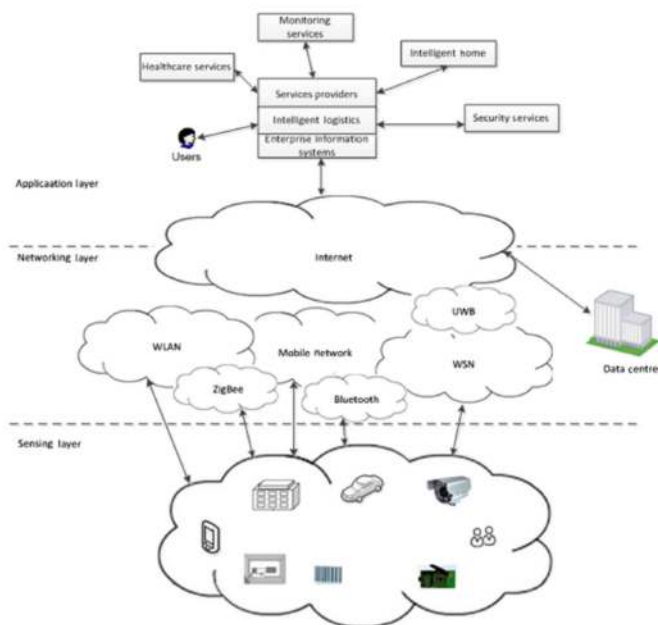


Figura 9. Arquitectura básica de la IoT (Li, et. al., 2014, p. 3)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

En la capa de aplicación, el esquema de programación de calidad de servicio explora la composición óptima de los servicios compatibles con QoS utilizando el conocimiento de cada uno de los componentes de servicio. En la capa de red, el modelo tiene como objetivo hacer frente a la programación del entorno de redes heterogéneas; en la capa de detección (percepción), se trata de la programación de adquisición de información y la asignación de recursos a diferentes servicios (ibídem, p. 1).

La calidad del servicio se optimiza al establecer procesos de toma de decisiones tanto en la capa de red como en la capa de detección, tal y como se observa en las figuras siguientes:

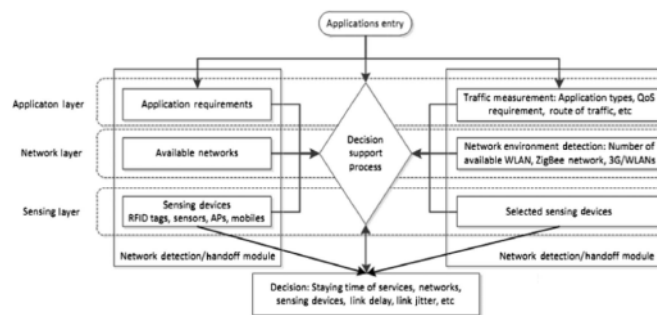


Figura 10. Decisión de QoS en la capa de red de IoT (Li, et. al., 2014, p. 4)

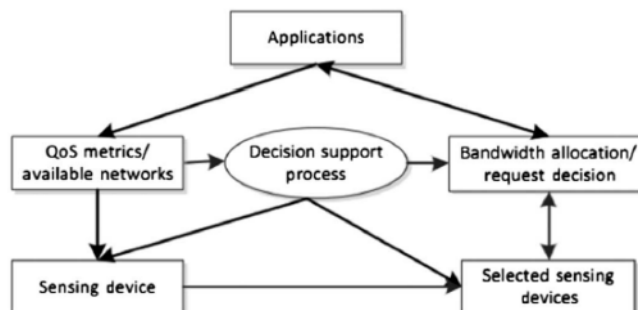


Figura 11. Decisión de QoS en la capa de detección de IoT (ibíd.)

A partir de métricas de calidad de servicio definidas en la capa de detección de IoT, como son la exactitud de la información, la precisión de la detección, el consumo de energía, el tiempo de vida de la red de percepción y el costo, se establece un módulo de monitoreo para la toma de decisiones bajo un modelo Markoviano estocástico, con lo cual se optimiza la calidad del servicio QoS en la IoT orientada a servicios. Bajo la misma política, se definen métricas para la capa de aplicación como costo de servicio, tiempo de desempeño, carga del servicio y confiabilidad. Para la capa de red siguiendo dicha política, sus métricas atañen a la asignación de ancho de banda, la capacidad de la red bajo restricciones, la tasa de retraso, de rendimiento y de distorsión y por último, la eficiencia energética (ibíd., 8).

Lo anterior les permite concluir a los autores que la tasa de éxito en la composición de la calidad del servicio decrece linealmente a medida que aumenta la cantidad de nodos en la IoT y la cantidad de servicios. Igualmente, la tasa promedio de exactitud de la información en la calidad del servicio decrece linealmente a medida que aumenta la cantidad de nodos en la IoT y la cantidad de servicios (ibídem).

2.1.6 Red de sensores inalámbricos WSN

Dado el gran desarrollo que han presentado las Redes de Detección Inalámbricas (Wireless Sensor Network WSN) y la facilidad que brindan para conectar sensores inalámbricos a la nube a través de Internet (IoT), ha llevado a un grupo de investigadores liderados por Syarifah Ezdiani en 2015 a adaptar la calidad del servicio (AQoS) de la IoT mediante una arquitectura de sistema de un banco de pruebas (testbed).

El concepto banco de pruebas con AQoS para redes de sensores en la nube, proporciona una vía para un sistema de experimentación flexible que es capaz de reaccionar a los cambios dinámicos de las condiciones de la red. Mediante el empleo de herramientas de simulador de red para obtener la información de rendimiento de la misma basado en los datos históricos de un banco de pruebas físicas, la predicción realista del rendimiento de la red se puede llevar a cabo y el ajuste de la red física puede ser propuesto (Ezdiani, 2015, pág. 1). La siguiente gráfica presenta dicho “testbed” de AQoS.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

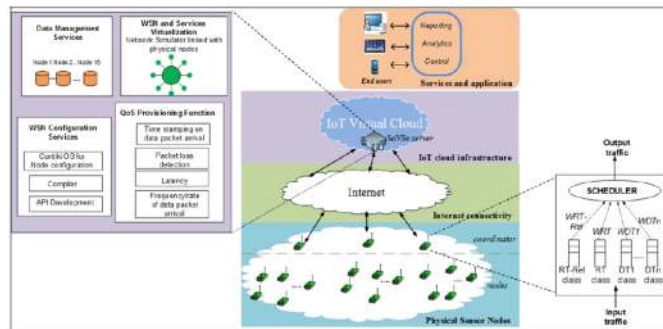


Figura 12. Arquitectura de Banco de Pruebas AQoS (Ezdiani, et. al., 2015, p. 3)

2.1.7 Red de sensores con transporte de cola de mensajes por telemetría MQTT-SN

La calidad del servicio en IoT tiene otro enfoque arquitectónico desarrollado por Govindan y Prakash (2015), consistente en una Red de Sensores con Transporte de Cola de Mensajes de Telemetría (Message Queue Telemetry Transport-Sensor Network MQTT-SN), la cual brinda un eficiente ancho de banda para el transporte de datos bajo un entorno inalámbrico no confiable en cuanto a un limitado ancho de banda.

A continuación se aprecian diversas arquitecturas MQTT-SN para que los dispositivos se comuniquen con el servidor (PUBLISH):

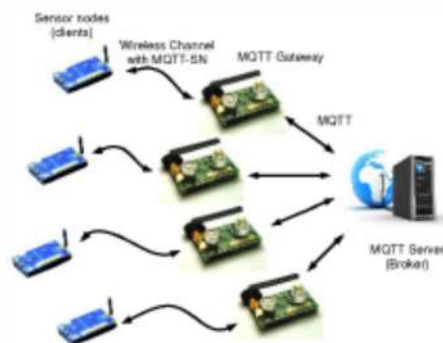


Figura 13. Puerta de enlace transparente o sencilla (Govindan, et. al., 2015, p. 2)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

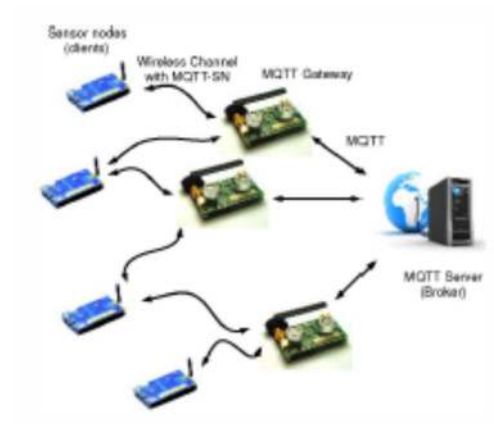


Figura 14. Puerta de enlace híbrida o distinta (ibídem)

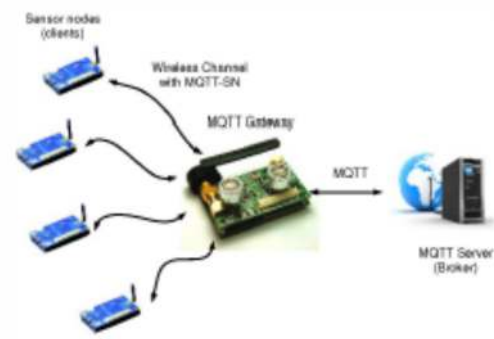


Figura 15. Puerta de enlace agregada o unida (ibíd.)

Ahora se observa el modelo de calidad de servicio de nivel 1 (QoS-1 o at-least-once) para un sistema de red extremo a extremo (end-to-end) (ibíd., pág. 3):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

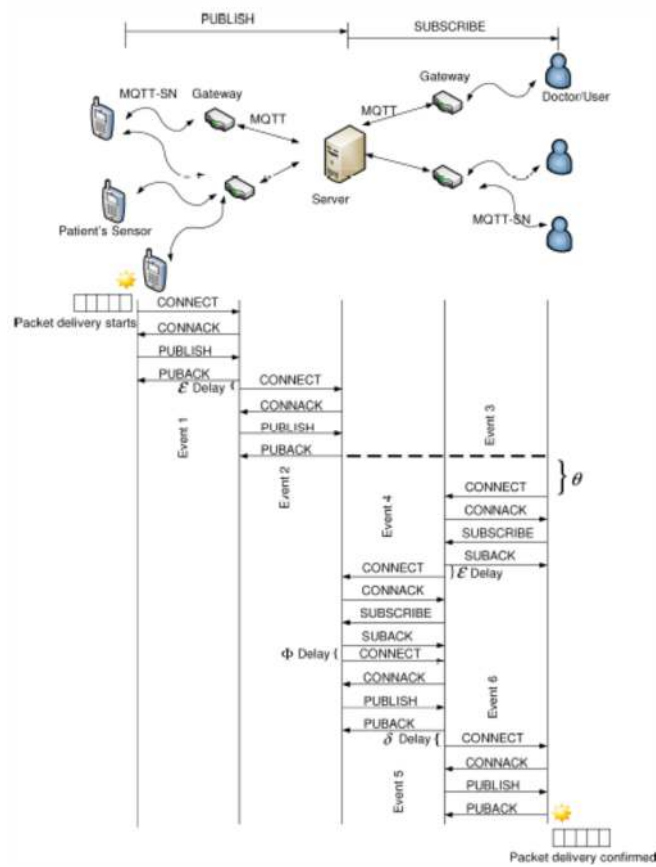


Figura 16. Modelo de calidad de servicio nivel 1 QoS-1 (Govindan, et. al., 2015, p. 3)

Se visualiza a continuación el modelo de calidad de servicio de nivel 0 (QoS-0 o at-most-once) para un sistema de red extremo a extremo (end-to-end):

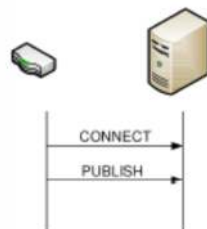


Figura 17. Modelo de calidad de servicio nivel 0 QoS-0 (ibid, pág. 5)

Por último se observa el modelo de calidad de servicio de nivel 2 (QoS-2 o exactly-once) para un sistema de red extremo a extremo (end-to-end) (ibíd.):

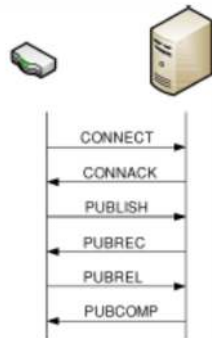


Figura 18. Modelo de calidad de servicio nivel 2 QoS-2 (ibídem)

2.1.8 Red de fibra e inalámbrica FiWi

Otro aspecto que se ha estudiado al interior de la calidad del servicio en IoT es la implementación de Redes de Fibra e Inalámbricas, (Fiber-Wireless FiWi), las cuales como una integración de la red de fibra óptica y red de acceso inalámbrico, han atraído el interés en investigación intensiva y han disfrutado de un crecimiento significativo en sus aplicaciones, en cuanto a la arquitectura de red, el aprovisionamiento de QoS, el mejoramiento en escalabilidad, fiabilidad y ahorro de energía (Liu, et. al., 2015, p. 1).

Se observa a continuación un par de arquitecturas FiWi para la IoT; una en la categoría de Radio sobre Fibra RoF y otra de tipo Radio y Fibra R&F:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

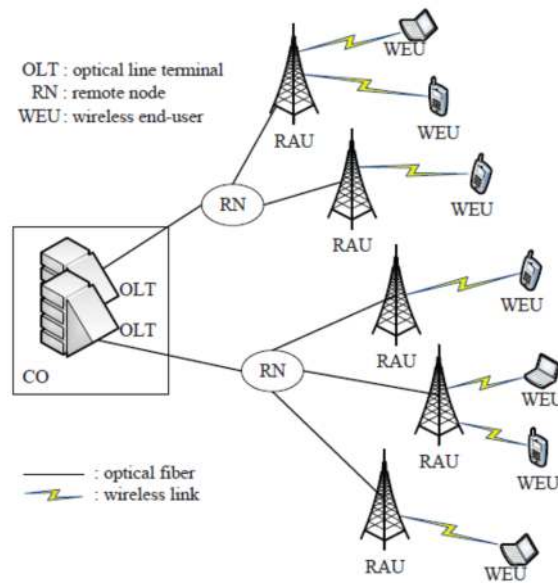


Figura 19. Integración de redes ópticas e inalámbricas en la categoría Radio sobre Fibra RoF (Liu, et. al., 2015, p. 2)

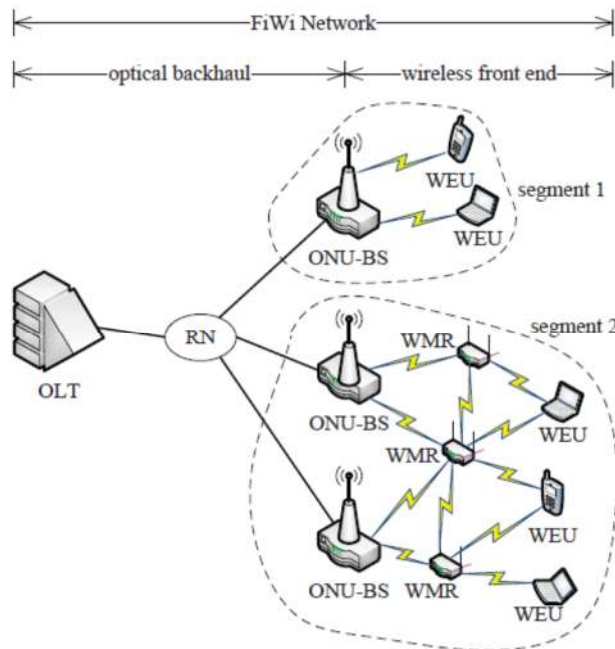


Figura 20. Integración de redes ópticas e inalámbricas en la categoría Radio y Fibra R&F (ib., p. 4)

2.1.9 Red inalámbrica definida por software

Para maximizar la implementación de la Internet de las Cosas IoT, se requiere un soporte de Calidad de Servicio (Quality of Service QoS) muy detallado que satisfaga: las necesidades diferenciadas de aplicación, la recuperación de información semántica sensible al contexto, y el despliegue rápido y fácil de los recursos, entre muchos otros objetivos. Estos objetivos sólo pueden alcanzarse si los componentes de la IoT se pueden gestionar de forma dinámica de extremo a extremo a través de: objetos heterogéneos, tecnologías de transmisión y arquitecturas de redes. Según El-Mougy et. al., las Redes Definidas por Software (Software-Defined Networking SDN) son un nuevo paradigma que proporciona potentes herramientas para abordar algunos de estos retos. Usando un plano de control basado en software, las SDNs introducen una flexibilidad significativa para la gestión de recursos y la adaptación de las funciones de red. (2015, pág. 1)

Se pueden observar a continuación los componentes de la Internet de las Cosas sugeridas por El-Mougy, et. al., con posibles arquitecturas de Redes Definidas por Software SDN:

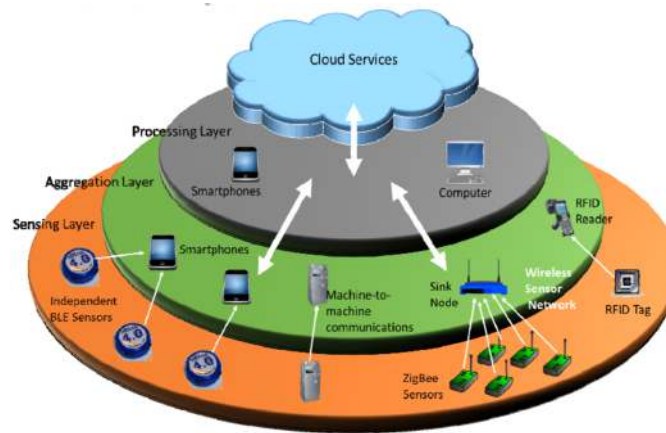


Figura 21. Componentes de la Internet de las Cosas (El-Mougy, et. al., 2015, p. 21)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

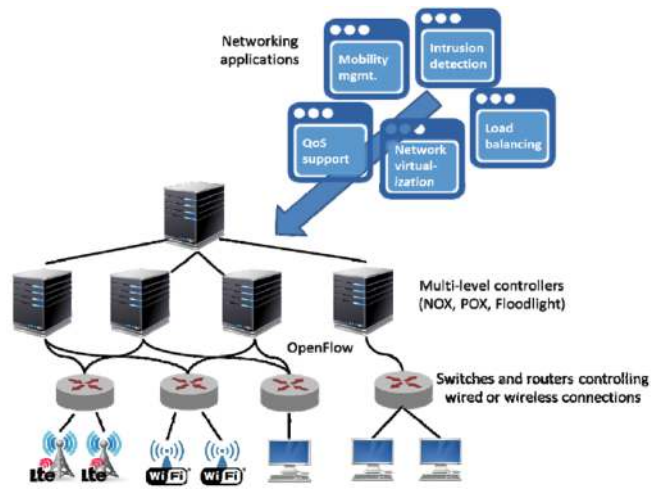


Figura 22. Componentes de las Redes Definidas por Software SDN (ibíd., pág. 3)

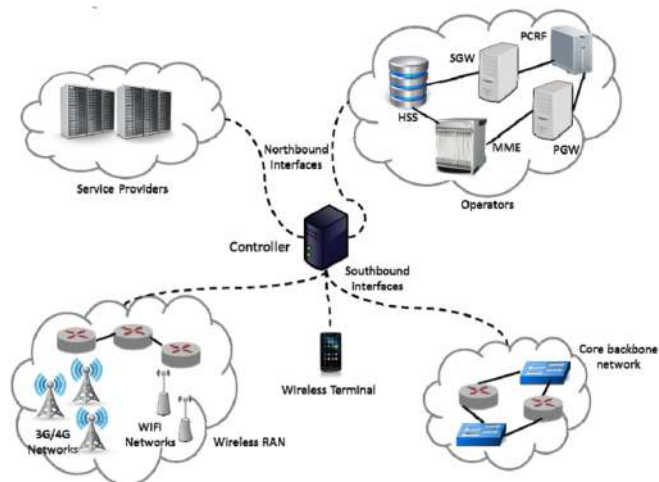


Figura 23. Arquitectura para una Red Inalámbrica Definida por Software (WSDN) (ibíd., p. 5)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

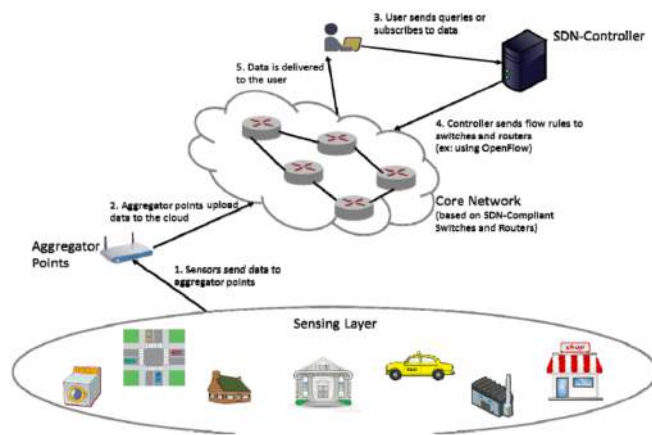


Figura 24. Arquitectura de una red definida por software SDN basada en Redes Centradas en la Información ICN para la IoT (ibíd., pág. 6)

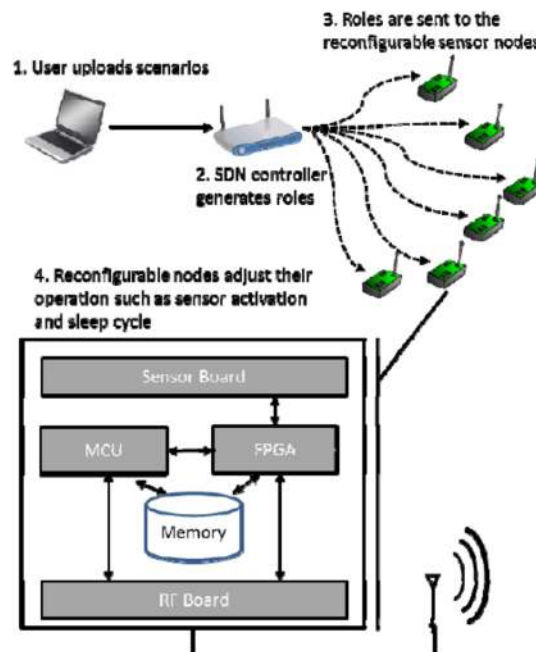


Figura 25. Reconfiguración de un nodo de sensores para gestionar la capa de detección en una Red SDN (ibídem., pág. 4).

2.1.10 Discusión de Arquitecturas Funcionales

A partir de la descripción de las diversas arquitecturas funcionales, se identifica como característica relevante la calidad de servicio (Quality of Service QoS), la cual funge como la métrica que facilita su estimación. En la tabla 2 se observa un balance de las mismas mediante las diversas características del indicador mencionado tales como el enfoque de diseño, el modelo de conectividad, la jerarquía de red, el procesamiento al interior de la red, la complejidad de la calidad de servicio y el progreso en la definición de esta última.

Se aprecia en la tabla de modelos de arquitecturas funcionales que la generalidad posee un enfoque de diseño evolutivo (75%) a excepción del modelo orientado a servicios, el de fibra FiWi y el definido por software, cuya perspectiva es de pizarra limpia (25%). Todas las arquitecturas tienen un modelo de conectividad IP ya sea total (50%) o compatible (50%), el cual es propio de la Internet. La jerarquía de red es relevante en el 75% de las arquitecturas, excepto en las orientadas a servicios mono y tri-capa, como también en la definida por software (25%).

El procesamiento al interior de la red existe en el 50% de las arquitecturas, específicamente en las que la funcionalidad por software es relevante tales como los modelos de capas (50%). En cuanto a la complejidad de la calidad de servicio los resultados son heterogéneos para las diversas arquitecturas, siendo alta en un 58%, media en un 17% y baja en un 25%. Por último, el progreso en la definición de QoS es de carácter mixto, con un 25% en etapa temprana, un 42% en intermedio y un 33% en avanzado.

Es de destacar que la arquitectura de red inalámbrica definida por software para la Internet de las Cosas establece un marco de trabajo para la conectividad de dispositivos, bajo una semántica que debe acordarse entre las diversas organizaciones de telecomunicaciones en el mundo, dando vía libre a la masificación de los servicios y funciones prestadas por las diversas máquinas inteligentes con su correspondiente gestión, dadas las características de diseño de pizarra limpia, enlace compatible con IP, jerarquía de red adaptable, calidad de servicio medianamente compleja y en etapa temprana, con un procesamiento en línea en tareas de descubrimiento, registro, monitoreo y control de dichos equipos electrónicos.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

No.	Autor	Arquitectura de red	Métrica de QoS					Progreso en la definición QoS
			Enfoque de diseño	Modelo de conectividad	Jerarquía de red	Procesamiento en la red	Complejidad QoS	
1	Inge Grønbaek	Conexión mediante API de servicios	Evolutivo	IP	Si	No	Baja	Intermedio
2	Duan R., Chen X., Xing T.	Estratificada con capa de servicios	Evolutivo	IP	Si	Si	Alta	Avanzado
3	Jin J., Gubbi J., et. al.	Autónoma	Evolutivo	Compatible con IP	Si	No	Baja	Intermedio
4	Jin J., Gubbi J., et. al.	Ubicua	Evolutivo	IP	Si	No	Alta	Intermedio
5	Jin J., Gubbi J., et. al.	Capa de aplicación superpuesta	Evolutivo	IP	Si	Si	Baja	Avanzado
6	Jin J., Gubbi J., et. al.	Orientada a servicios	Pizarra limpia	Compatible con IP	Adaptable	Si	Alta	Etapas tempranas
7	Wali P., Das D.	Evolución a largo término LTE Avanzada	Evolutivo	Compatible con IP	Si	Si	Alta	Avanzado
8	Li L., Li S., Zhao S.	Tri-capa orientada a servicios	Evolutivo	IP	Adaptable	No	Alta	Intermedio
9	Ezdiani S., Acharyya I., Sivakumar S., Al-Anbuky A.	Sensores inalámbricos WSN	Evolutivo	IP	Si	No	Alta	Avanzado
10	Govindan K., Prakash A.	Sensores con transporte de cola de mensajes por telemetría MQTT-SN	Evolutivo	Compatible con IP	Si	Si	Alta	Intermedio

11	Liu J., Guo H., et. al.	Fibra e inalámbrica FiWi	Pizarra limpia	Compatible con IP	Si	No	Alta	Etapa temprana
12	El-Mougy A., et. al.	Inalámbrica definida por software	Pizarra limpia	Compatible con IP	Adaptable	Si	Media	Etapa temprana

Tabla 2. Balance de las características de las arquitecturas funcionales a partir de la métrica de calidad de servicio QoS (Elaborada por el autor)

2.1.11 Conclusión de las de Arquitecturas Funcionales

Las arquitecturas funcionales de la Internet de las Cosas revisadas presentan –en cuanto a sus modelos– un soporte tanto en tratamiento como en transferencia de datos, siendo un 33% y un 67% respectivamente. Situación que conlleva a concentrar los esfuerzos de desarrollo en el aspecto lógico de la IoT, dado que en él se define la interoperabilidad sintáctica y semántica de la red para la conectividad de los dispositivos inteligentes.

Es posible llevar a cabo dicha mediante la definición de un marco conceptual que relacione las diversas abstracciones en un universo de discurso específico –lo cual se conoce con el nombre de ontología–, a pesar de las diversas tecnologías existentes en cuanto a tratamiento y transferencia de datos, lo que implica una revisión acerca de los distintos modelos ontológicos que atañen a las ciencias de la información y las comunicaciones como trabajo futuro.

Las arquitecturas funcionales que más se prestan para la labor de definición de un sistema conceptual concreto, son el modelo inalámbrico definido por software y el modelo de tres capas orientado a servicios, ya que permiten el alojamiento en servidores o puertas de enlace del marco de trabajo de los conceptos o clases y sus relaciones, ejecutando la ya mencionada interoperabilidad entre las diversas máquinas.

2.2 MODELOS ONTOLÓGICOS DE BASE MÁQUINA A MÁQUINA M2

Las arquitecturas de las Internet de las Cosas (IoT) facilitan la estructuración, la interacción y el funcionamiento de los componentes de dicha red de dispositivos, razón por la cual se hace necesario conocer las diversas maneras en que se puede organizar

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

formalmente tanto las entidades que configuran la IoT (Vermesan, 2013), como sus características y sus posibles relaciones entre sí dentro de un dominio específico. Esto último es lo que se conoce como “ontología” (Grønbæk, 2008).

Aproximadamente desde mediados del año 2008 a la fecha se han venido desarrollando, a partir de un marco conceptual (ontología), varias arquitecturas para la IoT desde el seno de iniciativas de proyectos de comunicaciones pertenecientes a Europa occidental, el lejano oriente y Norteamérica, las cuales se relacionan:

- European Telecommunications Standards Institute ETSI.
- International Telecommunications Union ITU
- Internet Engineering Task Force IETF
- Open Geospatial Consortium OGC
- One Machine To Machine oneM2M
- Internet of Things Architecture IoT-A

Los distintos modelos ontológicos que se integran en la arquitectura funcional de la Internet de las Cosas IoT, se detallan a continuación, mediante los artículos de investigación de la tabla 3 que son afines al presente trabajo y fungen como marco teórico del mismo:

No.	Autor	Artículo de investigación	Ontología
1	Compton, M. et. al. W3C (SSN-XG)	The SSN Ontology of the W3C Semantic Sensor Network Incubator Group	Semantic Sensor Network SSN
2	Gyrard, A. et. al.	Standardizing generic cross- domain applications in Internet of Things	Machine to Machine Measurement M3
3	oneM2M.	TS-0012 Base Ontology. Technical specification. V.0.8.0	one Machine to Machine oneM2M
4	Bermúdez-Edo, M. et. al.	IoT-Lite: A Lightweight Semantic Model for the Internet of Things	Internet of Things Lite IoT-Lite

Tabla 3. Trabajos relacionados con los modelos ontológicos de la IoT (Elaborada por el autor)

Cada modelo ontológico está soportado en un documento científico que lo describe detalladamente en cuanto a sus entidades, relaciones y restricciones al interior de un

universo de discurso específico, el cual es la tecnología Máquina a Máquina M2M aplicada al Internet de las Cosas IoT. En ellos el modelo conceptual de la red semántica de sensores SSN se halla patrocinado por la incubadora de grupos de trabajo de la W3C. Así mismo la ontología de medición máquina a máquina M3 para dominios transversales, se encuentra desarrollada por un grupo de científicos liderados por Gyrard A. Igualmente el modelo máquina a máquina monádico está construido por la alianza oneM2M, la cual consiste en la asociación de diversas organizaciones reconocidas mundialmente en el campo de las tecnologías de la información y la comunicación TIC. Por último, el esquema conceptual ligero para la semántica de la Internet de las Cosas IoT-Lite se ha elaborado por un conjunto de investigadores dirigidos por Bermúdez-Edo M.

2.2.1 Ontología de Red de Sensores Semánticos

La SSN (Semantic Sensor Network) del grupo de incubadoras de la W3C, presenta una ontología que describe sensores, observaciones, y conceptos relacionados. No se describen los conceptos de dominio, tiempo, ubicaciones, etc., que están destinados a ser incluidos en otras ontologías a través de importaciones de OWL (Compton, et. al., 2012) (W3C, 2011).

Esta ontología es desarrollada por el Grupo de Incubadoras de Redes de Sensores Semánticos del W3C (SSN-XG), quién discute los conceptos y la estructura de la misma. Dicha ontología se basa en conceptos de sistemas, procesos y observaciones. Soporta la descripción de la estructura física y de procesamiento de los sensores. Los sensores no están limitados a los dispositivos físicos de detección; más bien un sensor es cualquier cosa que puede estimar o calcular el valor de un fenómeno, por lo que un dispositivo o proceso computacional –o una combinación de ambos–, podría desempeñar el papel de un sensor. La representación de un sensor en la ontología enlaza lo que mide (los fenómenos del dominio), con el sensor físico (el dispositivo) y con sus funciones y procesamiento (los modelos) (ibídem).

La ontología está disponible como un único archivo OWL: la ontología SSN y una documentación generada de forma semiautomática derivada de ella también se proporciona como un documento independiente. Se han añadido anotaciones adicionales para dividir la ontología en "módulos" temáticos que se presentan a continuación. Para

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

que la ontología y su documentación sean más útiles, se proporcionan páginas de documentación separadas para cada módulo con fragmentos de ontología extraídos de los ejemplos desarrollados por los participantes de XG. Se incluyen cinco ejemplos trabajados para ilustrar diferentes partes de la ontología de SSN: despliegue universitario, producto inteligente, sensor de viento, meteorología agrícola y datos de sensores vinculados. Los archivos OWL para los ejemplos y para las ontologías importadas también están disponibles (ibíd.).

La ontología de la Red de Sensores Semánticos gira en torno al patrón central de estímulo-sensor-observación. Varios módulos conceptuales se construyen sobre el patrón para cubrir conceptos clave del sensor. Estos módulos se pueden ver en la Figura 26 y las relaciones entre ellos aparecen en la Figura 27 (ibíd.), que contiene una visión general de las principales clases y propiedades dentro de los módulos de la ontología.

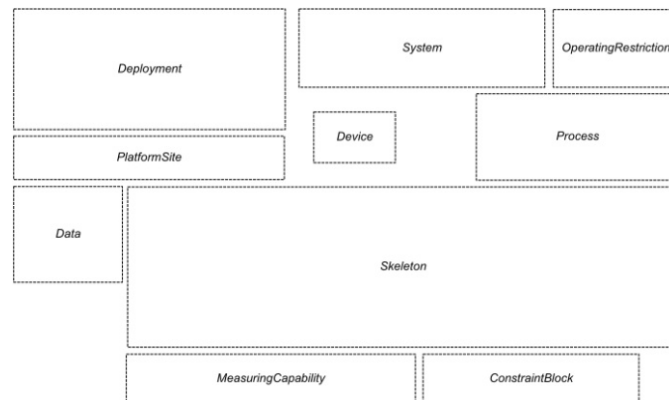


Figura 26. Módulos de la ontología SSN (Compton, et. al., 2012) (W3C, 2011)

La ontología puede utilizarse para centrarse en cualesquiera (o en una combinación) de una serie de perspectivas (ibídem):

- Una perspectiva del sensor enfocado en qué, cómo y cuándo percibe.
- Una perspectiva de datos u observación, con un enfoque en observaciones y metadatos relacionados.
- Una perspectiva del sistema, con un enfoque en sistemas de sensores.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Una perspectiva de característica y propiedad, con un enfoque en lo que se puede detectar de las mismas.

Los módulos, tal como se describen aquí, permiten refinar o agrupar más estas vistas en sensores y sensores. La descripción de los sensores puede ser detallada o abstracta. La ontología no incluye una jerarquía de tipos de sensores; Estas definiciones se dejan para los expertos de dominio, y por ejemplo podría ser una jerarquía simple o un conjunto más complejo de definiciones basadas en el funcionamiento de los sensores (ibíd.).

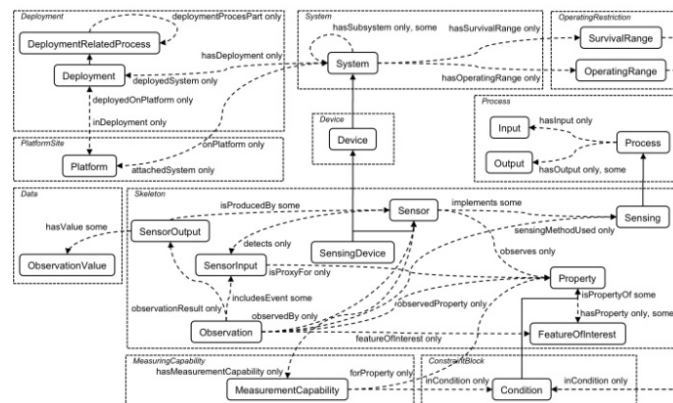


Figura 27. Clases y Propiedades de la ontología SSN (Compton, et. al., 2012) (W3C, 2011)

Los módulos contienen las clases y propiedades que pueden usarse para representar aspectos particulares de un sensor o sus observaciones: por ejemplo, sensores, observaciones, características de interés, el proceso de detección (es decir, cómo un sensor opera y observa), cómo son los sensores Las capacidades de medición de los sensores, así como sus propiedades ambientales y de supervivencia de los sensores en entornos particulares (ibíd.).

2.2.2 Ontología Machine to Machine Measurement M3

El marco de Medición Máquina a Máquina M3 ayuda a los desarrolladores en anotar semánticamente los datos M2M y en la construcción de nuevas aplicaciones por el razonamiento en los datos M2M procedentes de dominios IoT heterogéneos. El marco M3 se muestra en la Figura 28 y se compone de varias capas como sigue (Gyrard, et. al., 2014):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

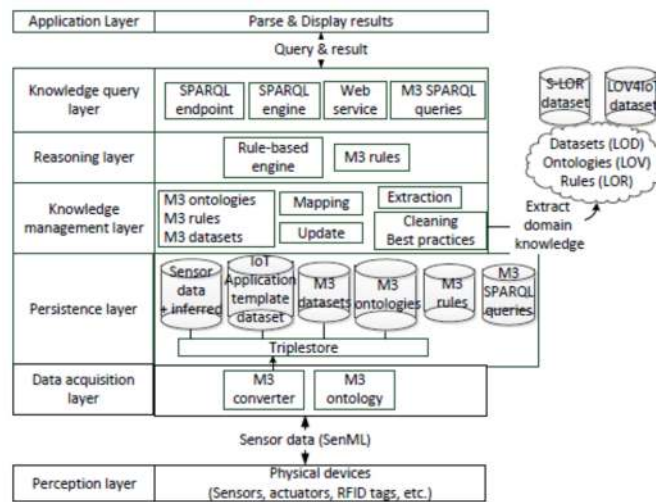


Figura 28. Marco de trabajo M3 (Gyrard, et. al., 2014)

- La capa de percepción está compuesta de dispositivos físicos como sensores, actuadores y etiquetas RFID (ibíd.).
- La capa de adquisición de datos recupera los datos de los sensores (SenML) de los dispositivos M2M y los convierte de forma unificada (RDF / XML) conforme a la ontología M3, una extensión del concepto de valor de observación SSN del W3C para proporcionar una base para el razonamiento (ib.).
- La capa de persistencia almacena ontologías de dominio M3, conjuntos de datos y datos de sensores semánticos en una tienda triple. Una tienda triple es una base de datos para almacenar datos de sensores semánticos. También almacenamos consultas y reglas de SPARQL (ibídem).
- La capa de gestión del conocimiento es responsable de encontrar, indexar, diseñar, reutilizar y combinar conocimientos específicos de dominio (por ejemplo, casa inteligente, sistemas de transporte inteligentes, etc.) como ontologías y conjuntos de datos para actualizar las ontologías, conjuntos de datos y reglas del dominio M3. Los vocabularios abiertos vinculados para IoT (LOV4IoT) son un gran conocimiento basado en ontologías de dominio, conjuntos de datos y reglas soportadas en tecnologías web semánticas que podrían ser reutilizadas para aplicaciones entre dominios (ibíd.).
- La capa de razonamiento infiere nuevos conocimientos usando motores de razonamiento y pautas M3 extraídas de las reglas abiertas ligadas a los sensores (S-LOR). Los criterios M3 son un conjunto de directrices que cumplen con la ontología M3 para obtener nuevos conocimientos sobre los datos de los sensores. Por ejemplo, con una luminosidad igual a 50000 lux, dichas reglas M3 indican que el ambiente exterior se encuentra bastante soleado (ib.).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- La capa de consulta de conocimientos ejecuta consultas SPARQL (un lenguaje similar a SQL) en datos de sensor inferidos (ibídem).
- La capa de aplicación emplea un programa (que se ejecuta en dispositivos inteligentes) que analiza y muestra los resultados a los usuarios finales. Por ejemplo, el marco de M3 sugiere dispositivos de seguridad para encender un coche inteligente de acuerdo con el pronóstico del tiempo (ibíd.).

Las descripciones uniformes mencionadas son una necesidad fundamental para desarrollar aplicaciones y servicios entre dominios. Una nomenclatura común se describe a continuación y las listas no son exhaustivas. Tales recomendaciones son relevantes para los organismos de normalización como oneM2M, ETSI M2M, W3C Web de Cosas y W3C SSN. Un ejemplo de ello se observa en la figura 29:

M3 or SenML domain	M3 or SenML sensor/ measurement name	Description, other names (synonyms)	M3 or SenML Unit
Weather	HumiditySensor/ Humidity	Hygrometer, humidity sensor, moisture sensor, soil moisture probes	Percent
Weather	WindDirectionSensor/ WindDirection	Wind direction	DegreeAngle
Weather	SunPositionDirectionSensor/ SunPosition	sun position direction to detect east, west, south, north	DegreeAngle
Weather	AtmosphericPressureSensor/ AtmosphericPressure	Atmospheric pressure sensor, Barometer, barometric pressure sensor	Pascal
Weather	CloudCoverSensor/ CloudCover	Cloud cover sensor	Okta
Weather	SunPositionElevationSensor/ SunElevation	sun position elevation to detect (twilight, day, night, etc.)	DegreeAngle
Weather	SolarRadiationSensor/ SolarRadiation	Solar radiation sensor, par (photo synthetically active radiation) sensor, sun light, solar sensors, sun's radiation intensity	WattPerMeterSquare
Weather	VisibilitySensor/ Visibility	Visibility sensor to detect fog	Miles, Meter
Weather	Thermometer, AirThermometer/ Temperature	Thermometer, temperature sensor, thermistor	DegreeCelsius
Weather	LightSensor/ Luminosity	Light, luminosity, illuminance, lighting	Lux
Weather	PrecipitationSensor/ Precipitation	Precipitation sensor, rainfall sensor, rain fall, pluviometer, rain, rainfall gauge	MilimeterPerHour
Weather	WindSpeedSensor/ WindSpeed	Wind speed sensor, wind velocity sensor, anemometer	MeterPerSecond

Figura 29. Descripción uniforme para sensores en un dominio de clima (Gyrard, et. al. 2014)

2.2.3 Ontología oneM2M

One Machine To Machine oneM2M es una iniciativa de asociación entre organizaciones líderes en el mundo en cuanto al tema de las TIC, la cual tiene como objetivo el desarrollo de especificaciones técnicas que abordan la necesidad de un capa común de servicios M2M, que se puede incrustar fácilmente dentro de una variedad de hardware y software, confiándole la conexión a una gran cantidad de dispositivos con los servidores de aplicaciones M2M a nivel global mediante Internet (oneM2M, 2018, pág. 2).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

El modelo de arquitectura planteada por oneM2M representa el modelo en capas de soporte de extremo a extremo (Extreme to Extreme E2E Services) M2M. Este modelo se compone de tres capas: la capa de aplicación, la capa de servicios comunes y la capa subyacente de servicios de red, como se observa en la figura 30 (ibídem, pág. 18).

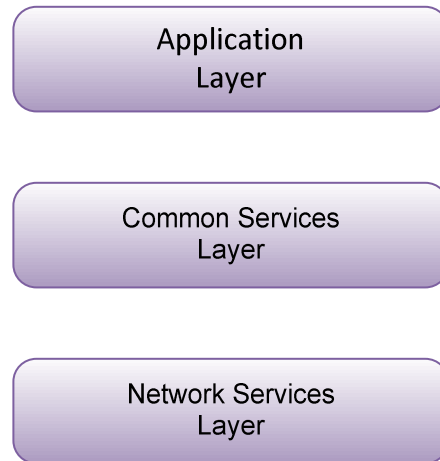


Figura 30. Modelo de capas oneM2M (oneM2M, 2018, pág. 18)

La arquitectura funcional de oneM2M, en la figura 31, comprende las siguientes funciones:

1. Entidad de aplicación (AE): es una función en la capa de aplicación que implementa una lógica de servicios de aplicaciones M2M. Cada lógica de servicio de aplicación puede ser residente de un número de nodos M2M o también de un solo nodo M2M. Cada instancia de ejecución de una lógica de servicio de aplicación se denomina "entidad de aplicación" (AE) y se identifica con un AE-ID único (ibíd. pág. 19). Los eventos que maneja esta entidad de aplicación incluyen instancias de seguimiento, de monitoreo remoto, de medición y de control.
2. Entidad de Servicios Comunes (CSE): Una entidad de servicios común representa una instancia de un conjunto de funciones de servicio "frecuentes" de los entornos M2M. Tales funciones de servicio están expuestas a otras entidades a través de los puntos de referencia o interfaces Mca y Mcc. El punto de referencia Mcn se utiliza para acceder a entidades subyacentes de servicios de red. Cada entidad de servicio común se identifica con una única CSE-ID (ib.).
 - Ejemplos de funciones de servicio ofrecidos por el CSE incluyen: gestión de datos, gestión de dispositivos, servicios de administración de suscripciones M2M y servicios de localización. Tales "sub-funciones" ofrecidas por un CSE pueden ser lógicas e informativamente conceptualizados como funciones de servicios comunes CSF. Los recursos normativos que implementan las funciones de servicio en un CSE pueden ser obligatorios u opcionales (ibídem).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

3. Entidad de Servicios de Red (NSE): Esta función provee servicios de la red subyacente a las entidades centrales de almacenamiento. Ejemplos de estos servicios incluyen la gestión de dispositivos, servicios de localización y dispositivo de desencadenamiento (ibíd.). Ninguna organización en particular asume las funciones de esta entidad, dado que trabaja con la pila de protocolos del modelo TCP/IP.

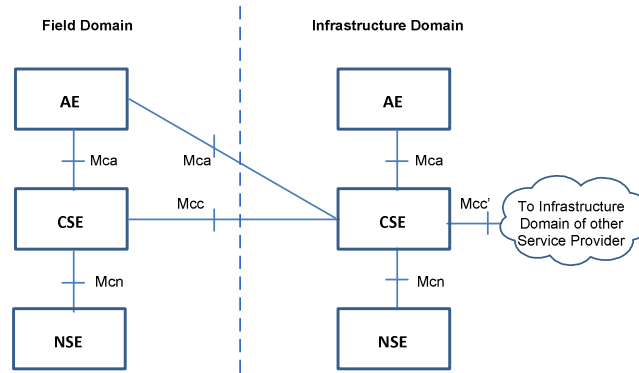


Figura 31. Arquitectura funcional de oneM2M (oneM2M, 2018, pág. 18)

Los puntos de referencia son conocidos con el nombre de interfaces entre las diferentes entidades M2M (Guevara, 2017). Los siguientes puntos de referencia están respaldados por los servicios entidad común (CSE). La nomenclatura “Mc(-)” se basa en los mnemotécnicos “comunicaciones M2M” (óp. cit. Pág. 19):

- a) Punto de referencia Mca: La comunicación fluye entre una entidad de aplicación (AE) y una entidad de servicios comunes (CSE) entre un campo y una infraestructura de dominio. Estos flujos permiten a la AE usar los servicios admitidos por el CSE para la comunicación con otra AE (ibídem).
- b) Punto de referencia Mcc: La comunicación fluye entre dos entidades de servicios comunes (CSE) cruzando desde un campo a una infraestructura de dominio. Estos flujos permiten a un CSE usar los servicios admitidos por otro CSE (ib.).
- c) Punto de referencia Mcn: La comunicación fluye entre una entidad común Servicios (CSE) y una entidad de servicios de red (NSE) dentro de un mismo campo o infraestructura de dominio. Estos flujos permiten a un CSE utilizar los servicios soportados (distintos de los servicios de transporte y conectividad) proporcionados por el NSE (ibídem).
- d) Punto de referencia Mcc': La comunicación fluye entre dos entidades de servicios comunes (CSE) en diferentes nodos de infraestructura (IN), siendo compatibles con la arquitectura oneM2M y residenciados en diferentes dominios M2M de otros proveedores de servicios SP. Estos flujos permiten un CSE de entrada de un residente en la infraestructura de dominio de un proveedor de servicios M2M para comunicarse con un CSE de otro SP en la infraestructura de dominios de un diferente proveedor de servicios M2M utilizando sus servicios soportados y viceversa. El Mcc' amplía la

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

accesibilidad de los servicios ofrecidos por encima del punto de referencia Mcc, o un subconjunto de los mismos. El detonante de estos flujos de comunicación puede ser iniciado en la red oneM2M o en cualquier otro lugar (ibíd.).

El sistema oneM2M para ser comprendido en su totalidad, requiere de una ontología de Base que ha sido diseñada con la intención de proporcionar un número mínimo de conceptos, relaciones y restricciones que son necesarias para el descubrimiento semántico de las entidades en dicho sistema. Para hacer este tipo de entidades detectable en el Sistema oneM2M es necesario una descripción semántica como clases (conceptos) en una tecnología Proveedor/Estándar específico de su ontología; de tal manera que estas clases (conceptos) estén relacionados con algunas clases de la ontología de base como subclases (oneM2M, 2016, pág. 11).

Además, la ontología de base permite tecnologías no oneM2M para construir ontologías derivadas, que describen el modelo de datos de dicha tecnología, con el propósito de que funcionen entre sí con el Sistema oneM2M (ibídem).

La ontología de base sólo contiene las clases y propiedades pero no instancias, debido a que tal ontología y los modelos derivados se utilizan en oneM2M, sólo para proporcionar una descripción semántica de las entidades que contienen (ibíd.).

La instanciación, es decir, los datos de las diferentes entidades representadas en el Sistema oneM2M –por ejemplo, los dispositivos, las cosas, etc.– se realiza a través de los recursos oneM2M (id.).

En la figura 32 se aprecia el esquema general de la ontología de base oneM2M en la que, su columna vertebral atañe a las entidades o clases “Cosa” (Thing), “Dispositivo” (Device) y “Servicio” (Service) (ibíd.).

The oneM2M Base Ontology

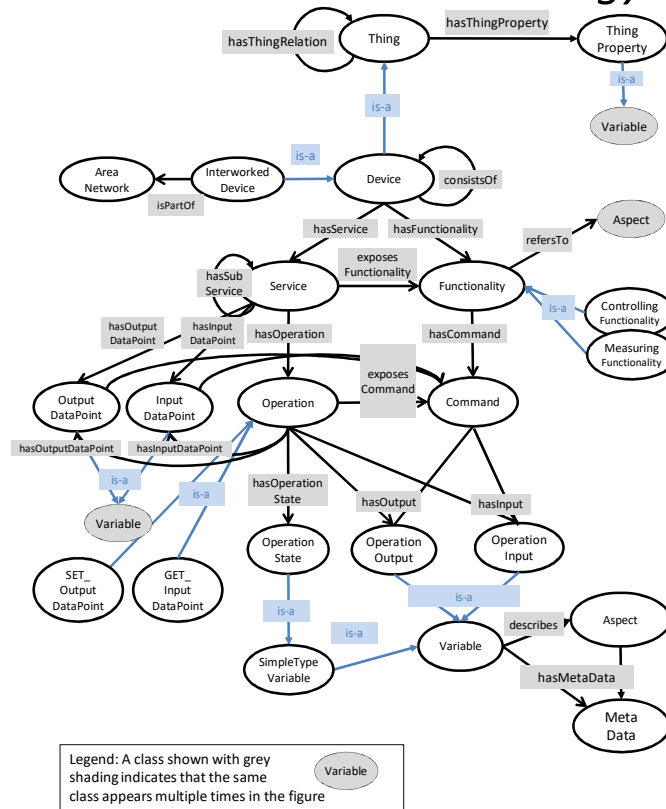


Figura 32. Ontología Base oneM2M (oneM2M, 2016, pág. 12)

2.2.4 Ontología IoT-Lite

IoT-Lite es una ontología ligera que representa los recursos, entidades y servicios de la Internet de las Cosas. Dicha simplicidad permite la representación y el uso de plataformas IoT sin consumir un exceso de tiempo de procesamiento al consultar la ontología. Sin embargo, también puede comportarse como un metamodelo que se puede extender para representar diversos conceptos de la IoT de una manera más detallada en diferentes dominios (Bermúdez-Edo, et. al., 2017).

Esta ontología describe los conceptos de IoT en tres clases: objetos, sistema o recursos y servicios. Los dispositivos también se dividen en –aunque no se limitan a estas– tres clases: dispositivos de detección, dispositivos de accionamiento y dispositivos de etiqueta. Los servicios se describen con un control de disponibilidad o de acceso y una cobertura.

2.2.5 Discusión de las Ontologías

Al interior de las diversas ontologías de base revisadas, es de resaltar que la calidad del servicio (Quality of Service QoS) entendida como el rendimiento de un sistema informático a nivel de tratamiento de datos, –i. e. la interoperabilidad sintáctica y semántica de los dispositivos inteligentes al estar conectado en red–, funge como una métrica adecuada y conveniente para la evaluación de las mismas a partir de los aspectos más sobresalientes que las definen en general y particular, como son la organización por capas, la estructura por módulos, el formateo de los datos, el patrón ontológico, la consulta de grafos, las reglas sintácticas y de inferencia.

En la tabla 4 que atañe a la métrica QoS de las ontologías de base se aprecia que el 100% de los modelos se caracterizan por poseer una compatibilidad semántica y el facilitar consultas de grafos tipo RDF. El 50% de las mismas poseen una organización por capas múltiple (M3 y oneM2M), coincidiendo en la proporción en cuanto la organización por módulos (SSN y oneM2M) y facilitar la obtención de reglas de inferencia (M3 y oneM2M). En cuanto al formateo de los datos, el 75% de los modelos lo hace en el lenguaje de ontologías web OWL a excepción del esquema M3, que lo realiza únicamente en RDF/XML, llamando la atención el que oneM2M lo lleva a cabo en ambos tipos de sintaxis; siendo esto de conformidad con la propiedad de interoperabilidad sintáctica, en la cual el 50% de dichos modelos se hallan en una etapa mediana de consolidación (SNN y IoT-Lite), en etapa temprana para M3 y en etapa avanzada para oneM2M.

En cuanto al formateo de los datos, el 75% de los modelos lo hace en el lenguaje de ontologías web OWL a excepción del esquema M3, que lo realiza únicamente en RDF/XML, llamando la atención el que oneM2M lo lleva a cabo en ambos tipos de sintaxis; siendo esto de conformidad con la propiedad de interoperabilidad sintáctica, en la cual el 50% de dichos modelos se hallan en una etapa mediana de consolidación (SNN y IoT-Lite), en etapa temprana para M3 y en etapa avanzada para oneM2M.

El patrón o plantilla implementado por cada ontología de base es singular, siendo el 75% enfocado en los aspectos del servicio y el dispositivo salvo el modelo M3; al igual que en lo tocante a la característica de entidad u objeto, excluyendo al esquema SSN. Es destacable por todo que el modelo ontológico oneM2M establece diferencia entre “cosa” y

“dispositivo”, al considerar este último como un agente inteligente capaz de tratar y/o transferir datos sin el concurso humano, mientras que el primero consiste en una entidad o elemento con propiedades y relaciones en un universo de discurso específico, mas no presenta incidencia alguna de tipo informático al interior del ámbito en que discurre.

No.	Autor	Ontología de base	Métrica de QoS							
			Estratificación	Modularización	Formato de los datos	Patrón	SPARQL	Compatibilidad semántica	Reglas de inferencia	Interoperabilidad sintáctica
1	W3C	SSN – Semantic Sensor Network	Monocapa	Si	OWL	Estimulo– Sensor– Observación	Si	Si	No	Etapa mediana
2	Gyrard, M. et. al.	M3 – Machine to Machine Measurement	Multicapa	No	RDF/XML	Registro– Razonamiento– Aplicación	Si	Si	Si	Etapa temprana
3	ETSI, et. al.	oneM2M – One Machine to Machine	Multicapa	Si	OWL RDF/XML	Cosa– Dispositivo– Servicio	Si	Si	Si	Etapa avanzada
4	Bermúdez-Edo, M. et. al.	Internet of Things Lite – IoT-Lite	Monocapa	No	OWL	Objetos– Recursos– Servicios	Si	Si	No	Etapa mediana

Tabla 4. Balance de las características de los modelos ontológicos de base según la métrica de calidad de servicio QoS (Elaborada por el autor)

2.2.6 Conclusión acerca de las Ontologías

Los modelos ontológicos descritos (M3, OneM2M, SNN y Lite) se caracterizan por considerar la estratificación a nivel de la arquitectura funcional (aplicación, de servicios y de red), en un 100%, incluyendo subcapas al interior de cada una de las mencionadas en un 50% –específicamente las ontologías M3 y oneM2M– sin comprometer la estructura modular y/o interoperabilidad sintáctica y semántica. Se infiere por lo tanto que la ontología más apropiada u oportuna es la OneM2M, dado que se encuentra en

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

permanente producción, actualización, definición y mantenimiento de las especificaciones y reportes técnicos enfocados en la capa de servicio de la tecnología M2M.

Así mismo, el hecho de que las organizaciones europeas, norteamericanas y australianas líderes en desarrollo de estándares en Tecnologías de la Información y la Comunicación (ICT, CSSA, TTA, ARIB, TTC, ETSI, ATIS) se hayan asociado desde el año 2014 en el proyecto OneM2M, ello se convierte en carta de garantía para aseguramiento del despliegue de los sistemas Máquina a Máquina en el futuro cercano.

Además, OneM2M se halla trabajando los aspectos de privacidad, seguridad, descubrimiento, interoperabilidad, suscripción, notificación, accesibilidad y gestión remota de dispositivos y aplicaciones a partir de casos de uso generales.

La ontología OneM2M, al reunir las propiedades relevantes de la tecnología Máquina a Máquina M2M de la ETSI con las características de otros modelos de referencia para dispositivos inteligentes como v.g. SAREF, logra establecer un alto grado de especificidad en las entidades y su conceptualización (Dispositivo, Servicio, Operación, Orden, Entrada, Salida, Método, Destino, Valor, Funcionalidad, Cosa), superando las diferenciaciones entre definiciones y abriendo el camino a la estandarización de la arquitectura.

A partir de una ontología robusta (OneM2M) y una estructura funcional de la tecnología M2M soportada en las capas ya mencionadas, la iniciativa de una arquitectura para el Internet de las Cosas coincide con el modelo de sistema informático de tres niveles (presentación, negocio y persistencia), en la que la capa de servicio (equivalente a la de negocio) y manejando la ontología mencionada, permite integrar los diversos dispositivos para su gestión así:

- Dispositivos de suscripción y notificación mediante MQTT.
- Dispositivos restringidos mediante CoAP
- Dispositivos autónomos mediante REST

De esta manera, cualquier dispositivo o máquina está facultado para conectarse permanentemente a Internet ya sea de manera directa (GSM) o indirecta (Gateway) y enviar peticiones y recibir respuestas en el momento apropiado (suscribe – publish), lo cual facilita su implementación en una gran cantidad de contextos.

2.3 MODELO ARQUITECTONICO FUNCIONAL M2M-IOT PROPUESTO

2.3.1 Definición de Ontología

Una ontología es una especificación explícita de una conceptualización (Gruber, 1993), siendo esta una visión abstracta y simplificada de un entorno que se desea representar para un propósito definido; facilitando la construcción de un cuerpo de conocimiento formalmente representado de manera declarativa mediante objetos, propiedades y relaciones descriptibles entre ellos, en un universo de discurso determinado (Liu, 2009) llamado también dominio, por lo que –expresa y concretamente–, una ontología es la manifestación estructurada de una teoría lógica (Gruber, 1995).

2.3.2 Arquitectura funcional de base ontológica

En la tabla número 5 se pueden observar las ontologías que conforman el modelo arquitectónico propuesto para la Internet de las Cosas IoT, la cual ha sido elaborada por el autor.

Ontologías de la arquitectura IoT propuesta		
Sigla	Significado	Traducción
oneM2M	One Machine To Machine	Máquina a Máquina Monádica
SAREF	Smart Appliances REFerence	Referencia de Dispositivos Inteligentes
SAREF4ENER	Smart Appliances REFerence For Energy	Referencia de Dispositivos Inteligentes para Energía
OM	Ontology of units of Measure	Ontología de unidades de Medida

Tabla 5. Ontologías que conforman el modelo arquitectónico propuesto para la Internet de las Cosas IoT (Elaborada por el autor)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

El modelo conceptual de una arquitectura de la Internet de las Cosas IoT propuesto se compone de las cuatro ontologías enunciadas en la tabla anterior (No. 5), siendo su esquema correspondiente el que se observa en la figura No. 35:

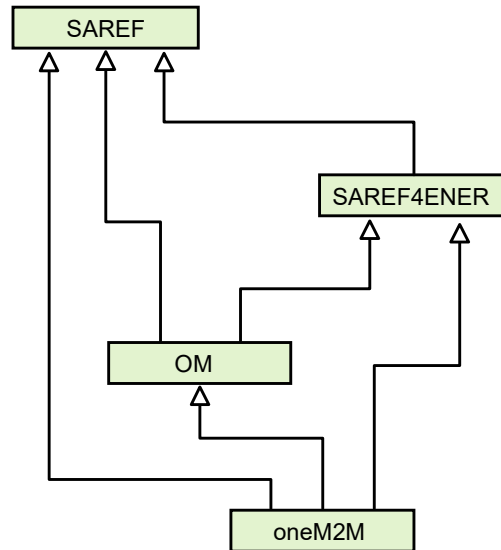


Figura 35. Modelo conceptual de una arquitectura de la IoT propuesto (Elaborado por el autor)

Dichas ontologías están desarrolladas y soportadas por Organizaciones reconocidas en el sector de las TIC, siendo cada una de ellas conformada por conceptos específicos con sus características y asociaciones distintivas dentro de un universo de discurso o dominio determinado, según se aprecia en la tabla 6.

Ontología	Universo de Discurso (Dominio)	Autor
oneM2M	Sistema M2M (Maquina a Maquina)	Alianza oneM2M
SAREF	Dispositivos inteligentes	TNO, ETSI
SAREF4ENER	Energía (Consumo/Producción)	TNO, ETSI
OM	Unidades de medida	Wageningen UR

Tabla 6. Dominio y desarrollador de las ontologías del modelo arquitectónico propuesto (Elaborado por el autor)

A continuación se describen de manera detallada las diversas ontologías que componen el modelo arquitectónico propuesto para la Internet de las Cosas IoT.

2.3.3 Ontología de unidades de Medida OM

La ontología de unidades de medida (en inglés *Ontology of units of Measure OM*), versión 2.0, es un esquema de conceptos y de relaciones esenciales para la investigación científica (Wageningen, 2016), el cual presenta un fuerte enfoque en unidades, cantidades, medidas y dimensiones.

Dichas nociones se organizan principalmente como se aprecia en la figura 2 (ibíd.), las cuales se comentan a continuación:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

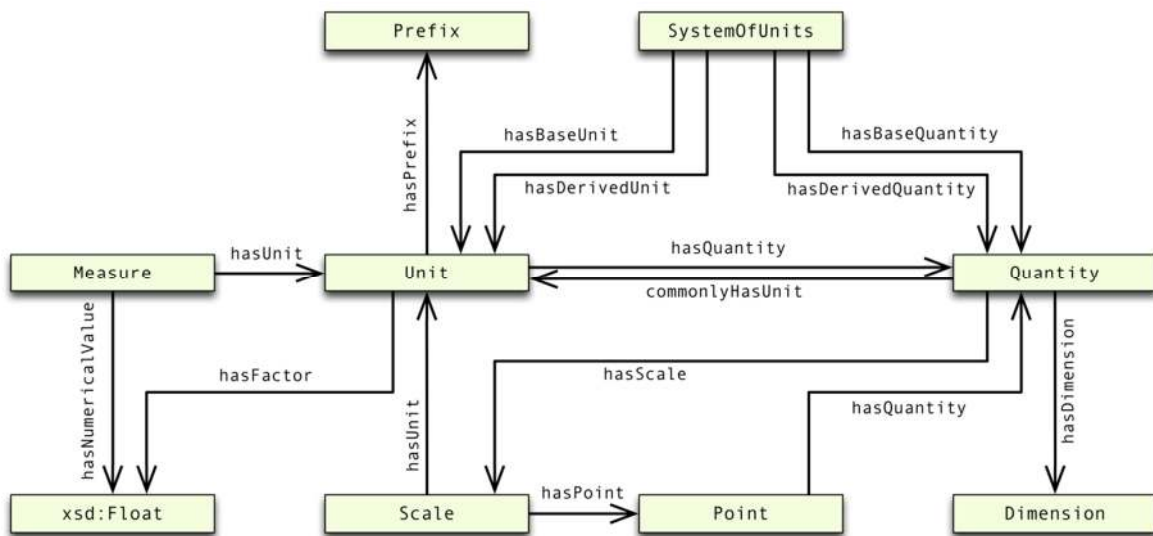


Figura 36. Estructura conceptual básica de la ontología OM (Wageningen, 2016)

- Sistema de unidades (SystemOfUnits): Un sistema de unidades se basa en un conjunto de magnitudes o patrones (Wageningen, 2013) elegidas por convención con el fin de que sean los elementos base de un sistema para comparación, considerándose estas mutuamente independientes, a fin de lograr un conjunto coherente e interdependiente de unidades de medida entre la alta diversidad de las mismas existentes.
 - Dentro de los diversos sistemas de unidades considerados por OM se ha elegido para la presente investigación el Sistema Internacional de Unidades SI, emitido por la Oficina Internacional de Pesos y Medidas BIPM de Francia (BIPM, 2006) y refrendado por el Instituto Nacional de Estándares y Tecnología NIST de Estados Unidos (NIST, 2008).
- Unidad (Unit): Una unidad de medida es una magnitud o canon definido de una cantidad, establecido como criterio o estándar para la cuantificación de la misma (Wageningen, 2013), donde cualquier otro valor de dicha cantidad se puede expresar como un múltiplo simple de la unidad en mención.
 - EL concepto de unidad está soportado en el estándar oficial emitido por la BIPM y respaldado por el NIST, estableciendo tanto las unidades fundamentales (BIPM, 2006, p. 26) y (NIST, 2008, p. 4) –v.g. el metro, el kilogramo, el segundo, el amperio, etc.–, como las derivadas (óp. cit., p. 28) y (ob. cit., p. 5) –p. ej. julio, vatio, culombio, voltio, kilovatio-hora, miliamperio-hora, etc.–, las cuales corresponden al SI o Sistema Internacional de unidades.
- Cantidad (Quantity): Una cantidad es una expresión numérica de un aspecto estandarizado, de un fenómeno o de una propiedad (Wageningen, 2013), clasificándose según su metrología como magnitud fundamental (BIPM, 2006, p. 26) y (NIST, 2008, p. 4) –p.ej. la Longitud, la Masa, el Tiempo, la Intensidad de Corriente Eléctrica, etc. –, o como magnitud derivada (óp. cit., p. 28) y (ob. cit., p. 5) –v.g.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Energía, Potencia, Carga eléctrica, Potencial eléctrico, etc.—, siendo estas pertenecientes al Sistema Internacional de unidades SI.

-
- Medida (Measure): Una medida combina un valor numérico con una unidad de medida, con un intervalo o con una escala proporcional (ibídem). Por ejemplo, "3 m" es una medida que junta el número "3" con la unidad "metro".
-
- Dimensión (Dimension): Una dimensión es la propiedad abstracta correspondiente a una unidad o una cantidad (ibíd.), sin considerar su carácter vectorial o tensorial, además de todos los posibles factores numéricos incluido su signo.
 - Al igual que los conceptos de cantidad y unidad, el concepto de dimensión está soportado en el estándar oficial emitido por la BIPM y respaldado por el NIST, estableciéndose según su metrología como magnitudes fundamentales (BIPM, 2006, p. 15) y (NIST, 2008, p. 22) de Longitud, Masa, Tiempo, Intensidad de corriente eléctrica, Temperatura, Cantidad de sustancia y por último, Intensidad luminosa, las cuales conforman la base del SI o Sistema Internacional de unidades.
-
- Escala (Scale): Una escala de medición es un concepto utilizado para la expresión de una cantidad. Se han definido cuatro tipos de escala de medición, las cuales son: nominal, ordinal, de intervalo y de proporción (óp. cit.). Las dos últimas escalas son llamadas escalas cardinales. Un ejemplo de una escala es la escala Celsius, la cual consiste en una escala de temperatura.
-
- Punto (Point): Un punto es un elemento de una escala de intervalo o de una escala de proporción (ibídem); v.g. el valor 273,16 en la escala Kelvin indica el punto triple de la temperatura termodinámica del agua.
-
- Prefijo (Prefix): Un prefijo es un nombre que precede a una unidad de medida básica (ibíd.), el cual facilita la indicación de un múltiplo o fracción decimal de la dicha magnitud o canon, caracterizándose mediante un símbolo único que se antepone al símbolo de la unidad en cuestión; por ejemplo. una corriente eléctrica de 0,000 000 001 amperios se escribe usando el prefijo "nano" del Sistema Internacional de unidades SI, como 1 nanoamperio o 1 nA.
-
- Número de coma flotante (xsd:Float): Un número de coma flotante es un tipo de dato primitivo perteneciente al Sistema de Tipos de Datos del Schema XML, conocido también como el Lenguaje de Definición de Esquema para XML (eXtensible Markup Language) desarrollado por la W3C, el cual es implementado por la ontología OM como el espacio de nombre (namespace) para el valor numérico real de precisión simple (W3C, 2012) de 32 bits definido por la IEEE. Su espacio o intervalo de valores consiste en un subconjunto de los números racionales, cuya utilidad estriba con frecuencia en aproximar valores decimales arbitrarios (ibídem).
 - Es de tener en cuenta que la IEEE ha establecido que un número de coma flotante (IEEE, 2008, p. 4) es un dato correspondiente a la expresión cuantificacional de una magnitud, tanto finita como infinita (incluyendo el valor indeterminado NaN), incluyendo su carácter positivo o negativo (signo); el cual es representable en un formato de tres componentes: un signo, un exponente y un significante; su valor

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

numérico es el producto –con signo– de su significante y su raíz o base elevada a la potencia de su exponente (ibíd., p. 7).

Las relaciones entre dichos conceptos se resumen de la siguiente manera a partir de la figura 36 (Wageningen, 2016):

Concepto (o clase) de dominio	Relación (o propiedad de objeto)	Concepto (o clase) de recorrido
Sistema de Unidades	Tiene una unidad fundamental	Unidad
	Tiene una unidad derivada	
	Tiene una cantidad fundamental	Cantidad
	Tiene una cantidad derivada	
Unidad	Tiene una cantidad	Cantidad
	Tiene un factor	xsd:Float
	Tiene un prefijo	Prefijo
Cantidad	Tiene una unidad (a menudo)	Unidad
	Tiene una dimensión	Dimensión
	Tiene una escala	Escala
Medida	Tiene una unidad	Unidad
	Tiene un valor numérico	xsd:Float
Escala	Tiene una unidad	Unidad
	Tiene un punto	Punto
Punto	Tiene una cantidad	Cantidad

Tabla 7. Relaciones básicas entre conceptos de la ontología OM (ibíd.)

La Ontología de unidades de Medida OM ha sido escogida como marco conceptual para la presente investigación, sin pretender restarle importancia a ontologías que igualmente se enfocan en metrología como EngMath, UCUM, MUO, QUDT, etc. No obstante, dado que OM es una ontología del dominio de cantidades, unidades, dimensiones y medidas,

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

dicho esquema igualmente permite que los datos cuantitativos de cualquier trabajo científico puro o aplicado sean más explícitos, con el propósito de que los mismos puedan integrarse, verificarse y reproducirse (Rijgersberg, 2013, p. 1). Para enfatizar lo anterior, el diseño de OM se ha comparado con la ontología QUDT (Quantities, Units, Dimensions and Data Types), la cual es otro esfuerzo activo de modelado en el Lenguaje de Ontologías Web OWL para el mismo ámbito (ibídem).

Es de tener en cuenta que la ontología de Cantidades, Unidades, Dimensiones y Tipos de Datos QUDT, ha sido desarrollada por la organización del mismo nombre, con el fin de suministrar especificaciones semánticas en metrología (QUDT, 2017), implementando estándares para cuantificar los datos expresados en RDF y JSON, mejorando la interoperabilidad de aquellos y la determinación precisa de estructuras de información, colocando dicho esquema conceptual a disposición del público mediante el proyecto de Modelos de Ontología de Exploración de la NASA NEXIOM, siendo este una iniciativa del Programa Constellation en el Centro de Investigación AMES ARC (ibídem).

Así mismo, la ontología QUDT define las cantidades o magnitudes como instancias o individuos, no como clases o conceptos (Rijgersberg, 2013, p. 8); por lo que dicha cantidad se relaciona con la medida y no con la expresión numérica de un aspecto estandarizado, de un fenómeno o de una propiedad que permita representar un registro completo de los datos, quedando dichos objetos asociados a una clase de tipo de cantidad mediante una generalización (ib.).

A pesar de que las unidades son definidas como instancias de la clase unidad, QUDT no define axiomas o restricciones de disyunción entre los tipos de unidades, por lo que no es posible verificar los registros de observación mediante OWL-DL tal y como se hace en OM (ibíd.).

Es de resaltar que el OWL-DL es uno de los sub-lenguajes o especies del Lenguaje de Ontología Web OWL, que facilita realizar inferencias a partir de implicaciones y equivalencias computables en tiempo finito (W3C, 2004) mediante la lógica descriptiva (Description Logic DL), llamada también lógica de predicados de primer orden –entendiendo el predicado como la expresión que designa características o relaciones de un individuo o instancia (Deaño, 2009, p. 176)–, cuyo propósito consiste en predecir el

comportamiento de un sistema (o validez del razonamiento) de manera consistente y completa en su totalidad; y a su vez decidible de modo parcial a nivel de enunciados monádicos o de algunas premisas poliádicas (íd. p. 295–297) en un universo de discurso determinado.

Por último, OM cubre más áreas de aplicación de las magnitudes y unidades que QUDT, haciendo que todas las cantidades tanto fundamentales como derivadas se extiendan de los estándares internacionales (Rijgersberg, 2013, p. 8), lo cual y teniendo en cuenta lo descrito en los párrafos anteriores, conduce a refrendar la elección de la ontología OM en el presente trabajo investigativo y cuyo modelado en OWL 2 se encuentra disponible al público en formato RDF (Wageningen, 2017), haciendo uso de espacios de nombres de rdf-schema, dublin core, xml-schema, skos, bibo y foaf.

2.3.4 Ontología de Referencia de Dispositivos Inteligentes SAREF

La ontología de Referencia de Dispositivos Inteligentes –en inglés Smart Appliances REFerence ontology SAREF– versión 2.1.1, es un sistema conceptual construido a partir de los diversos estándares, protocolos y modelos de datos existentes, llamados también “activos semánticos” (TNO, 2014) (TNO, 2016) (Daniele, 2015, p. 7), en el dominio de los dispositivos inteligentes.

La idea de “dispositivo inteligente” hace mención a un artefacto o máquina que se usa en el hogar para realizar un trabajo doméstico (ETSI, 2017, p. 6), caracterizándose por tener la capacidad de comunicarse con otros artefactos o dispositivos y permitiendo su control a través de la Internet.

Dichas máquinas pueden ser por ejemplo (ibíd.) (TNO, 2014) (Daniele, 2015, p. 31):

- Sensores para vivienda o edificaciones (temperatura, humedad, medidores de energía, etc.).
- Accionadores (ventanas, puertas, persianas, etc.).
- Línea blanca (lavadora, horno, calentador, etc.).
- Línea marrón (televisor, cámara, proyector, etc.).
- Línea gris (teléfono móvil, tableta, reproductor, etc.).
- Calefacción, ventilación y aire acondicionado.
- Iluminación (lámparas, reflectores, fotómetros, etc.).
- Micro-renovables (paneles y calentadores solares, turbinas de viento, etc.).

El concepto de dispositivo inteligente aplica no solo al dominio del hogar o viviendas privadas (óp. cit.) (ob. cit.) ; también se halla dirigido al ámbito de las oficinas y los edificios particulares y/o públicos comunes y corrientes, excepto equipos especiales como ascensores, instrumental médico, dispensadores, etc. Así mismo, el reconocimiento de los artefactos mencionados se realiza sólo

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

con respecto a los requerimientos semánticos para las operaciones energéticas relevantes (ibídem), tales como el encendido, en espera, etc.; mas no para la gestión de contenido, por ejemplo, elección de un canal, de una emisora, etc.

En la ontología SAREF tales representaciones semánticas se estructuran en esencia como se aprecia en la figura 37 (elaborada por el autor a partir de (TNO, 2016), (Daniele, 2015, p. 135–142) y (ETSI, 2017, p. 10–19)) y siguiendo los lineamientos del Lenguaje de Ontología Web OWL-DL, en el cual los “conceptos y roles” de la lógica descriptiva se asimilan a “clases y propiedades” (sean estas últimas atributos o relaciones) respectivamente (W3C, 2004).

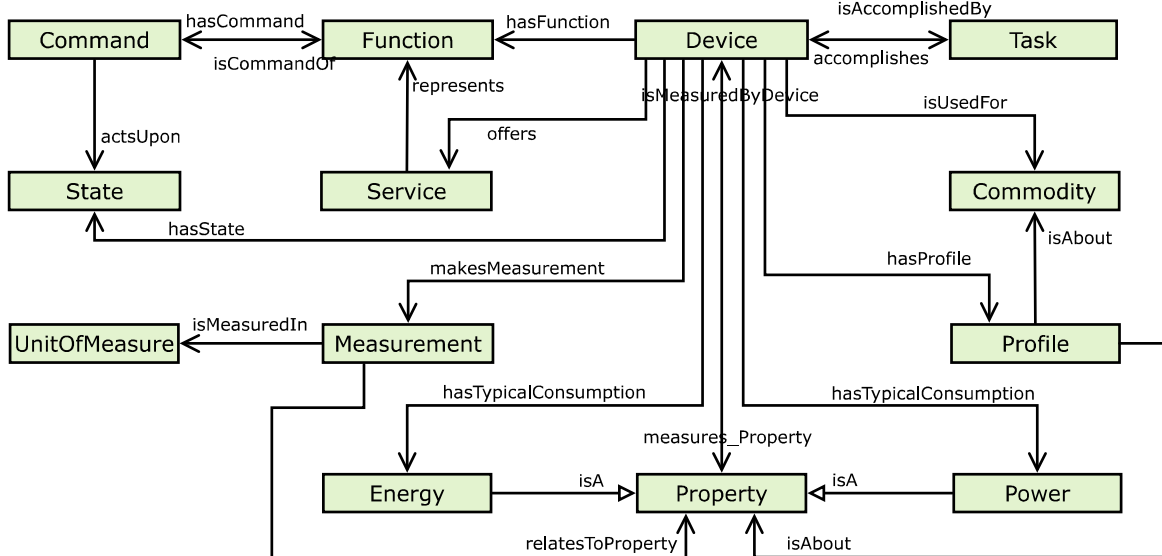


Figura 37. Estructura conceptual básica de la ontología SAREF (elaborada por el autor fundado en (TNO, 2016) (Daniele, 2015, p. 135–142) y (ETSI, 2017, p. 10–19)).

La notación de dichos conceptos o clases al interior de la ontología SAREF se establecen mediante el prefijo “saref”, el cual se separa del morfema que indica el concepto aludido utilizando el símbolo ortográfico “:”, por ejemplo “saref:Device”, “saref:Measurement”, “saref:Command”, etc. (ETSI, 2017, p. 10–19). Igualmente, los roles o propiedades cuentan con la escritura “prefijo:relación” (llamada también “ObjectProperties”) o “prefijo:atributo” (nombrada como “DataProperties”), ya sea una propiedad de categoría relación o de categoría atributo, de modo correspondiente (oneM2M, 2016, p. 8) (ETSI, 2016, p. 9). Se tiene por ejemplo: saref:offers”, “saref:actUpon”, saref:hasTypicalConsumption”, etc., para relaciones; “saref:hasManufacturer”, “saref:hasModel”, “saref:hasDescription”, etc., para atributos, siendo estos características de un tipo de dato predefinido (texto, numérico entero, etc.) que pueden variar en cuanto a su contenido (óp. cit.).

Para una mejor comprensión de la analogía existente entre la Lógica Descriptiva (Description Logic DL) o lógica de predicados de primer orden, el Lenguaje de Ontología Web para la Lógica Descriptiva OWL-DL y la ontología de Referencia para Dispositivos Inteligentes SAREF, se ha construido la tabla 8 (por parte del autor) de acuerdo a (W3C, 2004) (Deaño, 2009) (TNO, 2016) (Daniele, 2015) (ETSI, 2017) (W3C, 2004a) (oneM2M, 2016) y (ETSI, 2016):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Lógica Descriptiva	OWL-DL		SAREF
Concepto	Clase	Class	saref:Device
			saref:Measurement
			saref:Command
Rol	Propiedad (Relación)	ObjectProperty	saref:offers
			saref:actUpon
			saref:hasTypicalConsumption
	Propiedad (Atributo)	DataProperty	saref:hasManufacturer
			saref:hasModel
			saref:hasDescription
Individuo	Instancia	Instance	scheme://authority/path/query/fragment
			http://ontology.tno.nl/saref/washingmachine#45609WQ
			http://ontology.tno.nl/saref/lightswitch#890-09w
			http://ontology.tno.nl/saref/energymeter#EM5004-stv-16

Tabla 8. Analogía entre la Lógica Descriptiva, OWL-DL y SAREF (elaborada por el autor fundado en (W3C, 2004) (Deaño, 2009) (TNO, 2016) (Daniele, 2015) (ETSI, 2017) (W3C, 2004a) (oneM2M, 2016) y (ETSI, 2016))

El último aspecto a considerar en los diversos dispositivos o artefactos existentes es su ejemplarización como “individuos” desde la lógica descriptiva, siendo su símil en el lenguaje de ontología web OWL-DL una “instancia” (instance en inglés) e igual que en la ontología SAREF, su notación se realiza de acuerdo a un Identificador Uniforme de Recursos URI “esquema://autoridad/ruta/consulta/fragmento” (“scheme://authority/path/query/fragment” en inglés) (IETF, 2005, p. 16), siendo:

- **scheme** (esquema) = especificación que asigna identificadores como un sistema de nombres (ib., p. 17), incluyendo generalmente el protocolo de comunicación para acceder a un recurso como “http:”, “ftp:”, “mailto:”, etc.
- **authority** (autoridad) = elemento jerárquico que identifica un nombre de dominio, mediante una nominación o título registrado (ídem), como también a través de la dirección de un servidor, verbigracia, “//www.etsi.org”, “//ontology.tn.nl”, “//www.onem2m.org”, etc.
- **path** (ruta) = datos subordinados que identifican un recurso dentro del alcance del esquema y nombre de dominio a manera de un sistema de archivos (ibíd., p. 22), como en, “/saref/WashingMachine”, “/deliver/etsi_ts”, “/images/files/deliverables/Release2”, etc.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- query (consulta) = componente cuyos datos no están ordenados por niveles dependientes, sino en forma de pares “clave = valor” (ib., p. 23) al interior de un esquema, dominio y ruta determinado a partir del carácter de interrogación “?”; por ejemplo, “/c=GB?objectClass?one” (ibíd., p. 7).
- fragment (fragmento) = elemento que permite la identificación indirecta de un recurso secundario en referencia a un recurso principal (esquema y autoridad), a partir de una representación del mismo e indicándose mediante el carácter de signo numérico “#” (ib., p. 24). Dicha representación atañe a una parte, una vista, una definición o una descripción del recurso principal al que pertenece (i. e. el URI), siendo su semántica dependiente de la acción de recuperación de tal recurso (ídem).

Es de tener en cuenta que el formato y la resolución de un fragmento es dependiente del tipo de medio –“media type” en inglés– de la Extensiones Multipropósito de Correo en Internet MIME (Multipurpose Internet Mail Extensions en inglés) (ibídem), el cual corresponde a la categoría “application” e implementado para procesar datos discretos por parte de algún programa o aplicación antes de estar disponible para un agente, siendo de práctico empleo en la transferencia de archivos (IETF, 1996, p. 12).

El tipo de medio MIME utilizado para la instancia de un dispositivo atañe a la categoría “application/rdf+xml”, en razón a que la información a nivel sintáctico y semántico de dicha instancia se encuentra definida en un recurso “rdf/xml” (IETF, 2004, p. 1), i. e., estructurando los datos en el Marco de Descripción de Recursos RDF (Resource Description Framework) bajo la sintaxis del Lenguaje de Marcado Extensible XML (eXtensible Markup Language), lo cual facilita la codificación de tales datos en un archivo y transferirlos en la red, –procedimiento conocido igualmente como “serialización”– (ídem), al igual la descripción de dichos dispositivos como recursos y su respectiva interoperabilidad sobre la red.

El Marco de Descripción de Recursos RDF se utiliza para la instanciación de dispositivos como recursos con descripción semántica que permite la comunicación sin ambigüedades, dado que es un lenguaje para representar metadatos sobre la Internet acerca de recursos (W3C, 2004b) como un documento, una imagen, un video, un audio, una página web, etc., como también respecto de cosas tangibles como un artículo de consumo, un electrodoméstico, etc. Dicha representación RDF se visualiza como un grafo conformado por un par de nodos con una arista, además de establecerse a partir de una tripleta cuyos componentes son (W3C, 2004c):

- Un sujeto (concepto, clase, nodo), denotado por una referencia URI con nombre o sin él (nodo vacío).
- Un predicado (rol, relación, propiedad, arista), señalado por una referencia URI.
- Un objeto (dato, atributo o clase, nodo), indicado por un literal (tipificado) o una referencia URI, ya sea con nombre o sin él (nodo vacío).

La codificación de la instancia de un dispositivo como recurso descriptor semántico a partir de un archivo RDF/XML con sintaxis correcta y validez gramatical, permite el intercambio de información entre aplicaciones dispares con una semántica común (IETF, 2004, p. 3), convirtiendo este tipo de medio (application/rdf+xml) en neutral respecto a dispositivos, plataformas y proveedores, respaldado tanto por agentes (de usuario o de máquina) como por herramientas de desarrollo integrado (ídem).

La sintaxis normativa para codificar en RDF cualquier instancia de un dispositivo es RDF/XML (W3C, 2004b, inciso 3), lo cual lo convierte en el más oportuno tipo de medio para la comunicación

entre artefactos, ya que el Marco de Descripción de Recursos RDF tiene como objetivo hacer declaraciones procesables por máquina. Para efectos de lo descrito se requiere tener en cuenta un par de aspectos (ibídem, inciso 2.1):

- Un sistema de identificadores procesables por máquina para identificar en una declaración a un sujeto, un predicado y un objeto (i. e. una tripleta RDF), de tal manera que no haya lugar a ambigüedades al momento de ser utilizado dicho identificador por otro agente en el ciberespacio (idem).
- Un lenguaje informático que sea procesable por una máquina y así, poder representar cualquier declaración (tripleta RDF) e intercambiarla entre diversos dispositivos (ibídem).

El primer aspecto mencionado atañe a un Identificador Uniforme de Recurso URI (Uniform Resource Identifier en inglés), el cual permite identificar cualquier cosa en la red ya sea una imagen, un servicio, una persona, un concepto abstracto, un artefacto, etc. (ibíd.). El segundo aspecto se refiere al Lenguaje de Marcado eXtensible XML para RDF –el XML/RDF–, el cual facilita elaborar la estructura descriptora semántica de una entidad cualesquiera en la red (v.g. de la instancia de un dispositivo) en dicho lenguaje, para representar la información RDF de dicha entidad y transferirla entre máquinas (ib.).

No obstante, XML/RDF no es el único formato de serialización para escribir tripletas RDF, aunque las diversas formas existentes conducen exactamente a los mismos grafos por lo que son equivalentes a nivel lógico. Dichos formatos de codificación son N-Triples, N-Quads, Turtle, TriG, JSON-LD y RDFa (W3C, 2014, inciso 5), los cuales son susceptibles de convertir a XML/RDF mediante aplicaciones utilitarias, debiéndose su desarrollo a la facilidad de expresar los grafos de manera similar a la sintaxis de SPARQL (ib., inciso 7) –que es el lenguaje de consulta para RDF–, permitiendo elaborar criterios de búsqueda y por ende, reglas de inferencia.

Así mismo, el expresar los grafos RDF correspondientes a la instancia de un dispositivo con sus diversas clases, relaciones y atributos, conduce a la construcción de una ontología cuyo esquema es factible de serializar mediante el Lenguaje de Ontología Web OWL bajo la sintaxis de XML, lo cual redundaría en el tipo de medio “application/owl+xml” (W3C, 2004a) (W3C, 2014a). Dicha categoría se encuentra –por parte de la W3C– en proceso de registro y estandarización en la Autoridad de Asignación de Números de Internet IANA, por lo cual aun no ha adquirido el estatus de especificación técnica para ser implementada masivamente por aplicaciones herramientas de desarrollo (W3C, 2016). Sin embargo, actualmente se está haciendo uso de la serialización OWL/XML ya que OWL permite la optimización de RDF en cuanto a asociaciones de herencia simple y múltiple (agregación y composición) entre clases, dando relevancia a la interoperabilidad semántica de las entidades conectadas a la red (W3C, 2004) (W3C, 2012a).

Por último, dado que la instancia de un dispositivo se denota mediante un URI, siendo este una secuencia compacta de caracteres que identifica un recurso abstracto o concreto, cuya sintaxis es definida a partir de un conjunto de símbolos US-ASCII o UTF-8 (IETF, 2005, p. 8), se ha hecho necesario ampliar el repertorio de caracteres para manejar con éxito todas las posibles escrituras de URI sin ambigüedades, para lo cual se ha definido un elemento de protocolo llamado Identificador de Recurso Internacionalizado IRI (Internationalized Resource Identifier en inglés) (IETF, 2005a, p. 3), siendo el UCS (Universal Character Set en inglés) el conjunto de caracteres codificado para representar recursos de manera genérica, facilitando el mapeo entre URI e IRI (ibídem, p. 10).

Componentes de la ontología SAREF

Las descripciones de los principales conceptos de SAREF se despliegan como sigue:

- Dispositivo (Device):

Un dispositivo o artefacto es "un objeto tangible diseñado para llevar a cabo una tarea particular en hogares, oficinas o edificios públicos comunes y corrientes. Para lograr esta tarea, el dispositivo realiza uno o más funciones" (TNO, 2016) (Daniele, 2015, p. 135) (ETSI, 2017, p. 12), además de presentar características exclusivas como su modelo, fabricante y descripción del dispositivo, las cuales se establecen mediante las propiedades de objeto "saref:hasModel", "saref:hasManufacturer" y "saref:hasDescription" respectivamente (idem), siendo sus valores cadenas de caracteres. En la figura 38 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Dispositivo (saref:Device):

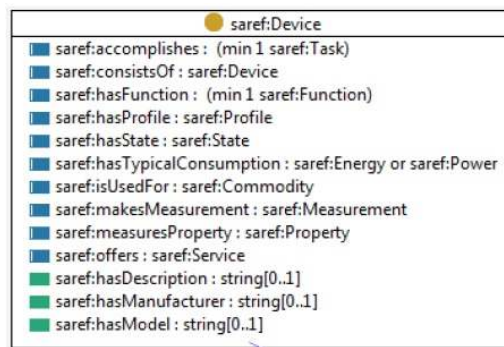


Figura 38. Clase Dispositivo (Device): propiedades de objeto y de datos (TNO, 2016) (Daniele, 2015, p. 135) (ETSI, 2017, p. 12).

Todo artefacto a lo menos, ha de estar soportado en su desempeño por mínimo una función o "saref:Function" mediante la propiedad de objeto "saref:hasFunction", con el propósito de que aquel se pueda utilizar en un recurso o "saref:Commodity" mediante la relación "saref:isUsedFor", para medir una propiedad física o magnitud "saref:Property" a través de la propiedad de objeto "saref:measuresProperty" (ibídem).

Un dispositivo es susceptible de clasificarse en categorías de acuerdo a su función de detección o accionamiento, a su eficiencia energética o a su espacio físico de ubicación. Dichas categorías corresponden en su orden "saref:FunctionRelated", "saref:EnergyRelated" y "saref:BuildingRelated", comportándose éstas como subclases de "saref:Device" (TNO, 2016) (Daniele, 2015, p. 136) (ETSI, 2017, p. 13). En la gráfica 39 se observan dichas categorías, las cuales organizan a los diversos dispositivos como sigue:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

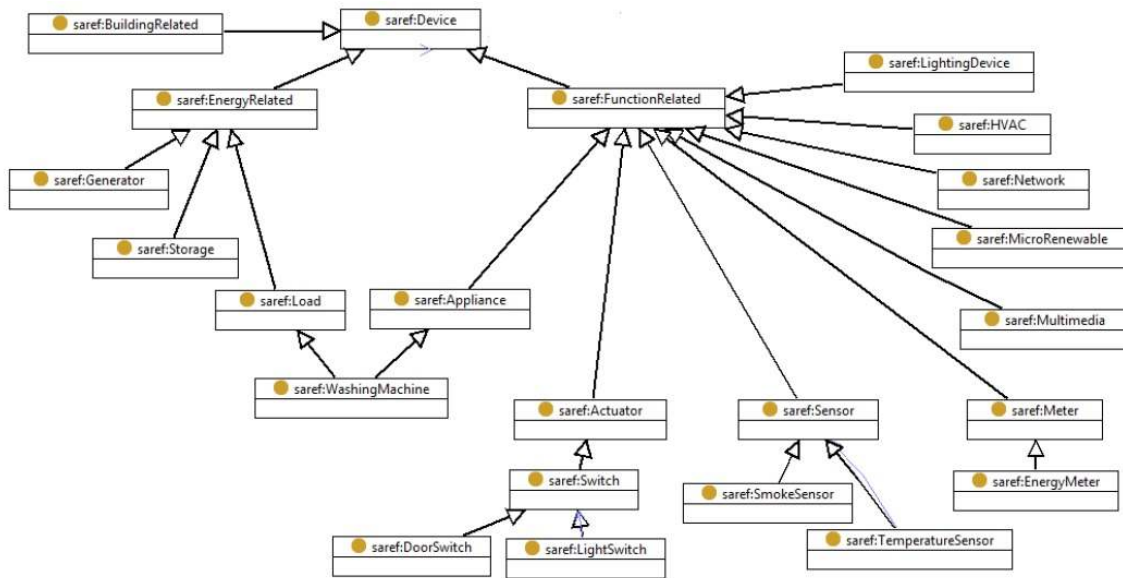


Figura 39. Categorías de un dispositivo SAREF (TNO, 2016) (Daniele, 2015, p. 136) (ETSI, 2017, pág. 13).

- Tarea (Task):



Figura 40. Clase Tarea (Task) (TNO, 2016) (Daniele, 2015, p. 144) (ETSI, 2017, p. 25)

Una tarea o “Task” consiste en el objetivo para el cual se diseña un dispositivo (desde la perspectiva del usuario). Por ejemplo, una lavadora está diseñada para la tarea de lavar (TNO, 2016) (Daniele, 2015, p. 144) (ETSI, 2017, p. 11). Dentro de las posibles tareas que son relevantes para SAREF están: iluminación (Lighting), eficiencia energética (EnergyEfficiency), lectura de medidor (MeterReading), etc. Dichas tareas actualmente funcionan como individuos o instancias de la clase “saref:Task” a partir de la versión 2.1.1 de la ontología, ya que en la versión anterior se manejaban como subclases de la misma (TNO, 2016) (ETSI, 2017, p. 25).

- Función (Function):

Una función es la capacidad de actuar propia por parte de un dispositivo, la cual es necesaria para llevar a cabo una tarea asignada y en cuyo diseño, se puede ejecutar un conjunto de dichas funcionalidades, siendo éstas organizadas en categorías de accionamiento “saref:ActuatingFunction”, de detección “saref:SensingFunction”, de medición “saref:MeteringFunction” y de suceso “saref:EventFunction” (TNO, 2016) (Daniele, 2015, p. 138) (ETSI, 2017, p. 14), tal y como se observa en la figura 41:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

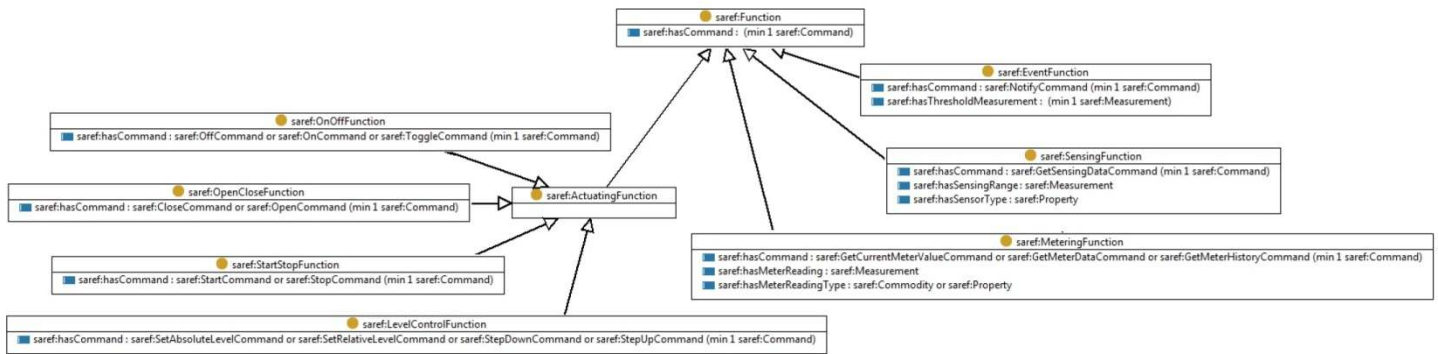


Figura 41. Categorías de la clase Función SAREF (TNO, 2016) (Daniele, 2015, p. 138) (ETSI, 2017, p. 14).

Dichas clasificaciones funcionan como subclases de la clase “Función”, las cuales se detallan a continuación (óp. cit.):

- La categoría de accionamiento permite transmitir a dispositivos “actuadores”, datos como nivel de configuración o ajuste de rango (v. g. temperatura), al igual que datos de conmutación binaria (p. ej. encendido/apagado), con el fin de que el dispositivo ejecute una acción o aplique una proporción de valores (ídem).
- La percepción es una categoría que facilita la transferencia de datos desde artefactos “sensores” para la medida de magnitudes (p. ej. temperatura) o detección de fenómenos (v. g. ocupación), con el propósito de que el dispositivo realice una lectura de cualesquiera de aquéllos en un momento determinado (íd.).
- La categoría de medición permite obtener datos desde un dispositivo medidor en cuanto al valor actual de una magnitud o recurso (v.g. energía o electricidad) durante una ventana de tiempo, i. e. la demanda instantánea de aquéllos (ídem).
- El suceso o evento es una categoría de función que facilita a un dispositivo notificar a otros, que un determinado valor de umbral ha sido alcanzado o excedido (íd.).

Para que una función pueda llevar a cabo una acción debe tener asociada a la misma un comando u orden, la cual se define mediante la propiedad de objeto “saref:hasCommand”, pudiendo vincular diversos comandos (ibídem), por lo que la cardinalidad de la relación entre las clases “Función” y “Comando” es del tipo uno a varios.

- Orden (Command):

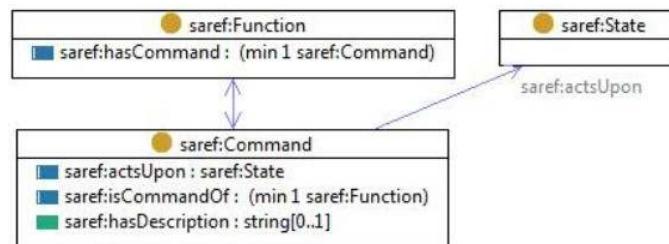


Figura 42. Clase Orden (Command) (TNO, 2016) (ETSI, 2017, p. 15) (Daniele, 2015, p. 139).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Un comando u orden es una directiva que un dispositivo deberá soportar para ejecutar una cierta función (TNO, 2016) (ETSI, 2017, p. 11). Un comando puede tener la capacidad de leer o recuperar el estado en que se encuentra un dispositivo, como también estar facultado para alterar o modificar dicho estado en un momento determinado (idem).

A continuación unos ejemplos (ETSI, 2017, p. 15) (Daniele, 2015, p. 139):

- Los comandos `saref:OnCommand`, `saref:OffCommand` y `saref:ToggleCommand` (que son subclases de `saref:Command`) están asociados a la función `saref:OnOffFunction` (la cual es una subclase de `saref:ActuatingFunction` y a la vez de `saref:Function`, teniendo en simultánea una propiedad de objeto `saref:hasCommand` que permite relacionarse con las subclases de tipo comando mencionadas al inicio).
-
- Los comandos `saref:SetLevelCommand` (que puede ser del tipo `saref:SetAbsoluteLevel` o `saref:SetRelativeLevel`, i.e. son subclases de la clase de comando asignar nivel), `saref:StepUpCommand` y `saref:StepDownCommand` (que son subclases de `saref:Command`), están asociados a la función `saref:LevelControlFunction` (la cual es una subclases de `saref:ActuatingFunction` y a la vez de `saref:Function`, teniendo en simultánea una propiedad de objeto `saref:hasCommand` que permite relacionarse con las subclases de tipo comando mencionadas al inicio).
-
- El comando `saref:GetCommand` (el cual es subclase de `saref:Command`) está asociado a la función `saref:SensingFunction` (la cual es subclase de `saref:Function`, teniendo en simultánea una propiedad de objeto `saref:hasCommand` que le permite relacionarse con la subclase de tipo comando mencionada al principio del párrafo).
-
- Los comandos `saref:GetCurrentMeterValueCommand`, `saref:GetMeterDataCommand`, `saref:GetMeterHistoryCommand` (que son subclases de `saref:Command`), están asociados a la función `saref:MeteringFunction` (la cual es una subclase de `saref:ActuatingFunction` y a la vez de `saref:Function`, teniendo en simultánea una propiedad de objeto `saref:hasCommand` que permite relacionarse con las subclases de tipo comando mencionadas al inicio del párrafo).
-
- El comando `saref:NotifyCommand` (el cual es subclase de `saref:Command`), está asociado a la función `saref:EventFunction` (la cual es subclase de `saref:Function`, teniendo en simultánea una propiedad de objeto `saref:hasCommand` que le permite relacionarse con la subclase de tipo comando mencionada al principio del párrafo).

La figura 42 muestra además que un comando (clase `saref:Command`) puede actuar sobre un estado o clase `saref:State` (mediante la relación o propiedad de objeto `saref:actsUpon`), para representar que la consecuencia o resultado de un comando puede ser un cambio de estado del dispositivo. Tener en cuenta que un comando puede actuar sobre un estado, pero no necesariamente incidir en el mismo. Por ejemplo, el comando `saref:OnCommand` actúa sobre el estado `saref:OnOffState`, pero el comando `saref:GetCommand` no actúa sobre ningún estado,

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

ya que solo da una directiva o instrucción para recuperar un cierto valor de estado o condición en la que se encuentra el dispositivo en el momento de ejecutar la orden.

- Estado (State):

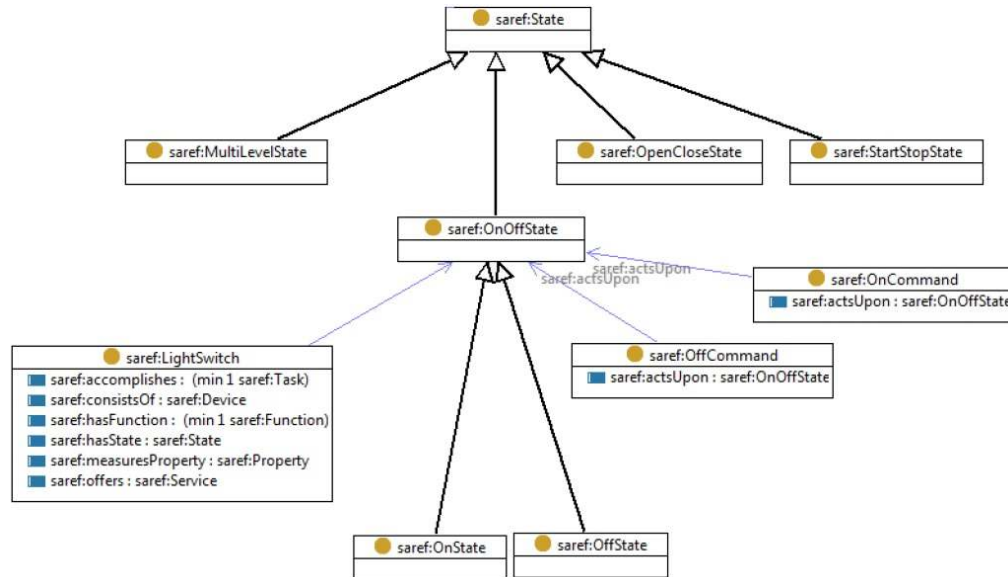


Figura 43. Clase Estado (State) (TNO, 2016) (ETSI, 2017, p. 16) (Daniele, 2015, p. 140)

Dependiendo de la(s) función(es) que realiza, un dispositivo se puede encontrar en un estado o condición correspondiente (clase saref:State) como se muestra en la Figura 7. Por ejemplo, un interruptor (clase saref:Switch) se puede encontrar en el estado saref:OnOffState, el cual se especializa más en las subclases "saref:OnState" y "saref:OffState" (ETSI, 2017, p. 16) (Daniele, 2015, p. 140). Es de observar que SAREF no está restringido a estados binarios (i. e. uno de dos valores posibles) como el caso de la clase "saref:OnOffState"; igualmente permite definir estados n-arios (esto es, uno de múltiples valores posibles, como por ejemplo, la clase saref:MultiLevelState). La figura 43 muestra además el ejemplo de un interruptor de luz que se puede encontrar en el estado encendido o apagado (clase saref:OnOffState) sobre el cual actuarán los comandos encender "saref:OnCommand" y apagar "saref:OffCommand" (idem).

- Servicio (Service):



Figura 44. Clase Servicio (Service) (ibídem)

Un servicio “saref:Service” atañe a una representación de una función o varias de ellas en la red, el cual permite que la función sea reconocible, registrable y controlable de manera remota por otros dispositivos en dicha red (TNO, 2016) (ETSI, 2017, p. 16) (Daniele, 2015, p. 140). Todo servicio debe especificar el dispositivo que está ofreciendo al mismo mediante la propiedad de objeto “saref:isOfferedBy”, como también la(s) función(es) a representar a través de la relación “saref:represents” (ídem).

Es de observar que el concepto de servicio se halla elaborado con mayor detalle en la ontología oneM2M (inciso 2.3.6 de la presente investigación), a la que se remite al lector para modelar los detalles de un servicio que están fuera del alcance de SAREF.

- Perfil (Profile):

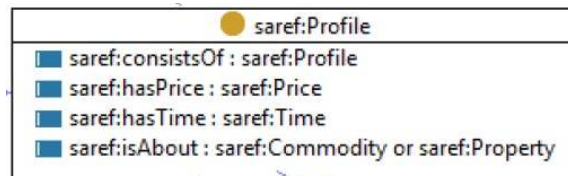


Figura 45. Clase Perfil (Profile) (TNO, 2016) (ETSI, 2017, p. 17)

Un perfil o configuración es una especificación asociada a un dispositivo para recopilar información sobre una determinada magnitud (p. ej. energía) o un recurso (v.g. agua), de tal manera que se optimice el uso de dicho artefacto según el sitio en que esté ubicado (vivienda, oficina o edificio) (TNO, 2016) (ETSI, 2017, p. 17). Dicho perfil se vincula a alguna magnitud o recurso mediante la relación “saref:isAbout”, pudiéndose establecer durante un lapso de tiempo a través de la propiedad de objeto “saref:hasTime” y en simultánea, asociarse a ciertos costos por medio de la “saref:hasPrice”. Un ejemplo de perfil es la configuración de potencia definida en la ontología de Referencia para Dispositivos Inteligentes para el dominio de Energía SAREF4ENER, la cual es una extensión de la ontología SAREF, siendo dicho perfil asociado a un dispositivo para optimizar su eficiencia energética (ibídem) y a la que se remite al lector (inciso 2.3.5 de la presente investigación).

- Recurso (Commodity):



Figura 46. Clase Recurso (Commodity) (TNO, 2016) (ETSI, 2017, p. 11) (Daniele, 2015, p. 143)

Consiste en un bien o medio susceptible de comercializar para el que existe demanda en un mercado, siendo ofrecido sin diferenciación cualitativa (TNO, 2016) (ETSI, 2017, p. 11) (Daniele, 2015, p. 143). En el universo de discurso energético, un recurso atañe a la electricidad, el gas natural, el carbón o el petróleo, las cuales son susceptibles de medir mediante la relación “isMeasuredIn” con la clase “saref:UnitOfMeasure” (óp. cit.). Además, un dispositivo “saref:Device” (y su propiedad de objeto “saref:isUsedFor”) se puede usar

con el propósito de exponer un recurso básico, por ejemplo “saref:Water” o “saref:Gas”, etc. (idem, p. 12).

- Medición (Measurement):

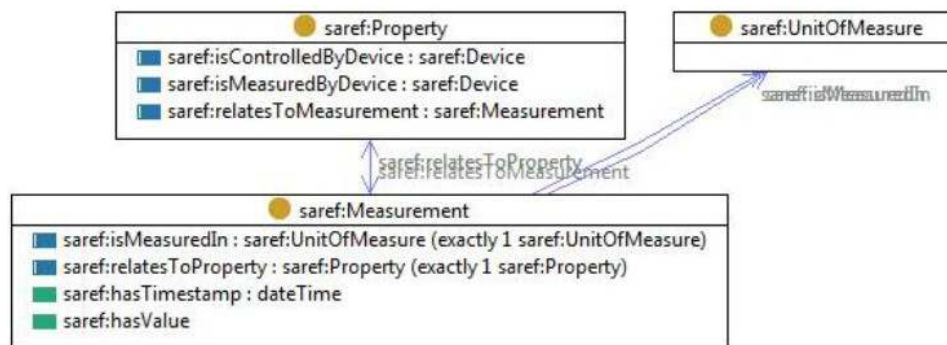


Figura 47. Clase Medición (Measurement) de SAREF (TNO, 2016) (ETSI, 2017, p. 18)

Corresponde al valor medido acerca de una propiedad o magnitud, la cual se expresa en una unidad de medida y una marca de tiempo o “timestamp” (TNO, 2016) (ETSI, 2017, p. 18). Es de tener en cuenta que una medida se refiere a una cuantificación; i. e., a la comparación de una cantidad con su respectiva unidad, con el fin de averiguar cuántas veces la segunda está contenida en la primera (RAE, 2014) (OUP, 2017).

Una Medición “saref:Measurement” se relaciona con una Unidad de Medida “saref:UnitOfMeasure” mediante la propiedad de objeto (ObjectProperty) de nombre “saref:isMeasuredIn”, con cardinalidad uno a uno (óp. cit.). Igualmente, se vincula a una Propiedad o magnitud “saref:Property” a través de la relación “saref:relatesToProperty” con idéntica cardinalidad de la anterior (1 a 1). También presenta un atributo “saref:hasValue” cuyo valor es un número de coma flotante “xsd:float” (TNO, 2016) (W3C, 2012) (IEEE, 2008, p. 4) (como se cita en el inciso 3.3.3, unidades de medida OM).

Por último, la característica “saref:hasTimeStamp” indica la marca de tiempo de la medición tomada en un momento específico en formato “xsd:dateTime” (TNO, 2016). Una marca de tiempo o “timestamp” atañe a una representación exacta y precisa de un instante de tiempo (OUP, 2017) (W3C, 2012, inciso 3.3.7) (IETF, 2002, p. 3), cuyo estructura consiste en la combinación de un par de datos: la fecha y la hora en que sucede un evento particular (i. e. una medición). Dicha marca de tiempo se registra en concordancia con el Tiempo Universal Coordinado UTC (óp. cit.) (ob. cit., p. 2) (ISO, 2004, p. 18), siendo su expresión una secuencia de caracteres al tenor del estándar 8601 de la ISO, tal y como se describe a continuación (idem, p. 19):

- Formato básico: “YYYYMMDDThhmmss±hhmm”, v.g. “19850412T101530+0400”.
- Formato extendido: “YYYY-MM-DDThh:mm:ss±hh:mm”, v. g. “1985-04-12T10:15:30+04:00”.

- Propiedad (Property):

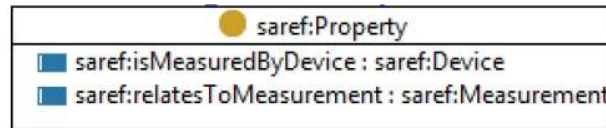


Figura 48. Clase Propiedad (Property) de SAREF (TNO, 2016) (ETSI, 2017, p. 18)

Una propiedad o magnitud “saref:Property” es cualquier cualidad física que pueda ser detectada, medida o controlada por un dispositivo ubicado en algún sitio (vivienda, oficina o edificio) (TNO, 2016), siendo la relación “saref:isMeasuredByDevice” la que permite vincular la magnitud con el dispositivo de clase “saref:Device”, como también la relación “saref:relatesToMeasurement” la que facilita el nexo de la propiedad con la medición de clase “saref:Measurement” (ETSI, 2017, p. 18).

Es de tener en cuenta que las clases de Medición “saref:Measurement”, de Propiedad “saref:Property” y de Unidad de Medida “saref:UnitOfMeasure”, permiten relacionar diferentes medidas de un dispositivo dado para diversas propiedades valoradas en distintas unidades (ibíd.); es decir, la clase de medición (saref:Measurement) describe una valoración numérica de una cantidad física o magnitud (utilizando la propiedad de tipo de dato o "DataTypeProperty" saref:hasValue) para una propiedad dada (saref:Property) y de acuerdo con una unidad de medida dada (saref:UnitOfMeasure). De esta manera, es posible diferenciar entre las propiedades y las mediciones hechas para tales propiedades, además de almacenar las mediciones para una propiedad concreta en diferentes unidades de medida (ibídem).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

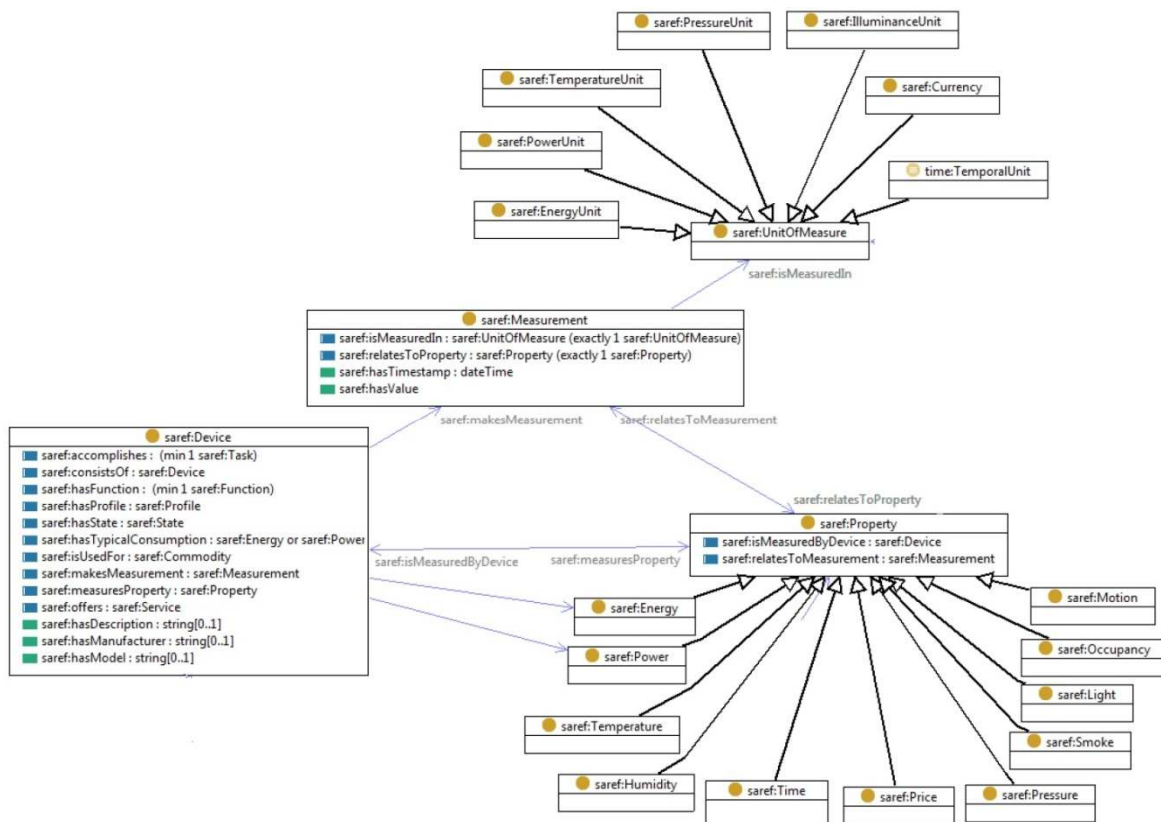


Figura 49. Nexos entre las clases Dispositivo, Propiedad, Medición y Unidad de Medida (id., p. 19)

Por otra parte, se puede agregar una marca de tiempo (utilizando la propiedad de tipo de dato "DataTypeProperty" `saref:hasTimeStamp`) para identificar cuándo se aplica la medición a la propiedad, la cual se puede usar ya sea para mediciones individuales o para series de mediciones (por ejemplo, flujos de medición). La Figura 49 muestra que un dispositivo "saref:Device" puede medir una propiedad "saref:Property", que a su vez se relaciona con una medición "saref:Measurement" y en simultánea, mide en una determinada unidad de medida "saref:UnitOfMeasure" (óp. cit.). Se observa que también es posible seguir la dirección inversa: en la que un dispositivo "saref:Device" realiza una medición "saref:Measurement" en una cierta unidad de medida "saref:UnitOfMeasure" (utilizando la propiedad de objeto "ObjectProperty" `saref:makesMeasurement`), y esta medida o valoración numérica (ibíd.) puede relacionarse con una propiedad "saref:Property" (mediante la propiedad de objeto "ObjectProperty" `saref:relatesToProperty`).

Un ejemplo de lo anterior se detalla con las clases de Potencia y Energía "saref:Power" y "saref:Energy" respectivamente (que son subclases de "saref:Property") en la figura 49 (ibidem) pueden relacionarse con un cierto valor de medición de la clase "saref:Measurement" (utilizando la propiedad de tipo de dato "DataTypeProperty" `saref:hasValue`), el cual se mide en una cierta unidad de medida (saref:UnitOfMeasure) tal como kilovatios de potencia (clase `saref:PowerUnit`) y kilovatios-hora de energía (clase `saref:EnergyUnit`), siendo estas dos últimas, subclases de "saref:UnitOfMeasure" (ib.).

- Unidad de Medida (Unit of Measure):

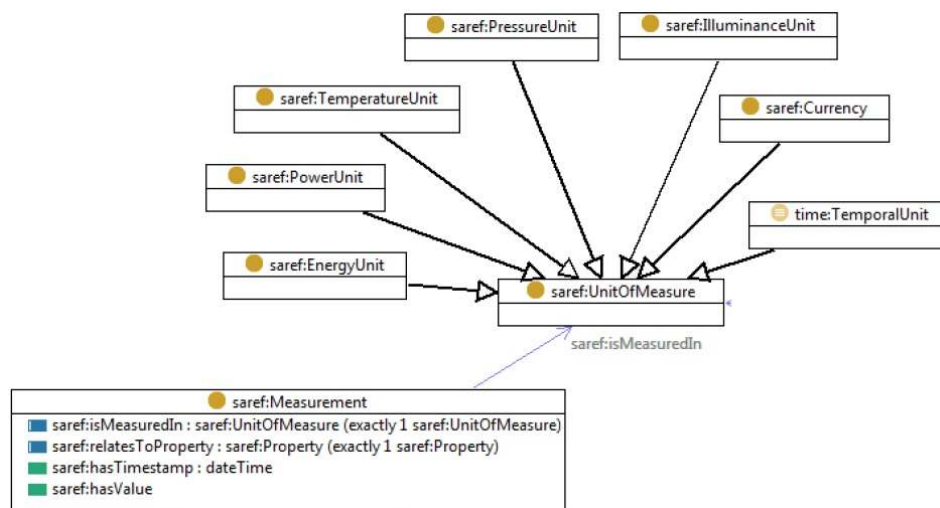


Figura 50. Clase Unidad de Medida (UnitOfMeasure) de SAREF (óp. cit.)

La unidad de medida es un estándar para medir una cantidad en referencia a una propiedad o magnitud “saref:Property”. Por ejemplo, la potencia (Power) es una propiedad y el vatio es una unidad de potencia que representa un valor determinado de dicha magnitud: cuando se dice 10 vatios, en realidad significa 10 veces la unidad de potencia llamada vatio o “watt” (TNO, 2016) (ETSI, 2017, p. 11). La definición de la unidad de medida en SAREF se refiere a la definición de la misma en la Ontología de unidades de medida (inciso 2.3.3 de la presente investigación), a la que se remite al lector para modelar los detalles de la medición en cuanto a la unidad apropiada a partir de la clase “saref:Measurement” (ibídem).

En la sección anterior (inciso Propiedad o “Property”) se detalla el vínculo existente entre la clase Unidad de Medida (UnitOfMeasure) con la clase Medición (Measurement), al igual que de esta con las clases Propiedad (Property) y Dispositivo(Device). Análogamente, la clase de precio “saref:Price” puede relacionarse con un determinado valor de medición que se mide utilizando un cierto valor de moneda “saref:Currency”, la cual es una subclase de la clase “saref:UnitOfMeasure” (ibídem, p. 18). Por último: la clase de tiempo “saref:Time” permite especificar el concepto de “tiempo” en términos de entidades temporales (es decir, instantes o intervalos) de acuerdo con la ontología de tiempo W3C® existente que se importó en SAREF para evitar definir este concepto desde cero. En la Figura 49, se hace referencia a la ontología W3C® Time con el prefijo “time:” para distinguir las clases y propiedades de SAREF, de las que se hace referencia usando el prefijo “saref:” (idem).

Como complemento de todo lo expuesto anteriormente, se incluye la definición de la ontología SAREF tomada textualmente de su sitio web correspondiente (TNO, 2014) (TNO, 2016), dado que dicha descripción es lo suficientemente sucinta y precisa.

“La ontología de referencia de dispositivos inteligentes (Smart Appliances REFERENCE SAREF) es un modelo compartido de consenso que facilita la comparación de los activos existentes (estándares/protocolos/modelos de datos/etc.) en el dominio de los dispositivos inteligentes. La ontología SAREF proporciona bloques básicos que permiten la separación y la recombinación de diferentes partes de la ontología dependiendo de las necesidades específicas. El punto de partida de

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

SAREF es el concepto de dispositivo (por ejemplo, un interruptor). Los dispositivos son objetos tangibles diseñados para llevar a cabo una tarea particular en hogares, oficinas o edificios públicos comunes. A fin de llevar a cabo dicha tarea, el dispositivo realiza (ejecuta) una o más funciones. Por ejemplo, una lavadora está diseñada para lavar (tarea) y para realizar esta tarea, realiza o ejecuta la función de inicio y parada. La ontología SAREF ofrece una lista de funciones básicas que pueden combinarse eventualmente para tener funciones más complejas en un solo dispositivo. Por ejemplo, un interruptor ofrece una función de activación de tipo 'encendido/apagado'. Cada función tiene algunos comandos asociados, que también pueden recogerse como bloques de construcción de una lista. Por ejemplo, el 'encendido/apagado' está asociado con los comandos 'encender', 'apagar' y 'alternar'. Dependiendo de la(s) función(es) que realice, se puede encontrar un dispositivo en algunos estados correspondientes que también están listados como bloques de construcción. Cuando se conecta a una red, un dispositivo ofrece un servicio, que es una representación de una función en una red que hace que la función sea detectable, registrable y remotamente controlable por otros dispositivos en la red. Un servicio puede representar una o más funciones. En concreto, un dispositivo ofrece un servicio cuando requiere que sus funciones —o un subconjunto de ellas— sean detectables, registrables y controlables remotamente por otros dispositivos en la red. Un servicio debe especificar el dispositivo que está ofreciendo el servicio y la(s) función(es) a representar. Un dispositivo en la ontología SAREF también se caracteriza por un perfil que se puede utilizar para optimizar algunas propiedades, como la energía, en un hogar u oficina que son parte de un edificación" (óp. cit.)(ob. cit.).

Motivación para la escogencia de la ontología SAREF

Dentro de todas las ontologías existentes sobre dispositivos inteligentes, se ha escogido la ontología SAREF por las siguientes razones:

- Según la oficina europea de estadística Eurostat en su informe del año 2012 sobre eficiencia energética, el sector residencial y terciario cuya mayor parte son edificaciones, representó más del 40% del consumo final de energía en la Unión Europea en el año 2010 (Eurostat, 2013, p. 11) (Eurostat, 2018), al igual que en Colombia dicha demanda para el mismo año alcanzó el 60% (UPME, 2018); una tendencia que seguramente aumentará el consumo de energía eléctrica en dicha comunidad y por lo tanto, sus emisiones de dióxido de carbono (Daniele, 2015, p. 25). No son tanto los edificios como tales los que consumen energía y producen gases de efecto invernadero GEI, sino los denominados Productos que utilizan y producen Energía (Energy using and producing Products EupP en inglés), también llamados "electrodomésticos", inherentemente presentes en los ecosistemas de los edificios y las personas que los utilizan (ídem).
- Un electrodoméstico es un instrumento o dispositivo diseñado para un uso o función particular, por lo general doméstica, siendo clasificados como línea blanca, marrón y electrónica de consumo, facultados para conectarse en red y trabajar de manera interactiva (ibídem), conformando sistemas de demanda, gestión y generación de energía eléctrica, lo que conduce a reducir el uso dicha potencia y la producción de GEI no solo aumentando la eficiencia de los dispositivos individuales, sino regulando y optimizando la utilización de energía a nivel de sistema (ibíd.).
- El llevar a cabo dicha labor requiere de interfaces abiertas estandarizadas de comunicación, para que los fabricantes de los diversos dispositivos y aplicaciones puedan conectarlos con las plataformas de servicio existentes o por desarrollarse (ib.).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Con ello se garantiza la interoperabilidad sintáctica y semántica para el interfuncionamiento de los dispositivos y las redes dentro del sistema, siendo la tecnología Máquina a Máquina (Machine-to-Machine M2M en inglés) la que permite a las computadoras, los procesadores integrados, los sensores inteligentes, los actuadores y los dispositivos móviles comunicarse entre sí, realizar medidas y tomar decisiones, sin necesidad de la intervención humana (LBNL, 2004, p. 1) (Daniele, 2015, p. 32).

- Son bastantes las tecnologías de dispositivos, aplicaciones y redes de alcance local y personal existentes en el mercado –tales como UPnP, Sep-2, Z-Wave, PowerOnt, OMS, KNX, SSN, ZigBee, CoAP, NFC, HGI, Bluetooth, Wi-Fi Alliance, IEEE 802.15.4, etc.–, por lo que el análisis de las mismas realizado por Daniele et al. en el año 2015 a partir de sus activos semánticos (modelos de datos, protocolos, APIs, estándares, etc.) (pág. 35), facilitó la elaboración de una ontología de referencia consensuada y compartida por los diversos fabricantes, cubriendo las necesidades de todos los artefactos relevantes para la eficiencia energética y superando los problemas de conectividad confusa (ib., p. 26), la cual se conoce como la ontología de referencia para dispositivos inteligentes SAREF.
- Dicha ontología la caracteriza una serie de fundamentos tales como: reutilización y alineación de conceptos y relaciones, que se definen en los activos existentes; modularidad, para permitir la separación y recombinación de diferentes partes de la ontología según necesidades específicas; extensibilidad, para permitir un mayor crecimiento de la ontología; mantenibilidad, para facilitar el proceso de identificación y corrección de defectos, adaptarse a los nuevos requisitos y afrontar los cambios en (partes de) la ontología SAREF (ibídem, p. 154) (ETSI, 2017, p. 7).

Tales principios de la ontología SAREF revelan tendencias de uso de los dispositivos enfocadas en su funcionalidad, su configuración energética y el espacio construido en el cual se encuentran ubicados (óp. cit., p. 131) (ob. cit., p. 12). En cuanto a los perfiles de potencia que puede tener un artefacto inteligente, dicho aspecto adquiere relevancia al corroborar que el consumo de energía en el sector residencial y de servicios se mantiene, según los datos al año 2016 (Eurostat, 2018) (UPME, 2018) tanto en la Unión Europea como en Colombia que se observan en la tabla 5:

Consumo por Sector	Consumo de energía año 2016					
	Unión Europea (óp. cit.)			Colombia (ob. cit.)		
	GWh	KToe	%	GWh	KToe	%
Total (final)	12.882.118	1.107.663	100	57.239	4.922	100
Residencial	3.311.792	284.763	25.7	22.690	1.951	39,6
Servicios	1.744.953	150.039	13.5	13.217	1.136	23,1

Suma Resid. y Serv.	5.056.745	434.802	39.3	35.907	3.087	62.7
---------------------	-----------	---------	------	--------	-------	------

Tabla 9. Consumo de energía por sector en la Unión Europea y Colombia año 2016 (Eurostat, 2018) (UPME, 2018).

No sobra aclarar que la unidad “GWh” corresponde a giga vatios hora (Giga Watt hour), como también “KToe” atañe a kilo toneladas de petróleo equivalente (Kilo Tonne oil equivalent), siendo la conversión entre dicho par de unidades así: 1 KToe = 11,63 GWh. El uso de la unidad “GWh” a pesar de no ser una unidad fundamental o derivada del Sistema Internacional SI, refleja de manera apropiada la naturaleza de la magnitud de energía (NIST, 2008, p. 40). Así mismo, la unidad “KToe” se usa a menudo al expresar la cantidad de energía producida o consumida en términos de una fuente o energético como lo es el petróleo, facilitando la correspondencia con las unidades básicas de energía, en este caso, el gigavatio hora (APS, 2018).

Todo lo anterior va de la mano con los cambios en los patrones de uso de energía tomando diversas formas en la actualidad. Los "energy resources" recursos energéticos incluyen combustibles fósiles como el carbón, el petróleo y el gas natural, además de flujos de energía renovables como la eólica y la solar (UPME, 2015, p. 8) (O'Connor, 2010, 8), siendo estos llamados fuentes energéticas primarias. Tales fuentes se transforman en "carriers" o portadores de energía, como la electricidad o la gasolina, conocidos también como fuentes energéticas secundarias. Dichos portadores se suministran luego a "converters" o convertidores de energía, como una bombilla fluorescente compacta o un automóvil (i.e. dispositivos o máquinas) y últimamente, se utilizan para proporcionar "services" o servicios de energía como iluminación o transporte (ibidem), por ejemplo el alumbrado público o la red ferroviaria eléctrica.

Un conjunto particularmente significativo de cambios en los patrones de uso de energía en una sociedad, es conocido como “transición energética” (UPME, 2015, p. 8) (O'Connor, 2010, 8); la cual puede afectar cualquier paso de dicho patrón o cadena como también varios de ellos. Dicha transición adquiere relevancia ante la preocupación mundial por el cambio climático y la concentración atmosférica de CO₂ (óp. cit.), por lo que la optimización de la eficiencia energética del sistema desde el concurso de los dispositivos inteligentes hace parte de tal metamorfosis, al suministrar un modelo conceptual compartido u ontología de referencia como lo es SAREF (Daniele, 2015, p. 25), estableciendo una apertura de la estandarización de interfaces entre los artefactos en mención con la semántica requerida de manera estricta.

2.3.5 Ontología de Referencia de Dispositivos Inteligentes para Energía SAREF4ENER

La ontología de Referencia de Dispositivos Inteligentes para el dominio de Energía –en inglés Smart Appliances REFERENCE ontology for the ENERGY domain SAREF4ENER– versión 1.0, es un esquema de conceptos y de relaciones principales que deriva o extiende de una ontología superior llamada SAREF (Smart Appliances REFERENCE ontology), caracterizándose aquella por manejar un universo de discurso específico consistente en la interconexión de dispositivos inteligentes para cubrir escenarios de respuesta a la demanda de energía sobre la red eléctrica inteligente (Smart Grid en inglés) (ETSI, 2017a, p. 11).

La ontología SAREF4ENER ha sido creada en colaboración con Energy@Home (<http://www.energy-home.it>) y EEBus (<http://www.eebus.org/en>), siendo la principal asociación italo-alemana de la industria, para permitir la interconexión de sus respectivos modelos de datos (ETSI,

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

2017b, p. 6) (TNO, 2016a). SAREF4ENER está enfocada en erigir y exponer un sistema conceptual primitivo que satisfaga el consumo final de electricidad, facilitando que los clientes puedan ofrecer flexibilidad a la red eléctrica inteligente (Smart Grid) para gestionar sus dispositivos domésticos inteligentes, mediante un gestor de energía de cliente (Customer Energy Manager CEM en inglés). El CEM es una función lógica que permite optimizar el gasto y/o la producción de energía, pudiendo residir tanto en una puerta de enlace doméstica como en la nube (ibídem).

Los diversos escenarios que cubre la ontología SAREF4ENER se describen a partir de los siguientes casos de uso:

- Caso de uso 1: atañe a la configuración de dispositivos que desean conectarse entre sí en la red doméstica; por ejemplo, registrar un nuevo lavaplatos en la lista de dispositivos administrados por el gestor de energía de cliente CEM (ETSI, 2017a, p. 13) (ETSI, 2017b, p. 6) (Daniele, et. al., 2015a, p. 1), tal y como se observa en la figura 51:

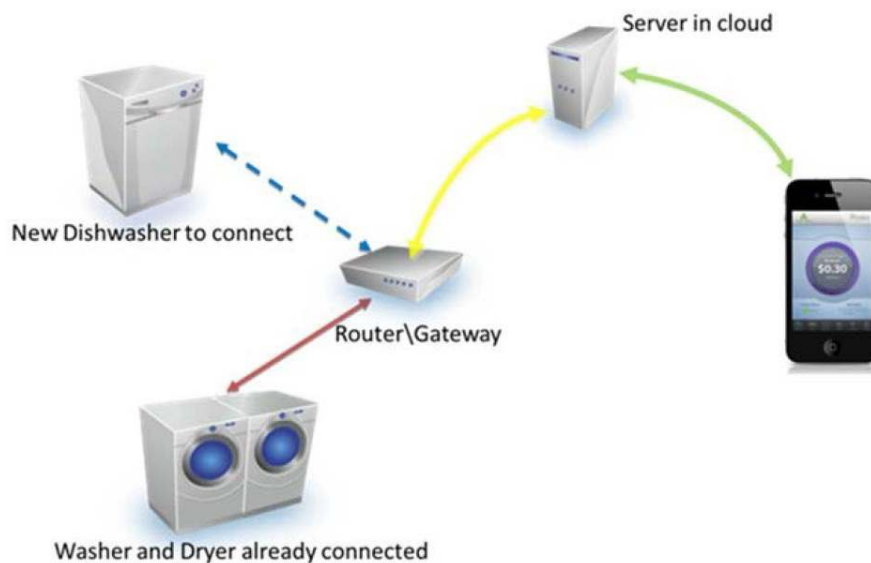


Figura 51. Caso de uso 1: registro de un nuevo dispositivo (ETSI, 2017a, p. 13) (Daniele, et. al., 2015a, pág. 1)

El nuevo dispositivo debe llegar a ser visible para el gestor de energía del cliente CEM, a fin de que este sea agregado a la lista de artefactos a administrar (lo cual se conoce como “aprovisionamiento”). Dicha máquina nueva debe suministrar al CEM tanto su ID como sus capacidades (ibídem). Igualmente, la aplicación móvil debe proporcionar los lineamientos sencillos al usuario acerca de cómo completar el proceso de aprovisionamiento, i. e. tramitación de registro (ibíd.).

- Caso de uso 2: una gestión inteligente de energía (re-)programando los dispositivos en ciertos modos y tiempos elegidos, utilizando perfiles de potencia para optimizar la eficiencia energética y satisfacer las preferencias del cliente (ETSI, 2017a, p. 13) (ETSI, 2017b, p. 7) (Daniele, et. al., 2015a, p. 2).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

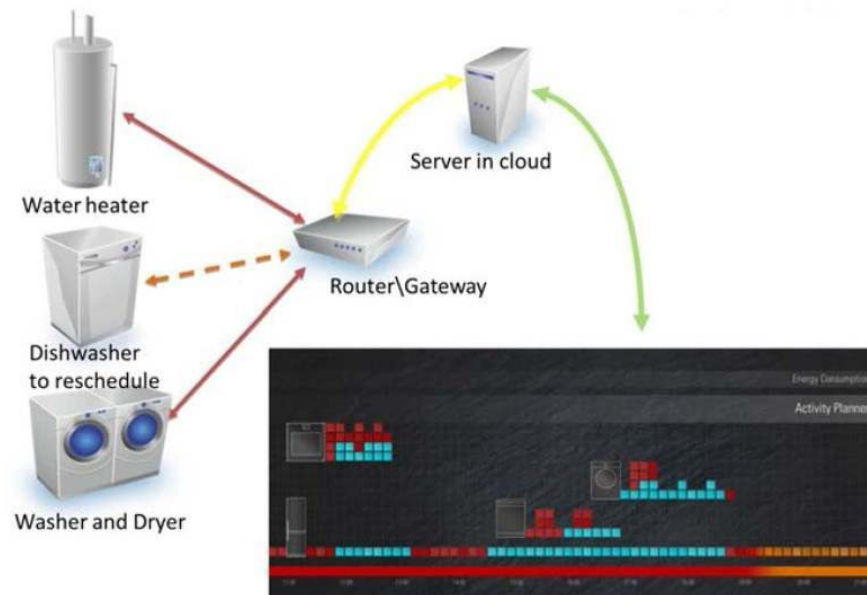


Figura 52. Caso de uso 2: Reprogramación de dispositivos mediante perfiles de potencia (ETSI, 2017a, p. 13) (Daniele, 2015a, pág. 2)

La programación del dispositivo se lleva a cabo cuando el CEM recibe la información de dicho artefacto que está listo para arrancar ya en modo de control remoto o de manera manual. El usuario puede ver en su aplicación móvil las tarifas para el día y las áreas de incentivos para el consumo o ahorro de energía, lo que le permite planificar cuándo iniciar el dispositivo a distancia y ver qué otra máquina ya se está ejecutando o está planificada para comenzar (ibíd.). Es factible igualmente realizar la reprogramación automática del dispositivo, ya que el CEM recibe la información de la empresa de servicios públicos o del intermediario, quién indica que existe un tiempo más conveniente para que el dispositivo ejecute su tarea. En la aplicación móvil, el usuario puede ver el nuevo horario para el inicio del artefacto elegido, como también determinar un nuevo plan para evitar horas pico (ibídem).

- Caso de uso 3: monitoreo y control del inicio y el estado de los dispositivos (ETSI, 2017a, p. 14) (ETSI, 2017b, p. 7) (Daniele, et. al., 2015a, p. 2).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

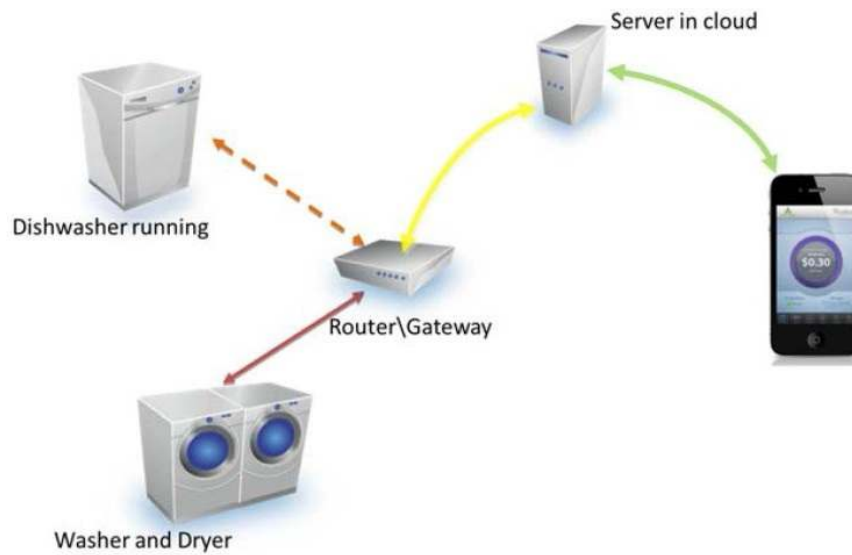


Figura 53. Caso de uso 3: Monitoreo y control del estado de un dispositivo (ETSI, 2017a, p. 14)
(Daniele, et. al., 2015a, pág. 2)

La información sobre el estado de un dispositivo, v. g. un lavavajillas (estados como: encendido/apagado, tiempo restante, perfil de potencia, hora programada para comenzar, etc.) se proporciona al gestor de energía de cliente CEM, incluso si la máquina se ha configurado manualmente para comenzar (idem).

- Caso de uso 4: respuesta a solicitudes especiales de la red eléctrica inteligente “smart grid”, por ejemplo, en incentivos para un menor consumo dependiendo de la disponibilidad de energía actual, o en situaciones de emergencia que requieren una reducción temporal del consumo de electricidad en cuanto a potencia (ETSI, 2017a, p. 14) (ETSI, 2017b, p. 7) (Daniele, et. al., 2015a, p. 3).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

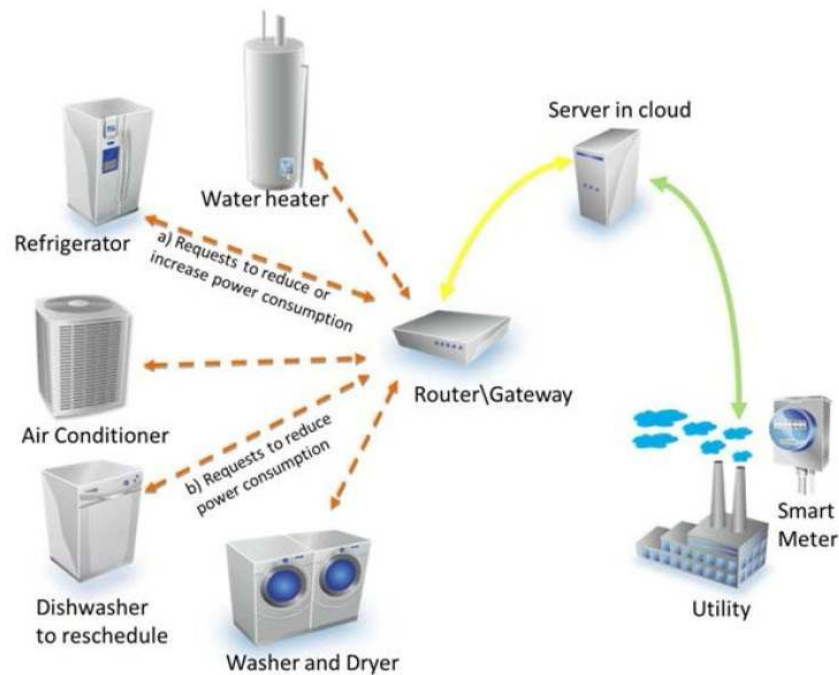


Figura 54. Caso de uso 4: Respuesta a solicitudes de la Red Eléctrica Inteligente (ETSI, 2017a, p. 14) (Daniele, et. al., 2015a, pág. 3)

El primer requerimiento atañe a que la CEM recibe la información de la empresa de servicios públicos sobre incentivos, ya sea para consumir más energía o para reducir el consumo total por debajo de un nivel definido (ibídem). Algunas máquinas cuyo comportamiento no está basado en ciclos (v. g. aire acondicionado, calentador de agua, etc.), pueden modificar su configuración para cumplir con la solicitud aumentando o disminuyendo la energía en función de su capacidad y lo que están haciendo, por lo que la notificación se proporciona al usuario a través de la interfaz de su aplicación móvil (ibíd).

Un segundo requerimiento consiste en que la CEM recibe la información de la empresa de servicio público de energía o del medidor inteligente, indicando que se ha presentado una emergencia y se solicita a los dispositivos la reducción del consumo durante algunos minutos (ibídem). Dichos máquinas disminuyen su gasto energético acorde a sus funciones y tareas, reportando al cliente mediante una interfaz de usuario. Por último, los artefactos que cuentan con una planificación para comenzar un ciclo durante dicha ventana de tiempo crítica serán reprogramados (ibíd.).

La ontología de referencia de dispositivos inteligentes para energía SAREF4ENER es una iniciativa desprendida de los trabajos realizados por Daniele L. –encargados por la Organización Neerlandesa para la Investigación Científica Aplicada TNO– acerca del modelo conceptual primitivo SAREF, quién elaboró una primera versión llamada SAREF4EE (Daniele, 2015a). La continuación de dicho esfuerzo fue asumida por el Instituto Europeo de Estándares en Telecomunicaciones ETSI al año siguiente, elaborando la especificación técnica ETSI TS 103 410-1 y publicándola dos años después (ETSI, 2017b), bajo el título “M2M inteligente; extensión de dispositivos inteligentes para SAREF; Primera parte: dominio de energía”, encargándose tanto de su mantenimiento como de su implementación y actualización anual.

Las representaciones semánticas al interior de la ontología SAREF4ENER, en esencia se estructuran como se observa en la figura 55 – elaborada por el autor a partir de (TNO, 2015), (Daniele, 2015a), (TNO, 2016a) y (ETSI, 2017b, p. 8)– y en consonancia con los lineamientos del Lenguaje de Ontología Web OWL-DL, en el cual los “conceptos y roles” de la lógica descriptiva se asimilan a “clases y propiedades”, sean ya estas últimas tanto atributos como relaciones respectivamente (W3C, 2004).

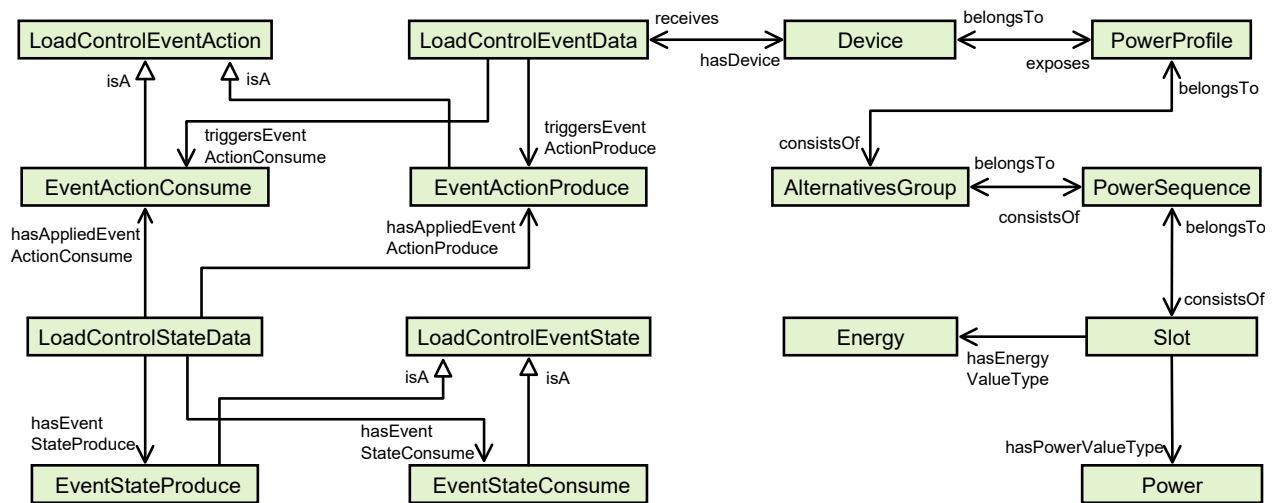


Figura 55. Estructura básica conceptual de la ontología SAREF4ENER (elaborada por el autor fundado en (TNO, 2015) (Daniele, 2015a) (TNO, 2016a) y (ETSI, 2017b, p. 8))

La notación de dichos conceptos o clases al interior de la ontología SAREF4ENER, se establecen mediante el prefijo “s4ener”, el cual se separa de la desinencia que indica el concepto aludido utilizando el símbolo ortográfico “:”, por ejemplo “s4ener:Device”, “s4ener:PowerProfile”, “s4ener:LoadControlEventData”, etc. (ETSI, 2017b, p. 7–17). Igualmente, los roles o propiedades cuentan con la escritura “prefijo:relación” (llamada también “ObjectProperties”) o “prefijo:atributo” (nombrada como “DataProperties”), ya sea una propiedad de categoría relación, o de categoría atributo, según corresponda (oneM2M, 2016, p. 8) (ETSI, 2017b, p. 7). Se tiene por ejemplo: s4ener:belongsTo”, “s4ener:hasEnergyValueType”, s4ener:”hasEventStateConsume”, etc., para relaciones; “s4ener:slotActivated”, “s4ener:hasTimeStamp”, “s4ener:sequenceRemoteControllable”, etc., para atributos, ya que estos últimos son características de un tipo de dato predefinido (texto, numérico entero, etc.) que pueden variar en lo referente a contenido (óp. cit.).

Todo lo anterior fue establecido por el ETSI mediante un Comité Técnico, que definió una estrategia para identificar y desarrollar extensiones de la ontología SAREF adicionales a la de respuesta a la demanda energética (o dominio de energía), tales como medio ambiente, edificios, ciber salud/envejecimiento y agricultura (ETSI, 2017a). La información sobre los requisitos de estos dominios (conocidos como extensiones) y la definición de las pautas para su mantenimiento y publicación, se compendió en el reporte técnico ETSI TR 103 411 del año 2017 con la nominación “M2M inteligente; Dispositivos inteligentes; investigación de la extensión SAREF”, lo cual facilita la recopilación de requisitos y la inclusión de los aportes de los principales actores de la industria en cuanto a la tecnología máquina a máquina (idem).

Componentes de la ontología SAREF4ENER

Las descripciones de los principales conceptos de SAREF4ENER (cuyo prefijo es “s4ener:”) se despliegan como sigue:

- Dispositivo (Device):

Un dispositivo en la ontología SAREF4ENER “s4ener:Device” es una subclase del mismo concepto en la ontología SAREF “saref:Device” (ETSI, 2017b, p. 10), cuya definición es “artefacto u objeto tangible diseñado para llevar a cabo una tarea particular en hogares, oficinas o edificios públicos comunes y corrientes” (inciso 2.3.4.1 de la presente investigación). Un dispositivo se caracteriza por tener un perfil energético con unas series de potencia específicas, las cuales expone a un gestor de energía de cliente CEM (TNO, 2016a) (TNO, 2015). Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a un “s4ener:Device” se observan en la tabla 10 (óp. cit.):

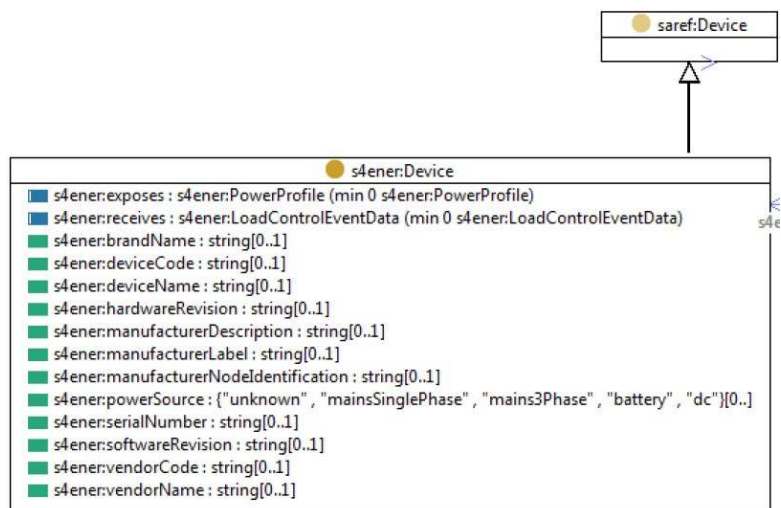


Figura 56. Clase Dispositivo (Device) de SAREF4ENER (ETSI, 2017b, pág. 10) (TNO, 2016a) (TNO, 2015)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:exposes	Vínculo consistente en la exhibición de un perfil de potencia “s4ener:PowerProfile” por parte de un dispositivo.
	s4ener:receives	Enlace referente a la admisión de los datos de un evento de control de carga “s4ener:LoadControlEventData” por parte de un dispositivo.
Atributo (DataProperty)	s4ener:brandName	El nombre de la marca del dispositivo.
	s4ener:deviceCode	Código del dispositivo definido por el fabricante.
	s4ener:deviceName	Nominación del dispositivo definida por el fabricante.
	s4ener:hardwareRevision	Actualización del hardware del dispositivo definida por el fabricante.
	s4ener:manufacturerDescription	Reseña del dispositivo hecha por el fabricante.
	s4ener:manufacturerLabel	Etiqueta del dispositivo establecida por el fabricante.
	s4ener:manufacturerNodeid	Identificación del nodo del dispositivo definido por el fabricante; útil cuando se retira el artefacto de la red y se reconecta con otra interfaz.
	s4ener:powerSource	Fuente de energía de un dispositivo tales como: red monofásica o trifásica, batería, corriente directa, etc.
	s4ener:serialNumber	Número serial del dispositivo establecido por el fabricante.

	s4ener:softwareRevision	Actualización del software del dispositivo conforme a lo definido por el fabricante.
	s4ener:vendorCode	Código para el vendedor del dispositivo determinado por el fabricante.
	s4ener:vendorName	Nombre del vendedor del dispositivo según lo definido por el fabricante.

Tabla 10. Propiedades de la clase Dispositivo (Device) (ETSI, 2017b, pág. 10)

- Perfil de Potencia (Power Profile):

Un perfil de potencia es una forma de modelar las curvas (comportamientos) de potencia y energía a lo largo del tiempo, que también proporciona definiciones para el modelado de la programación de energía, incluidos los planes alternativos (TNO, 2016a) (TNO, 2015). Con un perfil (configuración) de potencia, un dispositivo (o un servidor de series de potencia) expone las secuencias que son potencialmente relevantes para el gestor de energía del cliente CEM (o de un cliente de series de potencia) (óp. cit.).

Dicho "s4ener:PowerProfile" o perfil de potencia es una subclase de la clase "saref:Profile" perteneciente a SAREF como se observa en las figuras 45 y 57; es decir, hereda las propiedades de tal superclase y la extiende con propiedades adicionales que son específicas para SAREF4ENER (ETSI, 2017b, p. 11). Por lo tanto y de manera detallada, un perfil de potencia "s4ener:PowerProfile" es utilizado por un dispositivo "s4ener:Device" para exponer las series de potencia (clase s4ener:PowerSequence) que son significativas para el CEM, como también un "s4ener:Device" puede exponer un "s4ener:PowerProfile", que consiste en uno o más planes alternativos (clase s4ener:AlternativesGroup) (ibídem).

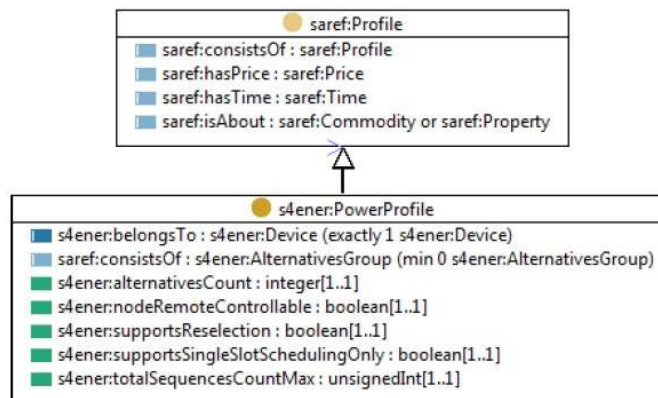


Figura 57. Clase Perfil de Potencia (Power Profile) de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 11) (TNO, 2015)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a un perfil de potencia “s4ener:PowerProfile” se observan en la tabla 11 (óp. cit., pág. 12):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:belongsTo	Vínculo consistente en la pertenencia de un perfil de potencia “s4ener:PowerProfile” a un y solo un dispositivo “s4ener:Device”.
	saref:consistsOf	Enlace referente a la conformación de un perfil de potencia “s4ener:PowerProfile” por uno o varios grupos de alternativas “s4ener:AlternativesGroup”. Esta propiedad es heredada de la clase perfil “Profile”, perteneciente a la ontología SAREF
Atributo (DataProperty)	s4ener:alternativesCount	Cantidad de grupos de alternativas proporcionados por un perfil de potencia.
	s4ener: nodeRemoteControllable	Característica booleana que establece si el dispositivo está configurado para ser controlado remotamente por el CEM. Un valor verdadero permite dicho control.
	s4ener:supportsReselection	Atributo lógico que determina si el CEM puede seleccionar de un grupo de alternativas, series de potencia de manera indistinta o una sola a la vez. Un valor falso no permite la re selección.
	s4ener: supportsSingleSlotScheduling Only	Característica booleana que establece para un dispositivo, el no permitir alterar la configuración de una fase (slot) de una serie de potencia perteneciente a un grupo de alternativas. Un valor falso permite la modificación.
	s4ener:totalSequencesCount Max	Sumatoria de las series de potencia soportadas por el dispositivo, pertenecientes a

		todos los grupos de alternativas.
--	--	-----------------------------------

Tabla 11. Propiedades de la clase Perfil de Potencia (Power Profile) (ETSI, 2017b, pág. 12)

- Grupo de Alternativas (Alternatives Group):

Un grupo de alternativas “s4ener:AlternativesGroup” consiste en una colección de series de potencia “s4ener:PowerSequence” para un perfil de energía determinado “s4ener:PowerProfile” (TNO, 2016a) (TNO, 2015). Esta clase se compone de un solo atributo o “DataProperty” de identificación del grupo de alternativas con tipo de dato entero (s4ener:alternativesGroupID), al igual que de un par de relaciones u “ObjectProperty” que le permiten enlazarse tanto con un único perfil de potencia (s4ener:belongsTo), como con una o varias series de energía (saref:consistsOf); siendo esta última propiedad heredada de la ontología de referencia para dispositivos inteligentes SAREF (óp. cit.) (ob. cit.).

A continuación se detallan las relaciones y atributos (“ObjectProperties” y “DataProperties”) de un grupo de alternativas, tanto en la figura 58 (ETSI, 2017b, pág. 11), como en la tabla 12 (ibídem, p. 12).

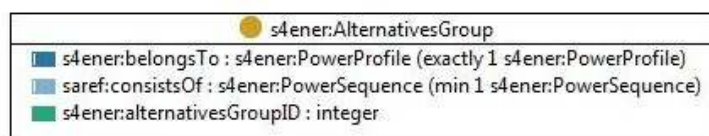


Figura 58. Clase Grupo de Alternativas (Alternatives Group) de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 11) (TNO, 2015)

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:belongsTo	Vínculo consistente en la pertenencia de un conjunto o Grupo de Alternativas “s4ener:AlternativesGroup” a uno y solo un Perfil de Potencia “s4ener:PowerProfile”.
	saref:consistsOf	Enlace referente a la conformación de un Grupo de Alternativas “s4ener:AlternativesGroup” por una o varias Series de Potencia “s4ener:PowerSequence”. Esta propiedad es heredada de la clase perfil “Profile”, perteneciente a la ontología SAREF

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Atributo (DataProperty)	s4ener: AlternativesGroupID	Atributo referente a un identificador único y absoluto para las todas las instancias de un Grupo de Alternativas "s4ener:AlternativesGroup", que es proporcionado por un Perfil de Energía "s4ener:PowerProfile".
----------------------------	--------------------------------	---

Tabla 12. Propiedades de la clase Grupo de Alternativas (Alternatives Group) (ETSI, 2017b, p. 12)

- Serie de Potencia (Power Sequence):

Una serie de potencia es la representación de todos los pasos individuales de una tarea completa; dicho de otro modo, es la especificación de una actividad, verbi gratia, la labor de secado en un dispositivo inteligente como la lavadora, acorde a las preferencias del usuario o la configuración predeterminada del fabricante (TNO, 2016a) (TNO, 2015). Dicha secuencia de acciones que conforman la serie de potencia se halla simbolizada por fases o estadios (slots), siendo manejada mediante el gestor de energía del cliente CEM (óp. cit.).

Es de tener en cuenta que una "s4ener:PowerSequence" o serie de potencia pertenece sólo y exactamente a un grupo de alternativas o "s4ener:AlternativesGroup" y en simultánea, esta última clase consta de una o más de tales secuencias energéticas, observándose tanto sus detalles en la figura 59 (ETSI, 2017b, p. 13), como sus relaciones y atributos ("ObjectProperties" y "DataProperties") en la tabla 13 (ibídem).

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:belongsTo	Vínculo consistente en la pertenencia de una serie de potencia "s4ener:PowerSequence" a un y solo un grupo de alternativas "s4ener:AlternativesGroup".
	saref:consistsOf	Enlace referente a la conformación de una serie de potencia "s4ener:PowerSequence" por una o varias fases "s4ener:Slot". Esta propiedad es heredada de la clase perfil "Profile", perteneciente a la ontología SAREF
	s4ener:hasEnergy	Vínculo consistente en la energía estimada para reanudar el funcionamiento normal del dispositivo, posterior a una pausa en una serie

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		de potencia con la entidad "s4ener:ResumeEnergyEstimated". Esta propiedad es heredada de la clase "Energy", perteneciente a la ontología SAREF.
	saref:hasPrice	Relación que atañe al costo adicional en la energía estimada, en que se incurre para reanudar el funcionamiento normal del dispositivo, posterior a una pausa en una serie de potencia con la entidad "s4ener:ResumeCostEstimated". Esta propiedad es heredada de la clase "Price", vinculada a la ontología SAREF.
	saref:hasState	Enlace referente al estado actual de la serie de potencia con la entidad "s4ener:powerSequenceState". Puede asumir uno de los siguientes valores: 'en ejecución', 'pausado', 'programado', 'programado en pausa', 'pendiente', 'inactivo', 'completado' o 'no válido'. Dicha propiedad es heredada de la clase "State", perteneciente a la ontología SAREF.
	saref:hasTime	Propiedad heredada de la clase "Time" atinente a la ontología SAREF. Dicha relación atañe al manejo del tiempo en una serie de potencia mediante diversas clases, las cuales se detallan en la tabla 14.
	saref:hasDescription	Relación que concierne a la especificación detallada para una serie de potencia "PowerSequence". Esta propiedad es heredada de la clase "Description", vinculada a la ontología SAREF.
Atributo (DataProperty)	s4ener:activeRepetitionNumber	Propiedad tipo atributo que contiene la cantidad actual de repeticiones de una secuencia de fases o estadios. Si el elemento "saref4ener:repetitionsTotal" posee un valor

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		mayor que uno, este atributo estará activo; de lo contrario, permanecerá ausente.
	s4ener: activeSlotNumber	Atributo referente al número de fase operante, el cual, si la clase "s4ener:PowerSequenceState" se encuentra configurado en "ejecución" o "pausa", este elemento deberá contener dicho guarismo de lapso activo; de lo contrario, será omitido.
	s4ener:cheapest	Atañe a la configuración más barata posible para una serie de potencia, definida por el Gestor de Energía de Cliente CEM al minimizar la factura de energía del usuario. Un valor verdadero para este elemento indica que se ha de aplicar dicha reducción.
	s4ener:greenest	Concierno a la configuración más óptima que maximice la disponibilidad de energía renovable para una serie de potencia, definida por el Gestor de Energía de Cliente CEM. Un valor verdadero para este elemento indica que se ha de aplicar dicha optimización.
	s4ener:isPausable	Corresponde a la posibilidad de pausar la secuencia energética, por parte del Gestor de Energía de Cliente CEM. Un valor verdadero para este atributo significa que se ha de aplicar dicha suspensión.
	s4ener:isStoppable	Es referente a la factibilidad de detener la serie de potencia, por parte del Gestor de Energía de Cliente CEM. Un valor verdadero para este elemento significa que se ha de aplicar dicha interrupción.
	s4ener: maxCyclesPerDay	Atañe a la cantidad máxima de arranques o ciclos por día, que el dispositivo permite llevar a cabo.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	s4ener:repetitionsTotal	Propiedad tipo atributo que contiene la cantidad de repeticiones de una secuencia de fases o estadios. Si este elemento se halla vacío, equivale a la existencia de la iteración con valor cero. Si el valor del elemento es 1, significa que no hay reiteraciones.
	s4ener:sequenceID	Consiste en un identificador de secuencia único y completamente definido para la serie de potencia. Debe ser un valor entero positivo.
	s4ener:sequenceRemoteControllable	Atañe a si la serie de potencia es modificable o no de manera remota. De requerirse alterar alguna secuencia energética y/o sus fases (slots) en cuanto a configuración o estado, el valor de este elemento debe ser verdadero.
	s4ener:taskIdentifier	Consiste en un identificador único de tarea, para los tipos recurrentes de series de potencia usados por un dispositivo, de tal manera que puedan ser reutilizados posteriormente.
	s4ener:valueSource	Se refiere a la fuente u origen de los valores de medición pronosticados para una serie de potencia en específico. Si dicho valor se encuentra ausente, indica que la fuente en cuestión no está definida.

Tabla 13. Propiedades de la clase Serie de Potencia (Power Sequence) (ETSI, 2017b, pág. 13)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

s4ener:PowerSequence	
s4ener:belongsTo	s4ener:AlternativesGroup (exactly 1 s4ener:AlternativesGroup)
s4ener:hasEnergy	s4ener:ResumeEnergyEstimated (max 1 s4ener:ResumeEnergyEstimated)
saref:consistsOf	s4ener:Slot (min 1 s4ener:Slot)
saref:hasPrice	s4ener:ResumeCostEstimated (max 1 s4ener:ResumeCostEstimated)
saref:hasState	s4ener:PowerSequenceState (min 1 s4ener:PowerSequenceState)
saref:hasTime	(max 1 s4ener:EarliestStartTime, max 1 s4ener:ActiveDurationSumMax, max 1 s4ener:LatestEndTime, max 1 s4ener:Ac
s4ener:activeRepetitionNumber	unsignedInt[0..1]
s4ener:activeSlotNumber	unsignedInt[0..1]
s4ener:cheapest	boolean[0..1]
s4ener:greenest	boolean[0..1]
s4ener:isPausable	boolean[0..1]
s4ener:isStoppable	boolean[0..1]
s4ener:maxCyclesPerDay	unsignedInt[0..1]
s4ener:repetitionsTotal	unsignedInt[0..1]
s4ener:sequenceID	unsignedInt[1..1]
s4ener:sequenceRemoteControllable	boolean[1..1]
s4ener:taskIdentifier	unsignedInt[0..]
s4ener:valueSource	{"measuredValue", "calculatedValue", "empiricalValue"}[0..]
saref:hasDescription	string[0..1]

Figura 59. Clase Serie de Potencia (Power Sequence) de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 13) (TNO, 2015)

El manejo del tiempo en una serie de potencia en la ontología SAREF4ENER, es una propiedad que se relaciona con diversos conceptos, los cuales han sido heredados de la clase "Time" de la ontología SAREF, los cuales se listan a continuación en la tabla 14 (ETSI, 2017b, p. 14):

Propiedad Tipo Relación (Object Property)		Definición
Nombre	Vinculada al Concepto	
saref:hasTime	s4ener:ActiveDurationMax	Consiste en la duración máxima activa de la serie de potencia (o secuencia energética) a la que puede funcionar el dispositivo sin interrupción.
	s4ener:ActiveDurationMin	Ataño a la duración mínima activa de la serie de potencia (o secuencia energética) a la que puede funcionar el dispositivo sin interrupción.
	s4ener:ActiveDurationSumMax	Se refiere a la duración máxima activa de la secuencia energética a la que puede funcionar el dispositivo a su capacidad total; i.e., la sumatoria de todos los tiempos activos).
	s4ener:ActiveDurationSumMin	Consiste en la duración mínima activa de la secuencia energética a la que puede funcionar el dispositivo a su capacidad exigua; i.e., la sumatoria

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		de todos los tiempos activos).
	s4ener:StartTime	Ataño a la hora de inicio de la serie de potencia para el dispositivo. Dicho valor debe estar siempre presente.
	s4ener:EarliestStartTime	Concierne a la hora de inicio más temprana posible para cualquier serie de potencia configurada para al dispositivo. Dicho valor debe estar siempre presente.
	s4ener:EndTime	Corresponde a la hora de finalización de la secuencia de energía. Si el valor está disponible, se indicará aquí. De lo contrario, el elemento debe omitirse.
	s4ener:LatestEndTime	Se refiere a la última hora de finalización posible para toda la secuencia de energía configurada para el dispositivo.
	s4ener:ElapsedSlotTime	Consiste en el tiempo transcurrido de la fase (slot). Si el estado de la secuencia de energía se configura en 'ejecución' o en 'pausa' y la fase o estadio (slot) se halla definida, este elemento puede contener el tiempo que, de la fase, ya ha estado en la instancia de 'ejecución' (esto también significa que el valor permanece constante durante el estado de 'pausa'). En caso contrario, el valor se omitirá.
	s4ener:RemainingSlotTime	Ataño al tiempo restante de la fase (slot). Si el estado de la serie de potencia se configura en 'ejecución' o en 'pausa' y la fase o estadio (slot) se halla definida, este elemento debe contener el tiempo que la fase aun requiere para alcanzar el estado de 'ejecución' (esto también significa que el valor permanece constante durante el estado de 'pausa'). En caso contrario, el valor se omitirá.
	s4ener:PauseDurationMax	Se refiere a la duración máxima que puede ser

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		pausada la secuencia de energía después del final de una actividad.
	s4ener:PauseDurationMin	Corresponde a la duración mínima que puede ser pausada la secuencia de energía después del final de una actividad.

Tabla 14. Elementos de la propiedad tipo relación “saref:hasTime” de la ontología SAREF4ENER para la clase “Serie de Potencia” o “PowerSequence” (ETSI, 2017b, pág. 14)

- Fase (Slot):

Una fase o “slot” es una ventana de tiempo o estadio, que representa un paso individual de una serie de potencia, el cual está asociado con un valor numérico entero positivo incremental, asignándose de manera singular para cada serie de potencia con la cual se relaciona (TNO, 2016a) (TNO, 2015), reconociéndose esta última mediante un identificador de secuencia energética.

Por lo anterior, los números de ventana de tiempo de dos series de potencia deben considerarse independientes entre sí; v.g., una fase o “slot” con valor numérico ‘a’ de la secuencia de energía ‘X’, describe un estadio diferente a la fase número ‘a’ de la serie de potencia ‘Z’. Por lo tanto, un lapso o “slot” sólo se identifica unívocamente en combinación con un ID de serie energética (ibídem) (óp. cit.).

Una ventana de tiempo en la ontología SAREF4ENER está definida por la clase “s4ener:Slot”, la cual pertenece a una y solamente una serie de potencia “s4ener:PowerSequence”. En simultánea, dicha secuencia energética consta de uno o más fases o “slots” (ETSI, 2017b, p. 14), tal y como se observa en la figura 60 (ibíd., pág. 15). Así mismo, sus relaciones y atributos (“ObjectProperties” y “DataProperties”) se detallan en la tabla 15 (ibídem, pág. 16).

The screenshot shows the class 's4ener:Slot' with the following properties and constraints:

- s4ener:belongsTo: s4ener:PowerSequence (exactly 1 s4ener:PowerSequence)
- s4ener:hasEnergyValueType: s4ener:Energy
- s4ener:hasPowerValueType: s4ener:Power
- s4ener:hasValueType: (min 1 s4ener:Energy or s4ener:Power)
- saref:hasTime: saref:Time (max 1 s4ener:StartTime, max 1 s4ener:DefaultDuration, max 1 s4ener:MinDuration, max 1 s4ener:EndTime, max 1 s4ener:MaxDuration)
- s4ener:optionalSlot: boolean[0..1]
- s4ener:slotActivated: boolean[0..1]
- s4ener:slotNumber: unsignedInt[1..1]
- saref:hasDescription: string[0..1]

Figura 60. Clase “Fase” o “Slot” de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 15) (TNO, 2015)

Propiedad (Property)		Definición
Tipo	Nombre	

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Relación (ObjectProperty)	s4ener:belongsTo	Vínculo consistente en la pertenencia de una fase o ventana de tiempo “s4ener:Slot” a una y solo una serie de potencia “s4ener:PowerSequence”.
	s4ener:hasEnergyValueType	Enlace referente a la representación de un tipo de valor energético. Esta propiedad es heredada de la clase “Energy”, perteneciente a la ontología SAREF.
	s4ener:hasPowerValueType	Vínculo que concierne a la representación de un tipo de valor de potencia. Esta propiedad es heredada de la clase “Power”, perteneciente a la ontología SAREF.
	saref:hasValueType	Relación correspondiente a la categoría de fuerza motriz representada por las clases de energía “saref:energy” o potencia “saref:power”, heredadas de la ontología SAREF. La energía puede ser del tipo “s4ener:EnergyMin”, “s4ener:EnergyMax”, “s4ener:EnergyExpected”, “s4ener:EnergyStandardDeviation” o “s4ener:EnergySkewness”. Así mismo la potencia podría ser del tipo “s4ener:PowerMin”, “s4ener:PowerMax”, “s4ener:PowerExpected”, “s4ener:PowerStandardDeviation” o “s4ener:PowerSkewness”.
	saref:hasTime	Propiedad heredada de la clase “Time” atinente a la ontología SAREF. Dicha relación atañe al manejo del tiempo en fase o ventana de tiempo mediante diversas clases, las cuales se detallan en la tabla 16.
	saref:hasDescription	Relación que concierne a la especificación detallada para una fase “Slot”. Esta propiedad es heredada de la clase “Description”, vinculada a la ontología SAREF.

Atributo (DataProperty)	s4ener:optionalSlot	Propiedad tipo atributo que toma el valor “verdadero” si la fase es factible de omitirse o si el numero de la fase se repite. De lo contrario, deberá establecerse como “falso”.
	s4ener:SlotActivated	Atributo que refleja el estado actual de la fase, el cual es verdadero si el elemento “s4ener:optionalSlot” también lo es. De ser falso este último, la activación de la fase no existirá.
	s4ener:SlotNumber	Esta propiedad de tipo atributo consiste en un identificador de fase único de serie de potencia según el Id correspondiente.

Tabla 15. Propiedades de la clase “Fase” o “Slot” (ETSI, 2017b, pág. 16)

La gestión del tiempo en una fase o ventana de tiempo (slot) en la ontología SAREF4ENER, es una propiedad que se relaciona con diversos conceptos, los cuales han sido derivados o heredados de la clase “Time” de la ontología SAREF (ETSI, 2017b, p. 14). Tales clases se catalogan como sigue en la tabla 16:

Propiedad Tipo Relación (Object Property)		Definición
Nombre	Vinculada al Concepto	
saref:hasTime	s4ener:DefaultDuration	Consiste en la duración de una fase determinada, cuyo valor debe ser el que se esté midiendo en el momento de su configuración, para una serie de potencia específica.
	s4ener:MaxDuration	Atañe a la duración máxima admitida en una fase para una secuencia energética, si la duración predeterminada existe; i.e., si la clase “s4ener:DefaultDuration” está configurada.
	s4ener:MinDuration	Se refiere a la duración mínima soportada en una fase para una secuencia energética, si la duración predeterminada existe; i.e., si la clase

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		“s4ener:DefaultDuration” está configurada. Este elemento aplica únicamente para la primera repetición del número de la fase.
	s4ener:DurationUncertainty	Consiste en la incertidumbre de la duración de la fase, cuyo valor figura al interior de la clase “s4ener:DefaultDuration”.
	s4ener:StartTime	Atañe a la hora de inicio de la fase para el dispositivo. Dicho valor debe estar siempre presente.
	s4ener:EarliestStartTime	Concierne a la hora de inicio más temprana posible para la fase configurada perteneciente a una serie de potencia del dispositivo. Dicho valor debe estar siempre presente.
	s4ener:EndTime	Corresponde a la hora de finalización de la ventana de tiempo configurada para la secuencia energética del dispositivo. Se debe aplicar la siguiente ecuación: $EndTime - StartTime = DefaultDuration$.
	s4ener:LatestEndTime	Se refiere a la última hora de finalización posible para la fase configurada perteneciente a una serie de potencia del dispositivo.
	s4ener:RemainingPauseTime	Atañe al tiempo que la fase (slot) actual permite a la serie de potencia estar en pausa. Este elemento estará presente sólo si la secuencia energética es interrumpible; i.e., si la clase heredada de la ontología SAREF “saref:isInterruptionPossible” posee valor verdadero.

Tabla 16. Elementos de la propiedad tipo relación “saref:hasTime” de la ontología SAREF4ENER para la clase “Fase” o “Slot” (ETSI, 2017b, pág. 16)

- Control de Carga (Load Control):

Todo dispositivo inteligente a la fecha, requiere de un suministro energético que le permita ejecutar su(s) tarea(s), tanto de percepción como de actuación. Dado que el propósito de la presente tesis se enfoca en la gestión eficiente de dicho recurso al ser consumido por el

artefacto en cuestión, la ontología SAREF4ENER define el modelado de supervisión de eventos concernientes a la administración de carga directa y a escenarios de reducción de energía (ETSI, 2017b, p. 16); un ejemplo de ello se denota en el caso 4, figura 54 (ETSI, 2017, p. 14)(ETSI, 2017b, p. 7)(Daniele, et. al., 2015a, p. 3).

Dicho modelado ha sido especificado en un conjunto de clases llamado Control de Carga o "Load Control", las cuales se denomina como Datos del Evento "s4ener:LoadControlEventData", Acciones del Evento "s4ener:LoadControlEventAction", Datos del Estado "s4ener:LoadControlStateData" y Estados del Evento "s4ener:LoadControlEventState" (TNO, 2016a) (ETSI, 2017b, p. 16). Cada uno de tales conceptos se describe a continuación:

- Control de Carga de Datos del Evento (Load Control Event Data):

La clase "s4ener:LoadControlEventData" consiste en la representación del nivel de gravedad de la advertencia de sobrecarga (overload) y los comandos de control de carga relacionados con un dispositivo. Tal evento o acontecimiento se caracteriza por poseer un identificador del evento (s4ener:eventID), una marca de tiempo (saref:hasTimestamp) que representa la hora en que se creó o recibió la instancia de información del evento, y un período de tiempo (saref:hasTime) que denota el lapso de validez del suceso según la figura 61 (ibídem, pág. 17) (óp. cit.).

El siguiente ejemplo explica mejor tales características mencionadas: si se recibió hace 5 minutos, un evento referente a la entrada en vigencia del mismo a partir de mañana desde las 14:00 hasta las 15:00 horas, la configuración del Control de Carga de Datos del Evento (LoadControlEventData) en cuanto a Marca de Tiempo (saref:hasTimestamp) será "hace 5 minutos" y el Período de Tiempo (saref:hasTime) quedará como "mañana de 14:00 a 15:30" (ETSI, 2017b, pág. 17).

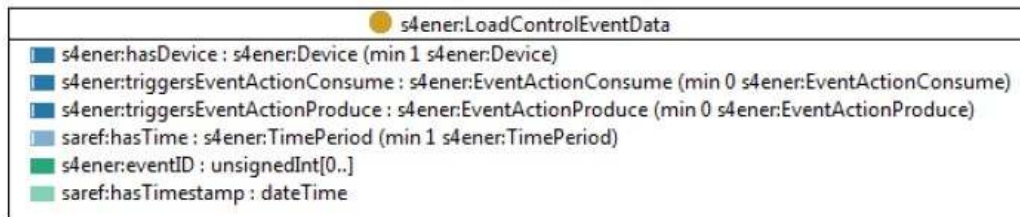


Figura 61. Clase Control de Carga de Datos del Evento o "LoadControlEventData" de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

Es de tener en cuenta que este concepto se encuentra asociado a la clase de Control de Carga de Acciones de Evento "s4ener:LoadControlEventAction", (véase figura 62), mediante las relaciones de desencadenamiento de sucesos de consumo o producción de energía o potencia "s4ener:triggersEventActionConsume" y "s4ener:triggersEventActionProduce" respectivamente (TNO, 2016a) (ETSI, 2017b, pág. 17). Así mismo, se detallan a continuación las relaciones y atributos ("ObjectProperties" y "DataProperties") de un control de carga para datos de un evento o "s4ener:LoadControlEventData", tanto en la figura 61, como en la tabla 17 (ibídem) (ob. cit).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:hasDevice	Vínculo consistente en la pertenencia de un Control de Carga de Datos de un Evento “s4ener:LoadControlEventData” a un y solo un dispositivo “s4ener:Device”.
	s4ener: triggersEventActionConsume	Enlace referente al desencadenamiento de uno o varios Eventos del tipo de Acción Consumir, con la subclase “s4ener:EventActionConsume”, la cual está heredando de la superclase de Control de Carga de Datos del Evento “s4ener:LoadControlEventAction”. La acción o suceso puede ser: pausar, reanudar, reducir, incrementar, emergencia o normal.
	s4ener: triggersEventActionProduce	Vínculo que atañe al lanzamiento de uno o varios Eventos del tipo de Acción Producir, con la subclase “s4ener:EventActionProduce”, la cual está heredando de la superclase de Control de Carga de Datos del Evento “s4ener:LoadControlEventAction”. La acción o acontecimiento puede ser la emergencia, la normalidad, la pausa, la reanudación, la reducción o el incremento.
	saref:hasTime	Enlace consistente en la disposición de un Período de Tiempo “s4ener:TimePeriod” o varios de ellos, por parte de un Control de Carga de Datos de un Evento “s4ener:LoadControlEventData”. Esta propiedad es heredada de la clase “Time” perteneciente la ontología SAREF.

Atributo (DataProperty)	s4ener:eventID	Consiste en un identificador único de Evento, para los tipos recurrentes de Control de Carga de Datos usados por un dispositivo, de tal manera que puedan ser reutilizados posteriormente por la clase "LoadControlStateData".
	saref:hasTimestamp	Ataño a la configuración de una marca de tiempo que establece la fecha y la hora del requerimiento de registrar el control de carga de un evento. Esta atributo es heredado de la clase "Time" perteneciente la ontología SAREF.

Tabla 17. Propiedades de la Clase "LoadControlEventData" de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

- Control de Carga de Acciones del Evento (Load Control Event Action):

La clase "s4ener:LoadControlEventAction" concierne al tipo de acciones o ejecuciones que se realizarán como consecuencia de un evento de control de carga, la cual funge como una superclase. Dichas acciones pueden ser categorizadas de consumo o de producción de energía o potencia, mediante las subclases "s4ener:EventActionConsume" o "s4ener:EventActionProduce" respectivamente, tal y como se aprecia en la figura 62 (TNO, 2016a) (ETSI, 2017b, p. 17). Los valores que pueden tomar los ejercicios de consumir y/o producir pueden ser: de emergencia "s4ener:emergency", de incremento "s4ener:increase", de normalidad "s4ener:normal", de pausa "s4ener:pause", de reducción "s4ener:reduce", y por último de reanudación "s4ener:resume" (ibídem) (óp. cit.).

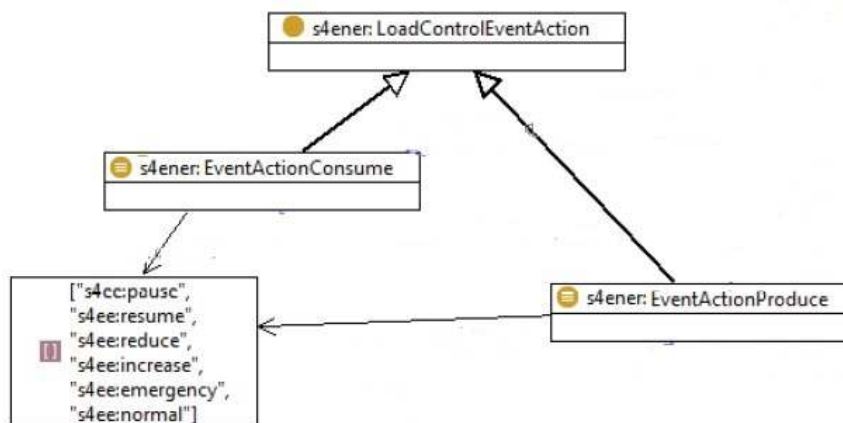


Figura 62. Clase Control de Carga de Acciones del Evento o "LoadControlEventAction" de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

Igualmente, en la figura anterior se puede observar la herencia existente de las subclases Acción del Evento de Consumir “s4ener:EventActionConsume” y Acción del Evento de Producir “s4ener:EventActionProduce” con la superclase Control de Carga de Acciones del Evento “s4ener:LoadControlEventAction”, como también la enumeración ya mencionada de ambas subclases (ibídem) (óp. cit.).

- Control de Carga de Datos del Estado (Load Control State Data):

La clase “s4ener:LoadControlStateData” atañe a los datos acerca del estado o condición de un evento, caracterizándose por poseer el mismo Identificador de Evento “s4ener:eventID” utilizado en la clase de Control de Carga de Datos del Evento (s4ener:LoadControlEventData), como también la Marca de Tiempo “saref:hasTimestamp” (TNO, 2016a) (ETSI, 2017b, p. 17). Así mismo, dicho concepto se encuentra asociado a un par de clases: la primera se refiere a “s4ener:LoadControlEventState” (véase figura 64), encargada de gestionar la carga en cuanto a estados del evento mediante las relaciones de configuración de sucesos de consumo o producción de energía o potencia “s4ener:EventStateConsume” y “s4ener:eventStateProduce” respectivamente (ibíd.) (óp. cit.).

La segunda clase con la que se encuentra asociado este concepto (i.e., s4ener:LoadControlStateData) compete a “s4ener:LoadControlEventAction” (véase figura 62), cuyo propósito es el de administrar la carga de acciones para un suceso a través de las relaciones de ejecución de eventos aplicados al consumo o la producción de energía o potencia “s4ener:EventActionConsume” y “s4ener:EventActionProduce” respectivamente (TNO, 2016a) (ETSI, 2017b, pág. 17).

Así mismo, se detallan a continuación las relaciones y atributos (“ObjectProperties” y “DataProperties”) de un control de carga para datos de un evento o “s4ener:LoadControlEventData”, tanto en la figura 61, como en la tabla 18 (ibíd.) (ob. cit.).

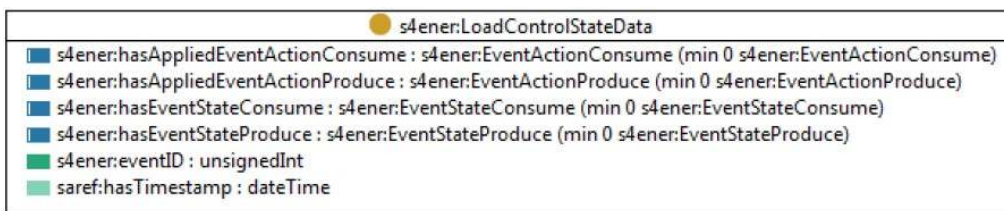


Figura 63. Clase Control de Carga de Datos del Estado o “LoadControlStateData” de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	s4ener:hasAppliedEventActionConsume	Enlace referente a la aplicación de una acción para uno o varios Eventos del tipo Consumir, con la

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		<p>subclase “s4ener:EventActionConsume”, la cual está heredando de la superclase de Control de Carga de Datos del Evento “s4ener:LoadControlEventAction”. La acción o suceso puede ser: pausar, reanudar, reducir, incrementar, emergencia o normal.</p>
	s4ener:hasAppliedEventActionProduce	<p>Vínculo que atañe a la puesta en marcha de una acción de uno o varios Eventos del tipo Producir, con la subclase “s4ener:EventActionProduce”, la cual está heredando de la superclase de Control de Carga de Datos del Evento “s4ener:LoadControlEventAction”. La acción o acontecimiento puede ser la emergencia, la normalidad, la pausa, la reanudación, la reducción o el incremento.</p>
	s4ener:hasEventStateConsume	<p>Enlace que expresa el estado de uno o varios Eventos del tipo Consumir, con la subclase “s4ener:EventStateConsume”, la cual está heredando de la superclase de Control de Carga de Estados del Evento “s4ener:LoadControlEventState”. El estado o condición puede ser: evento aceptado, evento iniciado, evento detenido, evento rechazado, evento cancelado o error de evento.</p>
	s4ener:hasEventStateProduce	<p>Vínculo consistente en el estado de uno o varios Eventos del tipo Producir, con la subclase “s4ener:EventStateProduce”, la cual está heredando de la superclase de Control de Carga de Estados del Evento “s4ener:LoadControlEventState”. El estado o configuración puede ser: suceso aceptado, suceso iniciado, suceso detenido, suceso rechazado, suceso cancelado o suceso errado.</p>

Atributo (DataProperty)	s4ener:eventID	Consiste en un identificador único del evento, para los tipos recurrentes de Control de Carga de Estados usados por un dispositivo, de tal manera que puedan ser reconocidos posteriormente por la clase "LoadControlEventData". Ambos "eventID" deben coincidir, asignándose en primera instancia el perteneciente a la clase "s4ener:LoadControlEventData" y replicándose este en la clase "s4ener:LoadControlStateData".
	saref:hasTimestamp	Ataño a la configuración de una marca de tiempo que establece la fecha y la hora del requerimiento de registrar el control de carga de los datos del estado, de tal manera que pueda ser reconocido posteriormente por la clase "LoadControlEventData". Esta atributo es heredado de la clase "Time" perteneciente la ontología SAREF. Ambos atributos "Timestamp" deben coincidir, asignándose en primera instancia el perteneciente a la clase "s4ener:LoadControlEventData" y replicándose este en la clase "s4ener:LoadControlStateData".

Tabla 18. Propiedades de la Clase "LoadControlStateData" de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

- Control de Carga de Estados del Evento (Load Control Event State):

La clase "s4ener:LoadControlEventState" expresa el tipo de estados o configuraciones que se determinan como consecuencia de un evento de control de carga, la cual funge como una superclase. Dichos estados pueden ser categorizadas de consumo o de producción de energía o potencia (de manera análoga a una acción de evento "LoadControlEventData" ya descrita dos incisos atrás), mediante las subclases "s4ener:EventStateConsume" o "s4ener:EventStateProduce" respectivamente, tal y como se aprecia en la figura 64 (TNO, 2016a) (ETSI, 2017b, p. 17). Los valores que pueden tomar las actividades de consumir y/o producir pueden ser: de evento aceptado "s4ener:eventAccepted", de evento iniciado "s4ener:eventStarted", de evento detenido "s4ener:eventStopped", de evento rechazado "s4ener:eventRejected", de evento cancelado "s4ener:eventCancelled", y por último de evento errado "s4ener:eventError" (ibídem) (óp. cit.).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

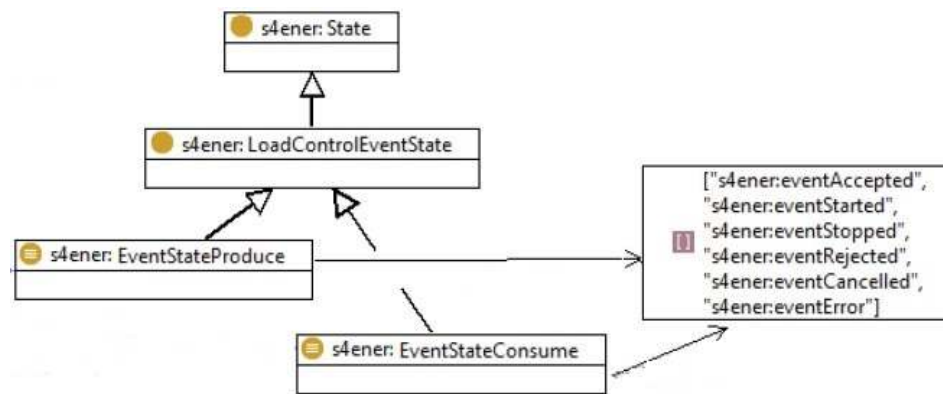


Figura 64. Clase Control de Carga de Estados del Evento o “LoadControlEventState” de SAREF4ENER (TNO, 2016a) (ETSI, 2017b, pág. 17)

De igual manera, en la figura anterior se puede apreciar la herencia existente de las subclases Estado del Evento de Consumir “s4ener:EventStateConsume” y Estado del Evento de Producir “s4ener:EventStateProduce” con la superclase Control de Carga de Estados del Evento “s4ener:LoadControlEventState”, al igual que la enumeración ya mencionada de ambas subclases (TNO, 2016a) (ETSI, 2017b, p. 17).

La tabla 19 compendia a continuación las clases referentes al control de carga en la ontología SAREF4ENER, tal y como las define la Organización Neerlandesa para la Investigación Científica Aplicada TNO o “Netherlands Organization for Applied Scientific Research” (TNO, 2016a), al igual que las nominaciones con las que contaba en la primera versión de la ontología conocida como SAREF4EE (TNO, 2015) (Daniele, 2015a, pág. 14).

Clases de Control de Carga (Load Control)	Reseña
Datos del Evento: “s4ener:LoadControlEventData”	En la ontología SAREF4ENER, dicha clase consiste en un evento utilizado para enviar el nivel de gravedad de la advertencia de sobrecarga y los comandos de control de carga relacionados con un dispositivo, por ejemplo, un electrodoméstico o un medidor inteligente (TNO, 2016a). En la primera versión de la ontología, conocida como SAREF4EE, la clase se titulaba “s4ee:DemandResponseEventData” o

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	<p>Respuesta a la Demanda de Datos del Evento (TNO, 2015) (Daniele, 2015a, pág. 14).</p>
<p>Acciones del Evento: "s4ener:LoadControlEventAction"</p>	<p>Dicha clase en la ontología SAREF4ENER, atañe a un tipo de acción utilizado para expresar la actividad que se llevará a cabo, como consecuencia de un evento utilizado para enviar el nivel de gravedad de la advertencia de sobrecarga, con los comandos de control de carga relacionados a un dispositivo; v.g., un electrodoméstico o un medidor inteligente (TNO, 2016a). En la primera versión de la ontología, nominada como SAREF4EE, la clase se titulaba "s4ee:DemandResponseEventAction" o Respuesta a la Demanda de Acciones del Evento (TNO, 2015) (Daniele, 2015a, pág. 14).</p>
<p>Datos del Estado: "s4ener:LoadControlStateData"</p>	<p>Al interior de la ontología SAREF4ENER, esta clase se refiere a la representación del estado de un evento, que se utiliza para enviar el nivel de gravedad de la advertencia de sobrecarga y los comandos de control de carga relacionados con un dispositivo, p. ej., un electrodoméstico o un medidor inteligente (TNO, 2016a). En la primera versión de la ontología, conocida como SAREF4EE, la clase se titulaba "s4ee:DemandResponseStateData" o Respuesta a la Demanda de Datos del Estado (TNO, 2015) (Daniele, 2015a, pág. 14).</p>
<p>Estados del Evento: "s4ener:LoadControlEventState"</p>	<p>En la ontología SAREF4ENER, dicha clase expresa los posibles estados de un evento utilizados para enviar el nivel de gravedad de la advertencia de sobrecarga, acompañado de los comandos de control de carga relacionados con un dispositivo; verbi gratia, un electrodoméstico o un medidor inteligente (TNO, 2016a). En la primera versión de la ontología, nominada como SAREF4EE, la clase se titulaba "s4ee:DemandResponseState" o Respuesta a la Demanda de Estados de un evento (TNO, 2015) (Daniele, 2015a, pág. 14).</p>

Tabla 19. Compendio de las clases de control de carga de SAREF4ENER (TNO, 2016a) (TNO, 2015) (Daniele, 2015a, pág. 14)

2.3.6 Ontología Máquina a Máquina Monádica oneM2M

Esta especificación conceptual ha sido diseñada por oneM2M, la cual es un proyecto de asociación entre organizaciones de desarrollo de estándares de TIC a nivel mundial, que desde el año 2012 funge como iniciativa para la normalización de las comunicaciones Máquina a Máquina y la Internet de las Cosas, conocidas igualmente como “Machine to Machine M2M” e “Internet of Things IoT” (oneM2M, 2017).

La finalidad de oneM2M es exponer con orden y amplitud los componentes, las características y el funcionamiento propios del intercambio de información digital máquina a máquina (oneM2M, 2018, pág. 2), de tal manera que estas dirijan el cumplimiento de las necesidades de una capa de servicios común entre máquinas, como lo son los dispositivos de bajo consumo energético. Dicha capa se ha de integrar de una manera sencilla en las diversas tecnologías informáticas (hardware y software) (óp. cit.), con la convicción de comunicar íntegra y fiablemente, la enorme cantidad de dispositivos electrónicos existentes en el mundo con servidores de aplicación M2M sin el concurso del ser humano (ob. cit.).

En el inciso 2.2 del presente trabajo (página 36), se encuentra la definición de ontología (Grønbaek, 2008); no obstante, oneM2M se permite enunciar que una ontología es un vocabulario con una estructura, el cual se aplica a un dominio de interés específico (por ejemplo, medición, electrodomésticos, medicina, etc.), conteniendo conceptos que se utilizan al interior de dicho universo de discurso (oneM2M, 2019, pág. 8) (ETSI, 2021, pág. 9).

Dado que una ontología debería capturar una comprensión compartida de un dominio de interés y en simultánea, proporcionar un modelo formal de dicho ámbito y que sea interpretable por cualquier máquina (óp. cit.) (ob. cit.), se hace necesario denominar y enumerar los conceptos atinentes al dominio en cuestión con unos significados bien definidos. Es por ello que los conceptos se denominan “Clases” según el estándar OWL (W3C, 2012a) (W3C, 2014a) –conocido como el Lenguaje de Ontologías Web–, ya que estos últimos identifican a los tipos, categorías o clases de individuos (oneM2M, 2019, pág. 8) (ETSI, 2021, pág. 9).

Según el OWL, la estructura de la ontología se establece mediante relaciones bien concertadas y definidas entre sus conceptos (W3C, 2012a) (W3C, 2014a); dichas relaciones se conocen como el nombre de “Propiedad de Objeto” (en inglés, ObjectProperty), las cuales permiten vincular un concepto de ‘sujeto’ con un concepto de ‘objeto’ (oneM2M, 2019, pág. 8) (ETSI, 2021, pág. 9). El esquema de dicha estructura sería el siguiente:

Concepto de Sujeto → Relación → Concepto de Objeto

En el Lenguaje de Ontologías Web OWL sería:

Clase de Dominio → Propiedad de Objeto → Clase de Recorrido

Un ejemplo de tal configuración conforme a la ontología SAREF es el siguiente:

Dispositivo (Device) → Lleva a cabo (Accomplish) → Tarea (Task)

Otro ejemplo bajo la ontología oneM2M sería:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Servicio (Service) → Expone (Exposes) → Función (Function)

Una sinopsis de la estructura ontológica y sus ejemplos en cuanto a la relación de Propiedad de Objeto, se observa en la siguiente tabla:

Ontología	Esquema (de vinculación)
Nominal	Concepto de Sujeto → Relación → Concepto de Objeto
OWL	Clase de Dominio → Propiedad de Objeto → Clase de Recorrido
SAREF	Dispositivo → Lleva a cabo una → Tarea (Device → Accomplish → Task)
oneM2M	Servicio → Expone una → Función (Service → Exposes → Function)

Tabla 20. Relación entre conceptos al interior de una ontología en cuanto a **Propiedad de Objeto** (Elaborada por el autor)

Existe un segundo tipo de propiedades en OWL conocido como "Propiedades de datos" (en inglés, DataProperty), la cual permite vincular a una Clase de Sujeto, una pieza de información. Es (oneM2M, 2019, pág. 8) (ETSI, 2021, pág. 9) opcional que dichos datos dispongan de tipología, mas es recomendable ya que facilita la gestión de los mismos. El modelo respectivo se presenta a continuación:

Clase de Dominio → Propiedad de Datos → Clase de Recorrido

Un ejemplo según la ontología SAREF es el siguiente:

Dispositivo (Device) → Tiene un fabricante (hasManufacturer) → Dato Literal (LiteralData)

Conforme a la ontología oneM2M, otro ejemplo sería:

Operación (Operation) → Dispone del procedimiento (Method) → Literal simple (PlainLiteral)

A continuación, se aprecia en la tabla No. 21, un compendio de la estructura ontológica y sus ejemplos en cuanto a la relación de Propiedad de Datos:

Ontología	Esquema (de vinculación)
Nominal	Concepto de Sujeto → Relación → Concepto de Objeto
OWL	Clase de Dominio → Propiedad de Datos → Clase de Recorrido
SAREF	Dispositivo → Tiene un Fabricante → Dato Literal (Device → Accomplish → Task)

oneM2M	Operación → Dispone del procedimiento → Literal Simple (Operation → Method → PlainLiteral)
--------	---

*Tabla 21. Relación entre conceptos al interior de una ontología en cuanto a **Propiedad de Datos** (Elaborada por el autor)*

Dado que toda ontología es susceptible de ser interpretada por máquinas y para lo cual, ha de ser codificable en un lenguaje informático, dicha característica –como ya se mencionó en línea atrás (inciso 2.3.4) –, permite construir sentencias de búsqueda de datos que cumplan con uno o varios criterios (v.g., el lenguaje de consulta SPARQL) (W3C, 2014, inciso 7), obteniendo con ello un conjunto de individuos o instancias de clases con relaciones específicas. Dichas sentencias son elaboradas siguiendo las normas de la lógica descriptiva y/o la lógica de predicados, razón suficiente para que la ontología oneM2M haya sido codificada en OWL, asegurando que tales argumentaciones o silogismos sean decidibles (oneM2M, 2019, pág. 9) (ETSI, 2021, pág. 10)

Otra particularidad de la ontología oneM2M y su representación en OWL, atañe a proporcionar interoperabilidad sintáctica y semántica con sistemas externos que cuentan con su respectivo universo de discurso codificado en dicho lenguaje (óp. cit.) (ob. cit.), garantizando con ello el mapeo de los mismos –en cuanto a conceptos y relaciones– a la ontología base de oneM2M mediante una subclasificación, una equivalencia, etc., al facilitar la caracterización de tipos específicos de dispositivos (como por ejemplo, en la ontología SAREF) o, más generalmente, la descripción de “Cosas” del mundo real (como edificios, habitaciones, automóviles, ciudades, etc.) al interior del ámbito de oneM2M con su respectiva gramaticalidad y significación (óp. cit.) (ob. cit., pág. 11).

Al mencionar la interoperabilidad sintáctica, se hace referencia al interfuncionamiento (en inglés, “interworking”) con dispositivos que no son oneM2M en redes (en inglés, “network”) de área tanto local como global. Ello se logra mediante una ontología representada como un archivo OWL, que contenga la tipificación de los parámetros de comunicación específicos de la red –verbi gratia, los nombres de operaciones, nombres de parámetros de entrada/salida, sus tipos y estructuras, etc.–, la cual se utiliza para la configuración de una Entidad Proxy de Interfuncionamiento (en inglés, Interworking Proxy Entity IPE) (oneM2M, 2019, pág. 9) (ETSI, 2021, pág. 11), facilitando la asignación de recursos del sistema oneM2M como una Entidad de Aplicación o un Contenedor (en inglés, Application Entity AE or Container), para llevar a cabo actividades que lean desde -o escriban hacia- estos recursos, de manera que el IPE pueda serializar los datos y enviarlos hacia -o recibirlos desde- los dispositivos en la red de área (óp. cit.) (ob. cit.).

En cuanto a la interoperabilidad semántica, esta aplica a la descripción de funciones que son suministradas por dispositivos que son compatibles con el sistema oneM2M –es decir, dispositivos M2M o “Machine to Machine appliances”–, los cuales se mapean en recursos oneM2M tales como contenedores o “Containers”, nombres de recurso o “resourceNames”, estructuras descendientes o “child-resources” y por último, tipos de contenido o “contentTypes” (oneM2M, 2019, pág. 10) (ETSI, 2021, pág. 11).

Un ejemplo de dicha interoperabilidad semántica atañe al dispositivo “lavadora”, el cual al ser compatible con oneM2M y contar con diversas versiones de tal artefacto, todas ellas pueden realizar funciones como “lavar”, “secar”, “elegir temperatura”, etc., independientemente de que estas tareas cuenten con diferentes nombres de recurso, estructuras descendientes y tipos de contenido, al estar definidos como recursos del sistema oneM2M (óp. cit.) (ob. cit.). Por lo tanto, un archivo OWL correspondiente a la ontología, contendrá las funciones anteriores agrupadas en un servicio común o de aplicación ofrecido por el dispositivo M2M.

De lo anterior se desprende un aspecto relevante de la ontología oneM2M, el cual atañe a la interpretación de una clase, no solo como una representación de un concepto o como un conjunto de individuos, sino también su definición mediante las propiedades –o relaciones– distintivas de las instancias de dicha clase. Dicho método se conoce con el nombre de “restricción”. Las clases que se especifican a partir de una restricción se les llaman “clases anónimas”, las cuales agrupan a todas las instancias que satisfacen tal condición (oneM2M, 2019, pág. 10) (ETSI, 2021, pág. 12). Existe una clasificación de las restricciones a saber; universal, existencial y de cardinalidad:

- La restricción universal describe a un dominio o clase de individuos que, para una propiedad determinada, solo tienen una y solo una relación según dicha propiedad, con individuos que son miembros de una clase recorrida. Por ejemplo, dado que una subclase “Válvula de agua” (en inglés, Watervalue) de la superclase “Dispositivo” (Class: Device), solo tiene una función o tarea (Object Property: hasFunction) de nombre “Abrir o cerrar válvula” (en inglés, Open_or_Close_Valve) y perteneciente a la subclase “Función” (Class: Function), entonces la subclase “Válvula de agua” (Class: Watervalue) es una superclase de la clase anónima formada por la propiedad de objeto “Tiene una función” (Object Property: hasFunction) y que posee una única relación con la clase “Abrir o cerrar válvula” (Class: Open_or_Close_Valve) (óp. cit.) (ob. cit.).
- La restricción existencial detalla a la clase de individuos –o conjunto de partida–, que participa como mínimo en una relación según una propiedad dada, con individuos que son miembros de la clase codominio. Verbigracia, dado que una clase “Dispositivo” (Class: Device) tiene por lo menos una función o actividad (Object Property: hasFunction), en relación con la clase “Función” (Class: Function) y que dicho artefacto realiza, entonces el dispositivo en mención (Class: Device) es una subclase de la clase anónima constituida por la propiedad de objeto “Tiene una función” (Object Property: hasFunction) y que dispone –cuando menos– de alguna asociación con la clase “Función” (Class: Function) (oneM2M, 2019, pág. 10) (ETSI, 2021, pág. 12).
- La restricción de cardinalidad reseña a la clase de individuos que funge como dominio –o conjunto de definición– y que, para una propiedad dada, solo tienen un número específico de relaciones a lo largo (o según) de dicha propiedad, con individuos que son miembros de la clase que ejerce como recorrido o conjunto de llegada (óp. cit.) (ob. cit.). P. ej., si para una clase “Servicio” (Class: Service), existe una operación es obligatoria (Class:Operation), entonces la propiedad de objeto relacionada “Tiene una operación” (Object Property: hasOperation), dispondrá de una restricción de propiedad (en inglés, Property Restriction) con cardinalidad de “tan solo una” (oneM2M, 2019, pág. 26) (ETSI, 2021, pág. 27).

Una última faceta del sistema oneM2M concierne a su Ontología Base (en inglés, “Base Ontology” BO), cuyo diseño procura suministrar una cantidad mínima de conceptos, relaciones y condicionamientos, las cuales se requieren para el descubrimiento semántico de las entidades con su correspondiente representación en clases, propiedades de objeto y restricciones (clases anónimas); i.e., en recursos oneM2M llamados también, contenedores. La detección de los elementos (o entidades) pertenecientes a la ontología de una tecnología oneM2M –con su respectiva especificidad– por parte de la BO de oneM2M, también ha de hacerse mediante la connotación descriptiva de las mismas como contenedores y a continuación, asociarlas a modo de subclases y por analogía, con las clases ya predefinidas en la Ontología Base, lo que facilita garantizar la interoperabilidad semántica (oneM2M, 2019, pág. 12) (ETSI, 2021, pág. 13).

Así mismo, el diseño de la Ontología Base BO proporciona un marco de trabajo para la creación de ontologías que pertenezcan a tecnologías que no son oneM2M (en inglés, “non-oneM2M”), las cuales mediante la descripción de su propio modelo de datos en contenedores (clases, propiedades de objeto y restricciones), permite asegurar la interoperabilidad sintáctica; o dicho de otro modo, afianzar el interfuncionamiento (en inglés, “interworked”) con y entre dispositivos “non-oneM2M” haciendo uso de una Entidad Proxy IPE inherente al sistema oneM2M (ob. cit.) (óp. cit.).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Adicionalmente, la Ontología Base de oneM2M se encuentra desarrollada en el Lenguaje de Ontologías Web OWL y disponible en un repositorio en la nube (oneM2M, 2021), con sus respectivas actualizaciones; por lo que –y para efectos del presente estudio–, dicha especificación se replica en el anexo 1 con su correspondiente esquematización en la figura No. 65 a manera de grafo, facilitando la visualización de los conceptos y las relaciones que conforman tal Ontología Base, id est, las clases y sus propiedades de objeto (oneM2M, 2019, pág. 13) (ETSI, 2021, pág. 14). Dichos conceptos están denotados mediante elipses (nodos), así como las relaciones se hallan indicadas por flechas (aristas). Es de tener en cuenta que en la ilustración se observa una clase con sombreado gris y de nombre “Variable”, significando que dicho concepto se repite múltiples veces en el modelo oneM2M.

The oneM2M Base Ontology

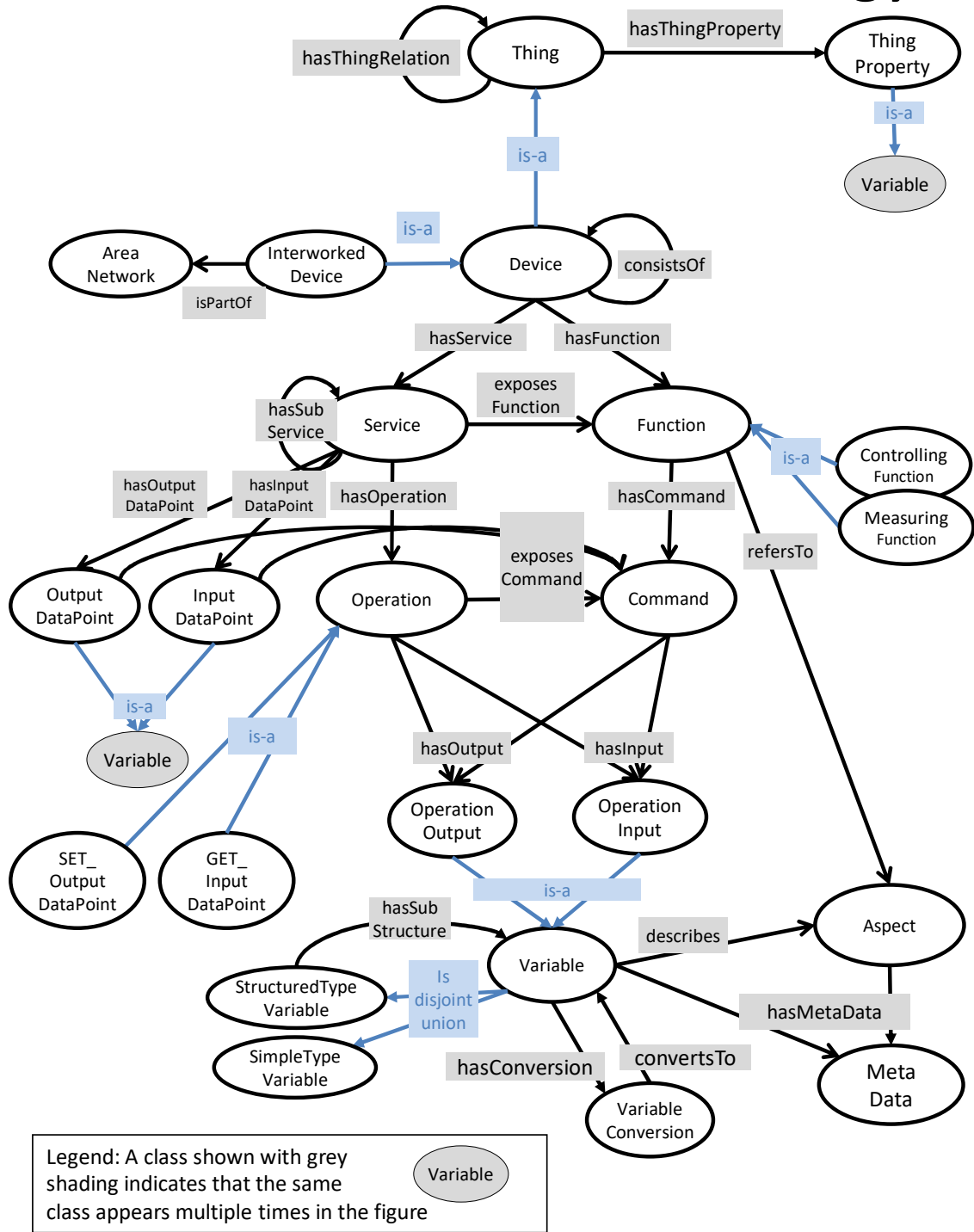


Figura 65. Ontología Fundamental oneM2M (oneM2M, 2019, pág. 13) (ETSI, 2021, p. 14)

Componentes de la Ontología Base oneM2M

Las descripciones referentes a las clases, las propiedades de objeto, las propiedades de datos y las restricciones, al interior de la Ontología Base de oneM2M, se despliegan como sigue:

- Cosa (Thing):

Una Cosa (Class:Thing) es una entidad o clase que se puede identificar en el sistema oneM2M, la cual puede tener propiedades como, por ejemplo, la Propiedad de Objeto “Tiene Propiedad de Cosa” (Object Property: hasThingProperty) (oneM2M, 2019, pág. 17) (ETSI, 2021, pág. 18).

Una Cosa puede tener relaciones con otras Cosas, verbi gratia, mediante la Propiedad de Objeto “Tiene Relación de Cosa” (Object Property: hasThingRelation), la cual también puede mencionarse como “Tiene Relación con otra Cosa” indistintamente (ibídem).

El siguiente ejemplo ilustra con más detalle lo anterior: una habitación o alcoba que se modele en oneM2M sería una Cosa (Class:Thing); dicha Cosa podría tener una temperatura ambiente, la cual funge como una Propiedad de Cosa (Class:ThingProperty). La habitación y la temperatura se relacionan a través de la Propiedad de Objeto “Tiene Propiedad de Cosa” (Object Property: hasThingProperty). Igualmente, la habitación se podría asociar con otra alcoba mediante la Propiedad de Objeto “Tiene Relación de Cosa” (Object Property: hasThingRelation), entendiéndose dicho vínculo como "está adyacente a" (isAdjacentTo) (ibíd.).

En general, se supone que una Cosa no puede comunicarse electrónicamente con su entorno; no obstante, la subclase de la clase Cosa (Class:Thing) que si le es factible interactuar de manera electrónica con dicho ámbito se denomina "Dispositivo" (Class:Device) (ib.).

En la figura 66 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Cosa (oneM2M:Thing):

Class: Thing

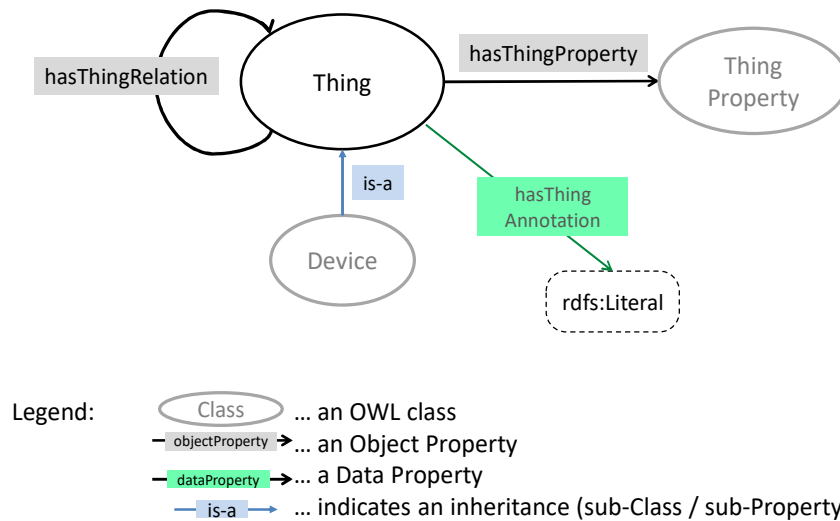


Figura 66. Clase Cosa o “Thing” de oneM2M (oneM2M, 2019, pág. 17) (ETSI, 2021, p. 18)

La clase Cosa o “Thing”, es superclase de la clase Dispositivo o “Device”, mas no es subclase de ninguna otra clase. No obstante, esta clase es una subclase anónima de la Propiedad de Objeto “hasThingProperty”, al relacionarse de manera unitaria con la clase “Propiedad de Cosa” o “ThingProperty”. Lo anterior se conoce como una restricción universal, la cual consiste en que una clase Cosa (Class: Thing) solo puede tener una y solo una relación o Propiedad de Objeto (Object Property: hasThingProperty) con otra clase –u otras clases– Propiedad de Cosa (Class: ThingProperty). (oneM2M, 2019, pág. 18) (ETSI, 2021, pág. 19).

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Cosa o “oneM2M:Thing” se detallan en la tabla 22 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:hasThingProperty	Vínculo consistente en la exhibición de una propiedad o cualidad de una Cosa o entidad que posee conectividad. Esta relación existe entre la clase "Thing" como dominio y la clase "ThingProperty" como recorrido; id est, la relación es inyectiva.
	oneM2M:hasThingRelation	Enlace referente a la relación o nexo que una Cosa le es posible tener con otra similar o diferente. La relación está definida entre la clase "Thing" como dominio y la clase "Thing" como recorrido; i.e. la relación es biyectiva.
Atributo (DataProperty)	oneM2M:hasThingAnnotation	Propiedad de datos en la que se establece una anotación de Cosa con un tipo de dato literal. Dicha característica posee como dominio a la clase "Thing" y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema).

Tabla 22. Propiedades de la clase Cosa o "Thing" de oneM2M (oneM2M, 2019, pág. 17) (ETSI, 2021, pág. 18)

- Propiedad de Cosa (ThingProperty):

Una Propiedad de Cosa (Class:ThingProperty) es una clase o entidad que denota una cualidad, característica o peculiaridad de un elemento específico, artículo o ejemplar (oneM2M, 2019, pág. 18) (ETSI, 2021, pág. 19). Entre sus mayores relevancias figura el que una Cosa (Thing) puede describirse con los valores de las Propiedades de la Cosa; aunque la clase Cosa, generalmente no puede influir en ese valor o ser influenciado por él (ibídem).

De igual manera, el valor de la Propiedad de Cosa (Class:ThingProperty) perteneciente a una Cosa (Class:Thing), puede ser establecido por una entidad que hace parte del sistema oneM2M como un dispositivo, un computador e incluso, un ser humano y contar además, con la posibilidad de leerlo mediante recuperación o actualización (ibíd.).

El siguiente ejemplo ilustra lo ya mencionado: la temperatura interior de una habitación podría ser un valor de Propiedad de Cosa referida a la Cosa "habitación". Igualmente, el nombre de un fabricante podría ser una Propiedad de Cosa (Class:ThingProperty) perteneciente a la Cosa (Class:Thing) "automóvil" (ib.).

Otro de los rasgos importantes consiste en que una Propiedad de Cosa (Class:ThingProperty) atinente a una Cosa (Class:Thing), puede describir una cierta faceta o Aspecto (Class:Aspect), como también tener datos tipificados y específicos conocidos como metadatos

(ClassMetaData); v.g. la temperatura interior de una habitación, describiría el aspecto "Temperatura" que podría medirse con un sensor ad hoc (ibíd.) y en simultánea, determinarse mediante el metadato "escala_celsius".

En la figura 67 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Propiedad de Cosa (oneM2M:ThingProperty):

Class: ThingProperty

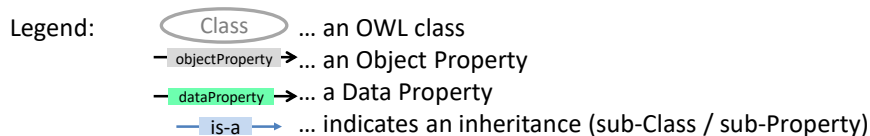
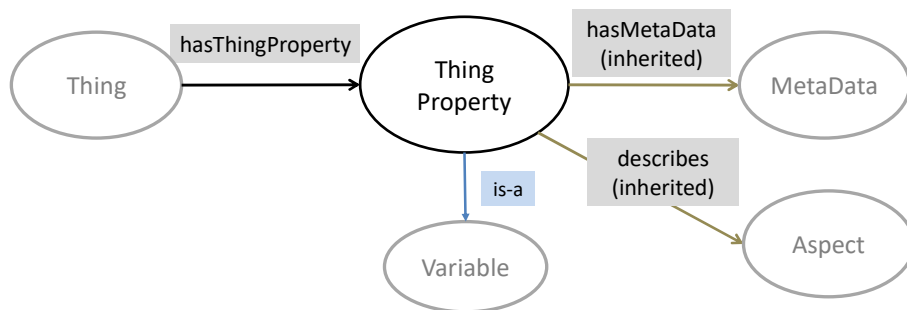


Figura 67. Clase Propiedad de Cosa o "ThingProperty" de oneM2M (oneM2M, 2019, pág. 18) (ETSI, 2021, pág. 19)

La clase Propiedad de Cosa o "ThingProperty", es subclase de la clase "Variable" (oneM2M:Variable); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (oneM2M, 2019, pág. 19) (ETSI, 2021, pág. 20), ya sea de tipo universal, existencial o de cardinalidad.

Las relaciones y atributos ("ObjectProperties" y "DataProperties") que caracterizan a una clase Propiedad de Cosa o "oneM2M:ThingProperty" se detallan en la tabla 23 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una cosa. Esta relación existe

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

(ObjectProperty)		entre la clase “ThingProperty” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva. Esta propiedad de objeto es heredada de la clase “Variable”, dado que la clase “ThingProperty” es una subclase de esta última (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una cosa. La relación está definida entre la clase “ThingProperty” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. La presente propiedad de objeto es heredada de la clase “Variable”, en razón a que la clase “ThingProperty” es una subclase de la anterior (ibíd.).
	oneM2M:hasThingProperty	Relación consistente en la exhibición de la propiedad de una Cosa, la cual posee conectividad. Esta relación existe entre la clase “Thing” que funge como dominio y la clase “ThingProperty” que se desempeña como recorrido; id est, la relación es inyectiva (ibídem.).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de una descripción semántica referida a la Propiedad de una Cosa con un tipo de dato literal, v.g., el nombre del fabricante de una cosa que posee conectividad. Dicha característica posee como dominio a la clase “ThingProperty” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “oneM2MTargetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).
	oneM2M:oneM2MMethod	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa, mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”,

		<p>“update” o “retrieve”-. Dicha característica posee como dominio a la clase “ThingProperty” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).</p>
	<p>oneM2M:oneM2MTargetURI</p>	<p>Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “ThingProperty” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).</p>

Tabla 23. Propiedades de la clase Propiedad de Cosa o “ThingProperty” de oneM2M (oneM2M, 2019, pág. 18) (ETSI, 2021, pág. 19)

- Aspecto (Aspect):

Un Aspecto (Class:Aspect) describe la faceta o matiz del mundo real con el que se relaciona una Función. El Aspecto también se utiliza para describir una cualidad o algún tipo de variable tales como: una Entrada de Operación (Class:OperationInput) o una Salida de Operación (Class:OperationOutput). Una faceta o “Aspecto” podría ser tanto una entidad física o abstracta, como también un fenómeno o una particularidad (oneM2M, 2019, pág. 19) (ETSI, 2021, pág. 20).

En la figura 68 se observan solamente las relaciones (ObjectProperties) de la clase “Aspecto” (oneM2M:Aspect), dado que la misma carece de atributos (DataProperties):

Class: Aspect

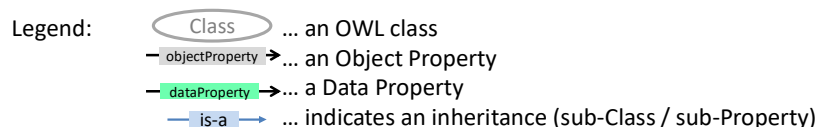
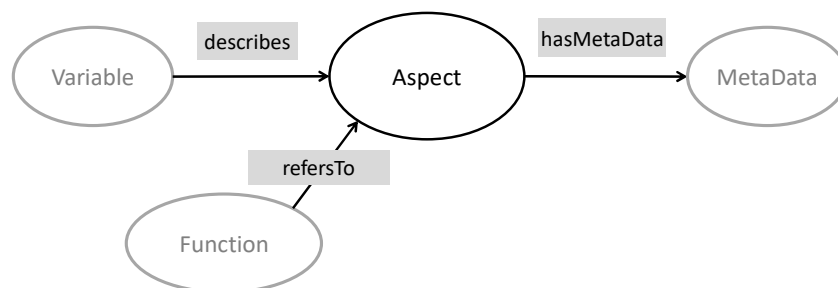


Figura 68. Clase Aspecto o “Aspect” de oneM2M (oneM2M, 2019, pág. 19) (ETSI, 2021, pág. 20)

La clase “Aspect” se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase, como tampoco ser subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no cuenta con restricciones de cardinalidad, de tipo existencial o universal (óp. cit.) (ob. cit.). Así mismo, carece de Propiedades de Datos o “DataProperties”, conocidos también como atributos.

Las relaciones u “ObjectProperties” que caracterizan a una clase “Aspecto” (oneM2M:Aspect) se detallan en la tabla 24 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una cosa. Esta relación existe entre la clase “Variable” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva (ibidem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de una variable. La relación está definida entre la clase “Aspect” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva (ibíd.).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	oneM2M:refersTo	Relación consistente en la referencia a una faceta o matiz atinente a una capacidad propia de actuar (por parte de un dispositivo). Este vínculo existe entre la clase "Function" que funge como dominio y la clase "Aspect" que se desempeña como recorrido; id est, la relación es inyectiva (ib.).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 24. Propiedades de la clase Aspecto o "Aspect" de oneM2M (oneM2M, 2019, pág. 19) (ETSI, 2021, pág. 20)

- Metadatos (MetaData):

Un Metadato (Class:MetaData) contiene datos –tales como unidades, rangos de precisión, etc.– acerca de los valores de una Cosa (Class:Thing) o de un Aspecto (Class:Aspect); dicho de otro modo, es registrar datos acerca de los datos (oneM2M, 2019, pág. 20) (ETSI, 2021, pág. 21), dado que en la informática, un dato es la expresión de la característica de una entidad, fenómeno o entorno.

A continuación un ejemplo de lo que significa dicha clase "MetaData": el Aspecto "temperatura" de un espacio interior o habitación (Cosa), podría tener como Metadatos una "Escala_Celsius" individual, la cual especificaría que dicha magnitud física se ha de comprender como "Grados Celsius" (óp. cit.) (ob. cit.).

En la figura 69 se observan solamente las relaciones (ObjectProperties) de la clase "Metadatos" (oneM2M:MetaData), dado que la misma carece de atributos (DataProperties):

Class: MetaData

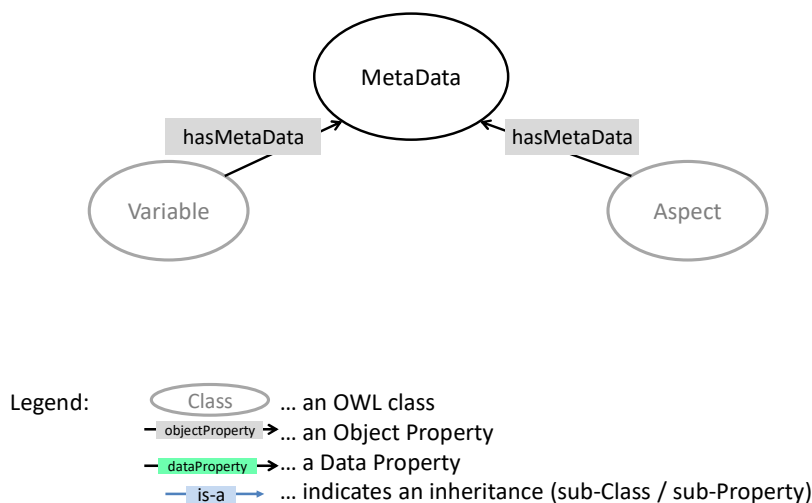


Figura 69. Clase Metadatos o “MetaData” de oneM2M (oneM2M, 2019, pág. 20) (ETSI, 2021, p. 21)

La clase “Metadatos” (oneM2M:MetaData) se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase, como tampoco ser subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no cuenta con restricciones de cardinalidad, de tipo existencial o universal (óp. cit.) (ob. cit.). Así mismo, carece de Propiedades de Datos o “DataProperties”, conocidos también como atributos.

Las relaciones u “ObjectProperties” que caracterizan a una clase “Metadatos” (oneM2M:MetaData) se detallan en la tabla 25 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de una variable. La relación está definida entre las clases “Variable” y “Aspect” que se desempeñan como dominio y la clase “MetaData” que obra como recorrido de las mismas; i.e. dicha relación con ambas clases es inyectiva (ibíd.).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 25. Propiedades de la clase Metadatos o “MetaData” de oneM2M (oneM2M, 2019, pág. 20) (ETSI, 2021, pág. 21)

- Dispositivo (Device):

La clase “Dispositivo” (Class:Device) es un artefacto o Cosa (específicamente, una subclase de Class:Thing) que está diseñado para realizar una tarea en particular, a través de las Funciones (Class:Function) que realiza dicha máquina (oneM2M, 2019, pág. 21) (ETSI, 2021, pág. 22).

Dicho Dispositivo o artilugio en el sistema oneM2M –por lo general, capaz de tratar y/o transferir datos–, presenta las siguientes características (óp. cit.) (ob. cit.):

- Puede interactuar electrónicamente con su entorno a través de una red informática.
- Contiene una lógica formal y es productor y/o consumidor de datos, los cuales se intercambian a través de sus Servicios (Clase:Service) con otras entidades (Dispositivos o Cosas) en la red.
- Interactúa a través de los Puntos de Datos y/o las Operaciones de sus Servicios. Dicho de otra manera, todo Dispositivo interactúa mediante los "DataPoints" (Class:InputDataPoint, Class:OutputDataPoint), como también por medio de las "Operations" (Class:Operation) que pertenecen a sus "Services" (Class:Service).
- En el contexto de oneM2M, siempre se asume que un Dispositivo está facultado para comunicarse electrónicamente a través de una red oneM2M u otra no-oneM2M, con la que se encuentre interfaccionando (interworked).
- Para llevar a cabo su tarea, el Dispositivo realiza una o más Funciones mediante su Propiedad de Objeto "tiene una función" (Object Property:hasFunction, relacionada con Class:Function); siendo dichas Funciones expuestas en la red como Servicios ofrecidos por dicho Dispositivo.
- Puede estar compuesto por varios sub-Dispositivos a través de su Propiedad de Objeto "consta De" (Object Property:consistOf, relacionada con Class:Device); aunque igualmente, es factible constar de un solo Dispositivo.
- Todo Dispositivo y subDispositivo debe ser direccionable individualmente en la red; i.e., ha de contar con una URI.
- A continuación un ejemplo diferencial: un interruptor de luz o "lightswitch", sería un dispositivo; un refrigerador/congelador combinado (i.e. una nevera o frigorífico) sería un dispositivo que consta de un par de subdispositivos: un refrigerador y un congelador.

En la figura 70 se observan solamente las relaciones (ObjectProperties) de la clase “Dispositivo” (oneM2M:Device), dado que la misma carece de atributos (i.e., DataProperties):

Class: Device

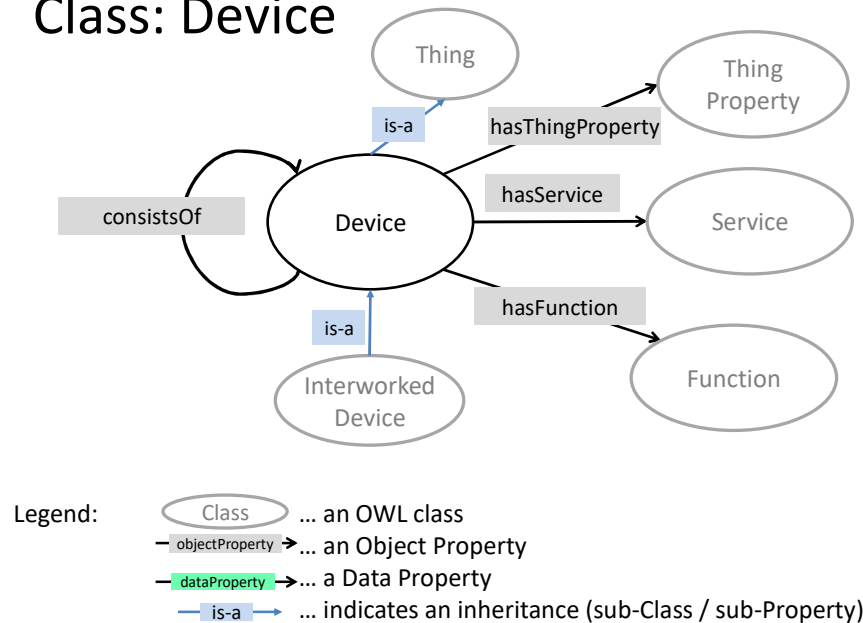


Figura 70. Clase Dispositivo o “Device” de oneM2M (oneM2M, 2019, pág. 21) (ETSI, 2021, p. 22)

La clase “Dispositivo” (oneM2M:Device) es en simultánea, una superclase de la subclase “Dispositivo Interfuncionando” (oneM2M:InterworkedDevice), como también es una subclase de la superclase “Cosa” (oneM2M:Thing). No obstante, esta clase es una subclase anónima de la Propiedad de Objeto “hasService”, al relacionarse de manera unitaria con la clase Servicio o “Service”. Lo anterior se conoce como una restricción universal, la cual consiste en que una clase Dispositivo (Class:Device), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasService) con otra clase –u otras clases– Servicio (Class:Service) (oneM2M, 2019, pág. 22) (ETSI, 2021, pág. 23).

Es de tener en cuenta que esta clase igualmente es una subclase anónima de la Propiedad de Objeto “hasFunction”, al relacionarse de manera indefectible con la clase Función o “Function”, comprendiéndose ello como una condición sine qua non (i.e., todo Dispositivo debe tener al menos una Función). Lo anterior se conoce como una restricción existencial, la cual consiste en que una clase Dispositivo (Class:Device), dispone como mínimo de una relación o Propiedad de Objeto (Object Property:hasFunction) con otra clase Función (Class:Function) (óp. cit.) (ob. cit.). Por último, esta clase Dispositivo, carece de Propiedades de Datos o “DataProperties”, conocidos también como atributos.

Las relaciones u “ObjectProperties” que caracterizan a una clase “Dispositivo” (oneM2M:Device) se detallan en la tabla 26 (óp. cit.):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:hasThingProperty	Vínculo consistente en la exhibición de una propiedad o cualidad de una Dispositivo o artefacto que posee conectividad. Esta relación existe entre la clase “Device” que funge como dominio y la clase “ThingProperty” que actúa como recorrido; id est, la relación es inyectiva (ibídem) . Esta propiedad de objeto es heredada de la clase Cosa o “Thing”.
	oneM2M:hasService	Enlace referente a la prestación ofrecida por parte de un dispositivo para tratar o transferir datos. La relación está definida entre la clase “Device” que se desempeña como dominio y la clase “Service” que obra como recorrido; Como consecuencia de su restricción universal, dicha relación es inyectiva (ibíd.).
	oneM2M:hasFunction	Vínculo consistente en la facultad de actuar propia de un Dispositivo para ejecutar la labor encomendada, sin la cual dicha máquina perdería propósito. La relación está definida entre la clase “Device” que funge como dominio y la clase “Function” que actúa como recorrido. Dada su restricción existencial, dicha relación es biyectiva (ib.).
	oneM2M:consistsOf	Enlace referente a la relación o nexo que un Dispositivo le es posible tener con otro similar o diferente, en cuanto a que dicha máquina está compuesto por más artefactos (es decir, subDispositivos). La relación está definida –en ambos sentidos– entre la clase “Dispositivo” que se desempeña como dominio y la clase “Dispositivo” que obra como recorrido; i.e. la relación es suprayectiva (ibídem).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 26. Propiedades de la clase Dispositivo o “Device” de oneM2M (oneM2M, 2019, pág. 22) (ETSI, 2021, pág. 23)

- Dispositivo Interfuncionado (InterworkedDevice):

La clase “Dispositivo Interfuncionado” o interconectado (Class:InterworkedDevice) atañe a un artefacto que hace parte de una Red de Área o "AreaNetwork". (oneM2M, 2019, pág. 23) (ETSI, 2021, pág. 24) Dicha máquina –que se asume facultada para tratar y/o transferir datos– se caracteriza por no ser un Dispositivo oneM2M, y solo se puede acceder a ella desde el sistema oneM2M, comunicándose con un dispositivo "proxy" (virtual) o intermediario, que ha sido creado por y para una Entidad Proxy de Interfuncionamiento (Interworking Proxy Entity IPE).

Es de tener en cuenta que, en el universo de discurso de oneM2M, La clase Dispositivo Interconectado (InterworkedDevice) describe a un dispositivo "proxy" (de manera virtual), el cual está representado en el sistema oneM2M como un recurso individual de Entidad de Aplicación <AE> (Aplication Entity), o también mediante un recurso secundario (child resource) de la <AE> de su correspondiente IPE (oneM2M, 2019, pág. 16) (ETSI, 2021, pág. 17).

En la figura 71 se observan solamente las relaciones (ObjectProperties) de la clase “Dispositivo Interfuncionado” (oneM2M:InterworkedDevice), dado que la misma carece de atributos (DataProperties):

Class: InterworkedDevice

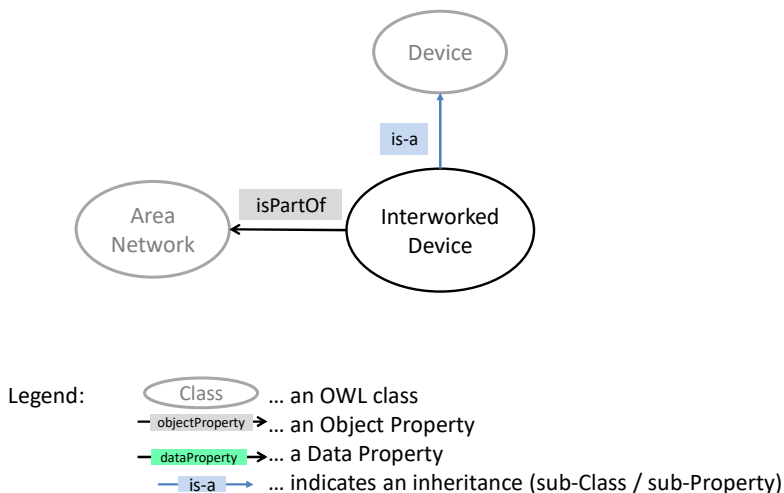


Figura 71. Clase Dispositivo Interfuncionado o “InterworkedDevice” de oneM2M (oneM2M, 2019, pág. 23) (ETSI, 2021, p. 24)

La clase “Dispositivo Interfuncionado” o (oneM2M:InterworkedDevice), es una subclase de la clase “Dispositivo” (oneM2M:Device); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (óp. cit.) (ob. cit.), ya sea de tipo universal, existencial o de cardinalidad.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Las relaciones u “ObjectProperties” que caracterizan a la clase “Dispositivo Interfuncionado” (oneM2M: InterworkedDevice) se detallan en la tabla 27 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:isPartOf	Vínculo consistente en la integración a una red de área informática, por parte de un dispositivo que se encuentra interconectado al mismo (Interworked). La relación está definida entre la clase “InterworkedDevice” que funge como dominio y la clase “AreaNetwork” que obra como recorrido; i.e. dicha relación es inyectiva (ibídem).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 27. Propiedades de la clase Dispositivo Interfuncionado o “InterworkedDevice” de oneM2M (oneM2M, 2019, pág. 23) (ETSI, 2021, pág. 24)

- Red de Área (AreaNetwork):

La clase “Red de Área” (Class:AreaNetwork) es atinente a una red informática de área (independientemente de su alcance), que proporciona el servicio de transferencia de datos entre un dispositivo interconectado (Class:InterworkedDevice) y el sistema oneM2M. Dichas redes puede ser diversas y hacer uso de tecnologías de interconexión heterogéneas, las cuales pueden o no admitir el acceso mediante IP (Internet Protocol) (oneM2M, 2019, pág. 24) (ETSI, 2021, pág. 25).

Es de tener en cuenta que la generalidad de las redes de área (Class:AreaNetwork) –sea cual fuere su topología, cobertura, conectividad, arquitectura, etc.–, se caracterizan principalmente por su tecnología en cuanto a (óp. cit.) (ob. cit.):

- Las propiedades físicas del medio, v.g., el estándar WPAN o de redes inalámbricas de área personal “IEEE_802_15_4_2003_2_4GHz”.
- El protocolo de comunicación inalámbrica, por ejemplo, el protocolo con balizas “ZigBee_1_0”.
- En segunda instancia, se distinguen potencialmente por disponer de perfiles, verbi gratia, la configuración de automatización doméstica “ZigBee_HA”.

En la figura 72 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Red de Área (oneM2M:AreaNetwork):

Class: AreaNetwork

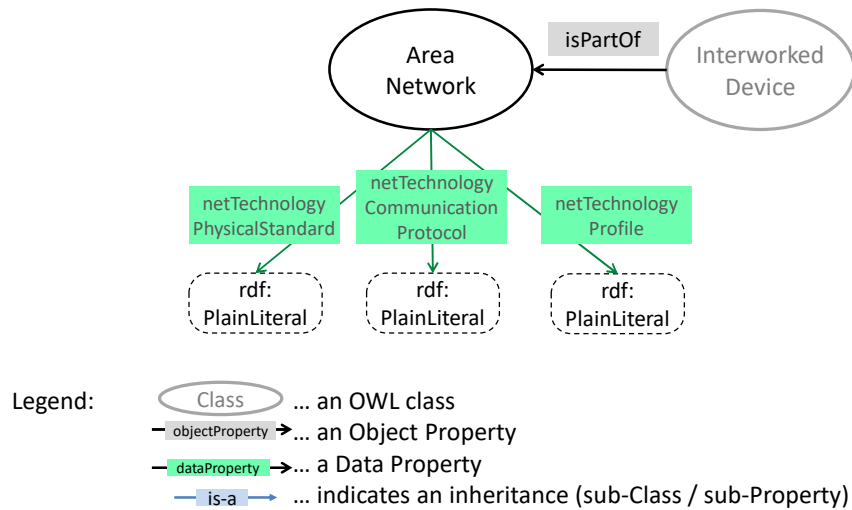


Figura 72. Clase Red de Área o “AreaNetwork” de oneM2M (oneM2M, 2019, pág. 24) (ETSI, 2021, pág. 25)

La clase “Red de Área” (oneM2M:AreaNetwork) se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase, como tampoco ser subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no cuenta con restricciones de cardinalidad, de tipo existencial o universal (óp. cit.) (ob. cit.).

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Red de Área o “oneM2M:AreaNetwork” se detallan en la tabla 28 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:isPartOf	Vínculo consistente en la integración a una red de área informática, por parte de un dispositivo que se encuentra interconectado al mismo (Interworked). La relación está definida entre la clase “InterworkedDevice” que funge como dominio y la clase “AreaNetwork” que obra como recorrido; i.e. dicha relación es inyectiva (ibidem).
Atributo (DataProperty)	oneM2M: netTechnologyPhysicalStandard	Propiedad de datos en la que se identifica las cualidades físicas de la tecnología de una red de área específica, mediante un dato de tipo texto, conocido también como dato literal llano. Dicha característica posee como dominio a la clase “AreaNetwork” y como

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		recorrido al tipo de dato Literal Simple (datatype:rdf:PlainLiteral), definido en el Marco de Descripción de Recursos RDF (Resource Description Framework); p. ej., “IEEE_802_15_4_2003_2_4GHz” (ib.).
	oneM2M: netTechnologyCommunicationProtocol	Propiedad de datos en la que se determina el protocolo de comunicación para una tecnología de red de área, mediante un dato de tipo texto, conocido también como dato literal simple. Dicha característica posee como dominio a la clase “AreaNetwork” y como recorrido al tipo de dato Literal Llano (datatype:rdf:PlainLiteral), definido en el Marco de Descripción de Recursos RDF (Resource Description Framework); v.g., “ZigBee_1_0” (ibíd.).
	oneM2M: netTechnologyProfile	Propiedad de datos en la que se establece la configuración o perfil de una tecnología de una red de área en específico, mediante un dato de tipo texto, conocido también como dato literal llano. Dicha característica posee como dominio a la clase “AreaNetwork” y como recorrido al tipo de dato Literal Simple (datatype:rdf:PlainLiteral), definido en el Marco de Descripción de Recursos RDF (Resource Description Framework); verbi gratia, “ZigBee_HA” (ibídem).

Tabla 28. Propiedades de la clase Red de Área o “AreaNetwork” de oneM2M (oneM2M, 2019, pág. 24) (ETSI, 2021, pág. 25)

- Servicio (Service):

La clase “Servicio” (Class:Service) concierne a una representación electrónica de una Función (Class:Function) en una red informática, que hace que la misma –y siendo perteneciente a un dispositivo– sea detectable, registrable y controlable de forma remota en la dicha red. He por ello que a un Servicio le es factible exponer u ofrecer una o más Funciones y en simultánea, una Función puede ser expuesta o exhibida por uno o más Servicios (oneM2M, 2019, pág. 25) (ETSI, 2021, pág. 26).

Si bien una Función (Class:Function) describe el significado –comprensible para el ser humano– de lo que realiza un Dispositivo (Class:Device), el Servicio (Class:Service) se utiliza para describir el cómo se representa dicha Función en una red computacional y se pueda acceder a la misma por medios electrónicos (óp. cit.) (ob. cit.). Por lo tanto, el Servicio y sus Operaciones (Class:Operation) pertenecientes al Dispositivo, se encuentran supeditados tanto al hardware y software de dicho artefacto, como a la tecnología de la red informática a la cual se interconecta.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Un ejemplo de lo anterior atañe a la Función de encendido o apagado de una luz “turn_light_On_or_Off”, la cual podría ser expuesta en la red por un Servicio de nombre Actualizar el Valor Binario o “Update Binary Value” (óp. cit.) (ob. cit.).

No sobra aclarar que un Servicio (Class:Service) puede estar compuesto por diversos subservicios, independientes y de menor tamaño –conocidos como módulos de servicio–, los cuales son reutilizables per se. Como ejemplo, un dispositivo “Atenuador” (Dimmer) podría contener un primer componente o módulo “Activador binario” (binaryActuator) para realizar la tarea de encendido/apagado y además, un segundo módulo “Establecer activador en entero 0-255” (setInteger0-255Actuator) que permita fijar el nivel de atenuación (oneM2M, 2019, pág. 26) (ETSI, 2021, pág. 27).

En la figura 73 se observan solamente las relaciones (ObjectProperties) de la clase “Servicio” (oneM2M:Service), dado que la misma carece de atributos (i.e., DataProperties):

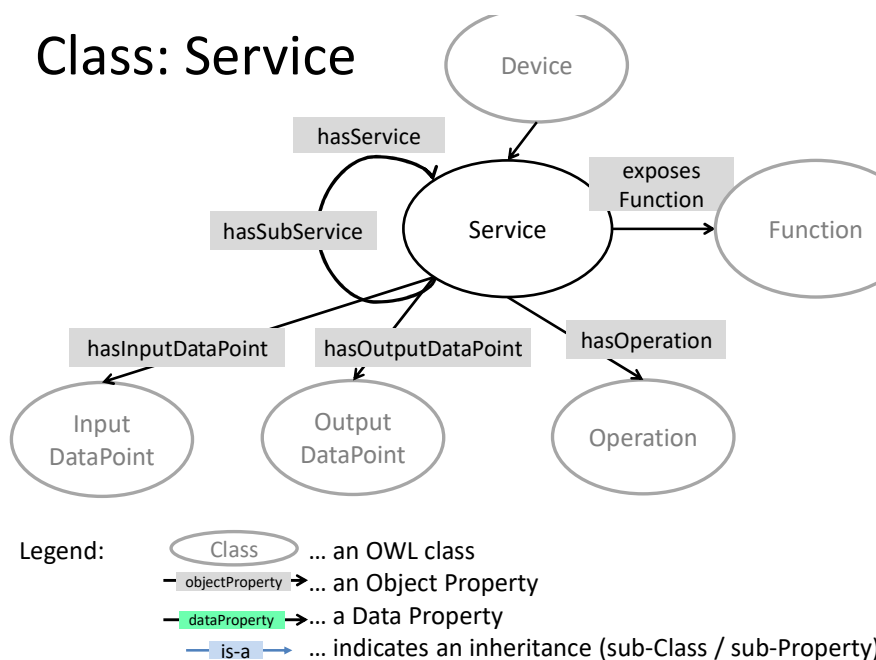


Figura 73. Clase Servicio o “Service” de oneM2M (oneM2M, 2019, pág. 25) (ETSI, 2021, p. 26)

Es de resaltar que la clase “Servicio” (oneM2M:Service) se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase. Así mismo dicha clase no cuenta con Propiedades de Datos o “DataProperties”, conocidos también como atributos (oneM2M, 2019, pág. 26) (ETSI, 2021, pág. 27).

No obstante, la clase “Servicio” (Class:Service) funge como una subclase anónima de diversas Propiedades de Objeto (ObjectProperties), las cuales se describen en la siguiente lista de restricciones (oneM2M, 2019, pág. 26) (ETSI, 2021, pág. 27):

- De la Propiedad de Objeto “tiene una Operación” (hasOperation), cuando la clase “Servicio” se relaciona de manera unitaria con la clase “Operación”, conociéndose ello como una restricción universal; la cual consiste en que una clase Servicio

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

(Class:Service), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasOperation) con otra clase –u otras clases– Operación (Class:Operation).

- De la Propiedad de Objeto “tiene un Punto de Datos de Entrada” (hasInputDataPoint), cuando la clase “Servicio” se relaciona de manera unitaria con la clase “Punto de Datos de Entrada”, conociéndose ello como una restricción universal; la cual consiste en que una clase Servicio (Class:Service), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property: hasInputDataPoint) con otra clase –o clases– Punto de Datos de Entrada (Class: InputDataPoint).
- De la Propiedad de Objeto “Expone la Función” (exposesFunction), cuando la clase “Servicio” se relaciona de manera indefectible con la clase “Operación”, comprendiéndose ello como una condición sine qua non (i.e., todo Servicio debe tener al menos una Función). Lo anterior se conoce como una restricción existencial, la cual consiste en que una clase Servicio (Class:Service), dispone como mínimo de una relación o Propiedad de Objeto (Object Property: exposesFunction) con otra clase –u otras clases– Función (Class:Function).
- De la Propiedad de Objeto “tiene un Punto de Datos de Salida” (hasOutputDataPoint), cuando la clase “Servicio” se relaciona de manera unitaria con la clase “Punto de Datos de Salida”, conociéndose ello como una restricción universal; la cual consiste en que una clase Servicio (Class:Service), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property: hasOutputDataPoint) con otra(s) clase(s) Punto de Datos de Salida (Class: OutputDataPoint).
- De la Propiedades de Objeto “hasOperation”, “hasInputDataPoint”, “hasOutputDataPoint” y “hasSubService”, cuando la clase “Servicio” se relaciona de manera obligatoria y particular con las clases “Operation”, “InputDataPoint”, “OutputDataPoint” y “SubService”, conociéndose ello como una restricción de cardinalidad. Dicha condición consiste en que la clase “Servicio” (Class:Service), solo puede tener a lo sumo, una relación o Propiedad de Objeto con otras clases (específicamente, con las ya referidas).

Las relaciones u “ObjectProperties” que caracterizan a una clase “Dispositivo” (oneM2M:Service) se detallan en la tabla 29 (óp. cit.):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:exposesFunction	Vínculo consistente en un Servicio que expone una Función a la red informática y la hace detectable, registrable y controlable de forma remota en dicha red. Esta relación existe entre la clase "Service" que funge como dominio y la clase "Function" que actúa como recorrido. Dada su restricción existencial, dicha relación es biyectiva (ibídem).
	oneM2M:hasService	Enlace referente a la prestación ofrecida por parte de un dispositivo para tratar o transferir datos. La relación está definida entre la clase "Device" que se desempeña como dominio y la clase "Service" que obra como recorrido; Como consecuencia de su restricción universal, dicha relación es inyectiva (ibíd.).
	oneM2M:hasOperation	Vínculo consistente en un Servicio que se comunica mediante Operaciones a través de la red, para transmitir datos hacia/desde otros dispositivos. La relación está definida entre la clase "Service" que funge como dominio y la clase "Operation" que actúa como recorrido. Dada su restricción universal, dicha relación es inyectiva (ib.).
	oneM2M:hasSubService	Enlace referente a la relación o nexo que un Servicio le es posible tener, con otro similar o diferente, en cuanto a que dicho Servicio está compuesto por más Servicios (es decir, subServicios). La relación está definida –en ambos sentidos– entre la clase "Dispositivo" que se desempeña como dominio y la clase "Dispositivo" que obra como recorrido; i.e. la relación es suprayectiva (ibídem).
	oneM2M:hasInputDataPoint	Vínculo concerniente a un Servicio que dispone de una lectura o recuperación de datos de entrada a un dispositivo, en un instante específico o periódico (i.e., recepción o "receipt"). La relación está definida entre la clase "Service" que funge como dominio y la clase "InputDataPoint" que actúa como recorrido. Dada su restricción universal, dicha relación es inyectiva (ibíd.).
	oneM2M:hasOutputDataPoint	Enlace que consiste en un Servicio que dispone de una escritura o suministro de datos de salida de un dispositivo, en un instante específico o periódico (i.e., entrega o

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		“delivery”). La relación está definida entre la clase “Service” que se desempeña como dominio y la clase “OutputDataPoint” que obra como recorrido. Ya que su restricción es universal, dicha relación es inyectiva (ib.).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 29. Propiedades de la clase Servicio o “Service” de oneM2M (oneM2M, 2019, pág. 26) (ETSI, 2021, pág. 27)

- Función (Function):

Una clase “Función” (Class:Function) representa la capacidad propia de actuar por parte de una Dispositivo (Class:Device), la cual es necesaria a fin de realizar la tarea para la que está diseñado dicho artefacto – y el cual puede ejecutar más de una función– (oneM2M, 2019, pág. 27) (ETSI, 2021, pág. 28).

La clase Función o “Function”, presenta (manifiesta o describe) el significado -comprensible para el ser humano- referente a lo que el dispositivo “hace”.Dicho de otro modo, una Función alude al observar o el influir en un determinado Aspecto (Class:Aspect) del universo de discurso, en el cual el artefacto se encuentra inmerso (óp. cit.) (ob. cit.).

A continuación un ejemplo: considerando un Dispositivo como “Interruptor de luz” (Light switch), una Función relacionada con el mismo podría ser “Controlar el Encendido y Apagado” (Controlling_On_Off) o también “Controlar el Brillo” (Controlling_Brightness) (óp. cit.) (ob. cit.). Esta Función se referiría a un Aspecto llamado “Iluminación” (Lighting), el cual estaría influenciado por el Dispositivo en cuestión, aunque igualmente, un usuario humano podría incidir en dicha faceta a través de los Comandos u órdenes (Commands) pertenecientes a la Función y disponibles para dicho usuario (ibídem).

En la figura 74 se observan solamente las relaciones (ObjectProperties) de la clase “Función” (Class:Function), dado que la misma carece de atributos (i.e., DataProperties):

Classes: Function, Controlling-, Measuring-

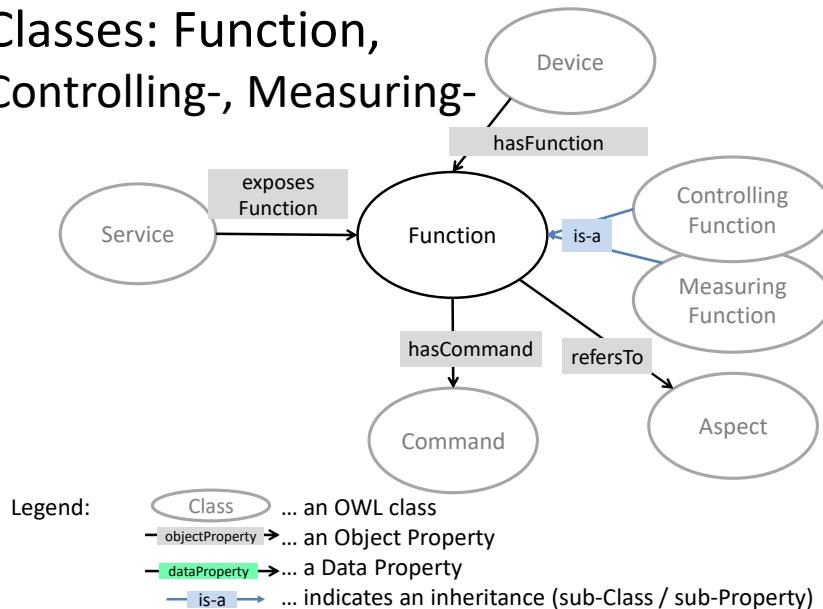


Figura 74. Clase Función o “Function” de oneM2M (oneM2M, 2019, pág. 27) (ETSI, 2021, p. 28)

Es de tener en cuenta que la clase “Función” (Class:Function), es superclase de un par de subclases: la “Función de Control” (Class:ControllingFunction) y la “Función de Medición” (Class:MeasuringFunction); mas dicha clase –Función– no es subclase de ninguna otra clase. Así mismo la clase Función no cuenta con Propiedades de Datos o “DataProperties”, conocidos también como atributos (oneM2M, 2019, pág. 27) (ETSI, 2021, pág. 28).

Las relaciones u “ObjectProperties” que caracterizan a una clase “Función” (oneM2M:Function) se detallan en la tabla 30 (óp. cit.) (ob. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:hasCommand	Enlace referente a las órdenes o comandos que pertenecen a una Función y que le permite a un usuario, incidir en un determinado Aspecto del universo de discurso, en el cual un Dispositivo se halla inmerso. Esta relación existe entre la clase “Función” que se desempeña como dominio y la clase “Command” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:refersTo	Vínculo consistente en la referencia a una faceta o matiz atinente a una capacidad propia de actuar (por parte de un dispositivo). Este vínculo

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		existe entre la clase “Function” que funge como dominio y la clase “Aspect” que se obra como recorrido; i.e., la relación es inyectiva (ib.).
	oneM2M:exposesFunction	Enlace consistente en un Servicio que expone una Función a la red informática y la hace detectable, registrable y controlable de forma remota en dicha red. Esta relación existe entre la clase “Service” que se desempeña como dominio y la clase “Function” que actúa como recorrido; id est, la relación es inyectiva (ibidem).
	oneM2M:hasFunction	Vínculo concerniente a la facultad de actuar propia de un Dispositivo para ejecutar la labor encomendada, sin la cual dicha máquina perdería propósito. La relación está definida entre la clase “Device” que se desempeña como dominio y la clase “Function” que obra como recorrido; i.e. la relación es inyectiva (ib.).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 30. Propiedades de la clase Función o “Function” de oneM2M (oneM2M, 2019, pág. 27) (ETSI, 2021, pág. 28)

- Función de Control (ControllingFunction):

La clase “Función de Control” (Class:ControllingFunction) representa un tipo de Función (Class:Function) que únicamente incide o influye en los Aspectos (Class:Aspect) del mundo real con los que se relaciona dicha Función (oneM2M, 2019, pág. 28) (ETSI, 2021, pág. 29).

No obstante, tal Función de Control (o capacidad de observancia) no recopila datos; cuenta con Comandos y/o Operaciones de sus Servicios relacionados, los cuales reciben Datos de Entrada, tal y como sucedería por ejemplo con un dispositivo activador llamado “Termostato”, cuya función de control consistiría en un “Ajuste de temperatura” (óp. cit.) (ob. cit.).

En la figura 74 se observa la herencia o asociación entre las clases “Función de Control” y “Función”, como también la ausencia de relaciones referentes a las Propiedades de Objeto (ObjectProperties) y las Propiedades de Datos (DataProperties) en la clase (Class:ControllingFunction) (óp. cit.) (ob. cit.).

Se confirma entonces en dicha imagen, que la clase Función de Control” o (oneM2M:ControllingFunction), es una subclase de la clase “Función” (oneM2M: Function); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (óp. cit.) (ob. cit.), ya sea de tipo universal, existencial o de cardinalidad.

- Función de Medición (MeasuringFunction):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

La clase “Función de Medición” (Class:MeasuringFunction) representa un tipo de Función (Class:Function) que únicamente detecta o mide los Aspectos (Class:Aspect) del mundo real con los que se relaciona dicha Función (oneM2M, 2019, pág. 29) (ETSI, 2021, pág. 30).

Es de resaltar que dicha Función de Control (o facultad de supervisión) lleva a cabo recopilación de datos; dispone de Comandos y/o Operaciones de sus Servicios relacionados, los cuales generan Datos de Salida, tal y como sucedería por ejemplo con un dispositivo detector de una magnitud llamado “Sensor de Temperatura”, cuya función de medición consistiría en una “Detección de temperatura” (óp. cit.) (ob. cit.).

En la figura 74 se aprecia la herencia o asociación entre las clases “Función de Medición” y “Función”, como también la ausencia de relaciones referentes a las Propiedades de Objeto (ObjectProperties) y las Propiedades de Datos (DataProperties) en la clase (Class:MeasuringFunction) (óp. cit.) (ob. cit.).

Se confirma entonces en dicha imagen, que la clase “Función de Medición” o (oneM2M:MeasuringFunction), es una subclase de la clase “Función” (oneM2M: Function); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (óp. cit.) (ob. cit.), ya sea de tipo universal, existencial o de cardinalidad.

- Operación (Operation):

Una clase “Operación” (Class:Operation) es el recurso con el que un Servicio (Class:Service) se comunica a modo de procedimiento, a través de una red informática; i.e., transmitiendo datos hacia/desde otros Dispositivos (Class:Device). Dicho de otra manera, una Operación es la exposición u ofrecimiento –interpretable por un Dispositivo– de una orden o Comando – comprensible por un usuario humano– en una red. (oneM2M, 2019, pág. 30) (ETSI, 2021, pág. 31).

Una Operación es transitoria o momentánea, id est, esta puede ser invocada, producir una salida eventualmente y por último, finalizar la transmisión. Esto es factible dado que una Operación es una representación de un Comando (Class:Command) en la red, para los siguientes casos (óp. cit.) (ob. cit.):

- Una “Entrada de Operación” (Class:OperationInput) o recepción de datos para activar el Dispositivo; lo cual significa que son datos consumidos por la máquina.
- Una “Salida de Operación” (Class:OperationOutput) o generación de datos para entregar por parte del Dispositivo, lo que se interpreta como datos producidos por el artefacto.

A diferencia de las clases ya descritas, la clase “Operación” cuenta con un método que describe cómo se invoca la operación a través de la red, al igual que con una correlación de los datos de salida de dicha clase, con los datos de entrada que se utilizaron en la invocación de la misma.

Dicha invocación la puede realizar un Dispositivo que no sea oneM2M, como también un artefacto que pertenezca al sistema oneM2M; v.g., una Entidad de Aplicación <AE> (óp. cit.) (ob. cit.).

En la figura 75 se observan tanto las relaciones (ObjectProperties), como los atributos (DataProperties) de la clase “Operación” (Class:Operation) (oneM2M, 2019, pág. 29) (ETSI, 2021, pág. 30):

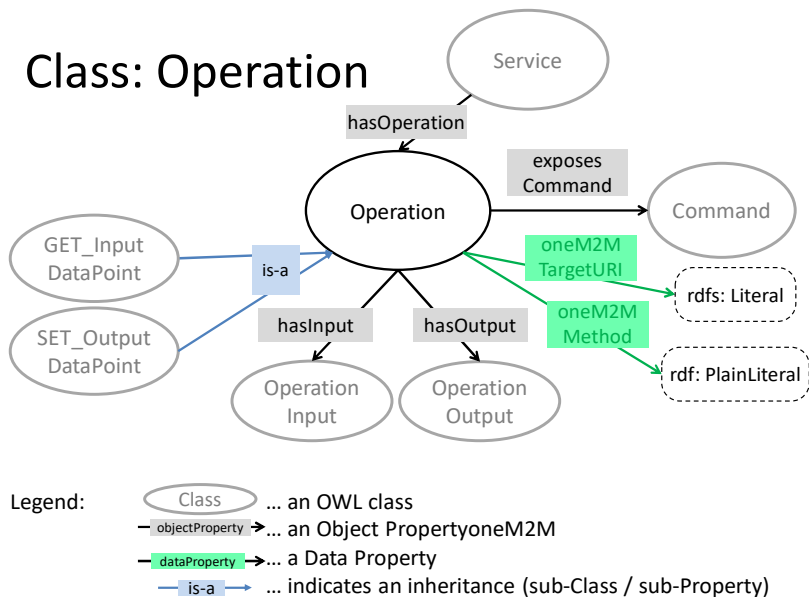


Figura 75. Clase Operación u “Operation” de oneM2M (oneM2M, 2019, pág. 29) (ETSI, 2021, p. 30)

Es de tener en cuenta que la clase “Operación” (Class:Operation), es superclase de un par de subclases a saber: la clase “Obtener Punto de Datos de Entrada” (Class:GET_InputDataPoint) y la clase “Establecer Punto de Datos de Salida” (Class:SET_OutputDataPoint); mas dicha clase –i.e., Operación– no es subclase de ninguna otra clase (oneM2M, 2019, pág. 30) (ETSI, 2021, pág. 31).

No obstante, la clase “Operacion” (Class:Operation) funge como una subclase anónima de diversas Propiedades de Objeto (ObjectProperties), las cuales se describen en la siguiente lista de restricciones (óp. cit.) (ob. cit.):

- De la Propiedad de Objeto “expone el Comando” (exposesCommand), cuando la clase “Operación” se relaciona de manera unitaria con la clase “Comando”, conociéndose ello como una restricción universal; la cual consiste en que una clase Operación (Class:Operation), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:exposesCommand) con otra clase –u otras clases– Comando (Class:Command).
- De la Propiedad de Objeto “tiene la Entrada” (hasInput), cuando la clase “Operación” se relaciona de manera unitaria con la clase “Entrada de Operación”, conociéndose ello como una restricción universal; la cual consiste en que una clase

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Operación (Class:Operation), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasInput) con otra(s) clase(s) “Entrada de Operación” (Class: OperationInput).

- De la Propiedad de Objeto “tiene la Salida” (hasOutput), cuando la clase “Operación” se relaciona de manera unitaria con la clase “Salida de Operación”, conociéndose ello como una restricción universal; la cual consiste en que una clase Operación (Class:Operation), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasOutput) con otra clase –u otras clases– “Salida de Operación” (Class: OperationOutput).

- De la Propiedades de Objeto “hasInput” y “hasOutput”, cuando la clase “Operación” se relaciona de manera obligatoria y particular con las clases “OperationInput” y “OperationInput”, conociéndose ello como una restricción de cardinalidad. Dicha condición consiste en que la clase “Operación” (Class:Operation), solo puede tener a lo sumo, una relación o Propiedad de Objeto con otras clases (específicamente, con las ya referidas).

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Operación (Class:Operation), se detallan en la tabla 31 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:exposesCommand	Enlace referente a la exposición u ofrecimiento –interpretable por un Dispositivo– de una orden o Comando – comprensible por un usuario humano– en una red informática. Esta relación existe entre la clase “Operación” que se desempeña como dominio y la clase “Command” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasInput	Vínculo consistente en la Entrada de datos que puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicho artefacto. Este vínculo existe entre la clase “Operation” que funge como dominio y la clase “OperationInput” que se obra como recorrido; i.e., la relación es inyectiva (ib.).
	oneM2M:hasOutput	Enlace referente a la Salida de datos que

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		<p>puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicha máquina. Esta relación existe entre la clase “Operation” que se desempeña como dominio y la clase “OperationOutput” que actúa como recorrido; id est, la relación es inyectiva (ibídem).</p>
	oneM2M:hasOperation	<p>Vínculo consistente en un Servicio que se comunica mediante Operaciones a través de la red, para transmitir datos hacia/desde otros dispositivos. La relación está definida entre la clase “Operation” que funge como dominio y la clase “Service” que actúa como recorrido. Dada su restricción universal, dicha relación es inyectiva (ib.).</p>
Atributo (DataProperty)	oneM2M:oneM2MTargetURI	<p>Propiedad de datos en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Operación, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “Operation” y como recorrido al tipo de dato literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con el atributo “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).</p>
	oneM2M:oneM2MMethod	<p>Propiedad de datos en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. El atributo en mención pertenece a la clase Operación, mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase “Operation” y como recorrido al tipo de dato literal llano (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).</p>

Tabla 31. Propiedades de la clase Operación u “Operation” de oneM2M (oneM2M, 2019, pág. 30) (ETSI, 2021, pág. 31)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

La Propiedad de Datos (DataProperty) correspondiente a “Método oneM2M” (oneM2MMethod), amerita una descripción aún más detallada, la cual se muestra a continuación:

El “Método oneM2M” (Data Property:oneM2MMethod) atañe una Propiedad de Datos que contiene un método CRUD del sistema oneM2M, a través del cual una instancia oneM2M del valor de la Variable (Class:Variable), puede ser manipulada por la entidad comunicante de las siguientes maneras:

- Al contener o abarcar la cadena "RETRIEVE" para recuperar la variable cuando el recurso oneM2M es de tipo <container> o <flexContainer>. Esto se aplica a las subclases: "OperationOutput", "OutputDataPoint" y "ThingProperty".
- Al envolver o contener la cadena "CREATE" para actualizar la variable cuando el recurso oneM2M es de tipo <container>. Esto se aplica a las subclases: "OperationInput", "InputDataPoint" y "ThingProperty".
- Al abarcar o envolver la cadena "UPDATE" para actualizar la variable cuando el recurso oneM2M es de tipo <flexContainer>. Esto se aplica a las subclases: "OperationInput", "InputDataPoint" y "ThingProperty".

Por último, esta Propiedad de Datos se encuentra definida entre las clases “Variable” y “Operation” que se desempeñan como dominio, y el tipo de dato literal simple (datatype:rdf:PlainLiteral) que obra como recorrido de las mismas, siendo aquel definido en el Marco de Descripción de Recursos RDF (Resource Description Framework). Por lo tanto, la relación de ambas clases con el tipo de dato es inyectiva (ibíd.).

- Obtener Punto de Datos de Entrada (GET_InputDataPoint):

La clase “Obtener Punto de Datos de Entrada” (Class: GET_InputDataPoint), representa un tipo de Operación (Class:Operation) que puede ofrecer un Dispositivo (Class:Device) para activar el mismo y así, **recuperar** los datos de un Punto de Datos de Entrada (Class:InputDataPoint) (oneM2M, 2019, pág. 31) (ETSI, 2021, pág. 32).

Un ejemplo de la obtención de un punto o momento de datos de entrada (GET_InputDataPoint), atañe a un Dispositivo que ha sido programado (schedule) para la adquisición, lectura o recepción de dichos datos de ingreso, requiriéndose activar dicho evento por fuera del agendamiento normal respectivo (óp. cit.) (ob. cit.).

En la figura 75, se observa la herencia o asociación entre las clases “Obtener Punto de Datos de Entrada” y “Operación”, como también la ausencia de relaciones referentes a las Propiedades de

Objeto (ObjectProperties) y las Propiedades de Datos (DataProperties) en la primera de las mismas (i.e., Class:GET_InputDataPoint) (óp. cit.) (ob. cit.).

Se confirma entonces en dicha imagen, que la clase “Obtener Punto de Datos de Entrada” o (oneM2M: GET_InputDataPoint), es una subclase de la clase “Operación” (oneM2M:Operation); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (óp. cit.) (ob. cit.), ya sea de tipo universal, existencial o de cardinalidad.

- Establecer Punto de Datos de Salida (SET_OutputDataPoint):

La clase “Establecer Punto de Datos de Entrada” (Class: SET_OutputDataPoint), representa un tipo de Operación (Class:Operation) que puede ofrecer un Dispositivo (Class:Device) para activar el mismo y así, **actualizar** los datos de un Punto de Datos de Salida (Class:OutputDataPoint) (oneM2M, 2019, pág. 31) (ETSI, 2021, pág. 32).

Un ejemplo del establecimiento de un punto o momento de datos de salida (SET_OutputDataPoint), atañe a un Dispositivo que ha sido programado (schedule) para la generación, producción o entrega de dichos datos de envío, requiriéndose activar dicho evento por fuera del agendamiento normal respectivo (óp. cit.) (ob. cit.).

En la figura 75, se observa la herencia o asociación entre las clases “Establecer Punto de Datos de Salida” y “Operación”, como también la ausencia de relaciones referentes a las Propiedades de Objeto (ObjectProperties) y las Propiedades de Datos (DataProperties) en la primera de las mismas (i.e., Class:SET_OutputDataPoint) (óp. cit.) (ob. cit.).

Se corrobora a partir de dicha figura, que la clase “Establecer Punto de Datos de Salida” o (oneM2M: SET_OutputDataPoint), es una subclase de la clase “Operación” (oneM2M:Operation); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (óp. cit.) (ob. cit.), ya sea de tipo universal, existencial o de cardinalidad.

- Comando (Command):

Una clase “Comando”, instrucción u orden (Class:Command), representa una acción que se puede realizar para respaldar a una Función (Class:Function). De manera adicional, un Comando es el nombre –comprensible para un usuario humano–, de la acción o tarea mencionada, que es **invocada** en un Dispositivo (Class:Device) o que es **reportada** por el mismo (oneM2M, 2019, pág. 32) (ETSI, 2021, pág. 33)

Así mismo, un Comando u orden es expuesto en la red informática por parte de una “Operación” (Class:Operation), como también una “Entrada de Operación” (Class:OperationInput) o una “Salida de Operación” (Class:OperationOutput) que estén relacionadas con dicha Operación, les es factible parametrizar tal instrucción o Comando (óp. cit.) (ob. cit.).

Un ejemplo de lo anterior sería el siguiente: una Función de nombre “Función de atenuación” perteneciente a un Dispositivo “Interruptor de luz”, que controla remotamente la iluminación en un

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

espacio, podría tener un Comando “Establecer Intensidad de Iluminación” (setLightIntensity), con un parámetro que tiene valores en el intervalo 0 - 100% (óp. cit.) (ob. cit.).

Igualmente, todo “Punto de Datos de Entrada” y “Punto de Datos de Salida” (InputDataPoint y OutputDataPoint), exponen Comandos en la red. Además, cuando un dispositivo se comunica bajo el estilo de arquitectura de Estado de Transferencia Representacional pleno RESTful (REpresentational State Transfer full), entonces la actualización en un Punto de Datos de Entrada desencadena una acción en el Dispositivo (i.e., un activador), una vez que dicho Dispositivo ha leído o recibido los datos del “InputDataPoint” (óp. cit.) (ob. cit.).

De manera similar, cuando un Dispositivo establece los datos de un Punto de Datos de Salida u “OutputDataPoint”, genera una acción en dicho Dispositivo (i.e., un sensor), para proporcionar información acerca del estado del artefacto en el universo de discurso que el mismo se encuentra inmerso (óp. cit.) (ob. cit.).

Es de tener en cuenta que en los sistemas RESTful, los nombres asignados a un Punto de Datos tanto de Entrada como de Salida –“InputDataPoint” y “OutputDataPoint” respectivamente–, generalmente se eligen de tal manera que expresen o describan el Comando; i.e., con un significado comprensible para el usuario humano, v.g., un “InputDataPoint” binario de un Dispositivo “Interruptor de luz” podría tener un nombre "Set_Light_Status"). Por ende, la actualización de dicho Punto de Datos se puede interpretar como la ejecución de un Comando (óp. cit.) (ob. cit.).

En la figura 76 se observan solamente las relaciones (ObjectProperties) de la clase “Comando” (oneM2M:Command), dado que la misma carece de atributos (DataProperties):

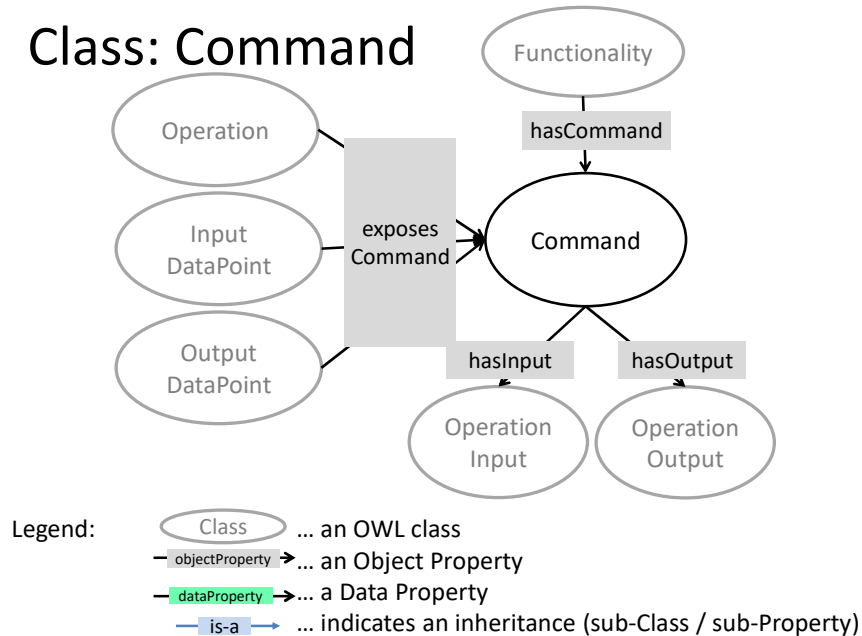


Figura 76. Clase Comando o “Command” de oneM2M (oneM2M, 2019, pág. 32) (ETSI, 2021, p. 33)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Es de resaltar que la clase “Comando” (oneM2M:Command) se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase. Así mismo dicha clase no cuenta con Propiedades de Datos o “DataProperties”, conocidos también como atributos (oneM2M, 2019, pág. 33) (ETSI, 2021, pág. 34).

No obstante, la clase “Comando” (Class:Command) funge como una subclase anónima de diversas Propiedades de Objeto (ObjectProperties), las cuales se describen en la siguiente lista de restricciones (óp. cit.) (ob. cit.):

- De la Propiedad de Objeto “tiene un Punto de Datos de Entrada” (hasInputDataPoint), cuando la clase “Comando” se relaciona de manera unitaria con la clase “Punto de Datos de Entrada”, conociéndose ello como una restricción universal; la cual consiste en que una clase Comando (Class:Command), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property: hasInputDataPoint) con otra clase –o clases– Punto de Datos de Entrada (Class: InputDataPoint).
- De la Propiedad de Objeto “tiene un Punto de Datos de Salida” (hasOutputDataPoint), cuando la clase “Comando” se relaciona de manera unitaria con la clase “Punto de Datos de Salida”, conociéndose ello como una restricción universal; la cual consiste en que una clase Comando (Class:Command), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property: hasOutputDataPoint) con otra(s) clase(s) Punto de Datos de Salida (Class: OutputDataPoint).
- De la Propiedad de Objeto “tiene la Entrada” (hasInput), cuando la clase “Comando” se relaciona de manera unitaria con la clase “Entrada de Operación”, conociéndose ello como una restricción universal; la cual consiste en que una clase Comando (Class:Command), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasInput) con otra(s) clase(s) “Entrada de Operación” (Class: OperationInput).
- De la Propiedad de Objeto “tiene la Salida” (hasOutput), cuando la clase “Comando” se relaciona de manera unitaria con la clase “Salida de Operación”, conociéndose ello como una restricción universal; la cual consiste en que una clase Comando (Class:Command), solo puede tener una y solo una relación o Propiedad de Objeto (Object Property:hasOutput) con otra clase –u otras clases– “Salida de Operación” (Class: OperationOutput).

Las relaciones u “ObjectProperties” que caracterizan a una clase “Comando” (oneM2M:Command) se detallan en la tabla 32 (óp. cit.):

Propiedad (Property)	Definición
-----------------------------	-------------------

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:exposesCommand	Enlace referente a la exposición u ofrecimiento –interpretable por un Dispositivo– de una orden o Comando – comprensible por un usuario humano– en una red informática. La relación está definida entre las clases “Operation”, “InputDataPoint” y “OutputDataPoint” que se desempeñan como dominio y la clase “Command” que obra como recorrido de las mismas; i.e. dicha relación con la triada de clases es inyectiva (ibíd.).
	oneM2M:hasCommand	Enlace referente a las órdenes o comandos que pertenecen a una Función y que le permite a un usuario, incidir en un determinado Aspecto del universo de discurso, en el cual un Dispositivo se halla inmerso. Esta relación existe entre la clase “Función” que se desempeña como dominio y la clase “Command” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasInput	Vínculo consistente en la Entrada de datos que puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicho artefacto. Este vínculo existe entre la clase “Command” que funge como dominio y la clase “OperationInput” que se obra como recorrido; i.e., la relación es inyectiva (ib.).
	oneM2M:hasOutput	Enlace referente a la Salida de datos que puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicha máquina. Esta relación existe entre la clase “Command” que se desempeña como dominio y la clase “OperationOutput” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 32. Propiedades de la clase Comando o “Command” de oneM2M (oneM2M, 2019, pág. 32) (ETSI, 2021, pág. 33)

- Entrada de Operación (OperationInput):

Una Entrada de Operación (Class:OperationInput) describe el tipo de entrada de una Operación hacia un Servicio del Dispositivo. La clase Entrada de Operación representa todos los valores posibles para esa entrada (v.g., tipos de datos, rangos de valores o una lista de individuos enumerados). Una operación puede tener múltiples Entradas de Operación y/o Salidas de Operación (oneM2M, 2019, pág. 33) (ETSI, 2021, pág. 34).

Si una instancia de una Operación es ejecutada, entonces el valor de entrada para (o hacia) esa Operación, es una instancia de sus clases de Entrada de Operación (por ejemplo, instancias enumeradas como "ON" u "OFF", para una clase de Entrada de Operación que determina el estado de un interruptor o en su defecto, un número real dentro de un cierto rango, para una clase de Entrada de Operación de "temperatura" perteneciente a un termostato (óp. cit.) (ob. cit.).

En la figura 77 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Entrada de Operación (oneM2M: OperationInput):

Class: OperationInput

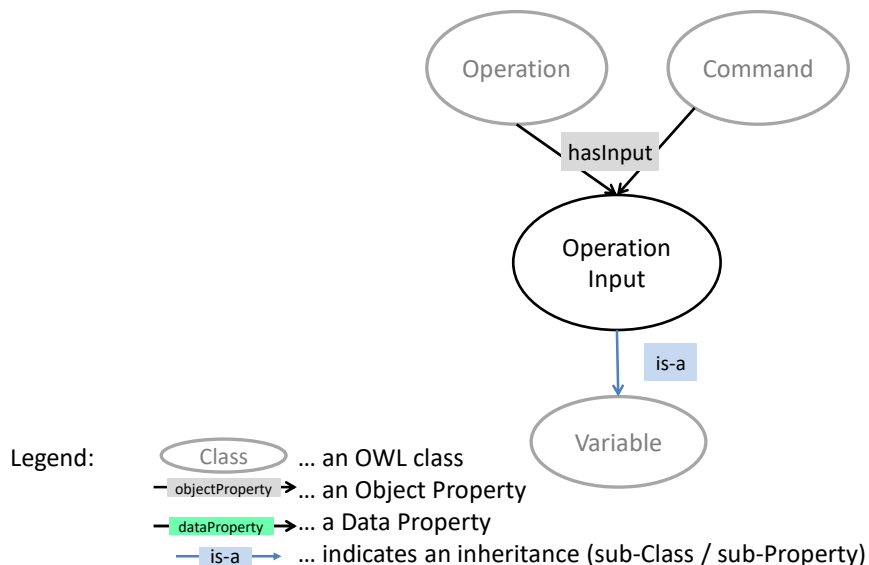


Figura 77. Clase Entrada de Operación o “OperationInput” de oneM2M (oneM2M, 2019, pág. 33) (ETSI, 2021, pág. 34)

La clase Entrada de Operación u “OperationInput”, es subclase de la clase “Variable” (oneM2M:Variable); mas no es superclase de ninguna otra clase. Así mismo, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (oneM2M, 2019, pág. 34) (ETSI, 2021, pág. 35), ya sea de tipo universal, existencial o de cardinalidad.

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Entrada de Operación o “oneM2M:OperationInput” se detallan en la tabla 33 (óp. cit.):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una Cosa o Dispositivo. Esta relación existe entre la clase “OperationInput” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva. Esta propiedad de objeto es heredada de la clase “Variable”, dado que la clase “OperationInput” es una subclase de esta última (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una Cosa o Dispositivo. La relación está definida entre la clase “OperationInput” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. La presente propiedad de objeto es heredada de la clase “Variable”, en razón a que la clase “OperationInput” es una subclase de la anterior (ibíd.).
	oneM2M:hasInput	Vínculo consistente en la Entrada de datos que puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicho artefacto. Este vínculo existe entre las clases “Operation” y “Command” que funcionan ambas como dominio y la clase “OperationInput” que se obra como recorrido; i.e., la relación es inyectiva (ib.).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de una descripción semántica referida a la Propiedad de una Cosa con un tipo de dato literal, v.g., el nombre del fabricante de una cosa que posee conectividad. Dicha característica posee como dominio a la clase “OperationInput” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “oneM2MTargetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).

	<p>oneM2M:oneM2MMethod</p>	<p>Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa, mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase “OperationInput” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).</p>
	<p>oneM2M:oneM2MTargetURI</p>	<p>Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase “Entrada de Operación”, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “OperationInput” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).</p>

Tabla 33. Propiedades de la clase Entrada de Operación u “OperationInput” de oneM2M (oneM2M, 2019, pág. 33) (ETSI, 2021, pág. 34)

- Salida de Operación (OperationOutput):

La clase “Salida de Operación” (Class:OperationOutput) describe el tipo de salida de una Operación hacia un Servicio del Dispositivo. La clase “OperationOutput” representa todos los valores posibles para dicha Salida (tipos de datos, rangos de valores o una lista de individuos enumerados) (oneM2M, 2019, pág. 35) (ETSI, 2021, pág. 36).

Es de tener en cuenta que una Operación (Class:Operation) puede tener múltiples Salidas de Operación y/o Entradas de Operación. Si se ejecuta una instancia de una Operación, entonces los valores de salida de esa Operación son instancias de sus clases Salida de Operación (Class:OperationOutput). Además, toda Salida de Operación es transitoria; i.e., una instancia de una “OperationOutput” se elimina cuando, en simultánea, la instancia de su respectiva Operación (Class:Operation) también es suprimida (óp. cit.) (ob. cit.).

En la figura 78 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Salida de Operación (oneM2M: OperationOutput):

Class: OperationOutput

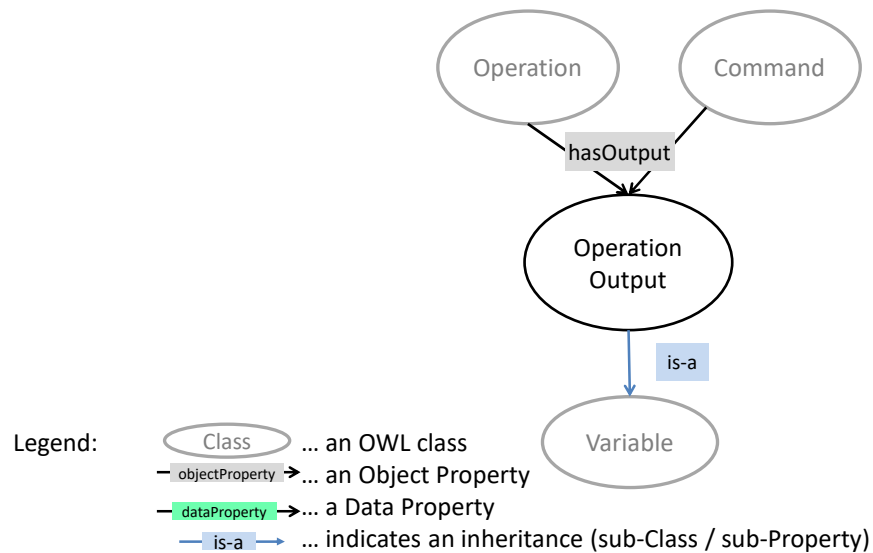


Figura 78. Clase Salida de Operación o “OperationOutput” de oneM2M (oneM2M, 2019, pág. 34) (ETSI, 2021, pág. 35)

La clase Salida de Operación u “OperationOutput”, es subclase de la clase “Variable” (oneM2M:Variable); mas la misma no es superclase de ninguna otra clase. Igualmente, esta clase no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty) y por ende, no se presenta restricción alguna (oneM2M, 2019, pág. 35) (ETSI, 2021, pág. 36), ya sea de tipo existencial, de cardinalidad o universal.

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que distinguen a una clase Salida de Operación (oneM2M:OperationOutput) se detallan en la tabla 34 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

(ObjectProperty)		propiedad de una cosa. Esta relación existe entre la clase “OperationOutput” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva. Esta propiedad de objeto es heredada de la clase “Variable”, dado que la clase “OperationOutput” es una subclase de esta última (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una cosa. La relación está definida entre la clase “OperationOutput” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. La presente propiedad de objeto es heredada de la clase “Variable”, en razón a que la clase “OperationOutput” es una subclase de la anterior (ibíd.).
	oneM2M:hasOutput	Vínculo consistente en la Salida de datos que puede tener, tanto una Operación de un Servicio del Dispositivo, como un Comando de una Función de dicho artefacto. Este vínculo existe entre las clases “Operation” y “Command” que fungen ambas como dominio y la clase “OperationOutput” que se obra como recorrido; i.e., la relación es inyectiva (ib.).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de una descripción semántica referida a la Propiedad de una Cosa con un tipo de dato literal, v.g., el nombre del fabricante de una cosa que posee conectividad. Dicha característica posee como dominio a la clase “OperationOutput” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “oneM2MTargetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).
	oneM2M:oneM2MMethod	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa,

		mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase “OperationOutput” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).
	oneM2M:oneM2MTargetURI	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase “Salida de Operación”, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “OperationOutput” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).

Tabla 34. Propiedades de la clase Salida de Operación u “OperationOutput” de oneM2M (oneM2M, 2019, pág. 35) (ETSI, 2021, pág. 36)

- Punto de Datos de Entrada (InputDataPoint):

La clase “Punto de Datos de Entrada” (Class:InputDataPoint) concierne a una Variable (Class:Variable) de un Servicio (Class:Service), que se encuentra determinada para un Dispositivo o “Device” RESTful (Estado de Transferencia Representacional pleno), el cual se halla inmerso en un universo de discurso. Dicho “InputDataPoint” es leído o **recuperado** por el Dispositivo (i.e., un activador) de manera autónoma (v.g., de manera periódica) (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37), para que a partir de dichos datos obtenidos, llevar a cabo la tarea asignada.

Con el propósito de permitir a un tercero que instruya a un Dispositivo para que recupere (por fuera de su programación) el valor actual de un “InputDataPoint”, los dispositivos también pueden ofrecer una Operación (Class:Operation) del tipo “Obtener_PuntoDeDatosDeEntrada” (Class:GET_InputDataPoint); ello con el fin último de activar el Dispositivo para adquirir los datos del “InputDataPoint” (óp. cit.) (ob. cit.).

Los Puntos de Datos de Entrada (y los de Salida también), suelen ser utilizados por Dispositivos (Application Entities AE's al interior del sistema oneM2M) que se comunican mediante RESTful; mientras que las Operaciones son las tareas sistemáticas (o rutinas) que se utilizan para la comunicación basada en procedimientos remotos (Remote Procedure Communication RPC). Sin embargo, las Operaciones también son necesarias en los sistemas RESTful para correlacionar la salida, que es generada por un dispositivo, con la entrada que desencadenó la producción de dicha salida (oneM2M, 2019, pág. 15) (ETSI, 2021, pág. 16).

En la figura 79 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Punto de Datos de Entrada (oneM2M:InputDataPoint) (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37):

Class: InputDataPoint

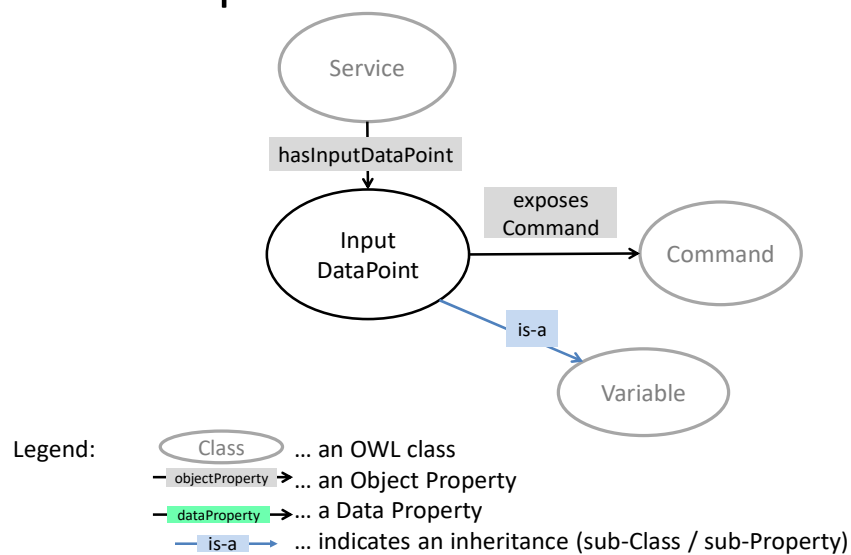


Figura 79. Clase Punto de Datos de Entrada o “InputDataPoint” de oneM2M (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37)

La clase Punto de Datos de Entrada (Class:InputDataPoint), es subclase de la clase “Variable” (Class:Variable); mas no es superclase de ninguna otra clase. Igualmente, la clase “InputDataPoint” no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no se presenta restricción alguna (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37), ya sea de tipo existencial, universal o de cardinalidad.

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Punto de Datos de Entrada (oneM2M:InputDataPoint) se detallan en la tabla 35 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Relación (ObjectProperty)	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una Cosa o Dispositivo. Esta relación existe entre la clase “InputDataPoint” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva. Esta propiedad de objeto es heredada de la clase “Variable”, dado que la clase “InputDataPoint” es una subclase de esta última (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una Cosa o Dispositivo. La relación está definida entre la clase “InputDataPoint” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. La presente propiedad de objeto es heredada de la clase “Variable”, en razón a que la clase “InputDataPoint” es una subclase de la anterior (ibíd.).
	oneM2M:exposesCommand	Enlace referente a la exposición u ofrecimiento –interpretable por un Dispositivo– de una orden o Comando – comprensible por un usuario humano– en una red informática. Esta relación existe entre la clase “InputDataPoint” que se desempeña como dominio y la clase “Command” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasInputDataPoint	Vínculo concerniente a un Servicio que dispone de una lectura o recuperación de datos de entrada a un dispositivo, en un instante específico o periódico (i.e., recepción o “receipt”). La relación está definida entre la clase “Service” que funge como dominio y la clase “InputDataPoint”, la cual obra como recorrido; Por lo tanto, dicha relación es inyectiva (ib.).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de una descripción semántica referida a la Propiedad de una Cosa con un tipo de dato literal, v.g., el nombre del fabricante de una cosa que posee conectividad. Dicha característica posee como dominio a la clase “InputDataPoint” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		Esta propiedad de datos junto con la propiedad “oneM2MTargetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).
	oneM2M:oneM2MMethod	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa, mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase “InputDataPoint” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).
	oneM2M:oneM2MTargetURI	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase “Punto de Datos de Entrada”, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “InputDataPoint” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).

Tabla 35. Propiedades de la clase Punto de Datos de Entrada o “InputDataPoint” de oneM2M (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37)

- Punto de Datos de Salida (OutputDataPoint):

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

La clase “Punto de Datos de Salida” (Class:OutputDataPoint) atañe a una Variable (Class:Variable) de un Servicio (Class:Service), que se encuentra determinada para un Dispositivo o “Device” RESTful (Estado de Transferencia Representacional pleno), el cual se halla inmerso en un universo de discurso. Dicho “OutputDataPoint” es **actualizado** o producido por el Dispositivo (i.e., un sensor) autónomamente (v.g., mediante series de tiempo) (oneM2M, 2019, pág. 37) (ETSI, 2021, pág. 38), para que a partir de dichos datos generados, realizar la tarea asignada.

Ante el objetivo de facilitarle a un tercero instruir a un Dispositivo para que recupere el valor actual de un “OutputDataPoint” independientemente de su programación, los dispositivos pueden igualmente ofrecer una Operación (Class:Operation) del tipo “Establecer_PuntoDeDatosDeSalida” (Class:SET_OutputDataPoint); ello con el propósito de activar el Dispositivo para actualizar los datos del “OutputDataPoint” (óp. cit.) (ob. cit.).

Indistintamente, los Puntos de Datos de Salida (y los de Entrada), a menudo se utilizan por parte de los Dispositivos (Application Entities AE's según el sistema oneM2M) que se comunican mediante RESTful; mientras que las Operaciones son las tareas sistemáticas (o rutinas) que se utilizan para la comunicación basada en procedimientos remotos (Remote Procedure Communication RPC). No obstante, las Operaciones también son necesarias en los sistemas RESTful para correlacionar la entrada, que es recibida por un dispositivo, con la salida que desencadenó la adquisición de datos de dicha entrada (oneM2M, 2019, pág. 15) (ETSI, 2021, pág. 16).

En la figura 80 se observan las relaciones (ObjectProperties) y los atributos (DataProperties) de la clase Punto de Datos de Salida (oneM2M:OutputDataPoint) (oneM2M, 2019, pág. 37) (ETSI, 2021, pág. 38):

Class: OutputDataPoint

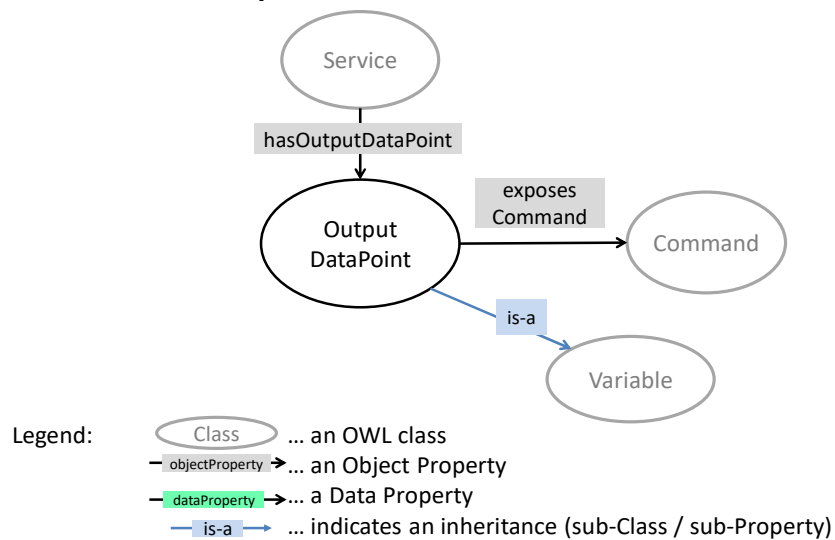


Figura 80. Clase Punto de Datos de Salida u “OutputDataPoint” de oneM2M (oneM2M, 2019, pág. 37) (ETSI, 2021, pág. 38)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

La clase Punto de Datos de Salida (Class:OutputDataPoint), es subclase de la clase “Variable” (Class:Variable); mas no es superclase de ninguna otra clase. Así mismo, la clase “InputDataPoint” no es una subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por tanto, no se presenta restricción alguna (oneM2M, 2019, pág. 36) (ETSI, 2021, pág. 37), ya sea de tipo universal, de cardinalidad o existencial.

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Punto de Datos de Salida (oneM2M:outputDataPoint) se detallan en la tabla 36 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una Cosa o Dispositivo. Esta relación existe entre la clase “OutputDataPoint” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva. Esta propiedad de objeto es heredada de la clase “Variable”, dado que la clase “OutputDataPoint” es una subclase de esta última (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una Cosa o Dispositivo. La relación está definida entre la clase “OutputDataPoint” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. La presente propiedad de objeto es heredada de la clase “Variable”, en razón a que la clase “OutputDataPoint” es una subclase de la anterior (ibíd.).
	oneM2M:exposesCommand	Enlace referente a la exposición u ofrecimiento –interpretable por un Dispositivo– de una orden o Comando – comprensible por un usuario humano– en una red informática. Esta relación existe entre la clase “OutputDataPoint” que se desempeña como dominio y la clase “Command” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasOutputDataPoint	Vínculo concerniente a un Servicio que dispone de una lectura o recuperación de datos de entrada a un dispositivo, en un instante específico o periódico (i.e., recepción o “receipt”). La relación está

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		definida entre la clase “Service” que funge como dominio y la clase “OutputDataPoint”, la cual obra como recorrido; Por lo tanto, dicha relación es inyectiva (ib.).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de una descripción semántica referida a la Propiedad de una Cosa con un tipo de dato literal, v.g., el nombre del fabricante de una cosa que posee conectividad. Dicha característica posee como dominio a la clase “OutputDataPoint” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “oneM2MTargetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).
	oneM2M:oneM2MMethod	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase Propiedad de una Cosa, mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar –“create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase “OutputDataPoint” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).
	oneM2M:oneM2MTargetURI	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando. Dicha propiedad de datos pertenece a la clase “Punto de Datos de Salida”, mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “OutputDataPoint” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource

		Description Framework Schema). Esta propiedad de datos junto con la propiedad "hasValue", son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).
--	--	--

Tabla 36. Propiedades de la clase Punto de Datos de Salida u "OutputDataPoint" de oneM2M (oneM2M, 2019, pág. 37) (ETSI, 2021, pág. 38)

- Variable (Variable):

Una Variable (Class:Variable) es una clase referente a una magnitud o característica medible de una entidad, la cual puede tener un valor cualquiera en un intervalo especificado. También constituye una superclase para las siguientes clases: Propiedad de la Cosa (ThingProperty), Entrada de la Operación (OperationInput), Salida de la Operación (OperationOutput), Punto de Datos de Entrada (InputDataPoint) y Punto de Datos de Salida (OutputDataPoint) (oneM2M, 2019, pág. 38) (ETSI, 2021, pág. 39).

Los miembros de una Variable son entidades que tienen algunos datos (por ejemplo, números enteros, texto, etc., o datos estructurados) que pueden cambiar con el tiempo. Estos datos de la clase "Variable", generalmente describen algunos Aspectos (Class:Aspect) del mundo real (por ejemplo, una temperatura) y pueden tener MetaDatos (Class:MetaData), como por ejemplo, unidades, precisión, etc. (óp. cit.) (ob. cit.).

Una Variable (Class:Variable) se puede estructurar, es decir, puede constar de (sub) Variables. Adicionalmente, las siguientes subclases están definidas en la ontología base de la siguiente manera (óp. cit.) (ob. cit.):

- Variable de Tipo Simple (SimpleTypeVariable): Una Variable de Tipo Simple (Class:SimpleTypeVariable) es una subclase de la clase Variable (Class:Variable) que solo consta de variables de tipos xml simples (o primitivas) como xsd:integer, xsd:string, etc., que potencialmente incluyen restricciones (condiciones o limitaciones).
- Variable de Tipo Estructurada (StructuredTypeVariable): Una Variable de Tipo Estructurada (Class:StructuredTypeVariable) es una subclase de la clase Variable (Class:Variable) que describe una variable que está estructurada (configurada u organizada), compuesta por un conjunto de otras variables. Este conjunto de otras variables puede contener a variables de tipo simple (Class: SimpleTypeVariable) o también a variables estructuradas (Class: StructuredTypeVariable).
- La clase Variable (Class:Variable) es la unión disjunta de las clases "SimpleTypeVariable" y "StructuredTypeVariable"; es decir, cualquier variable es una de dos, de tipo estructurado o de tipo simple.

En la figura 81 se observan las relaciones (ObjectProperties), los atributos (DataProperties) y las subclases de la clase Variable (Class:Variable):

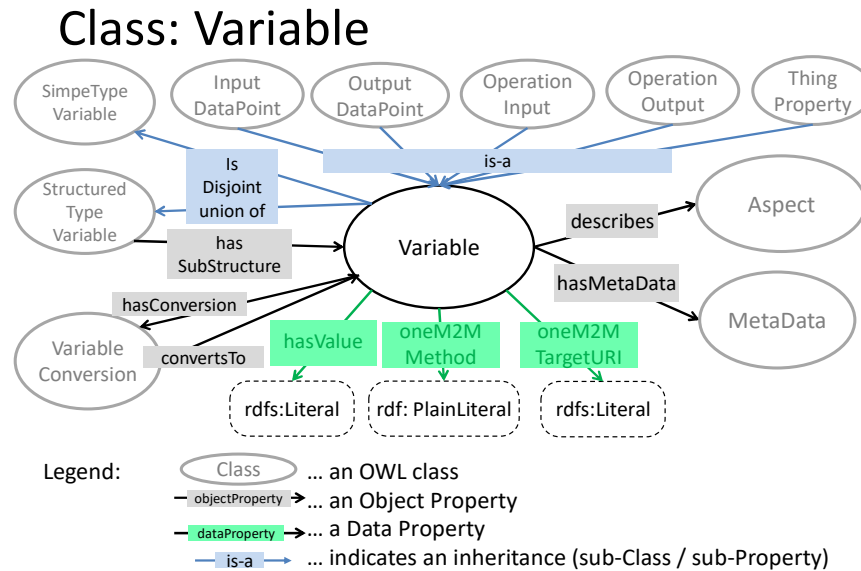


Figura 81. Clase Variable o “Variable” de oneM2M (oneM2M, 2019, pág. 38) (ETSI, 2021, pág. 39)

Es de aclarar que la Propiedad de Datos (DataProperty) correspondiente a “Método oneM2M” (oneM2MMethod), ya fue descrita detalladamente al interior de la clase Operación (Class:Operation); no obstante, se vuelve a reseñar a continuación con propósitos de completitud para la clase “Variable”:

El “Método oneM2M” (Data Property:oneM2MMethod) atañe una Propiedad de Datos que contiene un método CRUD del sistema oneM2M, a través del cual una instancia oneM2M del valor de la Variable (Class:Variable), puede ser manipulada por la entidad comunicante de las siguientes maneras (oneM2M, 2019, pág. 39) (ETSI, 2021, pág. 40):

- Al contener o abarcar la cadena "RETRIEVE" para recuperar la variable cuando el recurso oneM2M es de tipo <container> o <flexContainer>. Esto se aplica a las subclases: “OperationOutput”, “OutputDataPoint” y “ThingProperty”.
- Al envolver o contener la cadena "CREATE" para actualizar la variable cuando el recurso oneM2M es de tipo <container>. Esto se aplica a las subclases: “OperationInput”, “InputDataPoint” y “ThingProperty”.
- Al abarcar o envolver la cadena "UPDATE" para actualizar la variable cuando el recurso oneM2M es de tipo <flexContainer>. Esto se aplica a las subclases: “OperationInput”, “InputDataPoint” y “ThingProperty”.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Esta Propiedad de Datos (Data Property:oneM2MMethod) se encuentra definida entre las clases “Variable” y “Operation” que se desempeñan como dominio, y el tipo de dato literal simple (datatype:rdf:PlainLiteral) que obra como recorrido de las mismas, siendo aquel definido en el Marco de Descripción de Recursos RDF (Resource Description Framework). Por lo tanto, la relación de ambas clases con el tipo de dato es inyectiva (ibíd.).

Por último y al igual que el atributo anterior, se describirá con detenimiento la Propiedad de Datos (DataProperty) correspondiente a “URI de destino oneM2M” (oneM2MTargetURI) (oneM2M, 2019, pág. 39) (ETSI, 2021, pág. 40):

El “URI de destino oneM2M” (Data Property:oneM2MTargetURI), cuyo tipo de datos de recorrido es rdfs:Literal), corresponde a la propiedad de datos que contiene el URI (Uniform Resource Identifier) de un recurso oneM2M (<container> o <flexContainer>), a través del cual la instancia oneM2M del valor de la Variable puede ser manipulada por la entidad comunicante. Puede contener una dirección absoluta o una dirección relativa al recurso <semanticDescriptor> que contiene la descripción RDF de la variable (óp. cit.) (ob. cit.).

Dicha Identificador Uniforme de Recurso URI, el cual consiste en una dirección web de destino podría ser por ejemplo, el valor del ID principal (parentID) para el <container> o <flexContainer> de un Punto de Datos de Entrada o Salida que tiene un recurso secundario de tipo <semanticDescriptor>, el cual a su vez contiene la descripción del Marco de Descripción de Recursos RDF perteneciente al punto de datos (DataPoint) (óp. cit.) (ob. cit.).

Las relaciones y atributos (“ObjectProperties” y “DataProperties”) que caracterizan a una clase Variable o “oneM2M:Variable” se detallan en la tabla 37 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:describes	Vínculo consistente en la descripción o reseña de un aspecto correspondiente a la propiedad de una cosa. Esta relación existe entre la clase “Variable” que funge como dominio y la clase “Aspect” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasMetaData	Enlace referente a la tipificación y especificidad que posee un dato que hace parte de la propiedad de una cosa. La relación está definida entre la clase “Variable” que se desempeña como dominio y la clase “MetaData” que obra como recorrido; i.e. la relación es inyectiva. (ibíd.).
	oneM2M:hasSubStructure	Relación consistente en la existencia de una variable de tipo estructurado, conformada a partir de otras variables en conjunto (subvariables). Esta relación se presenta entre la clase “StructuredTypeVariable” que funge como dominio y la clase “Variable” que

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		se desempeña como recorrido; id est, la relación es inyectiva (ibídem.).
	oneM2M:hasConversion	Vínculo que atañe a la representación de una regla de conversión (o transformación) del rango de valores de una Variable (o magnitud), en otra imagen de datos de esa misma magnitud, o de una Variable diferente. Esta relación existe entre la clase “Variable” que funge como dominio y la clase “VariableConversion” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:convertsTo	Relación referente a la conversión de un rango de valores de una Variable (o magnitud), en otra imagen de datos de esa misma magnitud, o de una Variable diferente, aplicando una regla de transformación. Esta relación existe entre la clase “VariableConversion” que funge como dominio y la clase “Variable” que actúa como recorrido; id est, la relación es inyectiva (ibídem).
Atributo (DataProperty)	oneM2M:hasValue	Propiedad de datos en la que se almacena el valor de una descripción semántica referida a una Propiedad de una Cosa, una Operación de Entrada o Salida, una Variable de Tipo Simple o también, un Punto de Datos de Entrada o Salida. Dicha característica posee como dominio a la clase “Variable” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “oneM2M:targetURI”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (oneM2M, 2019, pág. 54) (ETSI, 2021, pág. 55).
	oneM2M:oneM2MMethod	Propiedad de datos en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando, el cual es referente a una Propiedad de una Cosa, una Operación de Entrada o Salida, una Variable de Tipo Simple o también, un Punto de Datos de Entrada o Salida. Dicho almacenamiento se realiza mediante un tipo de dato literal simple que alude a una de las tres operaciones CRUD: crear, actualizar o recuperar – “create”, “update” o “retrieve”–. Dicha característica posee como dominio a la clase

		<p>“Variable” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).</p>
	<p>oneM2M:oneM2MTargetURI</p>	<p>Propiedad de datos en la que se almacena el valor de un recurso contenedor que es susceptible de ser manipulado por un dispositivo que se está comunicando, el cual es atinente a una Propiedad de una Cosa, una Operación de Entrada o Salida, una Variable de Tipo Simple o también, un Punto de Datos de Entrada o Salida. Tal almacenamiento se efectúa mediante un tipo de dato literal que alude a una dirección de un Identificador Uniforme de Recurso URI, tanto absoluto como relativo a un descriptor semántico. Dicha característica posee como dominio a la clase “Variable” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad “hasValue”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (óp. cit., pág. 55) (ob. cit., pág. 56).</p>

Tabla 37. Propiedades de la clase Variable o “Variable” de oneM2M (oneM2M, 2019, pág. 39) (ETSI, 2021, pág. 40)

- Variable de Tipo Simple (SimpleTypeVariable):

La clase “Variable de Tipo Simple” (Class:SimpleTypeVariable) es una subclase de la clase Variable (Class:Variable), que solo consta de variables de tipos simples pertenecientes al Esquema de Lenguaje de Marcado Extensible XML (Extensible Markup Language Schema) –conocidas también con el nombre de primitivas–, tales como xsd:integer, xsd:string, etc., las cuales potencialmente incluyen restricciones (i.e., condiciones o limitaciones) (oneM2M, 2019, pág. 40) (ETSI, 2021, pág. 41).

No obstante, el sistema oneM2M admite para la clase “SimpleTypeVariable”, los tipos de datos simples y las restricciones contenidas en la especificación estructural y sintaxis de estilo funcional (segunda edición) del Lenguaje de Ontologías Web OWL2 (W3C, 2014a) (W3C, 2012a). Además, el identificador de espacio de nombres (namespace) para el esquema XML, hace uso del prefijo “xs” (W3C, 2012) (IEEE, 2008, p. 4).

En la figura 82 se observan únicamente las Propiedades de Datos o atributos (DataProperties) de la clase “Variable de Tipo Simple” (oneM2M:SimpleTypeVariable), dado que la misma carece de

Propiedades de Objeto o relaciones (ObjectProperties) con otras clases (oneM2M, 2019, pág. 40) (ETSI, 2021, pág. 41):

Class: SimpleTypeVariable

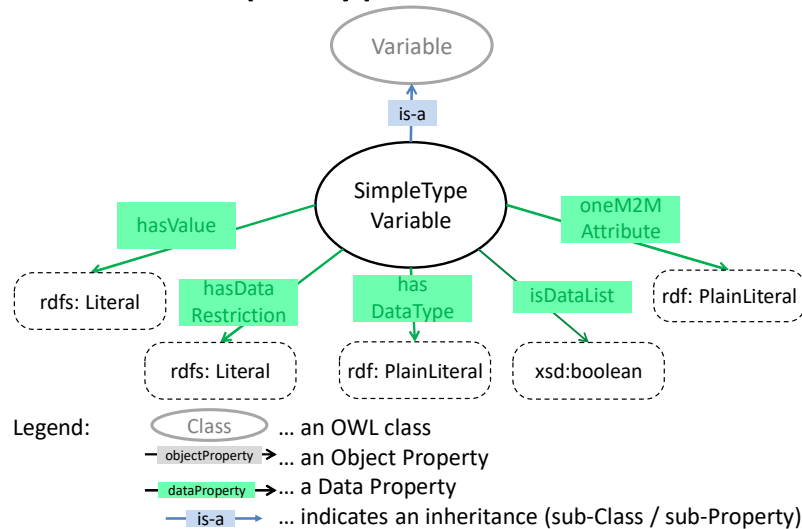


Figura 82. Clase Variable de Tipo Simple o “SimpleTypeVariable” de oneM2M (oneM2M, 2019, pág. 40) (ETSI, 2021, pág. 41)

Es de resaltar que, tal y como se observa en la figura 81, que la clase “Variable de Tipo Simple” (oneM2M: SimpleTypeVariable) a pesar de no ser superclase de ninguna clase, si es subclase de la clase Variable (Class:Variable); además de ser está última, la unión disjunta de las clases “SimpleTypeVariable” y “StructuredTypeVariable”. Dicho de otro modo, cualquier variable es una de dos, o de tipo simple o de tipo estructurado (óp. cit.) (ob. cit.).

No obstante, la clase “Variable de Tipo Simple” (oneM2M: SimpleTypeVariable) funge como una subclase anónima de diversas Propiedades de Datos o “DataProperties”, las cuales se describen en la siguiente lista de restricciones (óp. cit.) (ob. cit.):

- De la Propiedad de Datos “Tiene una Restricción de Datos” (hasDataRestriction), cuando la clase “SimpleTypeVariable” se relaciona de manera unitaria con el tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework), conociéndose ello como una restricción universal; la cual consiste en que una clase “Variable de Tipo Simple” (Class: SimpleTypeVariable), solo puede tener una y solo una relación o Propiedad de Datos (Data Property: hasDataRestriction) con un tipo de dato literal simple.
- De la Propiedad de Datos “Tiene un Tipo de Dato” (hasDataType), cuando la clase “SimpleTypeVariable” se relaciona de manera obligatoria y particular con el tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework), conociéndose ello como una restricción de cardinalidad. Dicha condición consiste en que la clase “Variable de Tipo Simple” (Class: SimpleTypeVariable), solo puede tener a lo sumo, una relación

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

de Propiedad de Datos (Data Property: `hasDataType`) con un tipo de dato literal simple.

- De la Propiedad de Objeto “Tiene una SubEstructura” (`hasSubStructure`), cuando la clase “SimpleTypeVariable” se relaciona de manera obligatoria y particular con con la clase “Variable”, conociéndose ello como una restricción de cardinalidad. Dicha condición consiste en que la clase “Variable de Tipo Simple” (Class: SimpleTypeVariable), no debe ni puede tener, ninguna relación de Propiedad de Objeto (Object Property: `hasSubStructure`) con la clase “Variable” (Class: Variable), dado que una variable de tipo simple tiene exclusivamente un tipo de datos, pero nunca otras variables tipo subestructura.

Los atributos (DataProperties) que caracterizan a una clase “Variable de Tipo Simple” (`oneM2M:SimpleTypeVariable`) se detallan en la tabla 38 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	Ninguno	Ninguno
Atributo (DataProperty)	<code>oneM2M:hasValue</code>	Propiedad de datos heredada de la superclase “Variable”, en la que se almacena el valor para un tipo de dato literal simple o llano. Dicha característica posee como dominio a la clase “SimpleTypeVariable” y como recorrido al tipo literal (<code>datatype:rdfs:Literal</code>) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema). Esta propiedad de datos junto con la propiedad o “ <code>oneM2M:targetURI</code> ”, son mutuamente excluyentes; i.e., una y solo una de ambas debería ser instanciada (<code>oneM2M</code> , 2019, pág. 54) (ETSI, 2021, pág. 55).
	<code>oneM2M:oneM2MAtribute</code>	Propiedad de datos en la que se almacena el valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se está comunicando. Esta propiedad de datos contiene el nombre del atributo del recurso <code>oneM2M</code> (de tipo <code><container></code> o <code><flexContainer></code>) al que se hace referencia con el URI de destino <code>oneM2M</code> (<code>oneM2M:targetURI</code>) y que almacena el valor de la variable de tipo simple. Dicha característica posee como dominio a la clase

		<p>“SimpleTypeVariable” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57).</p>
	oneM2M:hasDataType	<p>Propiedad de datos consistente en el repositorio del valor de un recurso contenedor susceptible de ser manipulado por un dispositivo que se halle transfiriendo. Esta propiedad de datos contiene el tipo de datos como cadena de texto, de la variable de tipo simple tales como: “integer”, “double”, “string”, “boolean”, “binary”, “datetime”, etc. Dicha característica posee como dominio a la clase “SimpleTypeVariable” y como recorrido al tipo de dato literal simple (datatype:rdf:PlainLiteral) definido en el Marco de Descripción de Recursos RDF (Resource Description Framework) (óp. cit., pág. 56) (ob. cit., p. 57), como también en el espacio de nombres del Esquema XML bajo el prefijo “xs” (W3C, 2012) (IEEE, 2008, p. 4).</p>
	oneM2M:hasDataRestriction	<p>Propiedad de datos que especifica las condiciones o limitaciones impuestas a los tipos de datos pertenecientes a la clase “Variable de Tipo Simple”, soportada en las siguientes especialidades: “Valores mayores e iguales qué” (minInclusive), “Valores mayores qué” (minExclusive), “Valores menores e iguales qué” (maxInclusive), “Valores menores qué” (maxExclusive), cantidad exacta de caracteres (length), cantidad mínima de caracteres (minLength) y cantidad máxima de caracteres (maxLength). Dicha característica posee como dominio a la clase “SimpleTypeVariable” y como recorrido al tipo literal (datatype:rdfs:Literal) definido en el Esquema del Marco de Descripción de Recursos RDFS (Resource Description Framework Schema) (óp. cit.)</p>
	oneM2M:isDataList	<p>Propiedad de datos que indica si una Variable de Tipo Simple contiene una lista de datos o un dato sencillo, a partir de un valor booleano. Si el valor lógico es “true”, significa que el dato es una lista de valores del tipo especificado en la propiedad de datos “tiene un Tipo de Datos”. En caso contrario (i.e., si es “false”), se interpreta como que el dato es un valor único, según lo establecido en “hasDataType”. Dicha característica posee como dominio a la clase “SimpleTypeVariable” y como recorrido al tipo de dato booleano o lógico (datatype:xsd:boolean), definido en el espacio de nombres del Esquema de Lenguaje de</p>

		Marcado Extensible XML –Extensible Markup Language Schema–, bajo el prefijo “xs” (W3C, 2012) (IEEE, 2008, p. 4).
--	--	--

Tabla 38. Propiedades de la clase Variable de Tipo Simple o “SimpleTypeVariable” de oneM2M (oneM2M, 2019, pág. 40) (ETSI, 2021, pág. 41)

- Variable de Tipo Estructurado (StructuredTypeVariable):

La clase “Variable de Tipo Estructurado” (Class:StructuredTypeVariable) es una subclase de la clase Variable (Class:Variable) que describe una variable que está estructurada, configurada o compuesta por un conjunto de otras variables. Dicha agrupación de variables diversas, puede contener a Variables de Tipo Simple (Class:SimpleTypeVariable) o también a Variables Estructuradas (Class:StructuredTypeVariable) (oneM2M, 2019, pág. 42) (ETSI, 2021, pág. 43).

En la figura 83 se observan únicamente la relación o Propiedad de Objeto (ObjectProperty) de la clase “Variable de Tipo Estructurado” (oneM2M: StructuredTypeVariable), dado que la misma carece de Propiedades de Datos o atributos (DataProperties) (óp. cit.) (ob. cit.):

Class: StructuredTypeVariable

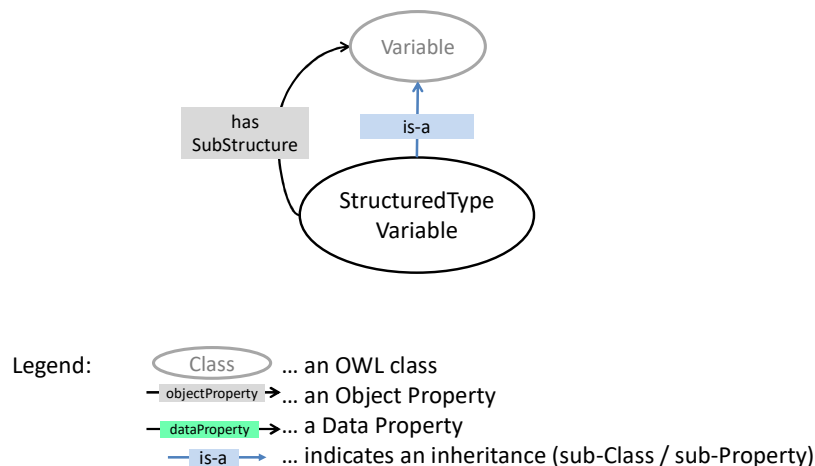


Figura 83. Clase Variable de Tipo Estructurado o “EstructuredTypeVariable” de oneM2M (oneM2M, 2019, pág. 41) (ETSI, 2021, pág. 42)

Un aspecto destacable en la figura 82, atañe a que la clase “Variable de Tipo Estructurado” (Class: StructuredTypeVariable), a pesar de no ser superclase de ninguna clase, si es subclase de la clase Variable (Class:Variable). Adicionalmente, dicha clase “Variable” es la unión disjunta de las clases “StructuredTypeVariable” y “SimpleTypeVariable”; id est, cualquier variable es una de dos, o de tipo estructurado o de tipo simple (oneM2M, 2019, pág. 42) (ETSI, 2021, pág. 43).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Sentado lo anterior, es de considerar que la clase “Variable de Tipo Estructurado” (Class: StructuredTypeVariable) es una subclase anónima de la Propiedad de Objeto “hasSubStructure”, al relacionarse indefectiblemente con la clase Variable, entendiéndose ello como una condición sine qua non (i.e., toda variable estructurada tiene al menos otra variable como subestructura). Lo anterior se conoce como una restricción existencial, la cual consiste en que una clase “Variable” (Class:Variable), dispone como mínimo de una relación o Propiedad de Objeto (Object Property:hasSubStructure) con otra clase “Variable de Tipo Estructurado” (Class:StructuredTypeVariable) (óp. cit.) (ob. cit.).

Las relación o Propiedad de Objeto (ObjectProperty) que caracteriza a la clase “Variable de Tipo Estructurado” (oneM2M:StructuredTypeVariable) se detalla en la tabla 39 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	
Relación (ObjectProperty)	oneM2M:hasSubStructure	Relación consistente en la existencia de una variable de tipo estructurado, conformada a partir de otras variables en conjunto (subvariables), las cuales pueden ser de Tipo Simple o de Tipo Estructurado. Esta relación se presenta entre la clase “StructuredTypeVariable” que funge como dominio y la clase “Variable” que se desempeña como recorrido; id est, la relación es inyectiva (ibídem.).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 39. Propiedades de la clase Variable de Tipo Estructurado o “EstructuredTypeVariable” de oneM2M (oneM2M, 2019, pág. 42) (ETSI, 2021, pág. 43)

- Conversión de Variable (VariableConversion):

La clase “Conversión de Variable” (Class:VariableConversion) representa una regla de conversión (transformación o cambio) del rango de valores de una Variable (Class:Variable) al rango de valores de otra Variable (Class:Variable) (oneM2M, 2019, pág. 43) (ETSI, 2021, pág. 44). La especificación de dicha regla en texto sin formato, se halla contenida en la propiedad de anotación “rdfs: comment”.

Dicha propiedad de anotación atañe a una característica que se puede usar para agregar información o metadatos (i.e., datos acerca de los datos), tanto para clases, como para individuos y propiedades de objetos/datos. He por ello que se le considera como un tercer tipo de propiedades en OWL nominada “AnnotationProperties”. Una Propiedad de Anotación se utiliza para proporcionar información adicional sobre elementos de ontología como clases e instancias, que normalmente son externos a la ontología y no se utilizarían para razonar. Los usos de ejemplo para dicha información adicional son para proporcionar un creador, una versión o un comentario. El

objeto de una propiedad de anotación es un literal de datos, una referencia de URI o un individuo (oneM2M, 2019, pág. 9) (ETSI, 2021, pág. 10).

En la figura 84 se pueden apreciar las relaciones o Propiedades de Objeto (ObjectProperties) de la clase “Conversión de Variable” (oneM2M: VariableConversion), dado que la misma carece de Propiedades de Datos o atributos (DataProperties) (óp. cit., pág. 43) (ob. cit., pág. 44):

Class: VariableConversion

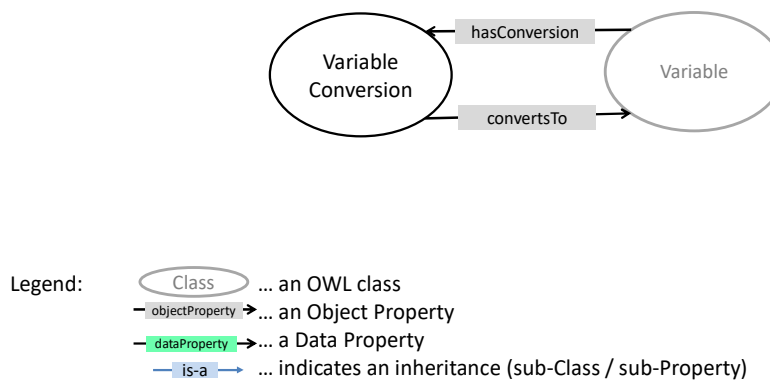


Figura 84. Clase Conversión de Variable o “VariableConversion” de oneM2M (oneM2M, 2019, pág. 43) (ETSI, 2021, pág. 43)

Es de resaltar que la clase “Conversión de Variable” (oneM2M: VariableConversion) se caracteriza por no presentar tipo de herencia alguno, i.e., no es superclase ni subclase de ninguna otra clase, como tampoco ser subclase anónima de ninguna Propiedad de Objeto (ObjectProperty); por ende, no cuenta con restricciones de cardinalidad, de tipo existencial o universal (oneM2M, 2019, pág. 43) (ETSI, 2021, pág. 44).

Las relaciones o Propiedades de Objeto (ObjectProperty) que caracteriza a la clase “Conversión de Variable” (oneM2M:VariableConversion) se detalla en la tabla 40 (óp. cit.):

Propiedad (Property)		Definición
Tipo	Nombre	

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Relación (ObjectProperty)	oneM2M:convertsTo	Relación referente a la conversión de un rango de valores de una Variable (o magnitud), en otra imagen de datos de esa misma magnitud, o de una Variable diferente, aplicando una regla de transformación. Esta relación existe entre la clase "VariableConversion" que funge como dominio y la clase "Variable" que actúa como recorrido; id est, la relación es inyectiva (ibídem).
	oneM2M:hasConversion	Vínculo que atañe a la representación de una regla de conversión (o transformación) del rango de valores de una Variable (o magnitud), en otra imagen de datos de esa misma magnitud, o de una Variable diferente. Esta relación existe entre la clase "Variable" que funge como dominio y la clase "VariableConversion" que actúa como recorrido; id est, la relación es inyectiva (ibídem).
Atributo (DataProperty)	Ninguno	Ninguno

Tabla 40. Propiedades de la clase Conversión de Variable o "VariableConversion" de oneM2M (oneM2M, 2019, pág. 43) (ETSI, 2021, pág. 44)

3. DISEÑO DE INVESTIGACIÓN

3.1 MARCO DE TRABAJO

Siendo el diseño de investigación según Hernández Sampieri et. al., el plan concebido para obtener la información que se desea, dicho marco de trabajo se adscribe a nivel de desarrollo de software al Proceso Unificado de Jacobson, Booch y Rumbaugh en sus fases de inicio, elaboración, construcción y transición; como también en cuanto a simulación del proyecto de tesis siguiendo a Shannon, dado que su definición de ser "el proceso de diseñar un modelo de un sistema real y realizar experimentos con este modelo con el fin de comprender el comportamiento del sistema y/o evaluar varias estrategias para el funcionamiento del sistema" es la más apropiada para alcanzar los objetivos propuestos (Hernández, et. al., 2006) (Jacobson, 1999) (Shannon, 1998).

La articulación entre el Proceso Unificado de desarrollo de software y el proceso de simulación se aprecia a continuación:

Proceso Unificado de Desarrollo de Software (Jacobson, et. al.)	Proceso de Simulación de Sistemas (Shannon R.)
Fase de Inicio.	Pasos 1 a 6 (Definición de problema, Planificación del proyecto, Definición del sistema, Formulación del modelo conceptual, Diseño experimental preliminar, Preparación de los datos de entrada).
Fase de Elaboración.	Pasos 7 a 8 (Modelo de traducción, Verificación y validación).

Fase de Construcción.	Pasos 9 a 10 (Diseño experimental final, Ejecución de la experimentación).
Fase de Transición.	Pasos 11 a 12 (Análisis e interpretación, Implementación y documentación).

Tabla 41. Articulación entre el Proceso Unificado de desarrollo de software de Jacobson, et. al. y el proceso de simulación de sistemas de Shannon (Elaborada por el autor)

El desglose de los pasos anteriores mencionados en la tabla, se observa como sigue:

3.1.1 Instancia de Inicio

Los pasos 1 a 6 del Proceso de Simulación de Sistemas (Shannon, 1998, pág. 9-10), conforman la fase de inicio del diseño de investigación (Jacobson et. al., 1999, pág. 327) cuya aplicación en la presente investigación se desglosan a continuación:

- **Paso 1: Planteamiento del problema:**

Según Shannon R. se ha de "definir claramente los objetivos del estudio para que sepamos el propósito, es decir, ¿por qué se estudia este problema y qué preguntas se espera responder?" (óp. cit.)

La situación atañe a la necesidad de gestionar la emisión de gases efecto invernadero – específicamente CO₂– en las ciudades, con el fin de potenciar las tareas de mitigación del calentamiento global y por ende, del cambio climático. Una de las maneras de llevar a cabo dicha gestión consiste en optimizar la eficiencia energética de los dispositivos inteligentes mediante la Internet de las Cosas IoT, ya que la mejora en el consumo apropiado de dichas máquinas en cuanto a su fuente de poder –la energía eléctrica generada por las centrales–, permitiría que tales complejos produjeran solamente la energía requerida para llevar a cabo las funciones de los dispositivos en mención, regulándose por defecto y en el tiempo la concentración de dióxido de carbono en el ambiente, ya que es conocido que son las centrales eléctricas las que arrojan al aire el CO₂ al generar la energía en el momento de la transformación del combustible fósil que se utiliza para su producción.

- **Paso 2: Planificación del proyecto**

De acuerdo con Shannon R., se ha de asegurar que se cuenta con el personal suficiente y adecuado, con el soporte de administración, con el hardware y con los recursos de software para hacer el trabajo (Shannon, 1998, pág. 9-10).

Dado que es un proyecto de investigación académica de maestría, el trabajo lo realiza individualmente el estudiante, capacitándose en las áreas de conocimiento necesarias para la concreción de la tesis, asumiendo los gastos de administración tanto fijos como variables en el tiempo, consiguiendo los recursos informáticos necesarios para realizar la fase experimental del proyecto tanto a nivel de tratamiento como de transferencia de datos.

- **Paso 3: Definición del sistema**

Atendiendo a Shannon R., se ha de determinar los límites y las restricciones que se utilizarán para definir el sistema (o proceso) e investigar cómo funciona el sistema (ibíd.).

El sistema extremo a extremo funciona a partir de un cliente (agente inteligente) consistente en un dispositivo autónomo que se conecta a internet o en su defecto, un componente electrónico que se conecta a una puerta de enlace, siendo dicha pasarela igualmente conectada a un servidor M2M mediante internet; por último y de manera opcional, una terminal (tableta, móvil, laptop, etc.; i.e. un agente de usuario o app) que se conecta mediante internet al servidor M2M para gestionar el consumo energético del cliente del extremo opuesto (id est, el agente inteligente).

Se pretende que el sistema prescindiera del concurso o intervención directa del ser humano en su funcionamiento, por ello el agente de usuario se ha de obviar en la medida de lo posible, ya que el agente inteligente (cliente) al ser una entidad software residente en una máquina y comunicarse con otra máquina que le ofrece un servicio o conjunto de los mismos (agente servidor) mediante una puerta de enlace (agente intermediario), pasarela o "gateway", lleva a cabo acciones a partir de la percepción de su entorno o como respuesta a los datos entregados al "hosting". Para este proyecto, la respuesta es optimizar la eficiencia energética del agente inteligente mediante la internet de las cosas, lo cual redundará en la gestión de CO₂ en las ciudades.

Dentro del amplio abanico de posibilidades para diseñar el sistema, se ha escogido el desarrollado hasta el momento por la alianza oneM2M, dado que su trabajo de generar especificaciones técnicas estandarizadas y de código abierto en TIC es de gran proyección a nivel mundial para establecer la interoperabilidad sintáctica y semántica entre los diversos agentes inteligentes, independientemente de su hardware y protocolos.

- **Paso 4: Formulación del modelo conceptual**

Shannon R, ha enunciado que se ha de desarrollar un modelo preliminar ya sea gráficamente (por ejemplo, diagrama de bloques o diagrama de flujo del proceso) o en pseudocódigo para definir los componentes, las variables descriptivas y las interacciones (lógica) que constituyen el sistema (Shannon, 1998, pág. 9-10).

El modelo conceptual del sistema para el proyecto de investigación –a manera de pseudocódigo– es el siguiente:

- **Entrada:** si el agente inteligente es un sensor, la entrada es un dato determinístico capturado por el mismo. En su defecto, si el agente inteligente es un accionador, la entrada es un dato estocástico generado por una tarea programada en el servidor M2M, o por un dato tomado por un sensor.
- **Proceso:** para un accionador, se registra la energía demandada para su activación durante el tiempo que ha de estar ejecutando su respectiva tarea. Para un sensor, se registra la energía demandada para permanecer activo en su tarea de percibir una señal.
- **Salida:** para un sensor, la salida es el envío de la señal al servidor y su respectivo registro en cuanto a energía demandada. Para un accionador la salida es la ejecución de la tarea respectiva con su correspondiente registro de la energía necesaria para llevar a cabo dicha labor.

Para que dicho modelo conceptual sea efectivo, se ha de garantizar la interoperabilidad sintáctica y semántica mencionada en el paso anterior, a partir de la definición de un corpus en el universo de discurso concerniente a la eficiencia energética de dispositivos inteligentes, que facilite la gestión de carbono atmosférico urbano. Dicho de otro modo, se requiere de la configuración de una ontología que funja como marco teórico interfuncional.

- ***Paso 5: Diseño Experimental Preliminar***

Según Shannon R., consiste en la selección de las medidas de efectividad a utilizar, los factores a variar y los niveles de esos factores a investigar, es decir, qué datos deben recopilarse a partir del modelo, en qué forma y en qué medida (ibídem).

El desarrollo del paso 5, se encuentra detallado en el inciso 4 de presente trabajo de investigación, dado que en el mismo se lleva a cabo la recopilación de las ontologías oneM2M, OM, SAREF4ENER y SAREF, en cuanto a sus enlaces respectivos a los recursos web en formato OWL o RDF/XML, con un fragmento de aquellos.

- ***Paso 6: Preparación de los datos de entrada***

De acuerdo a Shannon R., el paso 6 atañe a identificar y recopilar el insumo o valores de ingreso al sistema; es decir, los datos de entrada que necesita el modelo (ibíd.).

El despliegue de este paso (No. 6) del proceso de simulación de sistemas, se halla especificado en el inciso 5 de la presente tesis, a partir de la metodología de diseño de ontologías (Cifuentes, 2012) y soportada en el estudio pormenorizado en páginas anteriores de esta investigación, acerca de las ontologías oneM2M, SAREF4ENER, OM y SAREF.

3.1.2 Instancia de elaboración

Los pasos 7 y 8 del Proceso de Simulación de Sistemas (Shannon, 1998, pág. 11), establecen la fase de elaboración del diseño de investigación (Jacobson et. al., 1999, pág. 345), los cuales a continuación se detallan:

- ***Paso 7: Modelo de traducción***

Siguiendo a Shannon R. el paso No. 7 del proceso de simulación de sistemas, se ha de formular el modelo en un lenguaje de simulación apropiado (óp. cit.) o en su defecto, en un Lenguaje Unificado de Modelado como lo es el UML.

Es recomendable estructurar el modelo en UML, dado que el mismo facilita construir el sistema y representarlo de manera visual o gráfica, ante el hecho de que la ontología a proponer y simular, se compone de clases o conceptos, relacionados o asociados entre

sí, tanto a nivel de propiedades como de atributos en un dominio específico. Igualmente, UML permite el mapeo de las ontologías oneM2M, OM, SAREF4ENER y SAREF, las cuales se integran para conformar la ontología definitiva.

El desarrollo del paso 7, se encuentra detallado en el inciso 6 de presente trabajo de investigación, dado que en el mismo se lleva a cabo la combinación de las ontologías mencionadas bajo UML.

- ***Paso 8: Verificación y Validación***

De acuerdo con Shannon R., en el paso No. 8 del proceso de simulación de sistemas se confirma que el modelo funciona de la manera que pretende el analista (ibídem).

El desarrollo de este paso (No. 8), se encuentra detallado en el inciso 7 de presente trabajo de investigación, dado que una vez depurado el modelo ontológico que se observa en la figura 85, se ha de corroborar dicha configuración resultante (i.e., el modelo arquitectónico propuesto) es creíble y representativo de la salida del sistema real.

3.1.3 Instancia de construcción

Los pasos 9 y 10 del Proceso de Simulación de Sistemas (Shannon, 1998, pág. 12), son consecuentes con la fase de construcción del diseño de investigación (Jacobson et. al., 1999, pág. 367), cuya desagregación se aprecia acto seguido:

- ***Paso 9: Diseño experimental final***

Según Shannon R., en el paso No. 9 del proceso de simulación de sistemas se diseña un experimento que proporcione la información deseada y determine cómo se ejecutará cada una de las actividades de prueba especificadas en el diseño experimental preliminar (óp.cit.); i.e., el que corresponde paso 5 en la fase de inicio ya mencionado líneas atrás.

El despliegue de este paso (No. 9), se halla especificado en el inciso 8 de la presente tesis, a partir de la identificación de las clases del nivel inferior de la jerarquía oneM2M, y sus correspondientes asociaciones detalladas con las jerarquías SAREF4ENER, OM y

SAREF, de manera ídem; es decir, las clases o conceptos de sus niveles inferiores respectivos.

- ***Paso 10: La experimentación o simulación***

Siguiendo a Shannon R. el paso No. 10 del proceso de simulación de sistemas, se ha de ejecutar la experimentación para generar los datos deseados y realizar un análisis de sensibilidad (ibídem).

El desarrollo del paso 10, se encuentra detallado en el inciso 9 de presente trabajo de investigación, dado que en el mismo se efectúa la simulación a partir del validador de W3C para ontologías en OWL y/o RDF/XML, con sus respectivas tripletas del modelo ontológico propuesto incluyendo su grafo, los cuales arroja como resultado de su análisis gramatical (en inglés “parse”).

3.1.4 Instancia de transición

Los pasos 11 a 12 del Proceso de Simulación de Sistemas (Shannon, 1998, pág. 12-13), conforman la fase de transición del diseño de investigación (Jacobson et. al., 1999, pág. 382), cuya aplicación en la presente investigación se detalla a continuación:

- ***Paso 11: Análisis e interpretación***

Siguiendo a Shannon R., el paso No. 11 del proceso de simulación de sistemas, consiste en extraer inferencias acerca de los datos generados por las ejecuciones de la simulación o experimentación (óp. cit.).

El despliegue de este paso (No. 11), se halla especificado en el inciso 10 de la presente tesis, en el cual las deducciones conducen indefectiblemente a la optimización de la eficiencia energética de los agentes inteligentes mediante IoT y por ende, a la gestión de la emisión de CO2 urbano.

- ***Paso 12: Implementación y Documentación***

Según Shannon R., en el paso No. 12 del proceso de simulación de sistemas se ha de informar los resultados, disponerlos para su utilización, registrar los hallazgos y documentar el modelo con su implementación (ibídem).

El desarrollo de este último paso (No. 12), se halla especificado en el inciso 11 del presente trabajo de investigación, el cual consiste en la elaboración del documento escrito en cuanto a diseño, simulación, análisis, resultados, conclusiones, trabajos futuros.

4. DISEÑO EXPERIMENTAL PRELIMINAR

EL paso número 5 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), concierne al diseño experimental preliminar, el cual se refiere a la selección de las medidas de efectividad a utilizar, los factores a variar y los niveles de esos factores a investigar, es decir, qué datos deben recopilarse a partir del modelo, en qué forma y en qué medida.

Para efectos de lo anterior, se presenta a continuación en el lenguaje OWL las diversas ontologías que integran el modelo arquitectónico propuesto (oneM2M, SAREF4ENER, OM y SAREF). Dado la amplia extensión de las mismas, sólo se incluye un fragmento de cada una de ellas, acompañadas del enlace web correspondiente:

- **Ontología oneM2M**

Dirección web en formato OWL o RFD/XML (conocido como “en crudo” o “raw”):

https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl

Fragmento de ejemplo:

```
<Class
rdf:about="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#Thing">
  <rdfs:subClassOf>
    <Restriction>
      <onProperty
rdf:resource="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#hasThing
Property"/>
      <allValuesFrom
rdf:resource="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#ThingProp
erty"/>
    </Restriction>
  </rdfs:subClassOf>
  <rdfs:comment>A Thing in oneM2M (Class: Thing) is an entity that can be identified in the
oneM2M System. A Thing that is not a...
  </rdfs:comment>
</Class>
```

- **Ontología OM**

Dirección web en formato OWL o RFD/XML (en crudo):

<https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf>

Fragmento de ejemplo:

```
<owl:Class rdf:about="&om;Unit">
  <rdfs:label xml:lang="en">unit</rdfs:label>
  <om:alternativeLabel xml:lang="en">unit of measure</om:alternativeLabel>
  <om:alternativeLabel xml:lang="en">unit of measurement</om:alternativeLabel>
  <rdfs:comment xml:lang="en">A unit of measure is a definite magnitude of a quantity, defined and
adopted by convention or by law. It is used as a standard for measurement of the...
</rdfs:comment>
</owl:Class>
```

- **Ontología SAREF4ENER**

Dirección web en formato OWL o RFD/XML (en crudo):

<https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf>

Fragmento de ejemplo:

```
<owl:Class rdf:about="https://saref.etsi.org/saref4ener/LoadControlEventAction">
  <rdfs:label xml:lang="en">Load Control event action</rdfs:label>
  <rdfs:comment xml:lang="en">An action type used to express the action to be performed as a
consequence of an event used to send overload...</rdfs:comment>
</owl:Class>

<owl:ObjectProperty rdf:about="https://saref.etsi.org/saref4ener/...">
  :
  :
</owl:ObjectProperty>
```

- **Ontología SAREF**

Dirección web en formato OWL o RFD/XML (en crudo):

<https://saref.etsi.org/core/v3.1.1/saref.rdf>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Fragmento de ejemplo:

```
<owl:Class rdf:about="https://saref.etsi.org/core/Property">
  <rdfs:label xml:lang="en">Property</rdfs:label>
  <rdfs:comment xml:lang="en">A quality of a feature of interest that can be measured; an aspect
of a... >
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:about="https://saref.etsi..."/>
      </owl:allValuesFrom>
      :
      :
    </owl:Restriction>
  </rdfs:subClassOf>
  :
  :
</Class>
```


5. PREPARACIÓN DE DATOS DE ENTRADA

EL paso número 6 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), se refiere a la preparación de los datos de entrada; esto es, la identificación y recopilación de los valores de ingreso o insumo que necesita el modelo.

Dado que dichos datos hacen parte de la eficiencia energética (en inglés “energy efficiency”), entendiéndose esta como el consumo apropiado de potencia eléctrica para realizar las tareas –singulares o diversas– por parte del dispositivo (i.e. el gasto de energía), la configuración de su ámbito para optimizar dicho gasto atañe a la especificación de un modelo ontológico; el cual, siguiendo las pautas descritas por Cifuentes F. (2012), se describe a continuación.

1. Definición del propósito y el ámbito (Cifuentes, 2012)

1.1. ¿Sobre qué trata la ontología?

Trata acerca de la gestión de la emisión de CO₂ en las ciudades.

1.2. ¿Por qué se va a construir esta ontología?

Se pretende enlazar a la IoT cualquier dispositivo, independientemente de su protocolo de comunicación (MQTT, CoAP, o Rest), y poder regular su gasto energético.

1.3. ¿A quién está dirigida o quiénes serán los usuarios de la ontología?

A la comunidad de desarrolladores de dispositivos (electrónicos de consumo para domótica, edificios inteligentes, automoción, instrumentación, etc) y de protocolos de comunicación Máquina a Máquina (M2M).

2. Acotación del dominio (Cifuentes, 2012)

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

2.1. Definir el alcance, qué elementos se pueden modelar con los recursos disponibles y cuáles no es posible alcanzar.

No se modela el control de la máquina sino el seguimiento para reducir su gasto energético. Ello en razón a que cada fabricante, diseña los dispositivos (máquinas) con sus propiedades (tipo, autonomía y consumo energético, etc) y relaciones (protocolos de comunicación, suscripción y/o publicación, etc).

2.2. Intentar visualizar las fuentes de datos que permitirán poblar el conjunto de instancias acorde al modelo.

Los datos se obtienen directamente de los dispositivos al conectarse estos a la red e ir reportando su gasto energético según su desempeño.

2.3. Identificar los tipos de preguntas que se podrán responder con la ontología.

Mediante la ontología se dará respuesta a cuestiones como:

- Cuáles son los dispositivos que emiten mayor cantidad de CO₂ según su tipo y autonomía energética.
- Cuál es el consumo energético promedio por unidad de tiempo de un dispositivo específico.
- Qué cantidad excedente de CO₂ emite un dispositivo en particular.

3. Identificación de conceptos (Cifuentes, 2012)

3.1. Elaboración de un listado de conceptos presentes en el dominio de la ontología, tales como individuos o entidades (sujetos o clases) y acciones o relaciones (predicados o comportamientos).

- Es de tener en cuenta que las propiedades o atributos de los individuos corresponden a predicados de primer orden monádicos y/o poliádicos cuando dichos individuos se cuantifican. Así mismo, las acciones o las relaciones (o los eventos) realizadas, existentes (o sucedidos) por, entre (o a) las entidades, atañen a predicados de orden superior tanto monádicos como poliádicos al estimar la cuantía de los mismos (i.e. los sujetos).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Listado de conceptos: dispositivo (máquina), protocolo (de comunicación), consumo o gasto (energético), autonomía (energética), suscripción (a un servicio), publicación (de un servicio), Dióxido de Carbono (CO₂),
- El Internet de las Cosas (Internet of Things IoT) no es un concepto de la ontología, dado que el IoT es el dominio de la misma; i.e. su entorno, su orden, su ámbito, su ecosistema.
- El consumo o gasto energético de un dispositivo equivale a la cantidad de dióxido de carbono que se emite al ambiente por parte de dicha máquina para su funcionamiento (pergeñar sus tareas).

4. Clasificación de conceptos (Cifuentes, 2012)

4.1. Categorización de los conceptos mediante subconjuntos, a partir de aspectos comunes de los mismos. Un concepto puede figurar en más de un subconjunto.

- Dispositivo: móvil o celular, tableta, phablet, laptop, PC, reproductor mp3 o mp4, etc.
- Servicio: publicación, suscripción, etc.
- Protocolo: MQTT, CoAP, Rest, etc.
- GEI (Gas Efecto Invernadero): CO₂ (Dióxido de Carbono), CH₄ (Gas Metano), etc.
- Gasto energético:
- Autonomía energética:

4.2. Descripción de cada subconjunto.

- Dispositivo: máquina o equipo electrónico de consumo susceptible de conectar a la Internet.
- Servicio: prestación o asistencia ofrecida o demandada por un dispositivo.
- Protocolo: conjunto de reglas de comunicación acordadas entre dispositivos.
- GEI: fluidos gaseosos que arrojados a la atmósfera terrestre en grandes cantidades, inciden en el calentamiento global del planeta.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Gasto energético: consumo de energía eléctrica por parte de un dispositivo para realizar sus funciones.

- Autonomía energética: cantidad de energía eléctrica que puede almacenar un dispositivo para realizar sus funciones sin necesidad de recargarse de la misma.

4.3. Obtención de una lista inicial de conceptos con sus descripciones.

- Dispositivo: máquina o equipo electrónico de consumo susceptible de conectar a la Internet; v.g. móvil o celular, tableta, phablet, laptop, PC, reproductor mp3 o mp4, etc.

- Servicio: prestación o asistencia ofrecida o demandada por un dispositivo; v.g. publicación, suscripción, etc.

- Protocolo: conjunto de reglas de comunicación acordadas entre dispositivos; v.g. MQTT, CoAP, Rest, etc.

- GEI (Gas Efecto Invernadero): fluidos gaseosos que arrojados a la atmósfera terrestre en grandes cantidades, inciden en el calentamiento global del planeta; v.g. CO₂ (Dióxido de Carbono), CH₄ (Gas Metano), etc.

- Gasto energético: consumo de energía eléctrica por parte de un dispositivo para realizar sus funciones. Se asume como una propiedad del dispositivo.

- Autonomía energética: cantidad de energía eléctrica que puede almacenar un dispositivo para realizar sus funciones sin necesidad de recargarse de la misma. Se considera como un atributo de la máquina.

5. Identificación de relaciones (Cifuentes, 2012)

Reconocimiento de los enlaces o conexiones existentes entre pares de entidades o individuos.

Uno o varios dispositivos ofrecen o demandan uno o diversos servicios.

Uno o diversos dispositivos establecen o convienen uno o varios protocolos.

Uno o varios dispositivos emiten o arrojan uno o varios gases de efecto invernadero.

Uno o diversos dispositivos poseen o tienen un gasto energético.

Uno o varios dispositivos ostentan o poseen una autonomía energética.

6. Relacionamiento (Cifuentes, 2012)

6.1. Definición de Atributos o características (Propiedades de Datos).

6.2. Determinación de relaciones jerárquicas (Herencia múltiple y simple)

6.3. Establecimiento de asociaciones o comportamientos (Propiedades de Objeto)

7. Compilación (ibídem)

Integración o combinación de conceptos, relaciones o atributos.

8. Evaluación (ibíd.)

Verificación y corrección del modelo diseñado.

La aplicación de los pasos 6 a 8 con su respectivo refinamiento iterativo e incremental, dieron como resultado un modelo ontológico que se aprecia en la figura 85

Todo lo anterior se encuentra mapeado en una ontología diseñada como resultado y denominada como “OntologyModelingIoTCo2”, la cual a su vez asocia sus componentes con las ontologías oneM2M, SAREF4ENER, OM y SAREF en simultánea. El mapeo resultante de dicha estructura macro se observa en el inciso 6 “Modelo de traducción” bajo el modelado UML.

6. MODELO DE TRADUCCIÓN

El paso número 7 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), corresponde al modelo de traducción, el cual consiste en la formulación del modelo en un lenguaje de simulación apropiado o en su defecto, en un lenguaje unificado de modelado como lo es el UML.

Como ya se comentó en líneas atrás, es aconsejable estructurar el modelo ontológico en UML, dado su sencillez para la construcción del sistema y su correspondiente diagramación, ante el hecho de que la ontología a proponer y simular, se compone de clases o conceptos, relacionados o asociados entre sí, tanto a nivel de propiedades como de atributos en un dominio específico. Igualmente, UML permite el mapeo de las ontologías oneM2M, OM, SAREF4ENER y SAREF, las cuales se integran para conformar la ontología definitiva.

Todo lo correspondiente a la preparación de los datos de entrada detallados en el inciso anterior (paso No. 6 de Shannon), se encuentra mapeado en una ontología diseñada como resultado y denominada como “OntologyModelingIoTCo2”, la cual a su vez asocia sus componentes con las ontologías oneM2M, SAREF4ENER, OM y SAREF en simultánea. El mapeo resultante de dicha estructura macro se observa a continuación en UML:

Mapeo de las ontologías oneM2M, SAREF4ENER, OM y SAREF

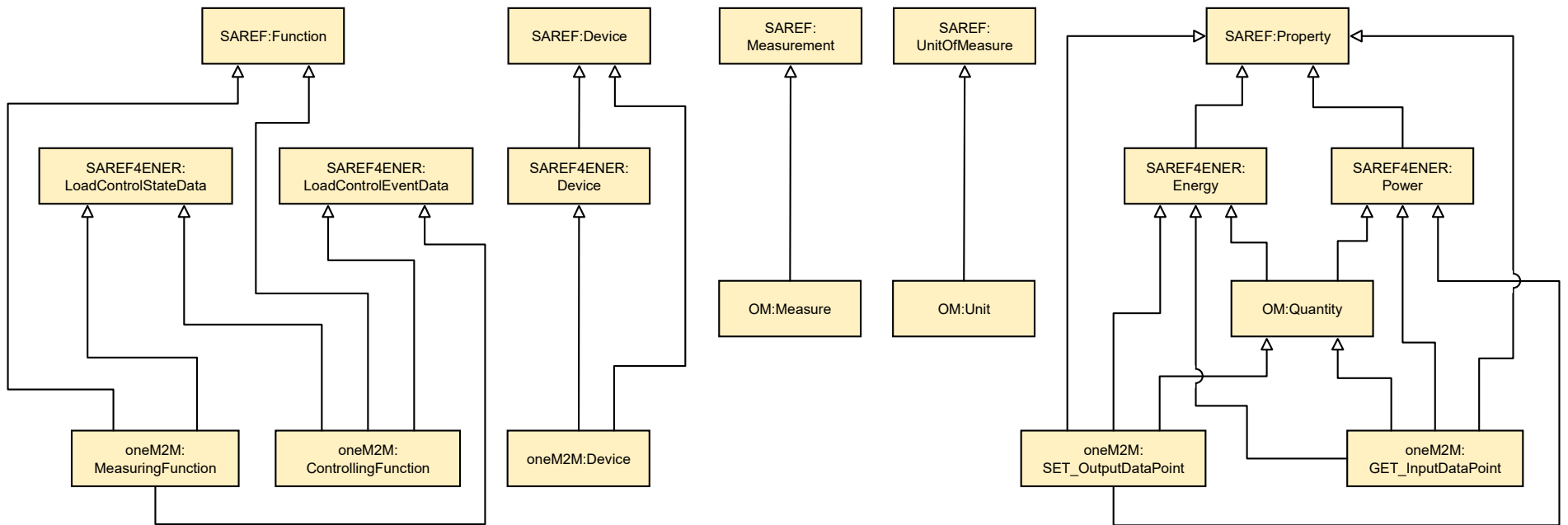


Figura 85. Estructura conceptual básica del mapeo ontológico propuesto (elaborada por el autor).

7. VERIFICACIÓN Y VALIDACIÓN

EL paso número 8 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), atañe a la verificación y validación; i.e., confirmar que el modelo funciona de la manera que pretendía el analista (depuración) y que la salida del modelo es creíble y representativa de la salida del sistema real:

El modelado de una arquitectura de la internet de las cosas para gestionar la emisión de carbono en las ciudades, atañe a la iniciativa o propuesta o especificación de implementar las ontologías oneM2M, SAREF, SAREF4ENER y OM, integrando las mismas de tal manera que sea factible optimizar la eficiencia energética de los dispositivos electrónicos con capacidad de comunicación a la red informática global, dado que la energía necesaria para que dichos artefactos realicen las tareas para las fueron diseñados, ya ha sido generada con antelación en las centrales respectivas y en su mayoría, a partir de fuentes no renovables, i.e. combustibles fósiles, los cuales de manera anticipada han arrojado el gas carbónico al ambiente en el momento de su ignición, incrementando por ende el efecto invernadero global con su cambio climático inherente.

A partir del estudio monográfico realizado en cuanto a las ontologías mencionadas, el modelo propuesto responde a tres preguntas esenciales:

- 1.) ¿Qué dispositivo va a ser gestionado en cuanto a su consumo energético? v.g., un interruptor, una luminaria, un termostato, etc.
- 2.) ¿Qué tarea estará realizando el dispositivo? i.e., una medición (mediante un sensor) o un control (a través de un activador).
- 3.) ¿Qué datos va a generar o a recuperar el dispositivo?
 - Si la tarea es medir, entonces el artefacto va a generar (producir) datos y entregarlos (sea a otro dispositivo o un usuario).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Si la tarea es controlar, entonces la máquina va a recuperar (obtener) datos y ejecutar la labor programada.

Dado que toda ontología se compone de conceptos o clases y las mismas se encuentran relacionadas mediante propiedades, dichas asociaciones entre clases permiten establecer una jerarquía en la cual, la noción de herencia -ya sea simple o múltiple- facilita la identificación de niveles en cuanto a superclases y subclases al interior de dicha categorización; por lo que, en el momento de efectuar la instanciación (individuos u objetos) de las clases -i.e., la creación de los dispositivos per se con sus peculiaridades-, lo más oportuno es realizar dichas instancias a partir de las subclases que se encuentre en el nivel inferior de la jerarquía.

Ante el hecho de que la jeraquía de clases que modela las ontologías combinadas entre sí, ha de satisfacer las preguntas formuladas anteriormente, se ha identificado que la ontología oneM2M presenta las clases que fungen como respuesta a los 3 interrogantes de la siguiente manera:

- Pregunta 1 (cuál dispositivo):
 - Se responde mediante la clase "oneM2M:Device".
- Pregunta 2 (tarea del dispositivo):
 - Se contesta con las clases "oneM2M:ControllingFunction" y "oneM2M:MeasuringFunction".
- Pregunta 3: (Datos a generar o recuperar):
 - Se responde a través de las clases "oneM2M:GET_InputDataPoint" y "oneM2M:SET_OutputDataPoint".

Las clases señaladas y que pertenecen a la ontología oneM2M, se encuentran en el nivel más bajo de la jerarquía respectiva, por lo que desde instancias de tales clases, se accede a todas las características y propiedades de cada una de las superclases que componen la jerarquía, independientemente del nivel en que se encuentren y del tipo de herencia, ya sea esta simple o múltiple.

Así mismo, al integrar la ontología oneM2M a las demás ontologías (SAREF, SAREF4ENER y OM), mediante la asociación con las subclases que exista equivalencia en cuanto a su rol y que además, se encuentren en el nivel inferior de sus respectivas

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

jerarquías, indefectiblemente se accederá también a los atributos y comportamientos propios de las superclases que constituyen a tales jerarquías y por ende, será posible gestionar la eficiencia energética de cualquier dispositivo electrónico enlazado a la red informática global, al poder registrar sus funciones, sus perfiles, sus magnitudes, sus mediciones, sus estados, sus servicios, sus eventos, sus consumos de potencia según ventanas de tiempo, sus operaciones, etc., todo ello de manera inmediata, sin retraso alguno; es decir, en caliente.

Las asociaciones entre las diversas ontologías a partir de las subclases del nivel más bajo de cada jerarquía, se listan a continuación a manera de "nombre_ontología:nombre_clase", indicando dichos niveles de jerarquía con el símbolo "➔" o lo contrario, señalando un nivel de jerarquía superclase con el signo "⬅":

- 1.) "oneM2M:Device" ➔ "saref4ener:Device"
- 2.) "oneM2M:Device" ➔ "saref:Device"
- 3.) "oneM2M:ControllingFunction" ➔ "saref4ener:LoadControlEventData"
- 4.) "oneM2M:ControllingFunction" ➔ "saref:Function"
- 5.) "oneM2M:ControllingFunction" ➔ "saref4ener:LoadControlStateData"
- 6.) "oneM2M:MeasuringFunction" ➔ "saref4ener:LoadControlEventData"
- 7.) "oneM2M:MeasuringFunction" ➔ "saref:Function"
- 8.) "oneM2M:MeasuringFunction" ➔ "saref4ener:LoadControlStateData"
- 9.) "oneM2M:SET_OutputDataPoint" ➔ ""saref:Property"
- 10.) "oneM2M:SET_OutputDataPoint" ➔ ""saref4ener:Energy"
- 11.) "oneM2M:SET_OutputDataPoint" ➔ ""saref4ener:Power"
- 12.) "oneM2M:SET_OutputDataPoint" ➔ ""OM:Quantity"

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- 13.) "oneM2M:GET_InputDataPoint" → "saref:Property"
- 14.) "oneM2M:GET_InputDataPoint" → "saref4ener:Energy"
- 15.) "oneM2M:GET_InputDataPoint" → "saref4ener:Power"
- 16.) "oneM2M:GET_InputDataPoint" → "OM:Quantity"
- 17.) "OM:Unit" → "saref:UnitOfMeasure"
- 18.) "OM:Measure" → "Saref:Measurement"
- 19.) "saref4ener:Energy" ← "OM:Quantity"
- 20.) "saref4ener:Power" ← "OM:Quantity"
- 21.) "saref4ener:Energy" → "SAREF:Property"
- 22.) "saref4ener:Power" → "SAREF:Property"

8. DISEÑO EXPERIMENTAL FINAL

EL paso número 9 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), atañe al diseño experimental final; i.e., el elaborar un experimento que proporcione la información deseada y determinar cómo se realizará cada una de las ejecuciones de prueba especificadas en el diseño experimental:

Para acceder a todas los conceptos de la ontología oneM2M, se identifican las clases que se encuentran en el nivel inferior de su propia jerarquía, las cuales se refieren a:

- Función de Medición (Class:MeasuringFunction).
- Funcion de Control (Class:ControllingFunction).
- Dispositivo (Class:Device).
- Establecer Punto de Datos de Salida (Class:SET_OutputDataPoint).
- Obtener Punto de Datos de Entrada (Class:GET_InputDataPoint).

A partir de las clases anteriores de la ontología oneM2M y al asociarsen las mismas como subclases a las ontologías SAREF4ENER, OM y SAREF, mediante las clases que se aprecian en la tabla 42 –configurándose estas últimas como superclases–, se garantiza el acceso absoluto a las propiedades y características (i.e., comportamientos y atributos) de todas las clases que componen cada una de las jerarquías pertenecientes a cada ontología.

(Ontología oneM2M) La Clase:	Es una subclase de la ontología		
	SAREF4ENER	SAREF	OM
Device	Device	Device	-----
ControllingFunction	LoadControlEventData	-----	-----
	-----	Function	-----

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	LoadControlStateData	-----	-----
MeasuringFunction	LoadControlEventData	-----	-----
	-----	Function	-----
	LoadControlStateData	-----	-----
SET_OutputDataPoint	Energy	-----	-----
	-----	Property	-----
	Power	-----	-----
	-----	-----	Quantity
GET_InputDataPoint	Energy	-----	-----
	-----	Property	-----
	Power	-----	-----
	-----	-----	Quantity

Tabla 42. Clases escogidas para ser asociadas desde la ontología oneM2M con las ontologías SAREF4ENER, OM y SAREF, acorde a sus respectivas jerarquías (Elaborada por el autor)

Por ende, cualquier instancia, individuo u objeto del que se desee gestionar su eficiencia energética, se asocia a alguna de las clases de la ontología oneM2M en cuanto a entidad (Dispositivo), ya sea en lo referente a controlar o medir (Función del Dispositivo) o por último, respecto a generar o recuperar datos (entrada o salida de los mismos), en consecuencia con los tres aspectos mencionados como interrogantes del modelo propuesto.

No obstante y como refrendación de la versatilidad y robustez de la ontología que integra a las ya mencionadas, en la tabla 43 se mapea con minuciosidad las clases de la ontología base oneM2M y las clases de las ontologías SAREF4ENER, OM y SAREF, con su correspondiente argumentación:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Clase de la ontología Base	Clase de la ontología mapeada	Argumentación del mapeo
oneM2M:Device (SubClase)	SAREF4ENER:Device (SuperClase)	En las tres ontologías, un Dispositivo (Device) se concibe como un artefacto que realiza una labor en específico –la cual se conoce con el nombre de función–, a partir de unos datos generados o recuperados. Lo anterior facilita asociar mediante herencia múltiple, a la subclase “Dispositivo” de la ontología oneM2M, con las dos superclases “Dispositivo”, ambas pertenecientes a las ontologías SAREF4ENER y SAREF respectivamente.
	SAREF:Device (SuperClase)	
oneM2M: ControllingFunction (SubClase)	SAREF4ENER: LoadControlEventData (SuperClase)	La labor de control que ha de efectuar un dispositivo, permite activar el mismo e incidir en el entorno. En dicho momento se ha de registrar la potencia que dicho artefacto produce, mediante el control de la carga de alguno de los eventos mencionados con sus datos respectivos, como del estado o condición en que se encuentre dicha máquina, lo cual depende de los datos que ingresen al mismo en el momento dado. Ello se logra asociando a través de herencia múltiple, desde la subclase “Función de Control” de la ontología oneM2M, hacia las superclases “Datos de Estado de Control de Carga” y “Datos del Evento de Control de Carga”, las cuales hacen parte de la ontología SAREF4ENER.
	SAREF4ENER: LoadControlStateData (SuperClase)	
	SAREF:Function (SuperClase)	La función de control se concibe como la capacidad de realizar una tarea por parte del dispositivo, en este caso de controlar, accionar o actuar en un ambiente en el cual dicho artefacto se encuentre inmerso. Por ende, asociar mediante herencia múltiple desde la subclase “Función de Control” de la ontología oneM2M, hacia la superclase “Función” de la ontología SAREF, permite acceder a las características y propiedades de esta última, relacionadas con tal facultad del dispositivo.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

<p>oneM2M: MeasuringFunction (SubClase)</p>	<p>SAREF4ENER: LoadControlEventData (SuperClase)</p>	<p>La tarea de medición que ha de realizar un dispositivo, permite activar el mismo para percibir el entorno. En dicho momento se ha de registrar la energía que dicho artefacto consume, mediante el control de la carga de alguno de los eventos mencionados con sus datos respectivos, como del estado o condición en que se encuentre dicha máquina, lo cual está supeditado a los datos que se generen en el mismo momento dado. Lo anterior se concreta a partir de la asociación de tipo herencia múltiple, desde la subclase “Función de Medición” de la ontología oneM2M, hacia las superclases “Datos de Estado de Control de Carga” y “Datos del Evento de Control de Carga”, las cuales hacen parte de la ontología SAREF4ENER.</p>
	<p>SAREF4ENER :LoadControlStateData (SuperClase)</p>	
	<p>SAREF:Function (SuperClase)</p>	<p>La función de medición se concibe como la capacidad de ejecutar una labor por parte del dispositivo, en este caso de percibir, medir o capturar en un ambiente en el cual dicha máquina se encuentra inmersa. Por lo tanto, asociar mediante herencia múltiple desde la subclase “Función de Medición” de la ontología oneM2M, hacia la superclase “Función” de la ontología SAREF, facilita el acceso a los atributos y comportamientos de esta última, relacionadas con dicha competencia del artefacto.</p>
<p>oneM2M: GET_InputDataPoint (SubClase)</p>	<p>SAREF4ENER:Energy (SuperClase)</p>	<p>El proceso de leer o recuperar los datos que ingresan a un dispositivo activador, conlleva a registrar tanto la energía que dicho artefacto consume, como la potencia que el mismo produce y en simultánea, a ejecutar la operación de entrada de tales datos en un momento o ventana de tiempo específica. Dicha rutina se alcanza a partir de la asociación de tipo herencia múltiple, desde la subclase “Obtener Punto de Datos de Entrada” de la ontología oneM2M, hacia las superclases “Energía” y “Potencia” pertenecientes a la ontología SAREF4ENER.</p>
	<p>SAREF4ENER:Power (SuperClase)</p>	
	<p>OM:Quantity (SuperClase)</p>	<p>El dato que es leído al momento de ingresar a un artefacto tipo accionador, atañe a un valor ya sea numérico, texto, booleano, etc., que resulta de una medida u operación respecto a una magnitud, en un universo de discurso preestablecido. Dicho valor se tipifica mediante la asociación de tipo herencia múltiple desde la subclase “Obtener Punto de Datos de Entrada” de la ontología</p>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		oneM2M, hacia la superclase “Cantidad” concerniente a la ontología OM.
	SAREF:Property (SuperClase)	La magnitud que es empleada por el dispositivo activador, corresponde a una propiedad física susceptible de ser medida, ya sea la energía consumida y/o la potencia producida por dicha máquina al momento de leer o recuperar los datos y realizar la tarea programada. Por ende, el asociamiento mediante herencia múltiple, desde la subclase “Obtener Punto de Datos de Entrada” de la ontología oneM2M, hacia la superclase “Property” de la ontología SAREF, facilita la cuantificación de la magnitud utilizada en el ingreso de datos al artefacto con el desencadenamiento respectivo de su acción.
oneM2M: SET_OutputDataPoint (SubClase)	SAREF4ENER:Energy (SuperClase)	El proceso de actualizar o generar los datos que se entregan por parte de un dispositivo sensor, conlleva a registrar tanto la energía que dicho artefacto consume, como la potencia que el mismo produce y en simultánea, a ejecutar la operación de salida de tales datos en un momento o ventana de tiempo específica. Dicha rutina se alcanza a partir de la asociación de tipo herencia múltiple, desde la subclase “Establecer Punto de Datos de Salida” de la ontología oneM2M, hacia las superclases “Energía” y “Potencia” pertenecientes a la ontología SAREF4ENER.
	SAREF4ENER:Power (SuperClase)	
	OM:Quantity (SuperClase)	El dato que es actualizado al momento de entregarse por parte de un artefacto tipo sensor, atañe a un valor ya sea numérico, texto, booleano, etc., que resulta de una medida u operación respecto a una magnitud, en un universo de discurso preestablecido. Dicho valor se tipifica mediante la asociación de tipo herencia múltiple desde la subclase “Establecer Punto de Datos de Salida” de la ontología oneM2M, hacia la superclase “Cantidad” concerniente a la ontología OM.
	SAREF:Property (SuperClase)	La magnitud que es empleada por el dispositivo sensor, corresponde a una propiedad física susceptible de ser medida, ya sea la energía consumida y/o la potencia producida por dicha máquina al momento de actualizar o generar los datos y realizar la tarea programada. Por ende, el asociamiento mediante herencia múltiple, desde la subclase “Establecer Punto de Datos de

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		Salida” de la ontología oneM2M, hacia la superclase “Property” de la ontología SAREF, facilita la cuantificación de la magnitud utilizada en la entrega de datos por parte del artefacto con el correspondiente desencadenamiento de su labor de percepción.
OM:Unit (SubClase)	SAREF:UnitOfMeasure (SuperClase)	La expresión correspondiente a la medición de una propiedad física o magnitud, es susceptible de clasificarse en alguna unidad fundamental o derivada del Sistema Internacional de Unidades SI. Dicha categorización se concreta a partir de la asociación de tipo herencia simple, desde la subclase “Unidad” de la ontología OM, hacia la superclase “Unidad de Medida” de la clase SAREF. Así mismo, al interior de la ontología OM, se presenta la asociación de tipo herencia simple entre la subclase “Cantidad” (Quantity) y la superclase “Unidad” (Unit). Similarmente, en la ontología SAREF existe la relación o Propiedad de Objeto (ObjectProperty) entre las clases “Medición” (Measurement) y “Unidad de Medida” (UnitOfMeasure). Ambas peculiaridades mencionadas facilitan el acceso a la totalidad de las características y propiedades de tales ontologías, ya que aquellas están disponibles desde la subclases “Establecer Punto de Datos de Salida” y “Obtener Punto de Datos de Entrada” de la ontología oneM2M, al asociarse con la subclase “Cantidad” de la ontología OM.
OM:Measure (SubClase)	SAREF:Measurement (SuperClase)	La comparación de una cantidad con su respectiva unidad, a fin de determinar una magnitud, facilita la medición de la misma, i.e., su dimensionamiento. Dicha labor de estimación se formaliza a partir de la asociación de tipo herencia simple, desde la subclase “Medida” de la ontología OM, hacia la superclase “Medición” de la clase SAREF. Adicionalmente, al interior de la ontología OM, se presenta la relación o Propiedad de Objeto (ObjectProperty) entre las clases “Medida” (Measure) y “Unidad” (Unit). De manera ídem, en la ontología SAREF existe la relación o Propiedad de Objeto (ObjectProperty) entre las clases “Medición” (Measurement) y “Unidad de Medida” (UnitOfMeasure). Ambas peculiaridades mencionadas facilitan el acceso a la totalidad de las características y propiedades de tales

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

		ontologías, ya que aquellas están disponibles desde la subclases “Establecer Punto de Datos de Salida” y “Obtener Punto de Datos de Entrada” de la ontología oneM2M, al asociarse con la subclase “Cantidad” de la ontología OM.
OM:Quantity (SubClase)	SAREF4ENER:Energy (SuperClase)	El consumo de electricidad por parte de un dispositivo electrónico que esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología oneM2M2, con las siguientes vinculaciones: - Mediante herencia múltiple, desde la subclase “Cantidad” de la ontología OM, hacia las superclases “Energía” y “Potencia”, las cuales hacen parte de la ontología SAREF4ENER. - A través de herencia simple, desde las subclases “Energía” y “Potencia” pertenecientes a la ontología SAREF4ENER, hacia la superclase “Propiedad”, la cual concierne a la ontología SAREF.
	SAREF4ENER:Power (SuperClase)	
SAREF4ENER:Energy (SubClase)	SAREF:Property (SuperClase)	
SAREF4ENER:Power (SubClase)		

Tabla 43. Detalle del mapeo y asociación de clases entre las ontologías oneM2M, SAREF4ENER, OM y SAREF, eligiendo a oneM2M como interfaz ontológica

9. SIMULACIÓN

EL paso número 10 del proceso de simulación de sistemas (Shannon, 1998, pág. 12), concierne a la experimentación; i.e., el ejecutar la simulación para generar los datos esperados o el resultado del modelado y así, realizar el estudio de sensibilidad que facilite comprender las incertidumbres, alcances y limitaciones del modelo arquitectónico propuesto, el cual se desarrolla como sigue:

A continuación se presenta en el lenguaje OWL (Ontology Web Language), la ontología de modelo arquitectónico propuesto denominado “OntologyModelingIoTCo2.owl”:

Dirección web en formato OWL o RFD/XML (en crudo o “raw”):

<https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl>

El código fuente completo de la ontología propuesta y desarrollado por el autor de la presente investigación (“OntologyModelingIoTCo2.owl”), se encuentra desplegado en el anexo 1. A continuación una fracción del mismo:

Fragmento de ejemplo:

```
<Class
rdf:about="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#ControllingFunction">
  <rdfs:subClassOf
rdf:resource="https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData"/>
  <rdfs:subClassOf rdf:resource="https://saref.etsi.org/core/v3.1.1/saref.rdf#Function"/>
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
<rdfs:subClassOf
rdf:resource="https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData"/>
  <rdfs:comment>La labor de control que ha de efectuar un dispositivo, permite activar el mismo e
incidir en el entorno... Ello se logra asociando a través de herencia múltiple, desde la subclase
'Función de Control' de la ontología oneM2M, hacia las superclases...</rdfs:comment>
</Class>
```

Al realizar la simulación en el validador de W3C (<http://www.w3.org/RDF/Validator>) de la ontología propuesta en formato OWL, se obtiene como resultado el grafo y las tripletas del modelo. Se presenta a continuación un fragmento tanto de la vista del grafo, como de las tripletas mencionadas. En el anexo 2 figuran la colección total de vistas del grafo y su vista general; en el anexo 3, se encuentra el cuadro completo de las tripletas RDF/XML.

Fragmento del grafo del modelo ontológico propuesto “OntologyModelingIoTCo2.owl”

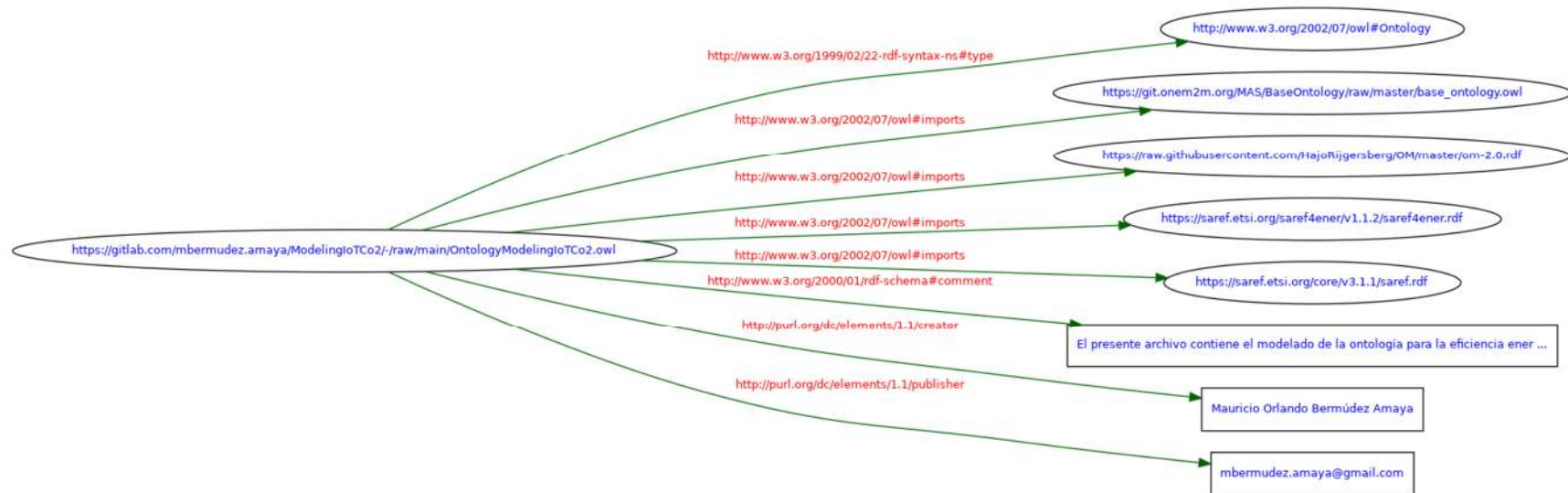


Figura 86. Fragmento del grafo de la ontología propuesta “OntologyModelingIoTCo2.owl” obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Fragmento de las Tripletas RDF de la ontología propuesta “OntologyModelingIoTCo2”

Num ber	Subject	Predicate	Object
1	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
2	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl
3	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf
4	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

5	<p>https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl</p>	<p>http://www.w3.org/2002/07/owl#imports</p>	<p>https://saref.etsi.org/core/v3.1.1/saref.rdf</p>
6	<p>https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl</p>	<p>http://www.w3.org/2000/01/rdf-schema#comment</p>	<p>"El presente archivo contiene el modelado de la ontología para la eficiencia energética de dispositivos electrónicos conectados a la red informática (Internet de las Cosas IoT), el cual facilita la gestión de dióxido de carbono en las ciudades. Dicho modelo hereda de manera múltiple, las características y propiedades de las ontologías OM (Ontología de Unidades de Medida), SAREF4ENER (Ontología de Referencia para Dispositivos Inteligentes en el Dominio de la Energía), SAREF (Ontología de Referencia para Dispositivos Inteligentes) y por último, oneM2M (Ontología Máquina a Máquina Monádica)"</p>
7	<p>https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl</p>	<p>http://purl.org/dc/elements/1.1/creator</p>	<p>"Mauricio Orlando Bermúdez Amaya"@es</p>
8	<p>https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl</p>	<p>http://purl.org/dc/elements/1.1/publisher</p>	<p>"mbermudez.amaya@gmail.com"@es</p>

Tabla 44. Fragmento de las tripletas RDF de la ontología propuesta "OntologyModelingIoTCo2.owl" obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

10. ANALISIS DE RESULTADOS

EL paso número 11 del proceso de simulación de sistemas (Shannon, 1998, pág. 12-13) consistente en el análisis e interpretación de resultados; conduce a la extracción de inferencias a partir de los datos generados por las ejecuciones de simulación.

La aplicación de dicha simulación o experimentación en referencia a la ontología propuesta denominada "OntologyModelingIoTCo2.owl", mediante el validador de W3C (<http://www.w3.org/RDF/Validator>), arrojó los siguientes resultados:

¿Qué se logró o qué se alcanzó al aplicar la simulación a la ontología propuesta?

De manera general, se obtuvo una consecuencia exitosa al compilar la ontología propuesta en formato OWL, dado que dicha validación no arrojó errores ni sintácticos ni gramaticales, permitiendo la generación del grafo y de las tripletas RDF.

En cuanto al archivo de código fuente denominado "OntologyModelingIoTCo2" y con extensión .OWL, se alcanzó lo siguiente:

- El código fuente es exacto y preciso sintáctica y gramaticalmente.
- Los espacios de nombres están bien definidos.
- Todo recurso web asociado a cada espacio de nombres se comprobó que estuviera disponible en la red informática.
- La ontología propuesta realiza la integración correcta de las demás ontologías a saber: oneM2M, SAREF4ENER, OM y SAREF.
- Al interior de la ontología propuesta, las subclases pertenecientes a oneM2M y que se encuentran en el nivel inferior de su jerarquía, se eligieron igualmente para que ellas mismas funjan como subclases de la ontología propuesta "OntologyModelingIoTCo2", quedando sin defectos al compilarse.
- La herencia múltiple configurada entre las subclases de oneM2M y las superclases de las ontologías SAREF4ENER, OM y SAREF es correcta.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- La definición de clases que funcionan como dominio al interior de cada una de las diversas ontologías, ha sido acertada.

En cuanto al grafo o diagrama de la ontología propuesta "OntologyModelingIoTCo2.owl", se logró lo siguiente:

- Todos los nodos coinciden con sus clases y datos respectivos del código fuente.
- Todas las aristas concuerdan con las asociaciones de herencia y de dependencia que figuran en código fuente.
- Las asociaciones de herencia múltiple que se observan en el grafo, son coherentes con el archivo de código fuente.
- Se comprobó el acceso a cada recurso web que figura en los nodos y las aristas, quedando todos de conformidad.
- Se contrastó cada nodo y arista del grafo con su respectiva clase, dato y asociación del código fuente en OWL, sin presentarse ningún defecto.

En lo referente a las tripletas RDF (unidades semánticas) de la ontología propuesta "OntologyModelingIoTCo2.owl", se alcanzó lo siguiente:

- Se codificaron en total 70 tripletas RDF
- Cada tripleta es consistente en cuanto al modelo de expresión de datos Sujeto–Predicado–Objeto del Marco de Descripción de Recursos RDF.
- No se presentó ningún nodo vacío (Sujeto u objeto desconocido), por lo tanto no fue necesario reemplazar por un ID generado "genid", lo cual se conoce como "skolemizar"
- Se comprobó el acceso a cada recurso web que figura en cada tripleta, tanto a nivel de Sujeto y Objeto, como de Predicado, quedando todos de conformidad
- Se contrastó cada tripleta en el grafo con su correspondiente nodo (Sujeto, Objeto) y arista (Predicado), sin presentarse faltantes o sobrantes.
- Se constató cada tripleta con su respectiva clase (Sujeto, Objeto), asociación (Predicado) y Dato (Objeto), sin ocurrir imperfecciones.

Se infiere como conclusión que el análisis de resultados ha arrojado un corolario favorable en la simulación realizada al modelo ontológico propuesto "OntologyModelingIoTCo2.owl".

Jerarquía de clases según ontologías oneM2M, SAREF4ENER, OM y SAREF

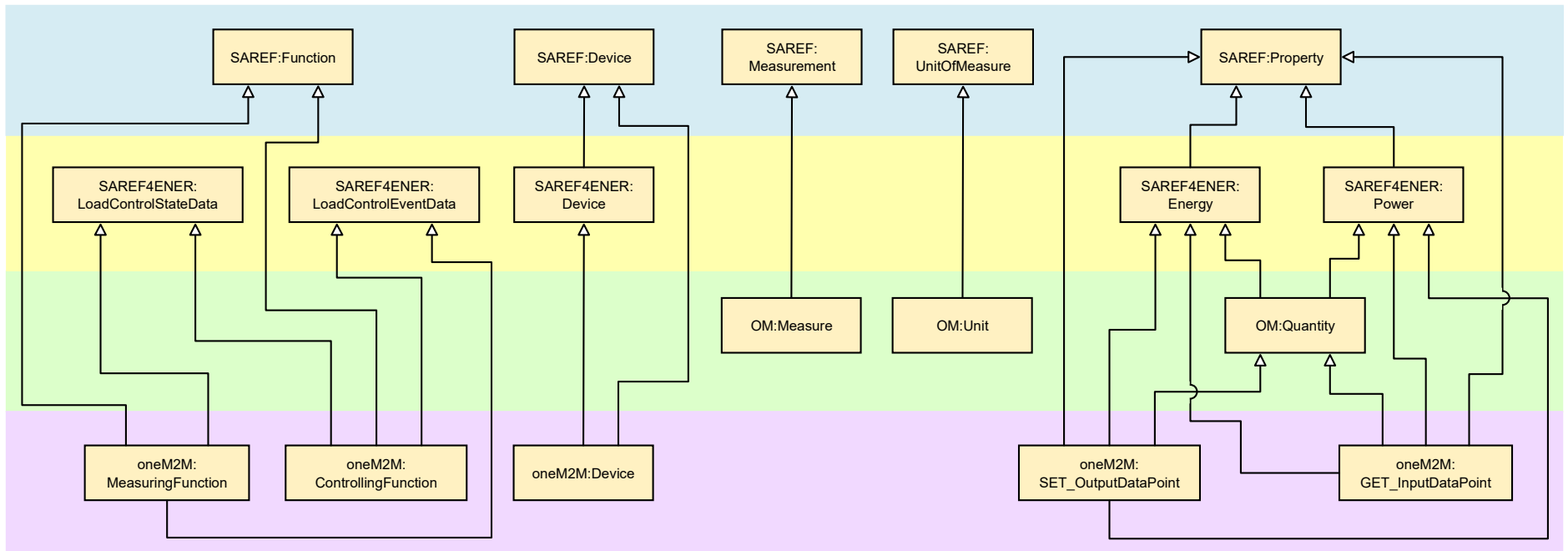


Figura 87. Jerarquía de clases según mapeo ontológico propuesto (elaborada por el autor).

11. CONCLUSIONES, TRABAJOS FUTUROS Y RECOMENDACIONES

EL paso número 12 del proceso de simulación de sistemas (Shannon, 1998, pág. 13), atañe a la implementación y documentación; i.e., la realización del informe acerca de los resultados, la puesta en marcha de los mismos, el registro de sus hallazgos y la elaboración escrita del trabajo efectuado con el modelo y su uso. Tales aspectos se listan como sigue:

11.1 CONCLUSIONES

- 1.) Por lo general en los países hispanohablantes, se hace uso de los anglicismos “smart” e “intelligent” de manera indistinta. No obstante, en el idioma inglés se establece la diferencia entre las expresiones “smart” e “intelligent” aplicadas a los dispositivos (appliances o devices). Según el diccionario Oxford (2017), el vocablo “smart” atañe al artefacto programado para ser capaz de una acción independiente; así mismo el término “intelligent” se refiere al artilugio apto para variar su estado o su accionar en respuesta a situaciones variables o experiencias pasadas.

Se puede inferir como conclusión que "smart" es a hardware como "intelligent" es a software; el primer alude a aspectos tecnológicos del dispositivo mientras el segundo concierne a elementos decisorios del mismo. Igualmente, "smart" se equipara con "perceptible" e "intelligent" se asemeja a "cognoscible". Por ende, el "smart appliance" responde a una instrucción dada (capturar una señal/mover un mecanismo); mientras el "intelligent device" toma una decisión respecto a un evento (regular la conducta del dispositivo según el valor de la magnitud medida). El dispositivo "intelligent" es capaz de aprender (machine learning); el artefacto "smart" ejecuta una tarea programada

(una y solo una). Por último, "smart" es a los atributos de un dispositivo; "intelligent" es a los comportamientos del mismo.

El "intelligent device" corresponde sin circunloquios, a un agente inteligente; el cual consiste en una entidad capaz de percibir su entorno, procesar tales percepciones y responder (actuar) en dicho ámbito de manera racional; es decir, de manera correcta y tendiendo a maximizar un resultado esperado, dado que la racionalidad no sugiere entendimiento como si lo hace la inteligencia (Russell y Norvig).

En consecuencia y como ejemplo, un dispositivo inteligente (smart appliance) atañe en su configuración básica a un circuito electrónico típico de led y pulsador, montado sobre un protoboard, el cual lleva a cabo una tarea particular de encendido/apagado ya sea por software o manualmente, siendo su estado reportado a todos los clientes que se encuentren suscritos para monitorear y/o actualizar el mismo.

Se deduce entonces que, el artefacto en cuestión (Smart appliance) no posee capacidad de decisión –de lo contrario sería un agente inteligente y por ende, programable–, haciéndose necesario conectarlo a una puerta de enlace (pasarela o "gateway", tal como una Placa Base tipo Raspberry, Arduino, etc.), la cual se encarga de llevar a cabo la función o funciones del dispositivo según se le programe.

En conclusión y para los propósitos del presente trabajo de investigación, un dispositivo inteligente (en inglés "Smart Appliance") se define como "un objeto tangible diseñado para llevar a cabo una tarea particular en hogares, oficinas o edificios públicos comunes y corrientes. Para lograr esta tarea, el dispositivo realiza uno o más funciones" (ETSI TS 103 264, 2017, pág 12) (Daniele, L. et. al., 2015, pág. 135) (oneM2M TS-0012, 2016, 2016, p. 75) (ETSI TS 118 122, 2016, pág 79).

- 2.) Al acceder a todas las características y propiedades de la jerarquía de clases de la ontología propuesta, a partir de las clases que se encuentran en el nivel inferior de toda la jerarquía (i.e., las subclases de oneM2M), se logra hacer uso eficiente de la herencia múltiple; i.e., se puede implementar las asociaciones de tipo agregación y composición.

La implementación de OWL para la ontología propuesta permite superar el problema del diamante propio de la herencia múltiple, a partir de la utilización de espacios de nombre (namespace, i.e., la nominación de cada ontología), la implementación de intersecciones o uniones (conocidas como interfaces en POO) y un modelado de datos eficiente.

Cuando el problema del diamante persiste, ello es indicativo de que el modelo de datos no satisface de manera estricta los requerimientos de tratamiento de los mismos, por lo que es necesario realizar un análisis y diseño de dicho modelo de manera exhaustiva, el cual a partir de una adecuada y conveniente recopilación, normalización, modelo y diccionario, se obtendrá un mapeo objeto-relacional apropiado, soportado en la jerarquía y modelo de clases con su correspondiente codificación y pruebas tanto unitarias (JUnit) como de integración.

Dado lo anterior se debe buscar el origen del problema en la normalización o en su defecto en la recopilación de los datos, lo cual refrenda que la toma de la muestra de datos no es lo suficientemente representativa de la población de los mismos, por lo que el procedimiento estadístico del cálculo de dicha muestra no fue el oportuno, corroborando que por parte de quien diseña el modelo (el analista del sistema o el arquitecto de software), existe desconocimiento acerca del universo de discurso respectivo; i.e. no comprende el ámbito del sistema a completitud.

3.) La ontología propuesta “OntologyModelingIoTCo2.owl” va muy de la mano con el porqué, el cómo, el qué y el para qué de la tecnología oneM2M; id est, la fundamentación, la composición, la conceptualización y la funcionalidad. Ello se describe a continuación:

- El porqué o fundamentación de oneM2M atañe a proveer una plataforma estándar común para M2M/IoT, que permita el interfuncionamiento (interworking) de dispositivos, sean o no inteligentes (smart appliances).
- El cómo o composición de oneM2M corresponde a la disponibilidad de un conjunto de especificaciones técnicas a manera de reportes en cuanto a arquitectura, requerimientos, modelos de dominio, protocolos, abstracción, semántica, seguridad y pruebas.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- El qué o conceptualización de oneM2M concierne a suministrar una solución de software intermediaria (middleware), que enlaza recursos de red y de aplicación.
- El para qué o rol (funcionalidad, papel, utilidad, etc.) de oneM2M se refiere al aprovisionamiento de interfaces de programación de aplicaciones APIs (funciones, rutinas, métodos, etc), que facilita la conexión entre dispositivos para que interactúen, independientemente de su tecnología subyacente.

Dado que la estructura lógica de oneM2M se compone de capas: de red, de servicio y de aplicación, la tecnología de oneM2M se enfoca en la capa de servicio; ofreciendo un conjunto esencial de interfaces de programación de aplicaciones APIs (Applications Programming Interfaces) para la interoperabilidad entre dispositivos, por lo que dicha capa se comporta como un "middleware" o proxy.

La utilización de oneM2M como software intermediario o "middleware", facilita la integración de tecnologías que no cumplen con el estándar global ofrecido (conocidas como non-oneM2M), que son por lo general sistemas heredados o ligeros a nivel LAN o PAN, efectuando la equivalencia de dichas plataformas con la de oneM2M en cuanto a ontologías.

La Implementación de oneM2M como una interfaz es pertinente, coherente y viable, dada que una de sus capas que conforman su arquitectura, es la capa de servicio, lo cual le permite establecerse como un estándar técnico universal para el interfuncionamiento de tecnologías tanto Máquina a Máquina M2M (Machine to Machine) como de la Internet de las Cosas IoT (Internet of Things). Dicho de otro modo, la capa de servicio le permite a oneM2M definir especificaciones técnicas que proporcionen el interfuncionamiento de artefactos mediante cualquier canal de comunicación remoto, ya sea a nivel de hardware o de software.

La ontología oneM2M es idónea para fungir como interface subyacente a todas las demás ontologías con las cuales se combina, en razón a su universalidad, su arquitectura e interoperabilidad sintáctica y semántica para conectar dispositivos de tecnología oneM2M como No-oneM2M.

Dicha universalidad de oneM2M como interfaz ontológica, permite enfocarse en el artefacto o máquina, en los datos o valores y por último, en la función o tarea a realizar con aquellos. No en balde es que, en el nivel inferior de la jerarquía de clases respectiva, se encuentran las subclases "Dispositivo" (Device), "Obtener Punto de

Datos de Salida” (GET_OutputDataPoint), “Asignar Punto de Datos de Entrada” (SET_InputDataPoint), “Función de Control” (ControllingFunction) y “Función de Medición” (MeasuringFunction).

- 4.) La ontología “OntologyModelingIoTCo2” propone que no se implemente a oneM2M como una supra-ontología, ni se intente mapear la mayoría cantidad de clases y propiedades. No se pretende la homologación de marcos conceptuales; al contrario, se busca integrarlos y que oneM2M funja como una infra-ontología para que, a partir de la identificación de las clases que se encuentran en el nivel inferior de cada ontología, sea factible acceder a todas las características y comportamientos de las mismas combinadas.

El modelo ontológico propuesto es una integración o combinación de ontologías que están consolidadas; no pretende convertirse en un nuevo corpus semántico de un determinado universo de discurso, adicional a los ya existentes. Solamente procura constituirse en un recurso alternativo para la gestión de la emisión de CO2 en las ciudades, soportado en la Internet de las Cosas.

Existe en el mundo de la informática una tendencia a implementar a oneM2M como un metamodelo; es decir, como un perfil, configuración o esquema teórico de un dominio en específico y del cual, se desprendan otros modelos o en su defecto, un marco conceptual textual y gráfico que reúna a diversos modelos ya existentes, ya probados y en uso.

Ante la miríada de definiciones acerca de lo que es una ontología, me permito formular de manera sucinta que una ontología es un corpus semántico de un tema particular. Es un sistema conceptual concreto. Es un ejercicio hermenéutico preciso. Es un modelo conceptual esencial de un determinado universo de discurso.

- 5.) Dada la versatilidad, robustez, idoneidad y respaldo que distinguen a las ontologías oneM2M, SAREF4ENER, OM y SAREF, fueron estas razones más que suficientes

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

para ser escogidas en la conformación de la ontología propuesta de gestión de CO2 urbano mediante IoT "OntologyModelingIoTCo2".

No obstante, dicha versatilidad ha de manejarse con precaución, dado que las ontologías que se han elegido e integrado (oneM2M, SAREF4ENER, OM y SAREF) han sido diseñadas con tanta flexibilidad, que se presenta el evento de pérdida de claridad y concisión; paulatinamente van dejando la exactitud y la precisión, van careciendo de rigurosidad a medida que se les van añadiendo propiedades de objeto para que su utilidad sea holística, lo cual las perjudica en cuanto a su comprensión y por ende, su aplicación para superar los problemas de comunicación abierta y estandarizada, lo cual puede llegar a conducir las a la impracticabilidad.

6.) La implementación como servicio de un Gestor de Energía de Cliente (Customer Energy Manager CEM), facilita el control o administración de la eficiencia energética de un dispositivo y por ende, la gestión de la emisión de CO2 en las ciudades, al registrar en caliente el consumo/producción de energía/potencia. Lo cual en asocio con la inherente interoperabilidad sintáctica y semántica que ofrece la ontología oneM2M y camino a la optimización, aportará a la mitigación del calentamiento global. Dicho de otra manera, si se controla el gasto energético de los dispositivos inteligentes soportado en la Internet de las Cosas, se gestiona por defecto la emisión de carbono atmosférico urbano.

7.) La validación de la ontología propuesta en formato OWL "OntologyModelingIoTCo2", arrojó unas tripletas consistentes en cuanto a Sujeto – Predicado – Objeto, denotando que las entidades de Sujeto y Objeto quedaron correctamente definidas en lo referente a los recursos web que enlazan, ya fuesen clases o datos. Igualmente, respecto a la entidad Predicado, también quedaron determinadas con exactitud y precisión en lo concerniente a la propiedad de objeto o relación que vinculan.

El grafo del modelo ontológico propuesto arrojado por la validación, se destaca por su coherencia esquemática en lo referente a los nodos origen (Sujeto, Clase), las aristas

(Predicado, Propiedad) y los nodos destino (Objeto, Clase o Dato), los cuales permiten constatarlo mediante las tripletas.

No existen nodos vacíos en el grafo de la ontología propuesta, al realizar la validación del código fuente en formato .owl; por lo tanto, no fue necesario reemplazar dichos nodos por un IRI que posea un ID generado –genid– por el validador. Dicho de otro modo, no fue necesario efectuar alguna “skolemización”.

8.) Los resultados arrojados por la experimentación, permiten corroborar que el modelo ontológico diseñado en los pasos 6 y 9 del proceso de simulación de un sistema de Shannon, es el adecuado y conveniente para gestionar la emisión de carbono atmosférico urbano a partir del modelado de una arquitectura de la Internet de las Cosas, al registrar y controlar el consumo/producción de energía/potencia por parte de un artefacto electrónico con conectividad a la red informática; i.e., un dispositivo inteligente. He por ello que en la figura 87 se refrenda el diagrama de la ontología propuesta (OntologyModelingIoTCo2), resaltando con bandas de colores a las ontologías integradas (oneM2M, SAREF4ENER, OM y SAREF).

11.2 TRABAJOS FUTUROS

Los siguientes trabajos futuros a plantear, son un complemento de las conclusiones descritas en el inciso 11.1 del presente trabajo de investigación, por lo que se sugiere poner en obra las mismas a partir de la validación del modelo ontológico para la gestión del carbono atmosférico en las ciudades implementando la Internet de las Cosas:

Un primer trabajo futuro atañe a la instalación y configuración de un servidor M2M para acceder a la arquitectura oneM2M, dado que la tecnología M2M consiste según el Lawrence Berkeley National Laboratory LBNL, en un sistema “que permite a las computadoras, procesadores integrados, sensores inteligentes, actuadores y dispositivos

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

móviles comunicarse entre sí, tomar medidas y tomar decisiones, a menudo sin intervención humana” (2004, pág. 1), al igual que lo define la European Telecommunications Standards Institute ETSI como una configuración “para adquirir datos de un sistema físico a gran escala y controlarlo a cambio, siendo los sensores y actuadores compartidos y distribuidos como puntos de monitoreo y control a través de dicha infraestructura” (ETSI TR 101 584, 2013, pág. 12) Por lo tanto, con una arquitectura Cliente-Servidor y Orientada a Servicios SOA se establece el sistema Máquina a Máquina en la que, el equipo informático encargado de procesar las peticiones y respuestas de los dispositivos (inteligentes o no) mediante la puerta de enlace, es un servidor de aplicaciones para gestionar la lógica de negocio y el acceso a datos.

Un segundo trabajo futuro concierne a la preparación de una puerta de enlace Máquina a Máquina o “Gateway M2M”, el cual consiste en un equipo informático que funge como una interfaz entre los dispositivos electrónicos –sean o no, inteligentes– y el servidor M2M a través del ciberespacio. Dicha pasarela se compone de los recursos mínimos y necesarios para su funcionamiento tanto a nivel de hardware (CPU, RAM, ROM, Flash, fuente de poder y puertos de comunicación) como de software (Sistema Operativo, controladores y aplicaciones). El más apropiado y conveniente para desempeñarse por sí mismo, es el computador de placa sencilla (Single Board Computer SBC), cuya arquitectura ha sido desarrollada por diversas organizaciones (Arduino, Raspberry, BeagleBoard, Gumstix, etc).

Existe predilección en el mundo de la informática, elegir una placa base sencilla SBC como puerta de enlace M2M; ello en razón a su bajo costo, buen rendimiento en entornos exigentes, sistema operativo de código abierto, bajo consumo de energía y versatilidad de implementación (educación, industria, prototipos, etc.). Así mismo, Su configuración como “gateway M2M” se efectúa mediante un contenedor “open source”, acompañado de una API de entrada y salida de datos orientada a objetos para gestionar la plataforma de la SBC, con integración nativa de bajo nivel y supervisión de interrupciones, lo que facilita concentrarse en el objetivo de la pasarela: desempeñarse como conexión física y lógica entre el servidor máquina a máquina y el dispositivo electrónico para su correspondiente gestión (monitoreo y/o accionamiento).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

Un tercer trabajo futuro se orienta al montaje de un dispositivo inteligente M2M tipo “smart appliance”, el cual consiste en un circuito electrónico típico de led y pulsador montado sobre un protoboard, el cual lleva a cabo una tarea particular de encendido/apagado ya sea por software o manualmente, siendo su estado reportado a todos los clientes que se encuentren suscritos para monitorear y/o actualizar el mismo.

Dado que el artefacto en cuestión no posee capacidad de decisión (de lo contrario sería un agente inteligente y por ende, programable), es necesario conectarlo a una puerta de enlace (pasarela o “gateway” tipo SBC), la cual se encarga de llevar a cabo la función o funciones del dispositivo según se le programe.

Un cuarto trabajo futuro consistiría en la configuración del dominio de eficiencia energética, entendiéndose la misma (en inglés “energy efficiency”), como el consumo apropiado de potencia eléctrica para realizar las tareas –singulares o diversas– por parte del dispositivo (i.e. el gasto de energía), la configuración de su ámbito para optimizar dicho gasto atañe a la especificación de un modelo ontológico (OntologyModelingIoTCo2) que facilite la gestión de la emisión de CO₂ en las ciudades mediante la Internet de las Cosas, el cual integre las ontologías oneM2M, SAREF4ENER, OM y SAREF.

Un quinto trabajo futuro consistiría en el registro del consumo energético del dispositivo inteligente, tanto de manera típica (sin restricciones) como eficiente (restringido) y en diversas unidades, tales como kilovatios hora (kVh), kilotoneladas de petróleo equivalente (kToe), gigatoneladas de dióxido de carbono equivalente (GtCO₂e), etc.

Un sexto trabajo futuro atañe a la optimización de la eficiencia energética, al articular el Gestor de Energía de Cliente, el servidor M2M con las API oneM2M de servicio, la puerta de enlace con las API tanto de proxy como de contenedores oneM2M y por último, el dispositivo (inteligente o no), todo ello en la red informática, para administrar el control de

carga y las series de potencia de dicho dispositivo al consumir/producir la energía/potencia que requiere para llevar a cabo su labor de percepción/activación; redundando ello en la gestión del CO2 atmosférico urbano.

11.3 RECOMENDACIONES

Las siguientes sugerencias o recomendaciones son un aditivo de los trabajos futuros descritos en el inciso 11.2 del presente trabajo de investigación, por lo que se propone tenerlas en cuenta ante la puesta en marcha de cualesquiera de aquellos. A continuación se describen:

- 1.) Efectuar un diseño experimental preliminar en el que los datos se recopilen por demanda de energía (cantidad), tanto en modo normal como en modo mejorado. Una manera de levantar tales datos es registrando en un servidor M2M dicha cantidad de energía según dicha demanda, ya sea en vatios-hora, en julios o en Toe.
- 2.) Preparar los datos de entrada al sistema registrando en un servidor M2M, tanto los datos referentes al dispositivo, como los concernientes a la marca de tiempo inicial, de tiempo final y por último, de cantidad consumida y/o producida de energía y/o potencia.
- 3.) Formular el modelo de traducción considerando el protocolo de comunicación de datos no solo entre los dispositivos; también con la puerta de enlace y el servidor M2M, dado que se hace uso de la red informática. Para una eventual comunicación entre el servidor y un agente de usuario, tener en cuenta el uso del protocolo HTTPS.
- 4.) En lo referente a la verificación y validación del sistema, Considerar la autenticidad y la fiabilidad de los datos almacenados en el servidor M2M, al igual que los del Gestor de Energía del Cliente CEM.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

5.) En la elaboración de un diseño experimental final, tener en cuenta que se han de hacer diversas simulaciones por rangos de tiempo, ya sea manejando unidades en segundos o en minutos, tanto en modo normal como en modo mejorado. Ello con el propósito de corroborar que sí se logra optimizar la eficiencia energética mediante el modelo ontológico “OntologyModelingIoTCo2”. Igualmente, se sugiere una cantidad de experimentos que conformen una muestra representativa, la cual se ha de inferir estadísticamente.

6.) En la realización de la simulación o experimentación, se sugiere llevarla a cabo en simultánea con el agente inteligente, la puerta de enlace y el servidor M2M y el Gestor de Energía de Cliente CEM, registrando los datos en un servidor de datos para su posterior análisis.

7.) En lo que respecta al análisis e interpretación de los resultados arrojados por la simulación del sistema, las inferencias deben conducir indefectiblemente a la optimización de la eficiencia energética de los agentes inteligentes mediante IoT y por ende, a la gestión de la emisión de CO2 urbano.

12. CONTRASTACIÓN DE OBJETIVOS, HIPÓTESIS Y APORTES ORIGINALES

A continuación se realiza la verificación o constatación tanto de los objetivos de la investigación, como de la hipótesis formulada, además de las contribuciones o aportes originales a saber.

12.1 CONTRASTACIÓN DE OBJETIVOS

En cuanto al primer objetivo "Identificar dentro los diversos tipos de arquitectura de la Internet de las cosas existentes, el más apropiado para el modelo a proponer", la discusión y conclusión de las arquitecturas funcionales que figuran en los incisos "2.1.10" y "2.1.11" de la presente tesis, permiten comprobar el cumplimiento de dicho propósito; además de que el mismo quedó sustentado en la publicación del artículo "Functional architectures for the Internet of Things: fundamental models", cuyo DOI es: <http://doi.org/10.24247/ijmperdddec202071>, divulgado en la revista "International Journal of Mechanical and Production Engineering Research and Development". Vol. 10, Num. 6 (dic. 2020), Págs. 655–672, siendo el autor de la presente investigación, igualmente el primer autor de dicho artículo científico.

En lo atinente al segundo objetivo "Determinar al interior de los diversos modelos ontológicos de base Máquina a Máquina existentes, el más adecuado y conveniente para la arquitectura a plantear", tanto la discusión como la conclusión acerca de las ontologías M2M que figuran en los incisos "2.2.5" y "2.2.6" de la presente trabajo de investigación, facilitan la corroboración del cumplimiento del fin último en cuestión; además de que el

mismo quedó sustentado en la publicación del artículo "Ontological Base Models Machine-To-Machine M2M Applied to the Internet of Things IoT", cuyo DOI es: <https://doi.org/10.36260/rbr.v10i12.1576>, divulgado en la revista "Revista Boletín Redipe". Vol. 10, Num. 12 (dic. 2021), Págs. 148–161 (Red iberoamericana de pedagogía), siendo el autor de la presente tesis, igualmente el primer autor de dicho artículo científico.

En lo alusivo al tercer objetivo "Reconocer de manera detallada las ontologías que conforman el modelo propuesto, las cuales han de corresponder tanto al universo de discurso Máquina a Máquina M2M como a la Internet de las Cosas IoT y a su vez, con arreglo al dominio de los dispositivos inteligentes y la optimización energética", la revisión monográfica exhaustiva de las diferentes ontologías que se describen en los incisos "2.3.3" a "2.3.6" de la presente tesis, permiten constatar la cumplimentación de dicho propósito; las cuales aunadas en cuanto a la conceptualización de dispositivo inteligente (ontología SAREF), producción/consumo de potencia/energía (Ontología SAREF4ENER), unidades de medida (ontología OM) y por último, interoperabilidad sintáctica y semántica Máquina a Máquina (M2M) sobre la Internet de las Cosas (IoT) -i.e., la ontología oneM2M-, facilitan la iniciativa de un modelo arquitectónico funcional flexible para la gestión de CO2 en las ciudades. Igualmente, en el inciso 4. "Diseño experimental preliminar", se complementa este fin último al recopilar en crudo las ontologías anteriores en formato OWL y RDF/XML.

En lo que concierne al cuarto objetivo "Definir el modelo arquitectónico funcional flexible IoT a proponer, a partir de la combinación de ontologías M2M que facilitan la gestión de CO2 atmosférico emitido por el uso de dispositivos inteligentes", con los incisos 5 y 6, "Preparación de los datos de entrada" y "Modelo de traducción" respectivamente y que figuran en la presente tesis, se comprueba la realización de dicho propósito; dado que se logra configurar el ámbito de la ontología integrada que sugiere o procura la eficiencia del gasto energético para ejecutar las tareas por parte de un dispositivo, al igual que el mapeo de la ontología resultante que funge como arquitectura funcional flexible al asociar

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

los componentes ontologías oneM2M, SAREF4ENER, OM y SAREF en simultánea, cuya estructura conceptual básica se observa en la figura 85.

En cuanto al quinto objetivo "Establecer el diseño experimental del modelo arquitectónico funcional flexible propuesto, que facilite su construcción y la comprensión de su comportamiento", mediante el inciso 8 "Diseño experimental final", que forma parte de la presente tesis, se constata la satisfacción de dicha meta; puesto que se identifican las clases que se ubicarán en nivel inferior de la jerarquía perteneciente a la ontología combinada (i.e., Dispositivo, Función de Control, Función de Medición, Establecer Punto de Datos de Salida, Obtener Punto de Datos de Entrada ["Device", "ControllingFunction", "MeasuringFunction", "SET_OutputDataPoint", "GET_InputDataPoint"]), junto con el mapeo y asociación correspondiente para garantizar el acceso omnimodo a las características y propiedades de todas las clases que componen cada una de las demás ontologías integradas (oneM2M, SAREF4ENER, OM, SAREF).

Respecto al sexto objetivo "Validar el modelo de arquitectura funcional propuesto mediante una simulación del mismo, acorde a las especificaciones oficiales de la W3C que permitan certificarlo", por medio del inciso 9 "Simulación", el cual hace parte del presente trabajo de investigación, se corrobora el cumplimiento del fin último en cuestión; ello en razón a la construcción de la ontología planteada en formato OWL, que combina las ontologías oneM2M, SAREF4ENER, OM y SAREF, la cual quedó certificada al simular la misma en el validador oficial de la World Wide Web Consortium W3C <http://www.w3.org/RDF/Validator>, obteniendo las tripletas RDF/XML y el grafo correspondientes satisfaciendo la especificaciones de la W3C (tripletas sin nodos vacíos), siendo disponible dicha ontología en la dirección <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl>, e igualmente en el anexo 1 de esta tesis.

12.2 CONTRASTACIÓN DE HIPÓTESIS

Es de tener en cuenta que, como lo enuncia Hernández Sampieri et. al. (2006, p. 142), una hipótesis no se prueba en cuanto a su veracidad o falsedad; en consecuencia un hipótesis se rechaza o se admite en razón a las evidencias aportadas a su favor. Mas dicha admitancia no es sinónimo de aceptación o acatamiento; es solamente una muestra de significancia de la misma.

Tal y como se describió en el apartado 3.1, "Marco de trabajo" perteneciente al inciso 3. "Diseño de investigación" de la presente tesis, en el que se articula la instancia o fase de construcción del Proceso Unificado de Desarrollo de Software (Jacobson et. al., 1999, pág. 367), con el Proceso de Simulación de Sistemas (Shannon, 1998, pág. 12) y sus correspondientes pasos 9 a 10 (i.e., el Diseño experimental final, acompañado de la Ejecución de dicha experimentación), son estos ítems los que permiten corroborar que el modelo arquitectónico funcional flexible propuesto ES VALIDADO por la simulación del sistema, sustentado en el corpus semántico combinado de ontologías para la gestión de la emisión de CO2 atmosférico por parte de dispositivos inteligentes. Por ende, se infiere como conclusión que la hipótesis de trabajo (Hi) es verdadera; o dicho de otro modo, la hipótesis de investigación se admite en razón a las evidencias aportadas a su favor. Igualmente, en el caso de que la hipótesis nula (Ho) hubiese sido la verdadera, el modelo ontológico tendría que fungir como un metamodelo, lo cual iría en contraposición con la flexibilidad de la arquitectura funcional y en consecuencia, la combinación de las ontologías de referencia para dispositivos inteligentes no facilitarían la versatilidad para la gestión de la emisión de CO2 atmosférico de la energía y potencia consumida/producida por los mismos a partir de combustibles fósiles.

12.3 APORTES ORIGINALES

1.) La iniciativa de la arquitectura funcional flexible a partir de un modelo ontológico de interoperabilidad semántica y sintáctica.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

2.) La integración de ontologías que permiten la gestión de la eficiencia energética para construir una propuesta o especificación de gestión del CO2 atmosférico urbano.

3.) La traducción al idioma español de las ontologías a combinar, dado que no existe material al respecto en idioma castellano.

4.) La identificación de las clases y/o conceptos de las diversa ontologías a integrar SAREF4ENER, OM y SAREF, para ser mapeadas desde la ontología oneM2M.

5.) La iniciativa de que la ontología oneM2M funga como un inframodelo integrador y no como un metamodelo, siendo esto último lo que se acostumbra a plantear.

6.) La construcción de la ontología propuesta en formato OWL, que combina las ontologías oneM2M, SAREF4ENER, OM y SAREF, disponible de manera pública en la dirección <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl>, al igual que en el anexo 1 de la presente tesis.

7.) La validación de la ontología propuesta en OWL, a partir de las tripletas RDF/XML y el grafo obtenidos de la simulación en el validador oficial de la World Wide Web Consortium W3C <http://www.w3.org/RDF/Validator>, las cuales se hallan en los anexos 2 y 3 de la presente tesis.

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

8.) El cumplimiento riguroso de las especificaciones de la W3C en OWL para la ontología propuesta, lo que garantiza el ámbito global y disposición pública de la misma, evitando con ello los nodos en blanco al interior del grafo y tripletas RDF/XML; de lo contrario, se correría el riesgo de tener que reemplazarlos por IRIs que posean un ID generado por el validador (i.e., "skolemizar"), lo cual limitaría el modelo de arquitectura funcional flexible a un alcance local.

13. REFERENCIAS

- Grønbaek, Inge. (2008). *Architecture for the Internet of Things (IoT): API and interconnect*. The Second International Conference on Sensor Technologies and Applications. Telenor R&I.
- Duan, Ren. Chen, Xiaojiang. Xing, Tianzhang. (2011). *A QoS architecture for IoT*. IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing. DOI 10.1109/iThings/CPSCoM.2011.125.
- Jin, Jiong. Gubbi, Jayavardhana. Luo, Tie. Palaniswami, Marimuthu. (2012). *Network architecture and QoS issues in the Internet of Things for a smart city*. International symposium on communications and information technologies ISCIT. Doi: 978-1-4673-1157-1.
- Wali, Prashant. Das, Debabrata. (2014). *A Novel Access Scheme for IoT Communications in LTE-Advanced Network*. IEEE ANTS 2014 1570023085.
- Li, Ling. Li, Shancang. Zhao, Shanshan. (2014). *QoS-Aware Scheduling of Services-Oriented Internet of Things*. IEEE Transactions on Industrial Informatics, Vol. 10, No. 2, May 2014.
- Ezdiani, Syarifah. Acharyya, Indrajit S. Sivakumar, Sivaramakrishnan. Al-Anbuky, Adnan. (2015). *An IoT Environment for WSN Adaptive QoS*. IEEE International Conference on Data Science and Data Intensive Systems. DOI: 10.1109/DSDIS.2015.28
- Govindan, Kannan. Prakash A., Amad. (2015). *End-to-end Service Assurance in IoT MQTT-SN*. 12th Annual IEEE Consumer Communications and Networking Conference (CCNC).

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Liu, Jiajia. Guo, Hongzhi. Nishiyama, Hiroki. Ujikawa, Hirotaka. Suzuki, Kenichi. Kato, Nei. (2015). *New Perspectives on Future Smart FiWi Networks: Scalability, Reliability and Energy Efficiency*. IEEE Communications Surveys & Tutorials. DOI: 10.1109/COMST.2015.2500960.
- El-Mougy, Amr. Ibnkahla, Mohamed. Hegazy, Lobna. (2015). *Software-Defined Wireless Network Architectures for the Internet-of-Things*. 40th Annual IEEE Conference on Local Computer Network. Florida, USA. DOI: 978-1-4673-6773-8.
- Evans, Dave. (2011). *Internet of Things. La próxima evolución de Internet lo está cambiando todo*. Informe Técnico. Grupo de Soluciones Empresarial para Internet (IBSG) de CISCO.
- Vermesan, Ovidio. Friess, Peter. (2013). *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. River Publishers. Denmark. URL: http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf
- W3C. (2011). *Semantic Sensor Network Incubator Group by W3C SSN-XG*. URL: <https://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>
- Compton, M. Barnaghi, P. Bermúdez, L. et. al. (2012). *The SSN Ontology of the W3C Semantic Sensor Network Incubator Group*. Journal of Web Semantics, vol 17, December 2012, pp 25-32.. DOI: <https://doi.org/10.1016/j.websem.2012.05.003>
- Gyrard, A. Kanti Datta, S. Bonnet, C, Boudaoud, K. (2014). *Standardizing Generic Cross-Domain Applications in Internet of Things*. Globecom 2014 Workshops – IEEE. DOI: <https://doi.org/10.1109/GLOCOMW.2014.7063496>
- oneM2M. (2018). *TS-0001-V1.18.0. Functional Architecture. Technical Specification. V1.18.0*. 2018-03-29. oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC). URL: <https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=26462>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- oneM2M. (2019). *TS-0012-V3.7.3. Base Ontology. Technical Specification. V.3.7.3.* 2019-02-28. oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC). URL: https://www.onem2m.org/images/pdf/TS-0012-Base_Ontology-V3_7_3.pdf
- Guevara, A. Salcedo, O. Pava, R. (2017). *M2M: Technological Basis for Internet of Things.* Contemporary Engineering Sciences, Vol. 10, 2017, no. 34, 1669 - 1676. DOI: <https://doi.org/10.12988/ces.2017.711183>
- Bermúdez-Edo, M. Elsaleh, T. Barnaghi, P. Taylor, K. (2017). *IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics.* Pers Ubiquit Comput. Springer-Verlag. . June 2017, Volume 21, Issue 3, pp 475–487. DOI <https://doi.org/10.1007/s00779-017-1010-8>
- Gruber, T. (1993). *A traslation approach to portable ontology especifications.* Stanford, California. Knowledge Acquisition, 5(2): 199-220.
- Gruber, T. (1995). *Tower principles for the design of ontologies used for knowledge sharing.* Stanford, California. Int. J. Human-Computer Studies, 43, 907-928.
- Liu, L. & Özsu, M. (Eds.). (2009). *Encyclopedia of Database Systems.* Springer US. Jensen, C. & Snodgrass, R. (Auth.). DOI: 10.1007/978-0-387-39940-9.
- oneM2M. (2017). *Standards for M2M and Internet of Things. About oneM2M.* oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC). Recuperado el 08 de Agosto de 2017. URL: <http://www.onem2m.org/about-onem2m/why-onem2m>
- Wageningen UR. (2016). *Ontology of units of Measure OM. Versión 2.0.* Rijgersberg, H. URL: <https://github.com/HajoRijgersberg/OM>.
- Wageningen UR. (2013). *Ontology of units of Measure OM. Versión 1.8.6.* Rijgersberg, H. van Assem, M. Willems, D. Wigham, M. Broekstra, J. Top, J. URL: <http://www.wurvoc.org/vocabularies/om-1.8/>.
- IEEE. (2008). *IEEE Std 754-2008. IEEE Standard for Floating-Point Arithmetic.* The Institute of Electrical and Electronics Engineers Computer Society. DOI:

0.1109/IEEESTD.2008.4610935.

URL:<http://ieeexplore.ieee.org/document/4610935/>

W3C. (2012). *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. W3C Recommendation 5 April 2012*. The World Wide Web Consortium W3C. URL: <https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/>

BIPM. (2006). *Le Système international d'unités*. The International System of Units. Bureau international des poids et mesures. 8e édition. URL: https://www.bipm.org/utis/common/pdf/si_brochure_8.pdf

NIST. (2008). *Guide for the Use of the International System of Units (SI)*. NIST Special Publication 811. 2008 Edition. National Institute of Standards and Technology. URL: <https://physics.nist.gov/cuu/pdf/sp811.pdf>.

Rijgersberg, H. van Assem, M. Top, J. (2013). *Ontology of Units of Measure and Related Concepts*. Semantic Web 4, 1. IOS Press. URL: http://www.semantic-web-journal.net/sites/default/files/swj177_7.pdf.

QUDT. (2017). *Quantities, Units, Dimensions and Data Types Ontologies*. QUDT.org. URL: <http://www.qudt.org/pages/HomePage.html>

W3C. (2004). *OWL Web Ontology Language. Guide. W3C Recommendation*. The World Wide Web Consortium W3C. URL: <https://www.w3.org/TR/owl-guide/>

Wageningen UR. (2017). *OM 2.0 modelled in OWL 2*. Rijgersberg, H. Willems, D. Wigham, M. Top, J. URL: <https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf>

Deaño, Alfredo. (2009). *Introducción a la lógica formal*. Alianza Editorial. Madrid.

TNO. (2014). *Smart Appliances. Study SMART 2013/077*. Netherlands Organization for Applied Scientific Research TNO. (January 2014 – March 2015). URL: <https://sites.google.com/site/smartappliancesproject/>

- TNO. (2016). *SAREF: the Smart Appliances REFerence ontology. Version 2.1.1 (2016-12-13)*. Netherlands Organization for Applied Scientific Research TNO. URL: <http://ontology.tno.nl/saref/>
- Daniele, L. den Hartog, F. Roes, J. (2015). *Study on Semantic Assets for Smart Appliances Interoperability. D-S4 Final Report. Version 1.0*. Netherlands Organization for Applied Scientific Research TNO and the European Commission EC. URL 1: <http://publications.tno.nl/publication/34616699/pLz0s2/daniele-2015-studyfinal.pdf>
- ETSI. (2017). *Technical Especification TS 103 264 V2.1.1 (2017-03). SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping*. European Telecommunications Standards Institute. URL: http://www.etsi.org/deliver/etsi_ts/103200_103299/103264/02.01.01_60/ts_103264_v020101p.pdf
- W3C. (2004a). *OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004*. The World Wide Web Consortium W3C. URL: <https://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- oneM2M. (2016). *Technical Especification TS-0012-V2.0.0. Base Ontology. 2016-August-30*. oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC). URL: http://www.onem2m.org/images/files/deliverables/Release2/TS-0012-oneM2M-Base-Ontology-V2_0_0.zip.
- ETSI. (2016). *Technical Especification TS 118 112 V2.0.0 (2016-09). oneM2M; Base Ontology (oneM2M TS-0012 version 2.0.0 Release 2)*. European Telecommunications Standards Institute. URL: http://www.etsi.org/deliver/etsi_ts/118100_118199/118112/02.00.00_60/ts_118112_v020000p.pdf
- IETF. (2005): *RFC 3986. Uniform Resource Identifier (URI): Generic Syntax*. The Internet Engineering Task Force. Berners-Lee, T. Fielding, R. Masinter, L. STD 66. URL: <https://www.rfc-editor.org/rfc/rfc3986.txt>

- IETF. (1996): *RFC 2046. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. The Internet Engineering Task Force. Freed, N. Borenstein, N. URL: <https://www.rfc-editor.org/rfc/rfc2046.txt>
- IETF. (2004): *RFC 3870. application/rdf+xml Media Type Registration*. The Internet Engineering Task Force. Swartz, A. URL: <https://www.rfc-editor.org/rfc/rfc3870.txt>
- W3C. (2004b). *RDF Primer. W3C Recommendation 10 February 2004*. The World Wide Web Consortium. Manola, F. Miller, E. URL: <https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>
- W3C. (2004c). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. The World Wide Web Consortium. W3C Recommendation 10 February 2004. Klyne, G. Carroll, J. URL: <https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- W3C. (2014). *RDF 1.1 Primer. W3C Working Group Note 24 Jun 2014*. The World Wide Web Consortium. McBride, B. Manola, F. Miller, E. URL: <https://www.w3.org/TR/rdf11-primer/>
- W3C. (2014a). *OWL 2 Web Ontology Language XML Serialization (Second Edition)*. W3C Recommendation 11 December 2012. The World Wide Web Consortium. Bechhofer, S. Cuenca, B. Fokoue, A. Hoekstra, R. URL: <https://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/>
- W3C. (2016). *Register an Internet Media Type for a W3C Spec*. The World Wide Web Consortium. Le Hégarét, P. Seltzer, W. URL: <https://www.w3.org/2002/06/registering-mediatype2014.html>
- W3C. (2012a). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. W3C Recommendation 11 December 2012. The World Wide Web Consortium. W3C OWL Working Group. URL: <https://www.w3.org/TR/owl2-overview/>
- IETF. (2005a): *RFC 3987. Internationalized Resource Identifiers (IRIs)*. The Internet Engineering Task Force. Duerst, M. Suignard, M. URL: <https://www.rfc-editor.org/rfc/rfc3987.txt>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Eurostat. (2013). *Manual for statistics on energy consumption in households*. Statistical office of the European Union. European Commission. Luxembourg. DOI: 10.2785/45686. URL: <http://ec.europa.eu/eurostat/documents/3859598/5935825/KS-GQ-13-003-EN.PDF/baa96509-3f4b-4c7a-94dd-feb1a31c7291>
- ETSI. (2017a). *Technical Report TR 103 411 V1.1.1 (2017-02). SmartM2M; Smart Appliances; SAREF extension investigation*. European Telecommunications Standards Institute. URL: http://www.etsi.org/deliver/etsi_tr/103400_103499/103411/01.01.01_60/tr_103411v010101p.pdf
- RAE. (2014). *Diccionario de la lengua española*. Real Academia Española RAE. 23ª edición. Actualización 23.1 (Dic. 2017). URL: <http://www.rae.es>
- OUP. (2017). *Oxford English Dictionary*. Oxford University Press OUP. Last update (Dec. 2017). URL: <https://en.oxforddictionaries.com/>
- IETF. (2002). *RFC 3339. Date and Time on the Internet: Timestamps. July 2002*. The Internet Engineering Task Force. Newman, C. Klyne, G. URL: <https://www.rfc-editor.org/rfc/rfc3339.txt>
- ISO. (2004). *International Standard ISO 8601. Data elements and interchange formats - Information interchange - Representation of dates and times. Third edition 2004-12-01*. the International Organization for Standardization. Technical Committee ISO/TC 154. Reference number ISO 8601:2004(E). URL: <https://www.iso.org/obp/ui/#iso:std:iso:8601:ed-3:v1:en>
- LBNL. (2004). *Machine to Machine (M2M) Technology in Demand Responsive Commercial Buildings*. Lawrence Berkeley National Laboratory LBNL. Watson, D. Piette, M. Sezgen, O. Motegi, N. ten Hope, L. (August 2004). URL: <https://eta.lbl.gov/sites/default/files/publications/lbnl-55087.pdf>
- Eurostat. (2018). *Simplified energy balances – annual data (nrg_100a)*. Statistical office of the European Union. European Commission. Luxembourg. Last update of data:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- 30/01/2018. URL:
http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_100a&lang=en
- UPME. (2018). *Balance Energético Colombiano BECO. Serie Energético*. Unidad de Planeación Minero Energética UPME. Ministerio de Minas y Energía. República de Colombia. Consultado el 30/01/2018. URL:
<http://www1.upme.gov.co/InformacionCifras/Paginas/BECOENERGTICO.aspx>
- UPME. (2015). *Plan Energético Nacional Colombia: Ideario Energético 2050*. Unidad de Planeación Minero Energética UPME. Ministerio de Minas y Energía. República de Colombia. Bogotá. Enero 2015. URL:
http://www.upme.gov.co/docs/pen/pen_idearioenergetico2050.pdf
- O'Connor, P. (2010). *Energy Transitions*. Boston University. The Frederick S. Pardee Center for the Study of the Longer-Range Future. The Pardee Papers. No. 12. November 2010. Massachusetts. URL:
<https://www.bu.edu/pardee/files/2010/11/12-PP-Nov2010.pdf>
- APS. (2018). *Energy Units. Source based conversion factors and units*. American Physical Society APS. The Current Energy Situation & Background Papers. The Panel on Public Affairs Reports POPA. New York. URL:
<https://www.aps.org/policy/reports/popa-reports/energy/units.cfm>
- ETSI. (2017b). *Technical Specification TS 103 410-1 V1.1.1 (2017-1). SmartM2M; Smart Appliances Extension to SAREF; Part 1: Energy Domain*. European Telecommunications Standards Institute. URL:
http://www.etsi.org/deliver/etsi_ts/103400_103499/10341001/01.01.01_60/ts_10341001v010101p.pdf
- TNO. (2016a). *SAREF4ENER: an extension of SAREF for the energy domain*. Netherlands Organization for Applied Scientific Research TNO. Last update: 2016-12-13. URL: <http://ontology.tno.nl/saref4ener/>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

- Daniele, L. (2015a). *SAREF4EE Documentation. Version 0.1, 28-12-2015*. Netherlands Organization for Applied Scientific Research TNO. URL: http://ontology.tno.nl/SAREF4EE_Documentation_v0.1.pdf
- TNO. (2015). *SAREF4EE: the extension of SAREF for EEBus and Energy@Home*. Netherlands Organization for Applied Scientific Research TNO. Last update: 2015-12-23. URL: <http://ontology.tno.nl/saref4ee/>
- Jacobson, I. Booch, G. Rumbaugh, J. (1999). *The Unified Software Development Process*. Pearson Education (Addison Wesley Longman). Boston, MA 02116.
- Shannon, R. (1998). *Introduction to the art and science of simulation*. Proceedings of the 1998 Winter Simulation Conference. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds. URL: <https://www.informs-sim.org/wsc98papers/001.PDF> .
- Cifuentes, Francisco. (2012). *Introducción al modelamiento de ontologías. Sistemas y Servicios de Información en Red*. Biblioteca del Congreso Nacional de Chile BCN. 01/02/12. URL: <https://es.slideshare.net/francisco.cifuentes/introduccion-curso-ontologas>
- Hernández, R. Fernández, C. Baptista, P. (2006). *Metodología de la investigación*. 4ª edición. McGraw-Hill Interamericana. México.
- IoT-A. (2013). *Internet of Things - Architecture*. URL: <http://www.ietf-a.eu/>. Consultado el 01 de Abril de 2016.
- ETSI. (2013). *ETSI TR 101 584 V2.1.1 (2013-12). Technical Report. Machine to Machine Communications (M2M); Study on Semantic support for M2M Data*. European Telecommunications Standards Institute. URL: https://www.etsi.org/deliver/etsi_tr/101500_101599/101584/02.01.01_60/tr_101584v020101p.pdf
- ETSI. (2021). *Technical Specification TS 118 112 V3.7.3 (2021-01). oneM2M; Base Ontology (oneM2M TS-0012 version 3.7.3 Release 3)*. European Telecommunications Standards Institute ETSI. URL:

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

https://www.etsi.org/deliver/etsi_ts/118100_118199/118112/03.07.03_60/ts_118112v030703p.pdf

oneM2M. (2021). *Base Ontology. The oneM2M Base Ontology. Version 3.6.0. (base_ontology.owl)*. oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC). URL: <https://git.onem2m.org/MAS/BaseOntology>

Bermúdez, M. (2021). *Modeling IoT CO2. Version 1.0 (OntologyModelingIoTCo2.owl)*. URL: <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/>

Bermúdez, M. Salcedo, O. Acosta, A. (2020). *Functional architectures for the Internet of Things: Fundamental models*. International Journal of Mechanical and Production Engineering Research and Development. Vol. 10, Num. 6 (Dec 31, 2020), Págs. 655–672 (TransStellar Journal Publications. Research Consultancy). DOI: <http://doi.org/10.24247/ijmperdddec202071>

Bermúdez, M. Salcedo, O. Rodríguez, J. (2021a). *Ontological Base Models Machine-To-Machine M2M Applied to the Internet of Things IoT*. Revista Boletín Redipe. Vol. 10, Num. 12 (dic. 2021), Págs. 148–161 (Red iberoamericana de pedagogía). DOI: <https://doi.org/10.36260/rbr.v10i12.1576>

ANEXOS

ANEXO 1:

Despliegue de la ontología propuesta titulada como "OntologyModelingIoTCo2.owl", la cual se encuentra disponible en la dirección web: <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl>

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/2002/07/owl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vann="http://purl.org/vocab/vann/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:wv="http://www.wurvoc.org/page/om-2"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:bibo="http://purl.org/ontology/bibo/"
  xmlns:time="http://www.w3.org/2006/time#"
  xmlns:saref="https://saref.etsi.org/core/"
  xmlns:s4ener="https://saref.etsi.org/saref4ener/"
  xmlns:om="http://www.ontology-of-units-of-measure.org/resource/om-2/"
  xmlns:oneM2M=
    "https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#"
  xmlns:OntIoTCo2=
    "https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-
    /raw/main/OntologyModelingIoTCo2.owl#"
>

  <Ontology rdf:about=
    "https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-
    /raw/main/OntologyModelingIoTCo2.owl">
    <imports rdf:resource=
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl"/>
  <imports rdf:resource=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf"/>
  <imports rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf"/>
  <imports rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf"/>
  <rdfs:comment>El presente archivo contiene el modelado de la
ontología para la eficiencia energética de dispositivos electrónicos
conectados a la red informática (Internet de las Cosas IoT), el cual
facilita la gestión de dióxido de carbono en las ciudades. Dicho modelo
hereda de manera múltiple, las características y propiedades de las
ontologías OM (Ontología de Unidades de Medida), SAREF4ENER (Ontología de
Referencia para Dispositivos Inteligentes en el Dominio de la Energía),
SAREF (Ontología de Referencia para Dispositivos Inteligentes) y por
último, oneM2M (Ontología Máquina a Máquina Monádica)</rdfs:comment>
  <dc:creator xml:lang="es">Mauricio Orlando Bermúdez
Amaya</dc:creator>
  <dc:publisher xml:lang="es">mbermudez.amaya@gmail.com</dc:publisher>
</Ontology>

<!--
https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#Devi
ce -->

  <Class rdf:about=
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#Dev
ice">
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Device"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device"/>
  <rdfs:comment>El Dispositivo es una subclase de la ontología oneM2M,
el cual se encuentra asociado a la superclase Dispositivo de la ontología
SAREF, como también a superclase Dispositivo de la ontología SAREF4ENER.
Esta clase se halla en el nivel más bajo de la jerarquía de clases
perteneciente a la ontología OntologyModelingIoTCo2. En las tres
ontologías, un Dispositivo (Device) se concibe como un artefacto que
realiza una labor en específico -la cual se conoce con el nombre de
función-, a partir de unos datos generados o recuperados. Lo anterior
facilita asociar mediante herencia múltiple, a la subclase 'Dispositivo'
de la ontología oneM2M, con las dos superclases 'Dispositivo', ambas
pertenecientes a las ontologías SAREF4ENER y SAREF respectivamente.
  </rdfs:comment>
</Class>

<!-- https://saref.etsi.org/core/v3.1.1/saref.rdf#Device -->

  <Class rdf:about="https://saref.etsi.org/core/v3.1.1/saref.rdf#Device">
  <rdfs:comment>El dispositivo es una superclase perteneciente a la
ontología SAREF. Esta clase se encuentra en el nivel más alto de la
jerarquía de clases perteneciente a la ontología
'OntologyModelingIoTCo2'</rdfs:comment>
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
</Class>

<!-- https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device -->

<Class rdf:about=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device">
  <rdfs:comment>El dispositivo es una superclase perteneciente a la
ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente
a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el
nivel superior a la clase Dispositivo de la ontología
oneM2M</rdfs:comment>
</Class>

<!--
https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#SET_
OutputDataPoint -->

<Class rdf:about=
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#SET_
_OutputDataPoint">
  <rdfs:subClassOf rdf:resource=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf#Quantity"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Property"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power"/>
  <rdfs:comment>La clase 'Asignar Punto de Datos de Salida'
(SET_OutputDataPoint) se halla en el nivel más bajo de la jerarquía de
clases perteneciente a la ontología 'OntologyModelingIoTCo2'. El proceso
de actualizar o generar los datos que se entregan por parte de un
dispositivo sensor, conlleva a registrar tanto la energía que dicho
artefacto consume, como la potencia que el mismo produce y en simultánea,
a ejecutar la operación de salida de tales datos en un momento o ventana
de tiempo específica. Dicha rutina se alcanza a partir de la asociación
de tipo herencia múltiple, desde la subclase 'Establecer Punto de Datos
de Salida' de la ontología oneM2M, hacia las superclases 'Energía' y
'Potencia' pertenecientes a la ontología SAREF4ENER, al igual que hacia
la superclase 'Quantity' de la ontología OM.</rdfs:comment>
</Class>

<!-- https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy -->

<Class rdf:about=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy">
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Property"/>
  <rdfs:comment>La clase 'Energía' es una superclase perteneciente a la
ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente
a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el
nivel superior a la clase 'SET_OutputDataPoint' de la ontología oneM2M.
El consumo de electricidad por parte de un dispositivo electrónico que
```


Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología onem2m2, con la siguiente vinculación: A través de herencia simple, desde las subclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, hacia las superclase 'Propiedad', la cual concierne a la ontología SAREF.</rdfs:comment>

```
</Class>
```

```
<!-- https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power -->
```

```
<Class rdf:about=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power">
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Property"/>
  <rdfs:comment>La clase 'Potencia' es una superclase perteneciente a
la ontologíaSAREF4ENER. Al interior de la jerarquía de clases
perteneciente a la ontología'OntologyModelingIoTCo2', esta clase se
encuentra en el nivel superior a la clase'SET_OutputDataPoint' de la
ontología onem2m2. El consumo de electricidad por parte de un dispositivo
electrónico que esté conectado a la red informática, es necesario
registrarlo para alcanzar la eficiencia energética del mismo al registrar
su energía y/o potencia generada en la realización de su tarea. Ello se
logra complementando y robusteciendo las asociaciones ya detalladas de la
ontología onem2m2, con la siguiente vinculación: A través de herencia
simple, desde las subclases 'Energía' y 'Potencia' pertenecientes a la
ontología SAREF4ENER, hacia las superclase 'Propiedad', la cual concierne
a la ontología SAREF.</rdfs:comment>
</Class>
```

```
<!-- https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf#Quantity -->
```

```
<Class rdf:about=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf#Quantity">
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy"/>
  <rdfs:subClassOf rdf:resource=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf#Unit"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power"/>
  <rdfs:comment>La clase 'Cantidad' es una superclase perteneciente a
la ontología OM. Al interior de la jerarquía de clases perteneciente a la
ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el nivel
superior a la clase 'SET_OutputDataPoint' de la ontología onem2m2. El dato
que es actualizado al momento de entregarse por parte de un artefacto
tipo sensor, atañe a un valor ya sea numérico, texto, booleano, etc., que
resulta de una medida u operación respecto a una magnitud, en un universo
de discurso preestablecido. Dicho valor se tipifica mediante la
asociación de tipo herencia múltiple desde la subclase 'Establecer Punto
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

de Datos de Salida' de la ontología onem2m, hacia la superclase 'Cantidad' concerniente a la ontología OM. El consumo de electricidad por parte de un dispositivo electrónico que esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología onem2m2, con la siguiente vinculación: Mediante herencia múltiple, desde la subclase 'Cantidad' de la ontología OM, hacia las superclases 'Energía' y 'Potencia', las cuales hacen parte de la ontología SAREF4ENER.</rdfs:comment>

```
</Class>
```

```
<!-- https://saref.etsi.org/core/v3.1.1/saref.rdf#Property -->
```

```
<Class rdf:about=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Property">
  <rdfs:comment>La clase 'Propiedad' es una superclase perteneciente a
la ontología SAREF. Al interior de la jerarquía de clases perteneciente a
la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el
nivel superior a la clase 'SET_OutputDataPoint' de la ontología onem2m.
La magnitud que es empleada por el dispositivo sensor, corresponde a una
propiedad física susceptible de ser medida, ya sea la energía consumida
y/o la potencia producida por dicha máquina al momento de actualizar o
generar los datos y realizar la tarea programada. Por ende, el
asociamiento mediante herencia múltiple, desde la subclase 'Establecer
Punto de Datos de Salida' de la ontología onem2m, hacia la superclase
'Property' de la ontología SAREF, facilita la cuantificación de la
magnitud utilizada en la entrega de datos por parte del artefacto con el
correspondiente desencadenamiento de su labor de
percepción.</rdfs:comment>
```

```
</Class>
```

```
<!--
```

```
https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#GET_
InputDataPoint -->
```

```
<Class rdf:about=
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#GET_
_InputDataPoint">
  <rdfs:subClassOf rdf:resource=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-
2.0.rdf#Quantity"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy"/>
  <rdfs:subClassOf
rdf:resource="https://saref.etsi.org/core/v3.1.1/saref.rdf#Property"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power"/>
  <rdfs:comment>La clase 'Obtener Punto de Entrada de Datos'
(GET_InputDataPoint) se halla en el nivel más bajo de la jerarquía de
clases perteneciente a la ontología 'OntologyModelingIoTCo2'. El proceso
de actualizar o generar los datos que se entregan por parte de un
dispositivo sensor, conlleva a registrar tanto la energía que dicho
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

artefacto consume, como la potencia que el mismo produce y en simultánea, a ejecutar la operación de salida de tales datos en un momento o ventana de tiempo específica. Dicha rutina se alcanza a partir de la asociación de tipo herencia múltiple, desde la subclase 'Establecer Punto de Datos de Salida' de la ontología onem2m, hacia las superclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, al igual que hacia la superclase 'Quantity' de la ontología OM.</rdfs:comment>

```
</Class>
```

```
<!--
```

```
https://git.onem2m.org/MAS/BaseOntology/raw/master/base\_ontology.owl#ControllingFunction -->
```

```
<Class rdf:about=  
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#ControllingFunction">
```

```
<rdfs:subClassOf rdf:resource=  
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventsData"/>
```

```
<rdfs:subClassOf  
rdf:resource="https://saref.etsi.org/core/v3.1.1/saref.rdf#Function"/>
```

```
<rdfs:subClassOf rdf:resource=  
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData"/>
```

```
<rdfs:comment>La labor de control que ha de efectuar un dispositivo, permite activar el mismo e incidir en el entorno. En dicho momento se ha de registrar la potencia que dicho artefacto produce, mediante el control de la carga de alguno de los eventos mencionados con sus datos respectivos, como del estado o condición en que se encuentre dicha máquina, lo cual depende de los datos que ingresen al mismo en el momento dado. Ello se logra asociando a través de herencia múltiple, desde la subclase 'Función de Control' de la ontología onem2m, hacia las superclases 'Datos de Estado de Control de Carga' y 'Datos del Evento de Control de Carga', las cuales hacen parte de la ontología SAREF4ENER. De igual manera, la función de control se concibe como la capacidad de realizar una tarea por parte del dispositivo, en este caso de controlar, accionar o actuar en un ambiente en el cual dicho artefacto se encuentre inmerso. Por ende, asociar mediante herencia múltiple desde la subclase 'Función de Control' de la ontología onem2m, hacia la superclase 'Función' de la ontología SAREF, permite acceder a las características y propiedades de esta última, relacionadas con tal facultad del dispositivo.</rdfs:comment>
```

```
</Class>
```

```
<!--
```

```
https://git.onem2m.org/MAS/BaseOntology/raw/master/base\_ontology.owl#MeasuringFunction  
-->
```

```
<Class rdf:about=  
"https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#MeasuringFunction">
```

```
<rdfs:subClassOf rdf:resource=
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEvent
Data"/>
  <rdfs:subClassOf
rdf:resource="https://saref.etsi.org/core/v3.1.1/saref.rdf#Function"/>
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlState
Data"/>
  <rdfs:comment>La tarea de medición que ha de realizar un dispositivo,
permite activar el mismo para percibir el entorno. En dicho momento se ha
de registrar la energía que dicho artefacto consume, mediante el control
de la carga de alguno de los eventos mencionados con sus datos
respectivos, como del estado o condición en que se encuentre dicha
máquina, lo cual está supeditado a los datos que se generen en el mismo
momento dado. Lo anterior se concreta a partir de la asociación de tipo
herencia múltiple, desde la subclase "Función de Medición" de la
ontología oneM2M, hacia las superclases "Datos de Estado de Control de
Carga" y "Datos del Evento de Control de Carga", las cuales hacen parte
de la ontología SAREF4ENER. En simultánea, la función de medición se
concibe como la capacidad de ejecutar una labor por parte del
dispositivo, en este caso de percibir, medir o capturar en un ambiente en
el cual dicha máquina se encuentra inmersa. Por lo tanto, asociar
mediante herencia múltiple desde la subclase "Función de Medición" de la
ontología oneM2M, hacia la superclase "Función" de la ontología SAREF,
facilita el acceso a los atributos y comportamientos de esta última,
relacionadas con dicha competencia del artefacto.</rdfs:comment>
  </Class>

<!--
https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventD
ata -->

  <Class rdf:about=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEvent
Data">
  <rdfs:comment>La clase 'LoadControlEventData' es una superclase
perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de
clases perteneciente a la ontología 'OntologyModelingIoTCo2', esta clase
se encuentra en el nivel superior a la clase 'ControllingFunction' de la
ontología oneM2M, como también de la clase 'MeasuringFunction' de la
misma ontología (oneM2M)</rdfs:comment>
  </Class>

<!-- https://saref.etsi.org/core/v3.1.1/saref.rdf#Function -->

  <Class
rdf:about="https://saref.etsi.org/core/v3.1.1/saref.rdf#Function">
  <rdfs:comment>La función de control se concibe como la capacidad de
realizar una tarea por parte del dispositivo, en este caso de controlar,
accionar o actuar en un ambiente en el cual dicho artefacto se encuentre
inmerso. Por ende, asociar mediante herencia múltiple desde la subclase
'Función de Control' de la ontología oneM2M, hacia la superclase
'Función' de la ontología SAREF, permite acceder a las características y
propiedades de esta última, relacionadas con tal facultad del
dispositivo.</rdfs:comment>
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
</Class>

<!--
https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData -->

<Class rdf:about=
"https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData">
  <rdfs:comment>La clase 'LoadControlStateData' es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el nivel superior a la clase 'ControllingFunction' de la ontología oneM2M, como también de la clase 'MeasuringFunction' de la misma ontología (oneM2M)</rdfs:comment>
</Class>

<!-- https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit -->

<Class rdf:about=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit">
  <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure"/>
  <rdfs:comment>La expresión correspondiente a la medición de una propiedad física o magnitud, es susceptible de clasificarse en alguna unidad fundamental o derivada del Sistema Internacional de Unidades SI. Dicha categorización se concreta a partir de la asociación de tipo herencia simple, desde la subclase 'Unidad' de la ontología OM, hacia la superclase 'Unidad de Medida' de la clase SAREF. Así mismo, al interior de la ontología OM, se presenta la asociación de tipo herencia simple entre la subclase 'Cantidad' (Quantity) y la superclase 'Unidad' (Unit). Similarmente, en la ontología SAREF existe la relación o Propiedad de Objeto (ObjectProperty) entre las clases 'Medición' (Measurement) y 'Unidad de Medida' (UnitOfMeasure). Ambas peculiaridades mencionadas facilitan el acceso a la totalidad de las características y propiedades de tales ontologías, ya que aquellas están disponibles desde la subclases 'Establecer Punto de Datos de Salida' y 'Obtener Punto de Datos de Entrada' de la ontología oneM2M, al asociarse con la subclase 'Cantidad' de la ontología OM.</rdfs:comment>
</Class>

<!-- https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure -->

<Class rdf:about=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure">
  <rdfs:comment>La clase 'Unidad de Medida' es una superclase perteneciente a la ontología SAREF. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el nivel superior a la clase 'Unidad' (Unit) de la ontología OM.</rdfs:comment>
</Class>
```

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

```
<!--      https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure -->

  <Class rdf:about=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure">
    <rdfs:subClassOf rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement"/>
    <rdfs:domain rdf:resource=
"https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit"/>
    <rdfs:comment>La comparación de una cantidad con su respectiva
unidad, a fin de determinar una magnitud, facilita la medición de la
misma, i.e., su dimensionamiento. Dicha labor de estimación se formaliza
a partir de la asociación de tipo herencia simple, desde la subclase
'Medida' de la ontología OM, hacia la superclase 'Medición' de la clase
SAREF. Adicionalmente, al interior de la ontología OM, se presenta la
relación o Propiedad de Objeto (ObjectProperty) entre las clases 'Medida'
(Measure) y 'Unidad' (Unit). De manera ídem, en la ontología SAREF existe
la relación o Propiedad de Objeto (ObjectProperty) entre las clases
'Medición' (Measurement) y 'Unidad de Medida' (UnitOfMeasure). Ambas
peculiaridades mencionadas facilitan el acceso a la totalidad de las
características y propiedades de tales ontologías, ya que aquellas están
disponibles desde la subclases 'Establecer Punto de Datos de Salida' y
'Obtener Punto de Datos de Entrada' de la ontología oneM2M, al asociarse
con la subclase 'Cantidad' de la ontología OM.</rdfs:comment>
  </Class>

<!-- https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement -->

  <Class
rdf:about="https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement">
    <rdfs:domain rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#Property"/>
    <rdfs:domain rdf:resource=
"https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure"/>
    <rdfs:comment>La clase 'Medición' es una superclase perteneciente a
la ontología SAREF. Al interior de la jerarquía de clases perteneciente a
la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el
nivel superior a la clase 'Medida' (Measure) de la ontología OM. Así
mismo, al interior de la ontología SAREF, la clase 'Medición' es el
dominio de la relación con las clases 'Propiedad' y 'Unidad de Medida',
las cuales fungen como recorrido</rdfs:comment>
  </Class>

</rdf:RDF>
```

ANEXO 2:

Grafo del modelo ontológico propuesto titulado como “OntologyModelingIoTCo2.owl”, el cual se encuentra disponible en la dirección web: <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/blob/main/GraphOntologyModelingIoTCo2.png>

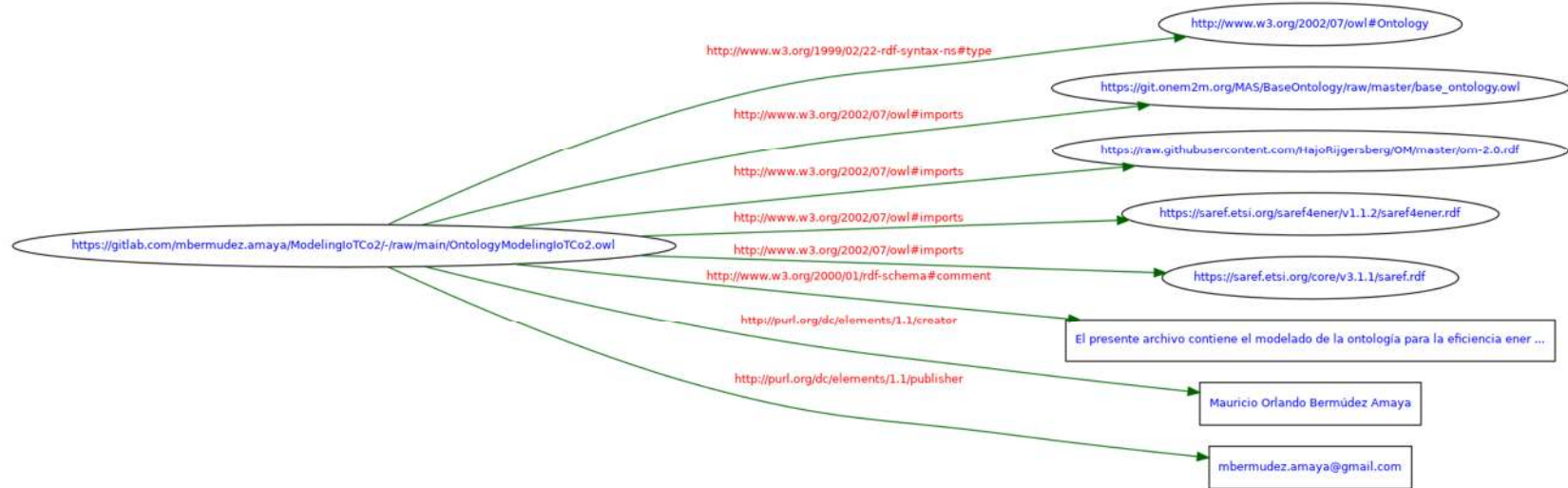


Figura 88. Primera parte del grafo del modelo ontológico propuesto “OntologyModelingIoTCo2.owl” obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades



Figura 89. Segunda parte del grafo del modelo ontológico propuesto “OntologyModelingIoTCo2.owl” obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades



Figura 90. Tercera parte del grafo del modelo ontológico propuesto “OntologyModelingIoTCo2.owl” obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

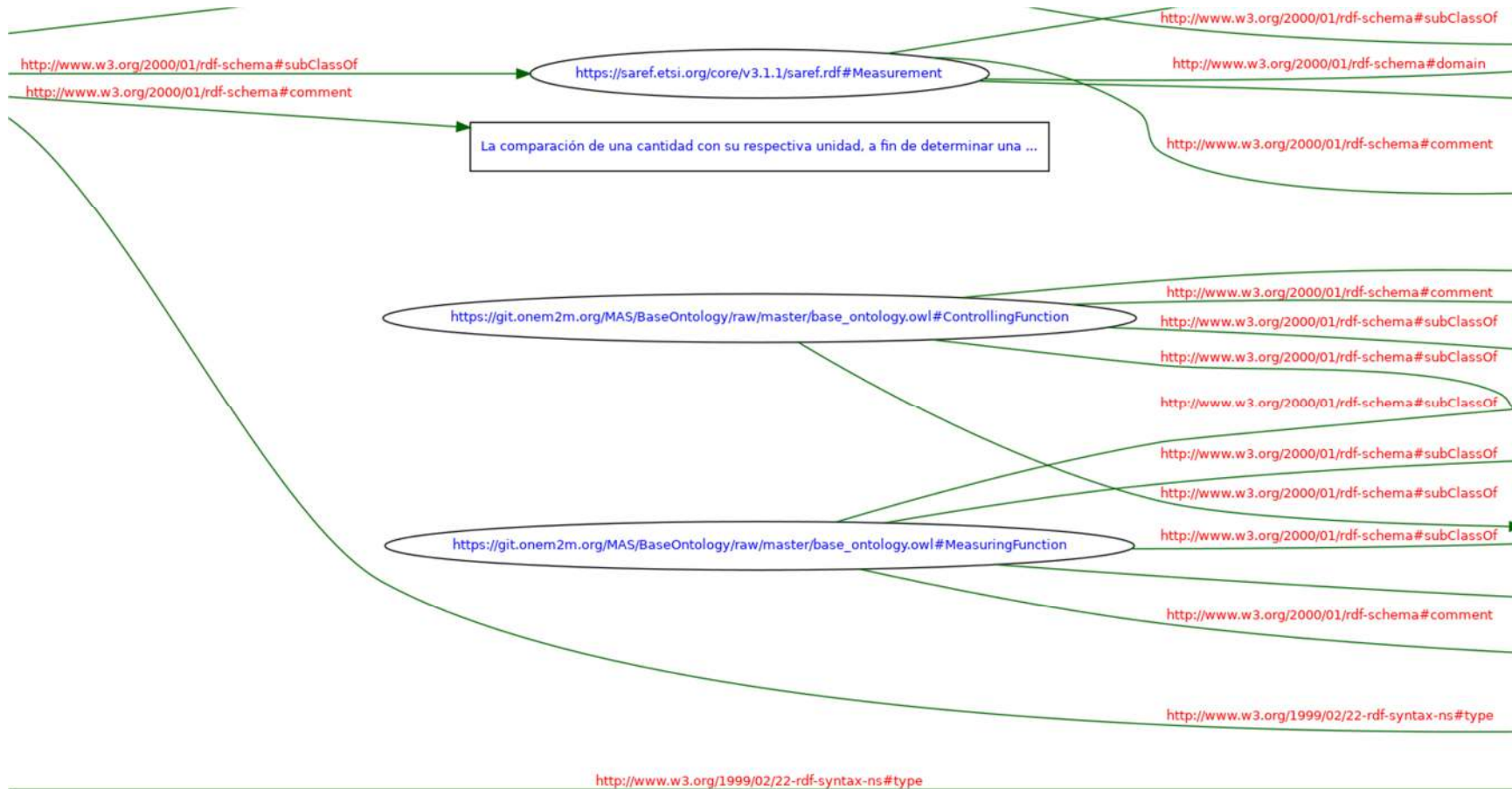


Figura 91. Cuarta parte del grafo del modelo ontológico propuesto "OntologyModelingIoTCo2.owl" obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades



Figura 92. Quinta parte del grafo del modelo ontológico propuesto "OntologyModelingIoTCo2.owl" obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades



Figura 93. Sexta parte del grafo del modelo ontológico propuesto "OntologyModelingIoTCo2.owl" obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>

ANEXO 3:

Tripletas del modelo ontológico propuesto titulado como “OntologyModelingIoTCo2.owl”, el cual se encuentra disponible en la dirección web: <https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/blob/main/TripletsOntologyModelingIoTCo2.pdf>

No.	Subject	Predicate	Object
1	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
2	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl
3	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf
4	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf
5	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2002/07/owl#imports	https://saref.etsi.org/core/v3.1.1/saref.rdf
6	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://www.w3.org/2000/01/rdf-schema#comment	"El presente archivo contiene el modelado de la ontología para la eficiencia energética de dispositivos electrónicos conectados a la red informática (Internet de las Cosas IoT), el cual facilita la gestión de dióxido de carbono en las ciudades. Dicho modelo hereda de manera múltiple, las

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	oTCo2.owl		características y propiedades de las ontologías OM (Ontología de Unidades de Medida), SAREF4ENER (Ontología de Referencia para Dispositivos Inteligentes en el Dominio de la Energía), SAREF (Ontología de Referencia para Dispositivos Inteligentes) y por último, oneM2M (Ontología Máquina a Máquina Monádica)"
7	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://purl.org/dc/elements/1.1/creator	"Mauricio Orlando Bermúdez Amaya"@es
8	https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl	http://purl.org/dc/elements/1.1/publisher	"mbermudez.amaya@gmail.com"@es
9	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#Device	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
10	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#Device	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Device
11	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#Device	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device
12	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#Device	http://www.w3.org/2000/01/rdf-schema#comment	"El Dispositivo es una subclase de la ontología oneM2M, el cual se encuentra asociado a la superclase Dispositivo de la ontología SAREF, como también a superclase Dispositivo de la ontología SAREF4ENER. Esta clase se halla en el nivel más bajo de la jerarquía de clases perteneciente a la ontología OntologyModelingIoTCo2. En las tres ontologías, un Dispositivo (Device) se concibe como un artefacto que realiza una labor en específico –la cual se conoce con el nombre de función–, a partir de unos datos generados o recuperados. Lo anterior facilita asociar mediante herencia múltiple, a la subclase 'Dispositivo' de la ontología oneM2M, con las dos superclases 'Dispositivo', ambas pertenecientes a las ontologías SAREF4ENER y SAREF respectivamente."

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

13	https://saref.etsi.org/core/v3.1.1/saref.rdf#Device	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
14	https://saref.etsi.org/core/v3.1.1/saref.rdf#Device	http://www.w3.org/2000/01/rdf-schema#comment	"El dispositivo es una superclase perteneciente a la ontología SAREF. Esta clase se encuentra en el nivel más alto de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCo2'"
15	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
16	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Device	http://www.w3.org/2000/01/rdf-schema#comment	"El dispositivo es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el nivel superior a la clase Dispositivo de la ontología oneM2M"
17	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#SET_OutputDataPoint	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
18	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#SET_OutputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity
19	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#SET_OutputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy
20	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#SET_OutputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property
21	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#SET_OutputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	DataPoint		
22	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#SET_OutputDataPoint	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Asignar Punto de Datos de Salida' (SET_OutputDataPoint) se halla en el nivel más bajo de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2'. El proceso de actualizar o generar los datos que se entregan por parte de un dispositivo sensor, conlleva a registrar tanto la energía que dicho artefacto consume, como la potencia que el mismo produce y en simultánea, a ejecutar la operación de salida de tales datos en un momento o ventana de tiempo específica. Dicha rutina se alcanza a partir de la asociación de tipo herencia múltiple, desde la subclase 'Establecer Punto de Datos de Salida' de la ontología oneM2M, hacia las superclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, al igual que hacia la superclase 'Quantity' de la ontología OM."
23	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
24	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property
25	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Energía' es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'SET_OutputDataPoint' de la ontología oneM2M. El consumo de electricidad por parte de un dispositivo electrónico que esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología oneM2M2, con la siguiente vinculación: A través de herencia simple, desde las subclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, hacia las superclase 'Propiedad', la cual concierne a la ontología SAREF."
26	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

27	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property
28	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Potencia' es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'SET_OutputDataPoint' de la ontología oneM2M. El consumo de electricidad por parte de un dispositivo electrónico que esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología oneM2M2, con la siguiente vinculación: A través de herencia simple, desde las subclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, hacia las superclase 'Propiedad', la cual concierne a la ontología SAREF."
29	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
30	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy
31	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit
32	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power
33	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Cantidad' es una superclase perteneciente a la ontología OM. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'SET_OutputDataPoint' de la ontología oneM2M. El dato que es actualizado al momento de entregarse por parte de un artefacto tipo sensor, atañe a un valor ya sea numérico, texto, booleano, etc., que resulta de una medida u operación respecto a una magnitud, en un universo de discurso

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

			<p>preestablecido. Dicho valor se tipifica mediante la asociación de tipo herencia múltiple desde la subclase 'Establecer Punto de Datos de Salida' de la ontología oneM2M, hacia la superclase 'Cantidad' concerniente a la ontología OM. El consumo de electricidad por parte de un dispositivo electrónico que esté conectado a la red informática, es necesario registrarlo para alcanzar la eficiencia energética del mismo al registrar su energía y/o potencia generada en la realización de su tarea. Ello se logra complementando y robusteciendo las asociaciones ya detalladas de la ontología oneM2M2, con la siguiente vinculación: Mediante herencia múltiple, desde la subclase 'Cantidad' de la ontología OM, hacia las superclases 'Energía' y 'Potencia', las cuales hacen parte de la ontología SAREF4ENER."</p>
34	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
35	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property	http://www.w3.org/2000/01/rdf-schema#comment	<p>"La clase 'Propiedad' es una superclase perteneciente a la ontología SAREF. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCo2', esta clase se encuentra en el nivel superior a la clase 'SET_OutputDataPoint' de la ontología oneM2M. La magnitud que es empleada por el dispositivo sensor, corresponde a una propiedad física susceptible de ser medida, ya sea la energía consumida y/o la potencia producida por dicha máquina al momento de actualizar o generar los datos y realizar la tarea programada. Por ende, el asociamiento mediante herencia múltiple, desde la subclase 'Establecer Punto de Datos de Salida' de la ontología oneM2M, hacia la superclase 'Property' de la ontología SAREF, facilita la cuantificación de la magnitud utilizada en la entrega de datos por parte del artefacto con el correspondiente desencadenamiento de su labor de percepción."</p>
36	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#GET_InputDataPoint	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
37	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#GET_InputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Quantity

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

38	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#GET_InputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Energy
39	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#GET_InputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property
40	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#GET_InputDataPoint	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#Power
41	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#GET_InputDataPoint	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Obtener Punto de Entrada de Datos' (GET_InputDataPoint) se halla en el nivel más bajo de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2'. El proceso de actualizar o generar los datos que se entregan por parte de un dispositivo sensor, conlleva a registrar tanto la energía que dicho artefacto consume, como la potencia que el mismo produce y en simultánea, a ejecutar la operación de salida de tales datos en un momento o ventana de tiempo específica. Dicha rutina se alcanza a partir de la asociación de tipo herencia múltiple, desde la subclase 'Establecer Punto de Datos de Salida' de la ontología oneM2M, hacia las superclases 'Energía' y 'Potencia' pertenecientes a la ontología SAREF4ENER, al igual que hacia la superclase 'Quantity' de la ontología OM."
42	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#ControllingFunction	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
43	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#ControllingFunction	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData
44	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#ControllingFunction	http://www.w3.org/2000/01/rdf-	https://saref.etsi.org/core/v3.1.1/saref.rdf#Function

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	unction	schema#subClassOf	
45	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#ControllingFunction	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData
46	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#ControllingFunction	http://www.w3.org/2000/01/rdf-schema#comment	"La labor de control que ha de efectuar un dispositivo, permite activar el mismo e incidir en el entorno. En dicho momento se ha de registrar la potencia que dicho artefacto produce, mediante el control de la carga de alguno de los eventos mencionados con sus datos respectivos, como del estado o condición en que se encuentre dicha máquina, lo cual depende de los datos que ingresen al mismo en el momento dado. Ello se logra asociando a través de herencia múltiple, desde la subclase 'Función de Control' de la ontología oneM2M, hacia las superclases 'Datos de Estado de Control de Carga' y 'Datos del Evento de Control de Carga', las cuales hacen parte de la ontología SAREF4ENER. De igual manera, la función de control se concibe como la capacidad de realizar una tarea por parte del dispositivo, en este caso de controlar, accionar o actuar en un ambiente en el cual dicho artefacto se encuentre inmerso. Por ende, asociar mediante herencia múltiple desde la subclase 'Función de Control' de la ontología oneM2M, hacia la superclase 'Función' de la ontología SAREF, permite acceder a las características y propiedades de esta última, relacionadas con tal facultad del dispositivo."
47	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#MeasuringFunction	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
48	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#MeasuringFunction	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData
49	https://git.onem2m.org/MAS/BaseOntology/raw/master/bas_e_ontology.owl#MeasuringFunction	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Function

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

50	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#MeasuringFunction	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData
51	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#MeasuringFunction	http://www.w3.org/2000/01/rdf-schema#comment	"La tarea de medición que ha de realizar un dispositivo, permite activar el mismo para percibir el entorno. En dicho momento se ha de registrar la energía que dicho artefacto consume, mediante el control de la carga de alguno de los eventos mencionados con sus datos respectivos, como del estado o condición en que se encuentre dicha máquina, lo cual está supeditado a los datos que se generen en el mismo momento dado. Lo anterior se concreta a partir de la asociación de tipo herencia múltiple, desde la subclase "Función de Medición" de la ontología oneM2M, hacia las superclases "Datos de Estado de Control de Carga" y "Datos del Evento de Control de Carga", las cuales hacen parte de la ontología SAREF4ENER. En simultánea, la función de medición se concibe como la capacidad de ejecutar una labor por parte del dispositivo, en este caso de percibir, medir o capturar en un ambiente en el cual dicha máquina se encuentra inmersa. Por lo tanto, asociar mediante herencia múltiple desde la subclase "Función de Medición" de la ontología oneM2M, hacia la superclase "Función" de la ontología SAREF, facilita el acceso a los atributos y comportamientos de esta última, relacionadas con dicha competencia del artefacto."
52	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
53	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'LoadControlEventData' es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'ControllingFunction' de la ontología oneM2M, como también de la clase 'MeasuringFunction' de la misma ontología (oneM2M)"
54	https://saref.etsi.org/core/v3.1.1/saref.rdf#Function	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
55	https://saref.etsi.org/core/v3.1.1/saref.rdf#Function	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	"La función de control se concibe como la capacidad de realizar una tarea"

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	1/saref.rdf#Function	2000/01/rdf-schema#comment	por parte del dispositivo, en este caso de controlar, accionar o actuar en un ambiente en el cual dicho artefacto se encuentre inmerso. Por ende, asociar mediante herencia múltiple desde la subclase 'Función de Control' de la ontología oneM2M, hacia la superclase 'Función' de la ontología SAREF, permite acceder a las características y propiedades de esta última, relacionadas con tal facultad del dispositivo."
56	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
57	https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'LoadControlStateData' es una superclase perteneciente a la ontología SAREF4ENER. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'ControllingFunction' de la ontología oneM2M, como también de la clase 'MeasuringFunction' de la misma ontología (oneM2M)"
58	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
59	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure
60	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit	http://www.w3.org/2000/01/rdf-schema#comment	"La expresión correspondiente a la medición de una propiedad física o magnitud, es susceptible de clasificarse en alguna unidad fundamental o derivada del Sistema Internacional de Unidades SI. Dicha categorización se concreta a partir de la asociación de tipo herencia simple, desde la subclase 'Unidad' de la ontología OM, hacia la superclase 'Unidad de Medida' de la clase SAREF. Así mismo, al interior de la ontología OM, se presenta la asociación de tipo herencia simple entre la subclase 'Cantidad' (Quantity) y la superclase 'Unidad' (Unit). Similarmente, en la ontología SAREF existe la relación o Propiedad de Objeto (ObjectProperty) entre las clases 'Medición' (Measurement) y 'Unidad de Medida' (UnitOfMeasure). Ambas peculiaridades mencionadas facilitan el acceso a la totalidad de las características y propiedades de tales ontologías, ya que aquellas están disponibles desde la subclases 'Establecer Punto de Datos de Salida' y 'Obtener Punto de Datos de Entrada' de la ontología oneM2M, al asociarse en

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

			con la subclase 'Cantidad' de la ontología OM."
61	https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
62	https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Unidad de Medida' es una superclase perteneciente a la ontología SAREF. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'Unidad' (Unit) de la ontología OM."
63	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
64	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement
65	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure	http://www.w3.org/2000/01/rdf-schema#domain	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Unit
66	https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf#Measure	http://www.w3.org/2000/01/rdf-schema#comment	"La comparación de una cantidad con su respectiva unidad, a fin de determinar una magnitud, facilita la medición de la misma, i.e., su dimensionamiento. Dicha labor de estimación se formaliza a partir de la asociación de tipo herencia simple, desde la subclase 'Medida' de la ontología OM, hacia la superclase 'Medición' de la clase SAREF. Adicionalmente, al interior de la ontología OM, se presenta la relación o Propiedad de Objeto (ObjectProperty) entre las clases 'Medida' (Measure) y 'Unidad' (Unit). De manera ídem, en la ontología SAREF existe la relación o Propiedad de Objeto (ObjectProperty) entre las clases 'Medición' (Measurement) y 'Unidad de Medida' (UnitOfMeasure). Ambas peculiaridades mencionadas facilitan el acceso a la totalidad de las características y propiedades de tales ontologías, ya que aquellas están disponibles desde la subclases 'Establecer Punto de Datos de Salida' y 'Obtener Punto de Datos de Entrada' de la ontología oneM2M, al asociarse con la subclase 'Cantidad' de la ontología OM."
67	https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class

Modelado de una arquitectura de la Internet de las Cosas para gestionar la emisión de carbono atmosférico en las ciudades

	1/saref.rdf#Measurement	syntax-ns#type	
68	https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement	http://www.w3.org/2000/01/rdf-schema#domain	https://saref.etsi.org/core/v3.1.1/saref.rdf#Property
69	https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement	http://www.w3.org/2000/01/rdf-schema#domain	https://saref.etsi.org/core/v3.1.1/saref.rdf#UnitOfMeasure
70	https://saref.etsi.org/core/v3.1.1/saref.rdf#Measurement	http://www.w3.org/2000/01/rdf-schema#comment	"La clase 'Medición' es una superclase perteneciente a la ontología SAREF. Al interior de la jerarquía de clases perteneciente a la ontología 'OntologyModelingIoTCO2', esta clase se encuentra en el nivel superior a la clase 'Medida' (Measure) de la ontología OM. Así mismo, al interior de la ontología SAREF, la clase 'Medición' es el dominio de la relación con las clases 'Propiedad' y 'Unidad de Medida', las cuales fungen como recorrido"

Tabla 45. Tripletas RDF/XML de la ontología propuesta "OntologyModelingIoTCO2.owl" obtenido de la simulación en el validador <http://www.w3.org/RDF/Validator>