UNIVERSIDAD DE MURCIA

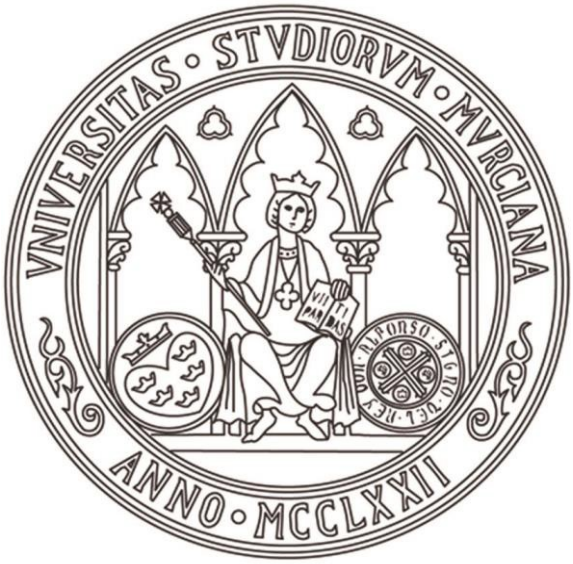
ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Arquitecturas de datos basadas en
Internet de las Cosas y su aplicación

D. Pedro González Gil

2023



UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Arquitecturas de datos basadas en
Internet de las Cosas y su aplicación

Autor: D. Pedro González Gil

Directores: D. Antonio F. Skarmeta Gómez y

D. Juan Antonio Martínez Navarro

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD
DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR
Aprobado por la Comisión General de Doctorado el 19-10-2022

D. Pedro González Gil

doctorando del Programa de Doctorado en

Informática

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

Arquitecturas de datos basadas en Internet de las Cosas y su aplicación

y dirigida por,

D. Antonio F. Skarmeta Gómez

D. Juan Antonio Martínez Navarro

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 22 de julio de 2023

Fdo.: Pedro González Gil

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

Dedicatoria y Reconocimientos

Resulta paradójico que, tras el extendido trabajo de investigación plasmado en este documento, resulte difícil poner en palabras la experiencia que lo ha acompañado durante su realización. Sin duda es el culmen de una carrera académica (no su final) que empieza, para todos los que tenemos la inmensa suerte de vivir en países desarrollados, a la vez que la conciencia. Es, sin duda, la más larga aventura a la que me he enfrentado. Una empresa en que he puesto esfuerzo y pasión y de la que he obtenido una invaluable cantidad de experiencias y conocimientos y que apropiadamente concluye devolviendo a la comunidad, participando en la creación de nuevo conocimiento.

Este bucle infinito y expansivo de creación de nuevo conocimiento, que puede ser resumido, tal vez de forma apropiada, con la frase: *"La ciencia nunca resuelve un problema sin crear otros 10 más"*, de George Bernard Shaw, no es una empresa solitaria, como resumía Isaac Newton con su frase: *"Si he visto más lejos, es poniéndome sobre los hombros de gigantes"*. Es por ello que tengo mucho que agradecer a quienes me han ayudado en este periplo, empezando por mis directores de tesis: Antonio y Juan Antonio, quienes me han acompañado y guiado por el camino de la producción científica, poniendo de relieve el inmenso valor del trabajo, del respeto por el conocimiento y del trabajo en equipo. Gracias por vuestro consejo, dirección y confianza.

Por suerte o desgracia, otros han tenido que sufrirme también durante esta andadura. Más notoriamente mis hijos, Pedro, Loreto y Valentina, a quienes dedico esta tesis con el anhelo de que les inspire, en el futuro, a perseguir este noble camino de sacrificio y descubrimiento y a quienes pido perdón por las horas de atención que les he robado en pos de esta meta.

A mi esposa y compañera Loreto, quien ha sufrido más intensamente mis ausencias, el cansancio y la ansiedad de este proceso y que lo ha soportado con estoicidad y paciencia, apoyándome y alentándome a seguir aun cuando el camino era más empinado. Loreto, esta tesis es tan tuya como mía. Gracias por acompañarme y no perder la fe.

A mis padres: Pedro y Lola, que con paciencia y perseverancia inculcaron en mi desde la infancia, un inmenso amor por la ciencia, el espíritu de superación y la capacidad de sacrificio que me empujaron desde un primer momento a iniciar y completar este periplo.

Por último, a mis compañeros de batalla: otros doctorandos y amigos que encontré en Pleiades, junto a los que los infinitos tropezones y errores se hicieron más llevaderos: Guillermo, Juan Andrés, Manolo, Pedro J., Jesús y en especial a Muss, con quien he compartido de cerca las tribulaciones de la tesis. Gracias a todos.

Tabla de Contenidos

ÍNDICE DE TABLAS

TABLA **Página**

Índice de Figuras

Figura **Página**

Índice de Listados

Listado de Acrónimos

ABAC Control de acceso basado en atributos (attribute-based access control) . . . . . 33 ff., 38 f., 41, 63, 67, 72, 84

API Interfaz de programación de aplicaciones (application programming interface) . . . . . 6, 8 f., 24, 27, 84, 87, 96 ff., 105, 109

CB Broker de contexto (context broker) . . . . . 80 f., 84, 87, 92, 102

CM Gestor de capacidades (capability manager) . . . 39, 63, 86, 92, 94

CP Proveedor de contexto (context provider) 84

CT Token de capacidad (capability token) . . . . . 86 f., 94

CUDA Arquitectura de computación unificada de dispositivos (Compute Unified Device Architecture) 110, 122

DAC Control de acceso a discreción (discretionary access control) . 32 f.

DCapBAC Control de acceso distribuido basado en capacidades (distributed capability-based access control) . . 38 f., 63, 67, 84, 86, 92, 94, 103

DER Recurso energético distribuido (distributed energy resource)19 f., 46, 78, 81, 97, 102

DNN Red neuronal profunda (deep neural network)26, 28, 105, 110, 113, 115, 122 f., 125, 129

DR Respuesta a la demanda (demand response) 19 f., 46, 48, 77 ff., 82, 101 ff., 129

EMC Componente de gestión de energía (energy management component) 77–82, 84, 86–89, 92, 94–98, 100, 102 f.

ETSI European Telecommunications Standards Institute . . . . 7 ff., 81

GDPR Reglamento General de Protección de Datos (General Data Protection Regulation) 2, 62, 65, 128

GPU Unidad de procesamiento de gráficos (graphics processing unit)105, 110, 119, 121 f.

HAS Sistema de automatización del hogar (home automation system)19 f., 77 ff., 81, 89, 96 ff., 100, 102 f.

HEC Controlador energético doméstico (home energy controller)78 ff., 88, 92, 94, 96 ff., 100

HEG Pasarela energética doméstica (home energy gateway) 79, 82 f., 98, 101, 103

HEMS Sistema de gestión energética del hogar (home energy management system) . . . . . 16 f., 19 ff., 78, 80 f.

HTML Lenguaje de marcas de hipertexto (hyper-text markup language) 11 f.

HTTP Protocolo de transferencia de hipertexto (hyper-text transfer protocol) 10, 86, 94, 111

HTTPS Protocolo de transferencia de hipertexto seguro (hyper-text transfer protocol secure) . . . . . 86 f.

HVAC Calefacción, ventilación y aire acondicionado (heating, ventilation and air conditioning) . . . . . 17 f., 79, 89, 91, 96

IdM Gestión de la identidad (identity management) . 38, 84, 86, 92, 94, 103

IdP Proveedor de identidad (identity provider) 37

IdT Token de identidad (identity token) 86, 94

IoT Internet de las cosas (internet of things) . 1 ff., 5 ff., 9 f., 14 ff., 18 f., 22 ff., 26–29, 38–41, 44 f., 49–56, 59–66, 70, 74–77, 83, 105, 109, 124, 127 ff.

IP Protocolo de Internet (Internet protocol) 27, 119

IPN Nodo de procesamiento de imagen (image processing node)107–111

JSON Notación de objetos de Javascript (Javascript object notation)11, 27

JSON-LD JSON para datos enlazados (JSON for linked data)8, 12, 49, 70, 72, 82, 84, 117

KB Base de conocimiento (knowledge base)78–84, 87, 92, 94, 96 ff., 100, 102

M2M Máquina a máquina (machine-to-machine) . . . 27, 38, 51, 61, 63

MAC Control de acceso obligatorio (mandatory access control) . . . 32 f.

MDR Repositorio de metadatos (metadata repository) . . . . . 59 f., 70

NGSI Interfaz de servicio de nueva generación (next generation service interface) 8, 24, 27, 106, 129

NGSI-LD NGSI para datos enlazados (NGSI for linked data) 8, 38, 59, 70 ff., 75, 81–84, 87, 92, 94, 102, 105 ff., 109, 111, 117, 124, 128

ODS Objetivos de desarrollo sostenible 1, 54

OWL Lenguaje de ontologías web (web ontology language) 11–14, 20, 41, 43, 50, 66, 71, 75, 83

PAP Punto de administración de políticas (policy administration point) 34, 61, 86, 92

PDP Punto de decisión de políticas (policy decision point)34 f., 39, 61, 86, 92, 95

PEP Punto de aplicación de políticas (policy enforcement point) 34 f., 61, 72, 86 f., 92, 94 f., 103

PV Fotovoltaico (photo-voltaic) . . . . . 89, 91, 97 f., 100–103

QoI Calidad de la información (quality of information) . . . . . 40 f., 70

QoS Calidad de servicio (quality of service) 22, 24, 26, 47

RBAC Control de acceso basado en roles (role-based access control)33 f., 41, 63, 67

RDF Marco de descripción de recursos (resource description framework) 11 f., 14

RDF-S RDF schema . . . . . 12 ff.

REST Transferencia de estado representacional (representational state transfer) . . . . . 9 ff., 20, 84, 87, 96, 98, 109

RTP Protocolo de transporte en tiempo real (real-time transfer protocol) 108, 111 f., 115, 117, 123

SC Ciudad inteligente (smart city) 61, 106

SG Red eléctrica inteligente (smart grid) 2, 19 ff., 26, 47 ff., 61, 77, 82, 103, 129

SH Hogar inteligente (smart home) . . . . . 14, 18 f., 21 f., 26, 81, 119

SHEMS Sistema de gestión energética del hogar inteligente (smart home energy management system) . . . 17 ff., 46, 77, 81 ff., 89, 91 f., 97 f., 100–103, 128 f.

URI Indicador de recurso universal (universal resource indicator) 10 ff.

URL Localizador de recurso universal (universal resource locator)65, 68, 72, 86, 112

USB Bus de serie universal (universal serial bus) 112, 122

USEF Marco universal para la energía inteligente (Universal Smart Energy Framework) 20, 49

W3C World Wide Web Consortium 7, 12, 14, 45

WoT Web de las cosas (web of things) . . . . . 8 ff., 14 f.

XACML Lenguaje extensible de marcas para control de acceso (extendible access control markup language) 20, 35, 38, 41, 86, 88, 94

XML Lenguaje extensible de marcas (extendible markup language)11, 35

Resumen

El internet de las cosas (IoT por sus siglas en inglés) se presenta como nuevo paradigma de los sistemas de información, capaz de ayudar en numerosos retos a los que se enfrenta nuestra sociedad: desde la lucha contra el cambio climático hasta el progreso por la consecución del derecho a una vida digna y feliz. Este nuevo paradigma plantea no solo posibilidades sino numerosos retos tecnológicos que deberán ser abordados aplicando el método científico, para el beneficio global.

La privacidad de los datos y la seguridad de las comunicaciones adquieren nuevas problemáticas en un marco tan dinámico como el IoT, donde los datos fluyen rápidamente de productores a consumidores, son procesados por agentes intermediarios y su valor incrementado gracias a relaciones entre datos de distintos orígenes.

Las características intrínsecas de los datos, ligadas frecuentemente a los propios dispositivos que los generan (como la precisión o la frecuencia de actualización), deben acompañar a estos datos en su viaje a través de Internet, aportando un contexto imprescindible para su correcta interpretación y utilización.

La búsqueda de dichos datos en las inmensas colecciones donde se almacenan, así como la comprensión de los mismos es un reto, donde el volumen exige la utilización de técnicas de procesado automatizado e incluso la aplicación de técnicas basadas en inteligencia artificial.

La distribución y transporte de estos grandes volúmenes de datos pueden suponer problemas para las redes de distribución, lo que unido a posibles exigencias de baja latencia en la respuesta, nos fuerzan a diseñar y aplicar novedosas estrategias para el procesado de dichos datos de forma local o próxima a donde han sido producidos.

Finalmente, la capacidad de interconectar plataformas y sistemas de manera fácil, rápida y segura mediante arquitecturas modulares y flexibles, supone una necesidad, más que una ventaja competitiva, en una nueva economía dirigida por los datos.

Estos problemas, atacados de forma individual, ofrecen interesantes escenarios de investigación y mejora, pero considerados de forma conjunta revelan una realidad aún más interesante, donde se hace evidente que el total de las partes es más que la suma de las mismas, revelando relaciones y sinergias entre los diseños de arquitecturas e interfaces, el modelado de los datos, la seguridad e incluso la distribución y orquestación de tareas.

Este trabajo de tesis doctoral explora estos aspectos de una manera holística, integrando tecnologías y estándares existentes en el diseño y construcción de arquitecturas de datos para

IoT seguras e interoperables, considerando modelado y representación ontológica de los datos y explorando también la distribución y orquestación de tareas entre el borde y la nube.

Como resultado, se presentan tres contribuciones principales, orientadas a la exploración de las arquitecturas interoperables y seguras, la distribución y orquestación de tareas entre el borde y la nube y finalmente sobre la representación de conceptos de seguridad y privacidad de los datos en este nuevo paradigma. En estas contribuciones se ofrecen diversos escenarios y bancos de prueba de las mismas, demostrando su relevancia y valor de cara al desarrollo y avance de estas tecnologías.

CAPÍTULO 1

INTRODUCCIÓN

En 2015, la Organización de las Naciones Unidas (ONU) aprobó la Agenda 2030 sobre el Desarrollo Sostenible [9]. En esta se planteó la oportunidad de que los países y sus sociedades emprendiesen un camino de mejora de la vida a nivel global, presentando 17 objetivos de desarrollo sostenible (ODS), que comprendían desde la eliminación de la pobreza hasta la lucha contra el cambio climático. En este contexto, el internet de las cosas (internet of things, IoT) se presenta como uno de los avances tecnológicos disruptivos [10] con un gran potencial de cara a su aplicación como parte de la estrategia para enfrentarnos a un gran número de los mencionados ODS.

A pesar del gran potencial presente en IoT, son numerosos los retos que aún quedan por superar, como por ejemplo la interoperabilidad. Esta puede ser definida como la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos. Mejorando la interoperabilidad de los sistemas IoT, habilitamos la explotación automatizada de la información por máquinas, así como el desarrollo de las economías basadas en los datos [11], plataformas de búsqueda de información IoT [3] e incluso el análisis y la explotación de los datos de manera ubicua y desatendida. La interoperabilidad vendrá facilitada por la aplicación de estándares, que facilitarán la integración y composición de soluciones, la reutilización de plataformas y componentes existentes, lo que flexibilizará las soluciones y al tiempo aportará seguridad y reducción de costes.

Esta estandarización está sucediendo tanto a nivel de comunicaciones e interfaces, como en los modelos de datos. Las aproximaciones ontológicas al modelado de datos abren la puerta a al procesado automatizado de información y a la aplicación de diversas tecnologías para el razonamiento semántico. Este paradigma, heredado de la web semántica, impulsa la creación de nuevas

ontologías y modelos de conocimiento para representar los diversos conceptos pertenecientes a los contextos de aplicación de IoT.

No menos importantes son las consideraciones referentes a la seguridad de dichas plataformas, sobre todo desde la perspectiva de la privacidad de datos, en un entorno donde la información fluye rápidamente entre diversas plataformas y donde el origen y/o el destino de los datos puede ser desconocido. Nuevas regulaciones, como el Reglamento General de Protección de Datos (General Data Protection Regulation, GDPR) imponen restricciones referentes a como deben ser manejados los datos por los intermediarios y abren nuevas vías para la exploración. En este contexto, la seguridad se convierte en parte de la información sobre la información (metainformación), que debe ser considerada en todos los pasos del camino y que debe acompañar a los datos.

Finalmente, la barrera entre el borde y la nube se ha desdibujado en los últimos años, dando pie al procesado de datos en "la niebla", aprovechando el principio de localidad de los datos de cara a su procesado de manera eficiente en términos de ancho de banda y consumo energético. Estas nuevas estrategias de procesado de la información abren otra vía de investigación en que la orquestación de tareas entre el borde y la nube presentan nuevos y excitantes retos y espacio para la innovación.

1.1 Organización de la tesis

Esta tesis doctoral está organizada de la siguiente forma: en este primer Capítulo se ofrece una introducción y objetivos generales, contextualizada a la realidad socioeconómica presente.

En el Capítulo 2 se explora el estado del arte, ahondando en las áreas cubiertas por este trabajo: arquitecturas para IoT, orquestación de tareas, seguridad y privacidad de datos y modelado ontológico de datos. Adicionalmente, en este capítulo, se expone la motivación y objetivos de investigación de esta tesis, planteados en el proyecto de doctorado y se resumen los resultados obtenidos durante su realización.

En los siguientes capítulos se desarrollan las principales contribuciones de esta tesis. El Capítulo 3 presenta una ontología ligera para la representación y anotación de conceptos de seguridad de los datos, con la novedad de realizarse desde la perspectiva de los propios datos e introduciendo conceptos nuevos, tales como regulaciones, certificaciones y procedencia a los conceptos clásicos tales como control de acceso y mecanismos de autenticación. Esta ontología sienta las bases del modelo de representación de la metainformación relativa a los esquemas de seguridad y privacidad, que son utilizados en las siguientes contribuciones presentadas en esta tesis.

El Capítulo 4 describe una arquitectura para la gestión energética del hogar inteligente, en la que se atiende a la seguridad y privacidad de los datos, así como la integración de servicios y dispositivos en el hogar, además de la integración del sistema de gestión energético en la red eléctrica inteligente (smart grid, SG), atendiendo a la interoperabilidad a través del uso de

interfaces y modelos de datos estandarizados.

Por último, tras sentar las bases para la representación de información y gestión energética del hogar inteligente, ahondamos en la eficiencia energética desde el punto de vista de la propia arquitectura. El Capítulo 5 describe una arquitectura para la orquestación de tareas en dispositivos IoT, capaz de aprovechar la potencia de computación mejorada (mediante hardware específico) de dispositivos en el borde, así como recursos en la nube, para la realización de tareas computacionalmente intensivas de procesado de imágenes. Esta arquitectura ofrece mejoras frente a enfoques tradicionales de procesado en la nube, en términos de latencia, eficiencia energética y privacidad y seguridad, entre otros.

El Capítulo 6 ofrece las conclusiones obtenidas como el resultado del trabajo realizado en las diversas áreas exploradas en esta tesis, así como posibles trabajos futuros que se puedan derivar de la misma.

Finalmente, en el Capítulo 7, se enumeran las principales contribuciones científicas derivadas de este proyecto de tesis doctoral, en forma de publicaciones de artículos de revista, publicaciones en congresos y un capítulo de libro.

CAPÍTULO 2

ESTADO DEL ARTE

La ingente cantidad de información generada por los dispositivos IoT produce que los sistemas y los frameworks utilizados para su gestión y proceso representen un ecosistema diverso y complejo. Estos sistemas consisten, de manera general, en complejas redes de dispositivos, así como plataformas de computación, habitualmente ubicadas en la nube y herramientas de análisis que trabajan al unísono con el objetivo común de recolectar información para su procesado y análisis y posterior almacenamiento histórico.

2.1 Arquitecturas de datos IoT

Para poder responder mejor a la elevada complejidad de los sistemas involucrados, las arquitecturas de datos para IoT se dividen en diversas capas. Los esquemas más comunes que podemos encontrar, contienen de 3 a 5 capas [12], [13] (Figura 2.1) en función de la complejidad de la solución representada.

El modelo más sencillo, de tres niveles, se divide en las siguientes capas:

1. *Percepción*: capa responsable de la recolección de datos del mundo real y la interacción con el mismo. Esta capa se compone de sensores y actuadores; dispositivos como cámaras, sensores de temperatura, detectores de movimiento y actuadores, como cerraduras electrónicas o electroválvulas. También es común encontrar dispositivos complejos consistentes en sensores y actuadores en un mismo paquete, como podría ser un termostato, que controla los elementos calefactores y bombas en base a las mediciones de diversos sensores. Esta capa es también conocida frecuentemente como *borde* (o “edge” en inglés) ya que se encuentra en el borde de la red, donde la información es producida.

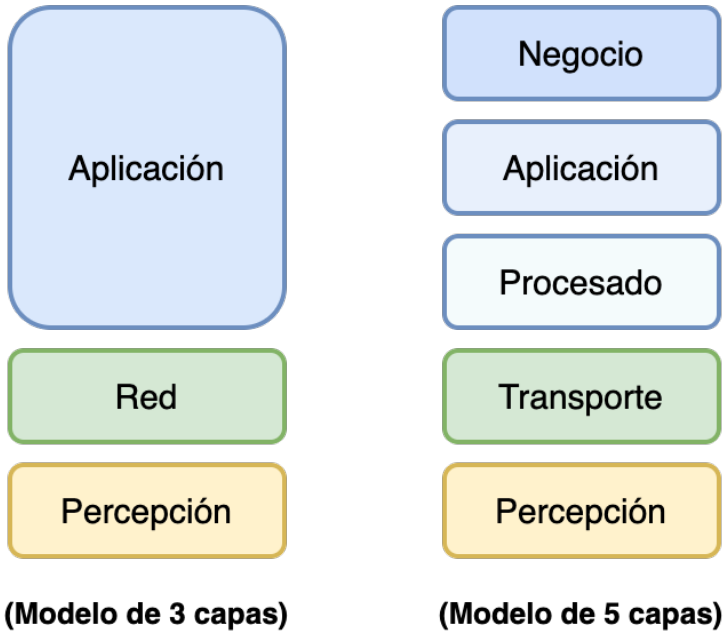


Figura 2.1: Estructura en capas de arquitecturas IoT

2. *Red*: esta capa es responsable de proveer conexión confiable y segura entre los diferentes componentes de la arquitectura, ofreciendo transporte de información entre la capa de percepción y las capas superiores de la arquitectura, por lo que también es frecuentemente denominada *capa de transporte*. En esta capa encontramos equipamiento de comunicaciones típico de arquitecturas de red (WiFi, 5G, 4G, GPRS), así como equipamiento especializado como pasarelas a diversas tecnologías de comunicación específicas para IoT (ZigBee, Z-Wave, Bluetooth).

3. *Aplicación*: finalmente, la capa de aplicación es la que actúa de intermediaria entre los usuarios finales y el resto de la arquitectura. Es la responsable de presentar la información recogida de la capa de percepción de forma que aporte valor, tenga sentido y resulte de utilidad al usuario. Paneles de datos y herramientas de monitorización son elementos típicos de esta capa. Otra de las funciones de esta capa es la de actuar de intermediaria, a través de interfaces de programación de aplicaciones (application programming interfaces, API), con otras aplicaciones pertenecientes a otros ámbitos, como planificadores de recursos empresariales (ERPs). También es frecuente que esta capa reaccione de forma autónoma a eventos relacionados con la capa de percepción, en función de diversas reglas programadas por los usuarios. Debido a la complejidad de esta capa, en la arquitectura de 5 capas, esta es refinada en dos capas adicionales: *Procesado*, responsable del procesado y análisis de datos, así como toma de decisiones autónomas y *Negocio*, responsable de la gestión general del sistema, sus relaciones con agentes externos y la seguridad.

2.1.1 Plataformas IoT

En la actualidad, numerosas compañías ofrecen sus plataformas para despliegues de soluciones IoT. Empresas como IBM, Microsoft y Amazon, ofrecen tecnologías destinadas a facilitar el despliegue de enjambres de dispositivos, así como las tecnologías de recolección, procesado y almacenamiento de datos para su análisis y explotación a través diversos paneles de datos y otras herramientas. Con frecuencia, estas soluciones hacen uso de tecnologías privadas y están limitadas a su uso dentro de dichas plataformas comerciales. Este abanico de ofertas y posibilidades, sin embargo, choca con la necesidad de poder compartir y acceder a la información y los recursos (dispositivos) desde otras plataformas [14].

Como resultado de la necesidad de interoperabilidad en IoT, diversos esfuerzos de estandarización entre las principales agencias mundiales han resultado en un número de plataformas y estándares diseñados específicamente para atacar esta problemática. Algunas de estas agencias de estandarización son:

- CEN: European Comittee for Standarization
- CENELEC: European Comittee for Electrotecnical Standarization
- ETSI: European Telecom Standards Institute
- W3C: World Wide Web Consortium
- ISO: International Organization for Standarization
- IEEE-SA: Institute of Electrical and Electronics Engineers Standards Activities
- OMA: Open Mobile Alliance

Dos plataformas IoT son especialmente relevantes debido al respaldo recibido tanto por la industria como por algunos de los organismos de estandarización mencionados: son FIWARE[1] [16] y oneM2M[2] [17]-[20].

FIWARE es desarrollada por la FIWARE Foundation, una organización sin propósito de lucro fundada en 2016 como el cuerpo legal independiente para el desarrollo de los objetivos de la misión, que dirige los objetivos y la adopción de estándares del proyecto (utilizando tecnologías "Open-Source"[3]). La fundación tiene entre sus miembros principales a Atos Engineering, NEC,

[1] "FIWARE - Open APIs for Open Minds". Sitio web de FIWARE, FIWARE Foundation. (2022), dirección: `https://www.fiware.org/` (visitado 18-02-2023).

[2] "oneM2M The IoT Standard, Published Specifications". Sitio web de oneM2M, oneM2M. (), dirección: `https://onem2m.org/technical/published-specifications` (visitado 18-02-2023).

[3] Código abierto (del inglés "Open-Source") o código libre, es software publicado bajo una licencia en la que el poseedor de los derechos de copia, permite el derecho de uso, estudio, cambio y distribución del software, así como de su código fuente a todo el mundo y para cualquier propósito.

Red Hat, Telefónica, Trigyn Technologies, Amazon Web Services (AWS) e incluso a la Universidad de Murcia.

FIWARE proporciona un conjunto de API y modelos de datos para desarrollar aplicaciones y servicios inteligentes. Gracias a su enfoque modular, basado en el código libre y la adopción de estándares, ha ganado en popularidad, siendo ampliamente utilizada en proyectos relacionados con Ciudad Inteligente, Industria 4.0 o Agricultura Inteligente [3], [5].

FIWARE ofrece un catálogo de componentes[4], llamados "Generic Enablers (GEs)", para (entre otros) la gestión y procesamiento de datos en tiempo real, así como la visualización y la seguridad, siendo el *Broker* el componente central de la arquitectura (Figura 2.2). Este elemento es el encargado de la recepción, almacenamiento y distribución de la información del sistema y es el principal responsable de la escalabilidad, flexibilidad e interoperabilidad de FIWARE. Está basado en el estándar NGSI para datos enlazados (NGSI for linked data, NGSI-LD) [22], definido por el ETSI-CIM[5]y heredero del estándar interfaz de servicio de nueva generación (next generation service interface, NGSI) previamente definido por la OMA. NGSI-LD no solo define las interfaces de comunicaciones, sino también los modelos de datos a utilizar, que están enriquecidos con información semántica utilizando JSON para datos enlazados (JSON for linked data, JSON-LD) como lenguaje de intercambio. Esta orientación hacia tecnologías habilitadoras de la Web Semántica, permiten a FIWARE beneficiarse del abundante catálogo de herramientas y servicios web existentes en su desarrollo, además de convertirlo en un sólido candidato para su explotación en la web de las cosas (web of things, WoT).

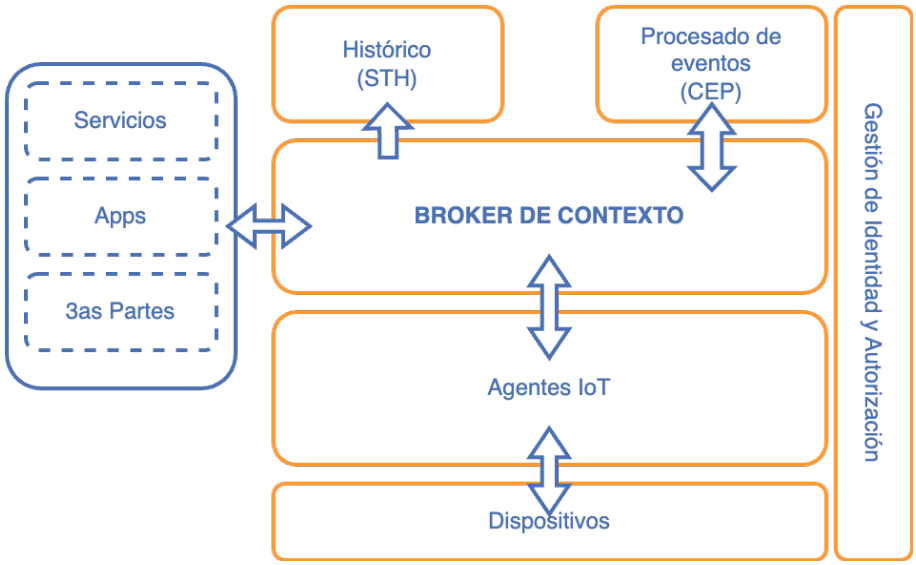


Figura 2.2: Estructura general de la arquitectura FIWARE

[4]"FIWARE Catalogue". Sitio web del catálogo de componentes FIWARE, FIWARE Foundation. (2022), dirección: `https://www.fiware.org/catalogue/` (visitado 06-03-2023).

[5]El ISG-CIM (Industry Specification Group on cross-cutting Context Information Management) es un grupo perteneciente al ETSI, al que pertenecen la FIWARE Foundation y NEC, entre otros, cuyo objetivo es facilitar el intercambio de información. Sitio web del ETSI-CIM: `https://www.etsi.org/committee/cim`

OneM2M, también estandarizado por el ETSI, proviene de un proyecto de cooperación global fundado en 2012 por ocho de las principales organizaciones de estándares de las TIC (tecnologías de la información y las comunicaciones) mundiales[6]. Ofrece una arquitectura IoT horizontal que permite la interacción entre aplicaciones y dispositivos incluso a través de silos y verticales[7](Figura 2.3), basada en estándares, que proporciona una infraestructura software para el desarrollo e implementación de soluciones IoT. Está diseñado con la interoperabilidad como objetivo, tanto a nivel de dispositivos y redes como otras plataformas de IoT, proporcionando un conjunto estandarizado de API y modelos de datos para la creación de aplicaciones, con un marcado enfoque orientado a servicios, organizados en capas funcionales que ofrecen distintos tipos de servicios, como administración de dispositivos, administración de datos y habilitación de servicios.

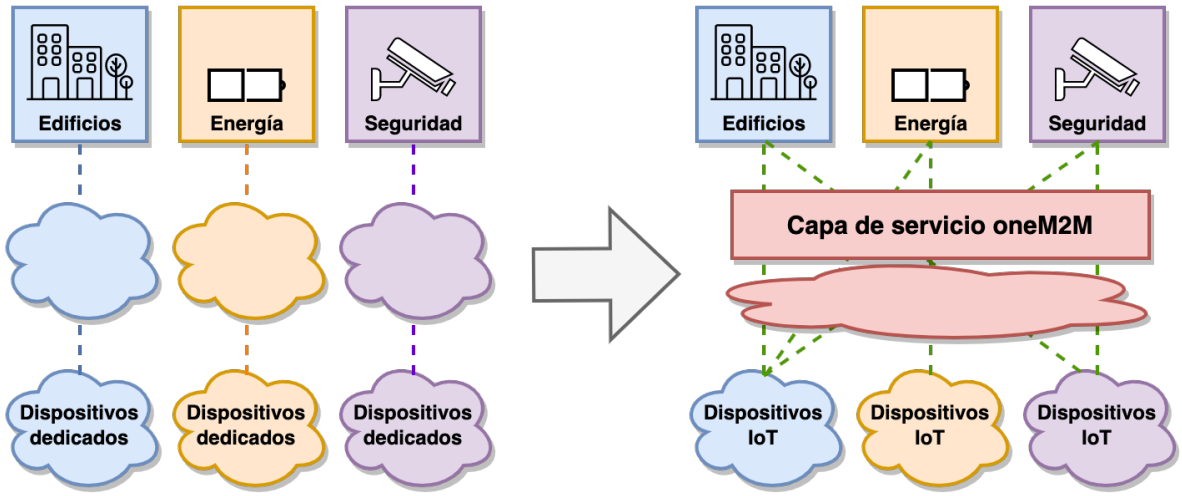


Figura 2.3: Estructura de la arquitectura oneM2M comparado con enfoque tradicional

2.1.2 La Web of Things

Como respuesta a los nuevos retos a los que se deben enfrentar los desarrolladores de soluciones para IoT, a principio del año 2000 comienzan a proponerse aproximaciones relacionadas con el uso de las tecnologías web [23]. A principios de 2010, Guirard, Tarifa y Wilde [24] presentan una arquitectura transferencia de estado representacional (representational state transfer, REST) para IoT. A lo largo de esta segunda década, se comienza a afianzar el nuevo enfoque, en el que se aprovecha las fortalezas de tecnologías y plataformas existentes para resolver un nuevo conjunto de problemas, dando luz al concepto de la web de las cosas (web of things, WoT) [25]-[27].

[6]Organizaciones fundadoras de oneM2M: ARIB (Japón), ATIS (EEUU), CCSA (China), ETSI (Europa), TIA (EEUU), TSDSI (India), TTA (Corea) y TTC (Japón).

[7]Vertical, en ciencia de la computación, se refiere a una aplicación o sistema que sirve una función específica de la industria o negocio. Diseñada para realizar una tarea específica dentro de un rango limitado de funciones y usualmente adaptada para satisfacer las necesidades específicas de una industria o negocio concreto.

Los retos a que se enfrentan los desarrolladores de soluciones IoT provienen principalmente de la necesidad de entender un escenario tecnológico heterogéneo, consistente en sistemas y servicios provenientes de diferentes fabricantes, que se comunican a través de diversos protocolos de comunicaciones, utilizando diferentes modelos de datos para el intercambio de información y con diversos requisitos de seguridad. El desarrollo de aplicaciones IoT implica dedicar un elevado esfuerzo para atacar la complejidad mencionada anteriormente. Un esfuerzo que es aplicado, frecuentemente, a un caso de uso específico y limitado que, además, durante su vida útil será difícil de mantener, extender y reutilizar.

En este escenario, la WoT se presenta como un candidato para resolver los problemas de interoperabilidad tanto a nivel de interfaces como de datos, así como atacar la diversidad de modelos existentes y al mismo tiempo ofrecer una solución competitiva para generar modelos de negocio que se basen en los datos.

Buscadores, plataformas de datos, seguridad, control de acceso son algunas de las características deseadas y fácilmente obtenibles con la WoT, aunque para entenderla primero es necesario explicar los dos desarrollos previos sobre los que se sustenta: la arquitectura REST y la Web Semántica.

2.1.2.1 Arquitectura REST

En el año 2000, Roy T. Fielding presenta su tesis doctoral [28], [29] definiendo los principios de arquitectura de la Web actual, conocidos como transferencia de estado representacional (representational state transfer, REST) en la que, a través del uso del protocolo de transferencia de hipertexto (hyper-text transfer protocol, HTTP) y los indicador de recurso universal (universal resource indicator, URI) establece los principios de ingeniería del software que guían el desarrollo de aplicaciones y servicios web. Esta arquitectura se basa en los siguientes principios:

1. *Arquitectura cliente-servidor:* este patrón impone la separación de intereses entre los dos extremos de la comunicación, permitiendo la simplificación de los mismos, así como su desarrollo independiente. De esta forma existen multitud de implementaciones diferentes de servidores web, así como clientes, navegadores y tecnologías que hacen uso de los servicios. Esta característica es imprescindible en la escala de Internet, donde el desarrollo conjunto de ambos extremos impondría una carga insostenible a la evolución de los sistemas.

2. *Carencia de estado:* en ciencia de la computación, la carencia de estado se refiere a que el servidor no almacena información relativa a la sesión. En su lugar, esta información es frecuentemente transmitida al cliente, quien deberá almacenarla y comunicarla al servidor en sus interacciones. Esta característica permite que estos sistemas sean altamente escalables, al reducir la carga de trabajo adicional del servidor en mantener la sesión y además permitir la adición de recursos de computación horizontalmente ya que cada interacción contiene toda la información necesaria para su tratamiento.

3. *Cacheado:* tanto clientes como intermediarios pueden almacenar temporalmente respuestas previas del servidor. Esta restricción, en parte facilitada por la carencia de estado, permite un uso mucho más eficiente de los recursos tanto a nivel de servidor como de comunicaciones, mejorando escalabilidad y rendimiento.

4. *Sistema por capas:* la arquitectura permite que los dos extremos de la comunicación realicen sus intercambios de manera totalmente ajena a los elementos intermedios que se encuentren. Por ejemplo, balanceadores de carga o proxies permiten almacenar temporalmente recurso para un acceso más rápido o permitir la escalabilidad del sistema de manera transparente a las partes. De igual forma, la seguridad puede ser introducida como capas intermedias en la arquitectura, forzando la aplicación de directivas de seguridad.

5. *Interfaz uniforme:* esta restricción, fundamental en el desarrollo de sistemas REST, se divide a su vez en cuatro principios: 1) Identificación de los recursos en las peticiones, utilizando URI. Además, el concepto de recurso es independiente del de representación, así podremos tener un mismo recurso representado como lenguaje extensible de marcas (extendible markup language, XML), notación de objetos de Javascript (Javascript object notation, JSON), o lenguaje de marcas de hipertexto (hyper-text markup language, HTML). 2) Manipulación de los recursos a través de representaciones: toda la información para poder manipular el recurso debe estar encapsulada en la representación. 3) Mensajes autodescriptivos, de tal forma que cada mensaje contenga toda la información para su interpretación (por ejemplo, indicando el tipo de representación) y finalmente 4) Utilización de hipermedios como motor del estado de la aplicación, por el que un cliente REST debe poder ser capaz de descubrir dinámicamente todos los recursos que necesite, por ejemplo, por medio del uso de hipervínculos.

2.1.2.2 Web Semántica

Posteriormente, a raíz de la explosión de información disponible en la Web, surge el concepto de la Web Semántica (a veces llamada Web 3.0). Se trata de una extensión de la WWW (World Wide Web) cuyo objetivo principal es permitir a las máquinas un mejor acceso a la información. Para ello es preciso que la información contenida en la Web no solo esté más estructurada sino que contenga metainformación[8], permitiéndoles entender el significado (semántica) del contenido. Como resultado de ese nuevo entendimiento de los datos, más profundo, las máquinas podrán ser intermediarias de la información, proveyendo apoyo a los humanos en la búsqueda y extracción de información, habilitando incluso la inferencia de nuevo conocimiento.

Para conseguirlo, la información en la Web debe ser organizada de tal forma que sea entendible por ambos, humanos y máquinas. Algunas de las tecnologías habilitadoras de la Web Semántica son RDF y OWL (explicados más abajo) como modelos para el almacenamiento de

[8]Metainformación o metadatos son, literalmente, datos que describen otros datos.

metainformación y el lenguaje SPARQL [30] que permite realizar consultas sobre conjuntos de datos RDF. Estas tecnologías permiten que en la Web Semántica, la información esté enlazada e integrada, permitiendo su consulta a través de múltiples orígenes y dominios y permitiendo a las máquinas procesar y razonar sobre la información. Las aplicaciones son diversas, desde mejoras en los resultados de motores de búsqueda al desarrollo de agentes inteligentes y sistemas de aprendizaje computacional.

El marco de descripción de recursos (resource description framework, RDF), introducido por Schreiber y Raimond [31] y posteriormente refinado por Mcbride [32], es un lenguaje para describir y modelar información en la Web. Permite hacer declaraciones sobre recursos, utilizando tripletas sujeto-predicado-objeto. Mediante estas se puede expresar relaciones entre cosas utilizando una estructura simple "X es Y de Z", por ejemplo: *"Fernando es padre de Alicia"* o *"París es la capital de Francia"*, donde *Fernando* y *París* son sujetos, *es padre* y *es la capital* son predicados y *Alicia* y *Francia* son objetos.

Dos características clave de RDF son su capacidad para utilizar cualquier URI como identificador de recursos, así como su gran capacidad para representar conocimiento, lo que permite su extensión o adaptación, permitiendo definir vocabularios y ontologías sobre él.

RDF schema (RDF-S) [33] es un vocabulario construido sobre RDF, consistente en un conjunto de clases y propiedades que proveen elementos básicos para la descripción de ontologías. Se puede decir que RDF provee el modelo para construir grafos de conocimiento, mientras que RDF-S define como deben relacionarse los nodos del grafo. Define, por ejemplo, los conceptos de clase y subclase (*"una Persona es una subclase de Mamífero"*), así como las relaciones que se permiten entre estas (*"las Personas pueden tener un predicado trabajaEn que se conecta a un objeto de clase PuestoDeTrabajo"*).

Otros lenguajes han derivado de RDF-S, inspirándose en él, siendo especialmente relevante "Schema.org": un conjunto de esquemas representados en diversos códigos (como RDF y JSON-LD) ampliamente utilizado en la Web para introducir metainformación de ayuda a navegadores y otras aplicaciones a través de atributos embebidos en el código HTML (ver Listado 1). Estos vocabularios fueron fundados inicialmente por Google, Microsoft, Yahoo y Yandex y son desarrollados en un proceso de comunidad abierta.

Por último, lenguajes como el lenguaje de ontologías web (web ontology language, OWL) aumentan la expresividad de RDF-S, utilizando a este último como base para su definición. OWL fue definido por el OWL Working Group, un grupo de trabajo que se cerró en 2004, concluyendo con la presentación de OWL 2, por Bao, Calvanese, Cuenca Grau et al. [34]; una "W3C Recommendation" (recomendación del W3C), en que se refina y extiende el OWL original.

Lenguaje de ontologías web (web ontology language, OWL) es una familia de lenguajes para la creación de ontologías, muy utilizado en la Web Semántica. Estas ontologías son representaciones formales del conocimiento, que permiten los objetivos mencionados al comienzo de este apartado: permitir un mejor acceso a la información por parte de las máquinas, habilitando una mayor

```
<div itemscope itemtype="https://schema.org/ScreeningEvent">
  <h1 itemprop="name">Tiburón</h1>
  <div itemprop="description">Cuando un tiburón asesino...</div>
  <p>Localización: <span itemprop="location" itemscope
  ↪  itemtype="https://schema.org/MovieTheater">
      <span itemprop="name">Cines ACME</span>
      <span itemprop="screenCount">10</span>
    </span>
  </p>
  <div itemprop="workPresented" itemscope itemtype="https://schema.org/Movie">
    <span itemprop="name">Tiburón</span>
    <link itemprop="sameAs" href="www.imdb.com/title/tt0073195/"/>
  </div>
  <p>Idioma: <span itemprop="inLanguage" content="es">Español</span></p>
  <p>Formato: <span itemprop="videoFormat">Panavision</span></p>
</div>
```

Listado 1: HTML anotado con microcode utilizando el vocabulario Schema.org

facilidad a la hora de compartir y reutilizar la información, así como realizar razonamiento automatizado sobre ella.

La familia de OWL incluye los sublenguajes (o dialectos) OWL Lite, OWL DL (o Description Logic) y OWL Full, siendo cada uno un subconjunto del siguiente. El motivo de la existencia de estas diferentes versiones del lenguaje radica en ofrecer diferentes niveles de expresividad a la hora de representar la información, así como distintos niveles de complejidad computacional a la hora de poder realizar razonamiento sobre los datos representados. La versión más simple es OWL lite, que solamente utiliza parte de las características del lenguaje OWL y presenta más limitaciones que sus hermanos. Fue originalmente definido bajo el pretexto de que resultaría más sencillo realizar herramientas que fueran compatibles con este estándar, en lugar de sus otros hermanos. En la práctica es poco utilizado, siendo prácticamente equivalente a OWL DL, que es más expresivo.

La principal diferencia entre OWL DL y OWL Full radica en que el primero consigue la máxima expresividad posible del lenguaje, reteniendo las propiedades de ser completo[9]y decidible[10], mientras que OWL Full fue diseñado para ser semánticamente compatible con RDF-S, permitiendo una mayor expresividad a costa de perder la decidibilidad: no es posible realizar un

razonamiento completo sobre él.

Finalmente, en OWL 2 encontramos tres perfiles; algo similar a lo que teníamos en OWL con sus 3 o dialectos: OWL 2 EL (EL++ logic), OWL 2 QL (Query Language) y OWL 2 RL (Rule Language). Estos perfiles son versiones reducidas de OWL 2 que sacrifican parte del poder expresivo del mismo, a cambio de una mayor eficiencia en el razonamiento. Las diferencias principales entre estos perfiles son que OWL 2 EL tiene una complejidad de razonamiento en tiempo polinomial al estar basado en la familia EL de lógicas de descripción. OWL 2 QL está diseñado para permitir un acceso y consulta más fácil a los datos almacenados en bases de datos: su acrónimo refleja que los algoritmos de consulta para este perfil pueden ser implementados mediante lenguajes de consulta. Finalmente OWL 2 RL es un subconjunto de OWL 2 diseñado para aplicaciones que prefieran canjear parte de la expresividad del lenguaje OWL 2 completo a cambio de una mayor eficiencia en el lenguaje, siendo posible realizar razonamiento sobre este utilizando lenguajes de reglas.

2.1.2.3 W3C y la Web of Things

Fundada en 1994 por Timothy John Berners-Lee, inventor de la World-Wide Web (WWW), el World Wide Web Consortium (W3C)[11] es una comunidad internacional donde organizaciones miembro, personal a tiempo completo y el público general trabajan juntos para desarrollar estándares para la Web, como RDF, RDF-S o OWL.

En la actualidad, además de los mencionados estándares, el W3C ofrece una "Candidate Recommendation" para la arquitectura de la WoT [36]. De acuerdo al sitio web dedicado a WoT en el W3C[12], esta arquitectura *"... busca contrarrestar la fragmentación existente en IoT utilizando y extendiendo los estándares web existentes. Al aportar metadatos estandarizados y otros componentes tecnológicos reutilizables, [esta arquitectura permite] una fácil integración entre plataformas IoT y los distintos dominios de aplicación."*

Los dominios de aplicación que plantea el W3C en el ámbito de la WoT incluyen (aunque no se limitan a):

- El espacio de consumidores: haciendo hincapié en el caso del hogar inteligente (hogar inteligente (smart home, SH)), la automatización, acceso remoto, control con voz, gestión energética, seguridad y la integración con servicios existentes.

[9] En lógica matemática, una teoría es completa si es consistente y para cada fórmula cerrada en el lenguaje de la teoría, o bien es demostrable o su negación lo es.

[10] En metalógica, la decidibilidad es una propiedad de los sistemas formales cuando, para cualquier fórmula en el lenguaje del sistema, existe un método efectivo para determinar si esa fórmula pertenece o no al conjunto de las verdades del sistema.

[11] "World Wide Web Consortium (W3C) - making the Web work". Sitio web de W3C, World Wide Web Consortium (W3C). (2023), dirección: `https://www.w3.org` (visitado 06-03-2023).

[12] "W3C Web of Things". Sitio web de W3C para Web of Things, World Wide Web Consortium (W3C). (2023), dirección: `https://www.w3.org/WoT/` (visitado 06-03-2023).

- Espacio industrial: presenta el ejemplo de la factoría inteligente (Smart Factory) y la integración con protocolos de comunicaciones como PROFINET, Modbus, OPC UA, TSN, EtherCAT o CAN. Sus objetivos incluyen la monitorización remota utilizando paneles de mandos, así como el análisis de los mismos para dictar pautas de mantenimiento preventivo e indicadores clave de productividad (KPI).
- Ciudades inteligentes: con la monitorización de equipamiento y mobiliario urbano para mantenimiento y reparación, así como la monitorización de vías de transporte y comunicación para optimización de tráfico, optimización y seguimiento de plazas de aparcamiento (Smart Parking), colección inteligente de basura, etc.
- Vehículo conectado: monitorización del estado de operación, predicción de necesidades de servicio, optimización del mantenimiento y otros servicios adicionales como monitorización y detección de patrones de tráfico.

También ofrece diversos patrones de despliegue comunes, que ilustran como las cosas (Things) interactuarán con controladores u otros dispositivos, agentes y servidores. Ejemplos de patrones para telemetría, acceso remoto, pasarelas para hogar inteligente, gemelos digitales y cosas virtuales son algunos de los patrones presentados en la arquitectura.

La arquitectura presentada (ver Figura2.4), gira entorno al concepto de *Thing* (cosa), una abstracción de una entidad física o virtual. Esta es a su vez descrita en metadatos estandarizados, por una *Thing Description (TD)* (descripción de cosa). Este modelo estandarizado de representación de descripciones de cosas, permite que sean interpretables por máquinas e incluso el descubrimiento de las capacidades de las mismas utilizando el lenguaje de representación y serialización de datos enlazados JSON-LD [38]. Para realizar dicho descubrimiento, un consumidor podrá consultar al propio dispositivo IoT directamente o consultar a un repositorio externo (*Thing Description Directory*).

La *Thing Description* es el bloque central de construcción en WoT, el punto de entrada de una instancia IoT, como sería la página “index.html” en una web común. En esta podemos encontrar información sobre funciones, protocolos y representación y estructura de la información y mecanismos de control de acceso y seguridad entre otros.

La arquitectura pretende ser agnóstica al protocolo de comunicación IoT subyacente (tales como MQTT, ZigBee, CoAP o HTTP), proveyendo de mecanismos para interaccionar con dichos protocolos a través de los *Binding Templates*. Estos ofrecen guías sobre como un cliente puede activar cada interacción abstracta, a través de la interfaz correspondiente al protocolo de dispositivo.

2.1.3 Sistemas de gestión energética en Smart Home

Como ya se ha mencionado con anterioridad en la introducción, uno de los puntos de interés generales de esta tesis es la gestión energética, específicamente en el entorno de los edificios

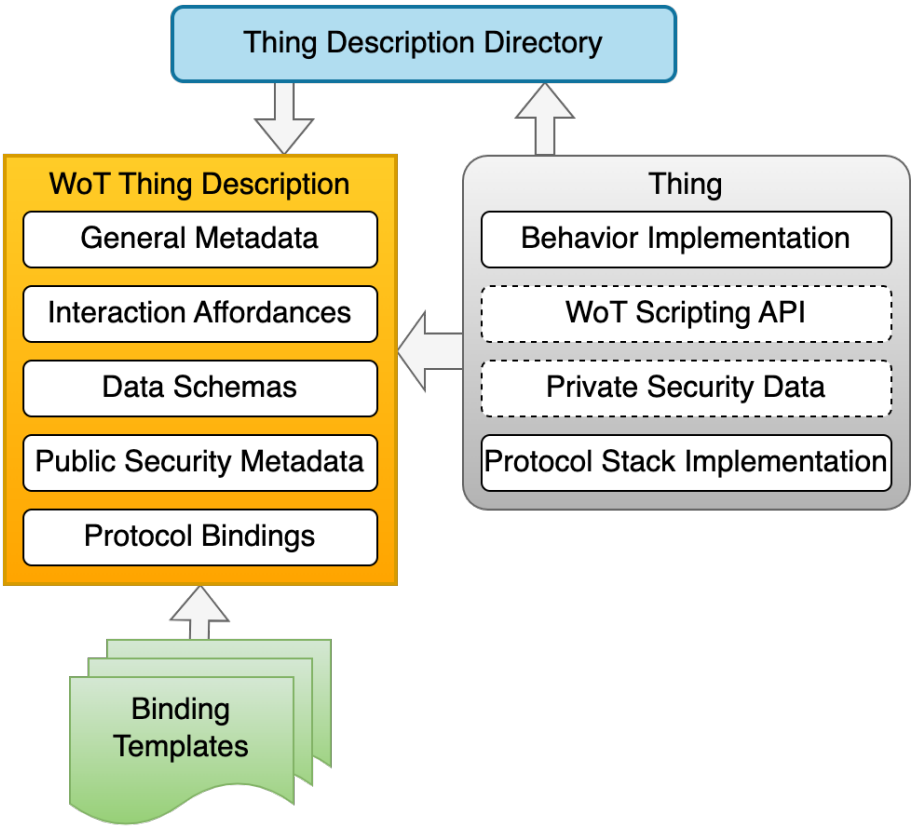


Figura 2.4: Arquitectura de la Web of Things según W3C

inteligentes y el hogar inteligente. En estos campos, la aplicación de plataformas y arquitecturas de IoT presenta nuevos retos y posibilidades de implementar aproximaciones novedosas para la gestión eficiente de los recursos energéticos, la optimización del consumo y la interacción con el sistema de distribución energético. Algunos de los retos que se enfrentan en estos escenarios, que deberán ser debidamente atacados en las propuestas de arquitectura a considerar, están relacionados con la gran diversidad de interfaces, dispositivos y agentes a considerar, así como consideraciones de privacidad y seguridad de los datos.

Los sistemas de gestión energética del hogar (home energy management systems, HEMS) [39], [40] tienen como objetivo ayudar a a gestionar y optimizar el uso de la energía, incrementando la eficiencia energética de los hogares, ayudando de esta forma a reducir costes a los propietarios y de una forma más amplia, incrementar la sostenibilidad energética a nivel global.

De forma general los HEMS consisten en uno o varios dispositivos instalados en el hogar y software de control, que monitorizan continuamente el uso de energía proporcionando datos en tiempo real al propietario o los ocupantes. Estos datos sirven potencialmente para identificar áreas de desperdicio energético, permitiendo en consecuencia el ajuste de su uso. La forma más común de interacción del usuario con el HEMS es a través de aplicaciones de teléfono móvil o portales web.

Los sistemas de gestión energética del hogar inteligente (smart home energy management systems, SHEMS) [41]-[43], por otro lado, difieren de los HEMS en capacidades y sofisticación, siendo característica la incorporación de algoritmos de inteligencia artificial y aprendizaje automático para la optimización del uso de energía. El aprendizaje de hábitos y preferencias de los ocupantes permite una mejor gestión energética, la reducción del consumo y el consiguiente ahorro económico.

Otro objetivo añadido de los SHEMS, es la integración con dispositivos domésticos inteligentes (domóticos), algunos de ellos con un elevado consumo energético asociado; como termostatos para sistemas de calefacción, ventilación y aire acondicionado (heating, ventilation and air conditioning, HVAC) y el vehículo eléctrico, así como otros más generales como electrodomésticos, iluminación, elementos de envolvente del hogar (persianas, cortinas, ventanas, etc...) y otros elementos para el ocio, como dispositivos multimedia. Estos sistemas utilizan una variedad de sensores y dispositivos para recopilar datos sobre los patrones de consumo de energía dentro del hogar. Posteriormente, estos datos pueden ser procesados y analizados mediante algoritmos de aprendizaje automático, para ofrecer información y recomendaciones a los ocupantes, sobre como optimizar el uso de la energía, e incluso es posible llegar más allá, ajustando automáticamente el consumo de energía según las preferencias del usuario, las condiciones climáticas y otros factores.

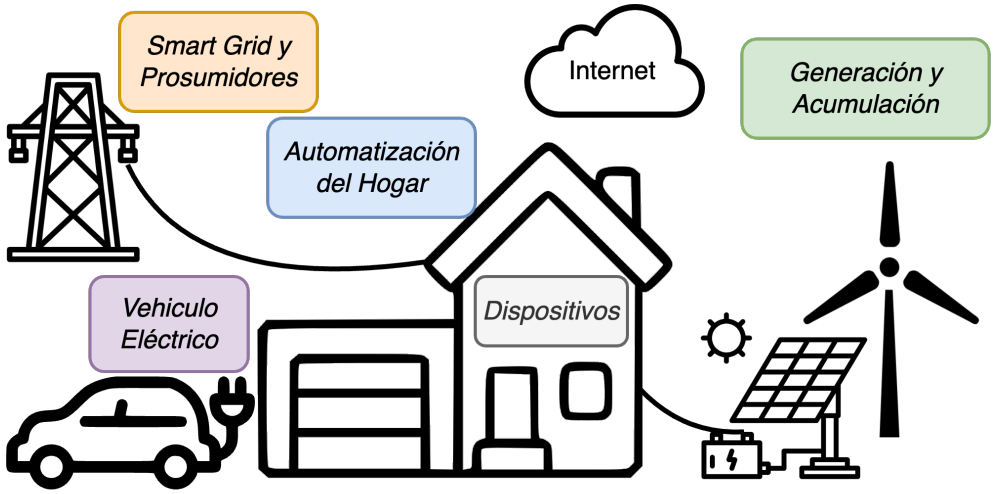


Figura 2.5: Visión global de la gestión energética en el hogar inteligente

Por último, la integración completa del SHEMS con el resto de los sistemas del hogar inteligente, permite no solo una gestión más avanzada del uso energético y la automatización de ciertos sistemas, sino un mayor y más profundo conocimiento del entorno del hogar y sus dependencias con agentes externos en materia energética, que permite nuevas sinergias no solo entre ocupantes y hogar inteligente, sino también con la red de producción y distribución eléctrica. Esto es aún más necesario tras la incorporación y rápido crecimiento del número de prosumidores, a raíz del

abaratamiento de los paneles fotovoltaicos y su consiguiente irrupción en el mercado doméstico. Una imagen conjunta de los conceptos mencionados puede verse en la Figura 2.5.

Los beneficios potenciales de los SHEMS son numerosos. Pueden ayudar a los propietarios de viviendas a reducir sus facturas de energía, disminuir su huella de carbono y aumentar su eficiencia energética general. Además, pueden proporcionar datos valiosos a los proveedores de energía y a los agentes generadores de políticas energéticas, para informar las estrategias de gestión de la energía y mejorar la infraestructura energética global, con el potencial de alterar profundamente la forma en que consumimos, producimos y gestionamos la energía.

2.1.4 Smart Home y Smart Grid

El hogar inteligente (smart home, SH), o sistema de automatización del hogar, es un ecosistema tecnológico avanzado que permite la automatización y el control de varios aspectos de una casa. Iluminación, sistemas de HVAC, sistemas de entretenimiento, seguridad y electrodomésticos son controlados mediante la integración de dispositivos y sensores, así como software de control, conectados a una red local que a su vez puede estar conectada a Internet, permitiendo el acceso y control tanto local como remoto.

Los hogares inteligentes utilizan el IoT como estructura principal para el control y gestión de datos y sobre este aplican la inteligencia artificial y el aprendizaje automático para extraer valor de la información y ofrecer servicios avanzados a los ocupantes. Aunque la computación en la nube[13] es una opción de cara a la gestión de datos y la oferta de servicios para los propietarios de SH, también hay una amplia oferta de soluciones hospedadas localmente. Entre las soluciones comerciales basadas en la nube más conocidas para Smart Hubs se encuentran las de Google, Amazon y Apple; mientras que la comunidad creciente de entusiastas del *Do It Yourself* (hazlo tú mismo), motivados por la preocupación respecto a la privacidad y seguridad de sus datos [44] en los entornos cloud, han optado por soluciones abiertas hospedadas en servidores locales, como Home Assistant[14], Domoticz[15] y OpenHab[16].

Estas soluciones abiertas, adicionalmente, se presentan como excelentes marcos para el rápido desarrollo de nuevas ideas más allá de la automatización del hogar, lo que permite aprovechar las integraciones ya existentes de diferentes dispositivos y plataformas; aunque para ello es necesario aportar soluciones nuevas e innovadoras, como el uso de tecnologías de la web semántica para cerrar la brecha entre los diferentes servicios y dominios de la gestión energética del hogar, interfaces estándar para facilitar la comunicación con otros agentes y mecanismos de seguridad para facilitar la poner límites al acceso a la información.

[13] Del inglés *cloud computing*: computación en la nube.

[14] "Home Assistant - awaken your home". Sitio web de Home Assistant. (), dirección: `https://www.home-assistant.io` (visitado 18-03-2023).

[15] "Domoticz - control at your fingertips". Sitio web de Domoticz. (), dirección: `https://www.domoticz.com` (visitado 18-03-2023).

[16] "openHAB - empowering the smart home". Sitio web de openHAB. (), dirección: `https://www.openhab.org` (visitado 18-03-2023).

2.1.5 Arquitecturas para SHEMS

La imagen actual de un SH moderno orientado al prosumidor es un ecosistema complejo en el que la gestión de la energía tiene que cumplir con una amplia gama de cometidos (Figura 2.6). La interacción con agentes externos, como SG, Smart Hubs y otros servicios externos, plantea altas expectativas de interoperabilidad tanto en el modelado de datos y las interfaces de comunicación, como en las preocupaciones de privacidad de los habitantes del hogar [48], [49].

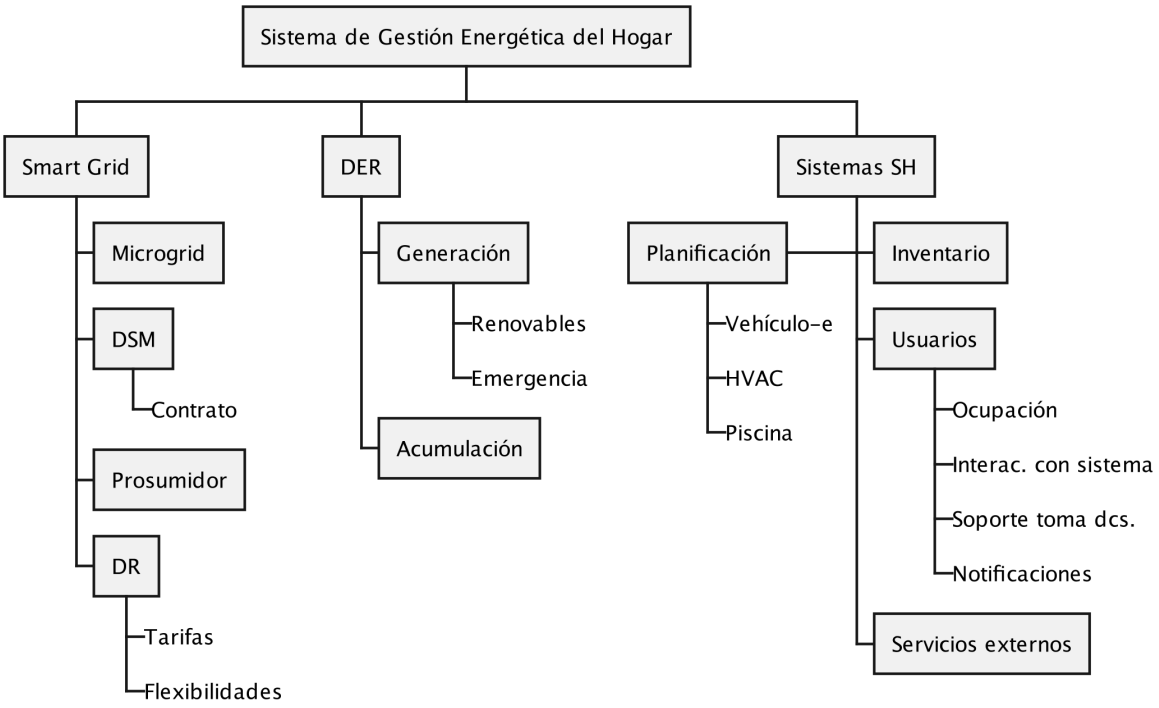


Figura 2.6: Cometidos de la gestión energética en el hogar

A continuación, procedemos a describir y analizar algunas de las contribuciones más relevantes en el ámbito de arquitecturas para SHEMS encontradas en la literatura.

En su trabajo, Machorro-Cano, Alor-Hernández, Paredes-Valverde et al. [50], presentan HEMS-IoT; un HEMS basado en el aprendizaje automático para el ahorro de energía, garantizando confort y seguridad a la vez que reduce el consumo de energía. La arquitectura propuesta consta de siete capas, desde la de presentación hasta el dispositivo, en las que se considera la seguridad de la información entre las capas de presentación, IoT servicios y gestión, contemplando tanto la autenticación como la autorización. No hace mención de la gestión de respuesta a la demanda (demand response, DR) o recurso energético distribuido (distributed energy resource, DER) como parte del HEMS propuesto, ni explora la posibilidad de su integración con sistema de automatización del hogar (home automation system, HAS) existentes, aunque su capa de gestión sí aplica una semántica enfoque a la gestión del hogar, utilizando una ontología de desarrollo propio para representar los principales conceptos domóticos. En general presenta una estructura

monolítica que va de dispositivo a presentación, donde no se ha abordado la interoperabilidad.

Elshaafi, Vinyals, Grimaldi et al. [51] exploran un enfoque para una DR automatizada y descentralizada, así como la gestión energética del hogar. La arquitectura propuesta se implementa utilizando un sistema multiagente con tres niveles bien diferenciados: hogar, agregador y operador del sistema de distribución (estando estos dos últimos relacionados con SG). A nivel de hogar, definen solo dos agentes: el agente de dispositivo y el agente HEMS. Su objetivo combinado es reducir la factura energética respetando las preferencias y el confort de los usuarios. Mientras que los agentes de dispositivos encapsulan las comunicaciones con los dispositivos domésticos, el agente HEMS es responsable del grueso del trabajo: planificar, optimizar y comunicarse con los agentes SG. Esta arquitectura no considera la existencia de un HAS anterior. La solución final se presenta en forma de un dispositivo *home gateway*, o pasarela, que interactuará directamente con los dispositivos inteligentes, aunque sí considera la gestión de DER a través de los diferentes agentes del dispositivo. Este trabajo también aborda la interoperabilidad al proponer el uso de interfaces de comunicación estándar OSGi[17] (iniciativa Open Services Gateway) e interfaces REST y el uso de OWL para el modelado de información. Finalmente, la arquitectura propuesta tiene en cuenta la privacidad y la seguridad de los usuarios domésticos mediante el uso de lenguaje extensible de marcas para control de acceso (extendible access control markup language, XACML) para el control de acceso basado en atributos, salvaguardando la visibilidad de los dispositivos domésticos para el agente HEMS.

El proyecto MAS2TERING [53] presenta una implementación del marco universal para la energía inteligente (Universal Smart Energy Framework, USEF)[18] [55]), definiendo un sistema multiagente que consta de los siguientes agentes: *Agente operador del sistema de distribución*, *Agente agregador*, *Agente central de gestión de energía doméstica*, *Agente de microgeneración*, *Agente de electrodomésticos* y *Agente de baterías* cuyas interacciones apuntan a brindar una gestión del lado de la demanda a lo largo de la cadena de suministro, desde la generación hasta los electrodomésticos de consumo. Esta arquitectura no cubre aspectos de seguridad, ni la integración con HAS existentes o la gestión de energía del hogar.

Zhang, Li y Schooler [56] proponen iHEMS, una infraestructura de comunicaciones de publicación/suscripción, utilizando Information-Centric Networking (específicamente Content Centric Networking) como la columna vertebral de la comunicación. No se se centra exclusivamente en proteger los canales de comunicación, sino en cifrar los datos en sí, utilizando un esquema de comunicaciones de grupo seguro por encima de la capa pub/sub. La arquitectura en sí contempla los diferentes dispositivos interconectados a través del sustrato pub/sub basado en Information-Centric Networking, así como un *Servicio de Directorio*, que utilizan los dispositivos para publicitar sus datos y un *Controlador de Grupo* a cargo de la clave. gestión para el

[17] "OSGi Working Group, The Dynamic Module System for Java". Sitio web de OSGi, Eclipse Foundation. (), dirección: `https://www.osgi.org` (visitado 11-03-2023).

[18] "Universal Smart Energy Framweork (USEF)". (jun. de 2021), dirección: `https://www.usef.energy` (visitado 22-04-2023).

cifrado/descifrado.

El proyecto Digital Environment Home Energy Management System (DEHEMS) [57] propone una arquitectura basada en servicios, que consta de un servidor remoto donde se implementa la base de conocimientos, que a su vez se alimenta de un *Data Collector* ubicado en el hogar, al que se conectan sensores, dispositivos, electrodomésticos y dispositivos de visualización mediante interfaces inalámbricas.

Rossello-Busquet, Soler y Dittmann [58], proponen una pasarela (home gateway) para un sistema HEMS, que controla los dispositivos en una red doméstica a nivel de servicio. Construido sobre el marco OSGi, utiliza la ontología DogOnt como base para su repositorio de datos de base de conocimiento. La arquitectura se compone de seis paquetes: *Base de conocimiento*, *Interfaz*, *Red*, *Administrador de redes*, *Administrador* y *Emulador de red*.

ThinkHome, de Reinisch, Kofler, Iglesias et al. [59], describe una arquitectura de sistema multiagente con dos premisas principales: garantizar la eficiencia energética en el hogar y optimizar el confort. Las estrategias de control realizadas por el sistema multiagente se dividen en aspectos del problema que se asignan directamente a los diferentes agentes del framework: *Control*, *Usuarios*, *Objetivos globales*, *Inferencia de contexto*, *Datos auxiliares*, *Interfaz de base de conocimiento* e *Interfaz del sistema de automatización de edificios*.

Tabla 2.1: Comparativa de arquitecturas para SHEMS

Arquitectura	**Seg.**	**Intrp.**	**HAS**	**HEMS**	**DR**	**Prosumer**
HEMS-IoT [50]	✓	✗	✗	✓	✗	✗
Elshaafi et al. [51]	✓	✗	✗	✓	✓	✓
MAS2TERING [53]	✗	✗	✗	✗	✓	✓
iHEMS [56]	✗	✗	✗	✓	✗	✗
DEHEMS [57]	✗	✗	✗	✓	✗	✗
Rossello-Busquet et al. [58]	✗	✗	✗	✓	✗	✗
ThinkHome [59]	✗	✗	✗	✓	✓	✗

Tras la revisión bibliográfica realizada en arquitectura y seguridad, resumida en la Tabla 2.1, podemos concluir que ninguno de los trabajos revisados considera en su arquitectura todas las características deseables anteriormente descritas para el SH moderno orientado al prosumidor. Ninguno aborda la interoperabilidad del modelo de datos y las comunicaciones de manera holística para permitir una interacción efectiva con otros proveedores de servicios, como plataformas de dispositivos externos, Smart Hubs y el SG. Finalmente, algunos trabajos cubren la seguridad, pero principalmente en el dominio de las comunicaciones, sin poner el foco en la privacidad de los datos, que como hemos mencionado anteriormente, representa una creciente preocupación en la

base de usuarios de SH.

2.1.6 Modelos de distribución de tareas y orquestación en IoT

En un contexto general dentro de IoT, la distribución y orquestación de tareas se refiere al proceso de administración y coordinación de tareas entre diferentes dispositivos, sensores y sistemas para lograr un objetivo específico. Esto significa, entre otras cosas, dividir tareas complejas en fragmentos más pequeños, que pueden ser más fácilmente distribuidos y ejecutados en diferentes nodos de una forma coordinada.

La distribución y orquestación de tareas es crítica en determinadas aplicaciones de IoT, ya que permiten la escalabilidad y la utilización eficiente de los recursos, así como la capacidad de manejar flujos de trabajo y procesos complejos. Esto es particularmente importante en implementaciones de IoT a gran escala donde hay muchos dispositivos y sistemas interconectados que deben funcionar juntos de manera confiable, así como sistemas donde la ejecución de la carga de trabajo ha de ser gestionada de cara a maximizar la calidad de servicio (quality of service, QoS) en términos de tiempo de ejecución y/o ancho de banda disponible.

2.1.6.1 Computación en el borde

En muchas aplicaciones emergentes de IoT, es esencial procesar datos de manera efectiva para cumplir con los requisitos rigurosos y diversos. Dadas las limitaciones en recursos de los dispositivos IoT; tanto en almacenamiento como computación, la computación en la nube se presenta como una gran alternativa para el procesamiento de datos.

Esta aproximación, sin embargo, puede generar retrasos significativos en las comunicaciones debido a que la ubicación física de los potentes equipos de la nube está lejos de los dispositivos IoT. Para abordar este problema, se ha desarrollado el "borde"[19], que acerca la computación a los dispositivos IoT (Figura 2.7). De esta forma, en lugar de enviar todos los datos a una ubicación central, el procesamiento y el almacenamiento se realizan "in situ"; en dispositivos o servidores locales, lo que puede reducir significativamente los requisitos de latencia y ancho de banda.

La computación en el borde es particularmente importante en aquellas aplicaciones donde el análisis de datos y la toma de decisiones sean sujeto de restricciones de latencia y/o ancho de banda; siendo especialmente interesante donde una respuesta más rápida y eficiente a eventos críticos sea necesaria.

La cooperación y orquestación entre nube y borde, supone un punto crucial en las arquitecturas para IoT. En [60], encontramos una fuente que proporciona a los investigadores una comprensión más profunda de la arquitectura de computación de orquestación entre el borde y la nube, así como una recopilación extensa y minuciosa de los principales trabajos en el estado del arte de arquitecturas borde para IoT, así como para procesamiento de datos en arquitecturas borde.

[19] Del inglés *edge computing*: computación en el borde de la red.

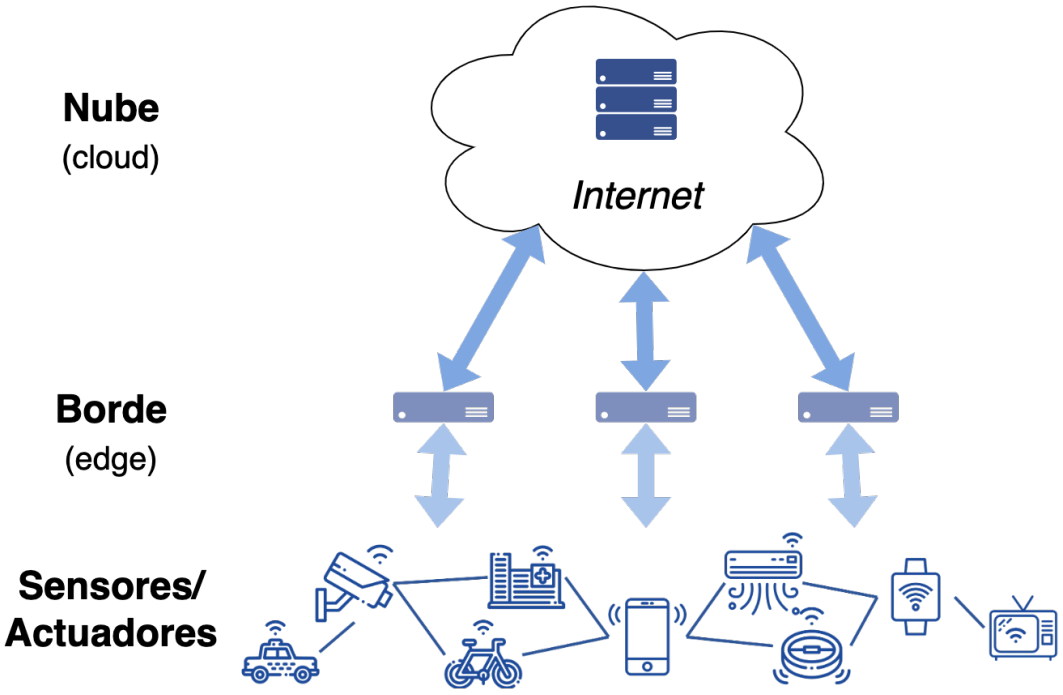


Figura 2.7: Computación en el borde: acercando la nube a IoT

Según este documento, la arquitectura y los algoritmos de procesamiento son aspectos clave que afectan el rendimiento de la orquestación borde desde la perspectiva de las aplicaciones de IoT. También investiga las propuestas de estado del arte de la orquestación IoT borde, dirigida por inteligencia artificial, detallando diferentes arquitecturas IoT de borde, para diferentes aplicaciones y problemas prácticos. Finalmente, este documento enumera y analiza los posibles desafíos de investigación y los problemas abiertos en este campo.

2.1.6.2 Computación en la niebla

En 2011 Bonomi [61] hace la primera referencia a *Fog Computing*, traducido como "computación en la niebla". Este término hace alusión a que tanto las nubes como la niebla están formadas de minúsculas gotas de agua en suspensión, la única diferencia es la altura respecto al suelo. En este sentido, la computación en la niebla supone que cualquier recurso de computación es una de esas minúsculas gotas y propone la implementación de arquitecturas que fomenten el aprovechamiento de recursos de computación tanto en términos de capacidad como de proximidad a los dispositivos productores/consumidores de datos (sensores/actuadores), dando una visión unificada tanto a la nube como a los recursos de computación en el borde y actuando de mediador entre ambos con varios propósitos, como por ejemplo el filtrado de datos. Uno de los beneficios inmediatos de esta estructura es la mejora en la eficiencia en el tráfico de datos y la reducción en la latencia de la toma de decisiones.

Posteriormente el trabajo de Bonomi, Milito, Zhu et al. [62] profundiza en el concepto y hace una caracterización de la computación en la niebla, describiendo los principios de la arquitectura, así como sus potenciales interacciones con otras tecnologías y ámbitos de aplicación, como la red eléctrica inteligente o las ciudades inteligentes.

Otro trabajo interesante es el de Wen, Yang, Garraghan et al. [63], en que presentan una arquitectura de orquestador para la niebla capaz de proporcionar la gestión centralizada del conjunto de recursos, asignando aplicaciones a solicitudes específicas y proporcionando un flujo de trabajo automatizado para la planificación y despliegue de recursos; así como gestión de la ejecución de la carga de trabajo con control de la QoS en tiempo de ejecución, aportando también la implementación de un framework basado en algoritmos genéticos paralelos (GA-Par) capaz de manejar escenarios de orquestación involucrando la composición de un gran conjunto de aplicaciones IoT.

De cara a la orquestación de tareas y microservicios, las tecnologías de contenedores (Docker, CRI-O, containerd, rktlet...) se presentan como grandes aliadas, ya que permiten el empaquetado y el despliegue de aplicaciones informáticas, así como sus dependencias, de una forma aislada y eficiente. El trabajo de Hoque, Brito, Willner et al. [64] evalúa el efecto que tienen las tecnologías de contenedores sobre el rendimiento general de las aplicaciones ejecutadas sobre nodos de computación en la niebla. También analizan diferentes herramientas de orquestación de contenedores que se presentan como candidatos prometedores y finalmente proponen un framework de orquestación de contenedores, basado en OpenIoTFog [65].

Especialmente relevante para la contribución realizada en esta tesis, es el trabajo de Cheng, Solmaz, Cirillo et al. [66], que presenta FogFlow, un framework basado en estándares para la computación en la niebla, con una marcada orientación hacia su aplicación en plataformas IoT para ciudades inteligentes. Su modelo de programación permite crear servicios IoT elásticos, sobre la nube y el borde, proponiendo el uso de interfaces estándar para compartir y reutilizar datos de contexto. El modelo de programación presentado en el documento y representado en la Figura 2.8, se basa en la familia de estándares NGSI [67] y está inspirado en Dataflow [68] de Google. Mediante el uso del modelo de datos abiertos y estandarizados y la API de NGSI, ofrecen una forma fácil de integración en muchas plataformas Smart City ya existentes basadas en NGSI.

2.1.6.3 Arquitecturas para procesado de imágenes mediante DNN en el borde

El trabajo más relevante, como resultado de la investigación bibliográfica para esta tesis, es el ofrecido por Rocha Neto, Silva, Batista et al. [69], en el que presentan MELINDA: una arquitectura para el procesamiento de flujo de vídeo distribuido, que tiene como objetivo mejorar el rendimiento computacional en sistemas inteligentes de Internet de las Cosas Multimedia (IoMT por sus siglas *Internet of Multimedia Things*) mediante la aplicación de técnicas de Fusión de Información Multinivel (MIF) sobre los flujos de vídeo.

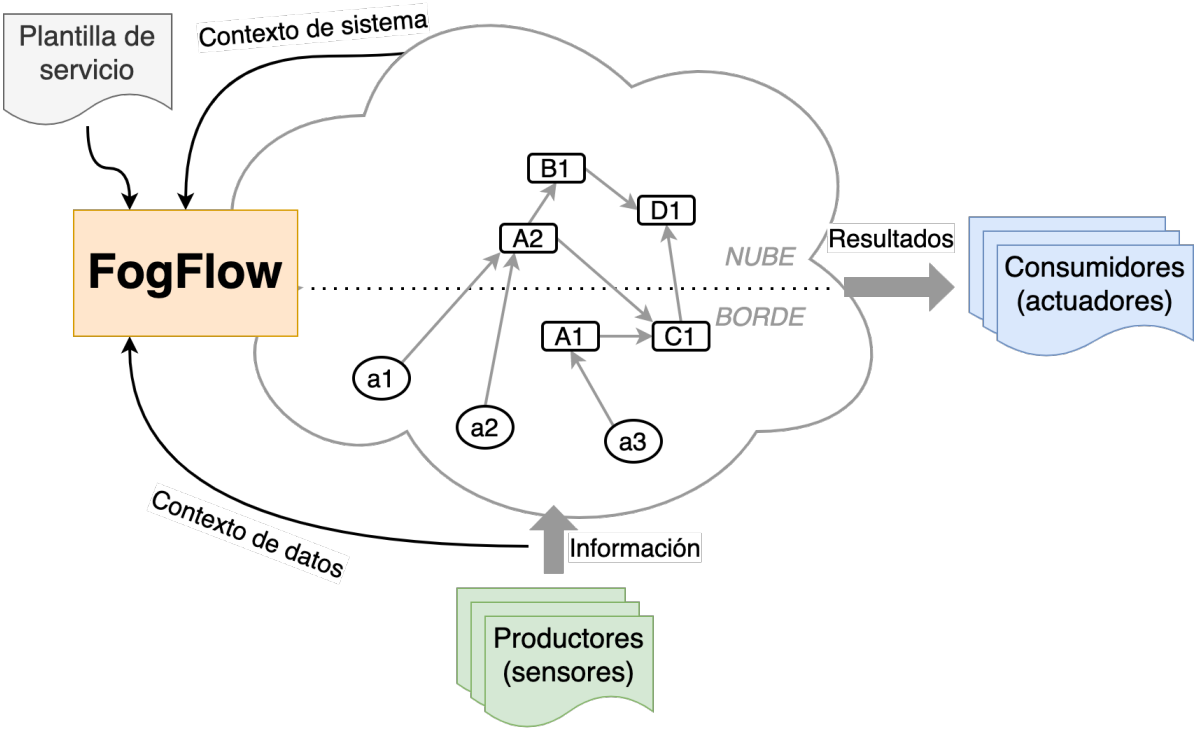


Figura 2.8: Orquestación de tareas entre nube y borde con FogFlow

La arquitectura propuesta aplica un flujo de procesamiento jerárquico de tres etapas, en el que cada flujo de entrada pasa por los siguientes pasos o tareas:

1. *Measurement Level Task (MLT)* (tarea de nivel de medición): en la que primero se procesa el flujo de vídeo de la cámara, seleccionando solo aquellas imágenes que tienen el objeto de interés.

2. *Feature Level Task (FLT)* (tarea de nivel de característica): en la que se identifican los objetos.

3. *Decision Level Task (DLT)* (tarea de nivel de decisión): en la que finalmente se interpreta el evento para la posterior toma de decisiones.

La arquitectura va más allá, distribuyendo la ejecución de esas tareas en diferentes nodos perimetrales, mediante el uso de contenedores Docker para encapsular la funcionalidad de la tarea, incluyendo un *Processing Task Repository Manager (PTRM)* (gestor de repositorio de tareas de procesado) para almacenar esas imágenes Docker. También propone aprovechar la potencia mejorada de los dispositivos que disponen de GPU para la aceleración del procesado de imagen, como las microcomputadoras Jetson Nano, para realizar un análisis de aprendizaje automático avanzado en las transmisiones de vídeo.

La orquestación de las tareas de procesamiento es realizada por el *Intelligence Orchestrator Operation Process* (proceso operativo de orquestación de inteligencia), que tiene en cuenta los parámetros de QoS para garantizar que se alcance un número mínimo de fotogramas por segundo, a fin de seleccionar los nodos de borde para ejecutar las tareas de procesamiento.

Finalmente, las comunicaciones entre las distintas capas o tareas del proceso de procesamiento de imágenes, que se alimentan de los datos del paso anterior, se realizan de forma asíncrona a través de los mecanismos de publicación/suscripción que ofrece el broker de mensajes *Data Communication Manager* (gestor de comunicaciones de datos).

En el apartado de procesado acelerado de imágenes en el borde, encontramos diversos trabajos. Empezando con el de Abdel Magid, Petrini y Dezfouli [70], en el que nos presentan un extenso estudio sobre la viabilidad y el rendimiento en la clasificación de imágenes utilizando dispositivos IoT, explorando las relaciones entre diferentes factores que pueden afectar al consumo energético en los nodos de procesamiento de imagen. Este estudio se centra en enfoques que no son redes neuronales profundas (deep neural networks, DNN) para el procesamiento de imágenes y revela que también podrían ser posibles otras alternativas utilizando hardware no acelerado.

Siguiendo con el procesado acelerado de imágenes en el borde, Lage, Santos, Junior et al. [71] proponen un sistema de vigilancia IoT de bajo coste, usando aceleración del procesado de imagen por hardware y redes neuronales convolucionales, usando como plataforma hardware la Raspberry Pi, junto con *Intel Movidius Neural Compute Stick (NCS)* y algoritmos de aprendizaje profundo de última generación, en una solución IoT de bajo costo, capaz de realizar detección de objetos in situ. Este trabajo demuestra exitosamente los beneficios de la aceleración hardware aplicada a las tareas de reconocimiento de imagen utilizando dispositivos embebidos, proporcionando rendimientos casi en tiempo real.

2.1.7 Conclusiones

Tras la revisión bibliográfica realizada en arquitectura y seguridad, podemos concluir que ninguno de los trabajos revisados considera en su arquitectura todas las preocupaciones descritas de un SH moderno orientado al prosumidor, descritas en la Sección 2.1.5. Ninguno aborda la interoperabilidad del modelo de datos y las comunicaciones simultáneamente para permitir una interacción exitosa con otros proveedores de servicios, como plataformas de dispositivos externos, Smart Hubs y el SG.

Adicionalmente, la seguridad es cubierta por algunos trabajos, pero principalmente en el dominio de las comunicaciones, sin poner el foco en la privacidad de los datos, lo que representa una gran preocupación en el complejo ecosistema de SH y su interacción con el SG.

Por otro lado, el estudio del arte en el campo de la distribución de tareas y orquestación para el procesado de imágenes en el borde revela pocos trabajos. La arquitectura MELINDA parece la candidata más prometedora, aunque algunos aspectos pueden ser sujeto de mejora.

Para empezar, la arquitectura MELINDA utiliza comunicaciones asíncronas entre operadores, basadas en mecanismos de publicación/suscripción centralizados. Otras tecnologías investigadas en el estado del arte establecen un mecanismo similar, aunque distribuido y basado en contexto. Esto puede parecer una pequeña diferencia, pero las implicaciones pueden ser sustanciales: un mensaje puede considerarse efímero, mientras que el contexto se almacena, lo que abre la posibilidad de consumir y utilizar los resultados del procesamiento de diferentes maneras. Además, el enfoque distribuido de FogFlow lo hace más robusto, al no depender de un único elemento centralizado. Finalmente, FogFlow hace uso de una API NGSI estandarizada, por lo que se mejora la interoperabilidad y su integración con otros sistemas.

Además, la arquitectura MELINDA ofrece recomendaciones como el uso de JSON como formato de intercambio de datos para comunicaciones entre tareas, pero no proporciona ni sugiere ningún modelo de datos para la implementación de descriptores de tareas, ni ningún tipo de integración con tecnologías para la web semántica, lo que también limita la interoperabilidad de la arquitectura.

Finalmente, MELINDA no proporciona ninguna implementación específica para sus componentes, siendo puramente hipotética, dejando muchos aspectos abiertos para la implementación.

2.2 Seguridad y privacidad de datos en IoT

El IoT es la columna vertebral de un gran número de aplicaciones que se han convertido en parte importante de nuestras vidas. Dispositivos, objetos y sensores son conectados para que puedan comunicarse con la nube y entre sí; intercambiar datos y realizar diversas tareas de forma autónoma. El avance de IoT ha acarreado un aumento exponencial en el volumen de datos que se generan y transmiten a través de la red, lo que además de nuevas oportunidades para empresas, gobiernos e individuos, también ha traído a nuevos desafíos de seguridad y privacidad.

De acuerdo con CISCO, en su informe anual sobre Internet [72] las conexiones máquina a máquina (machine-to-machine, M2M) representarán el 50 % en 2023 del total global de dispositivos conectados y conexiones, con un estimado de 14,7 miles de millones de conexiones M2M en 2023. El Ericsson Mobility Report [73], por por su parte, predice que llegaremos a 34,7 miles de millones de conexiones IoT en 2028.

El estado actual de IoT, aunque crece a un ritmo constante, aún adolece en el campo de la seguridad [74]. Ataques basados en el IoT, como el aun presente y famoso botnet Mirai [75], nos mostró que la seguridad no se puede pasar por alto, incluso en el caso de dispositivos aparentemente inofensivos con un poder de cómputo y capacidad de almacenamiento muy reducidos que no producen datos, o cuyos datos producidos ni siquiera se consideró que tuvieran algún tipo de valor estratégico, como cámaras IP públicas y routers ADSL (asymmetric digital subscriber line, linea asimétrica de subscriptor digital). Debido a su abundancia, esos dispositivos se pueden usar para producir ataques masivos capaces de causar grandes disrupciones en el

acceso a servicios en Internet. Quizás igualmente preocupante es el hecho de que técnicas modernas para el análisis de datos, tales como la aplicación de redes neuronales profundas (deep neural networks, DNN) abren nuevas posibilidades para inferir información privada siguiendo estrategias que son difíciles de predecir.

La posibilidad de reclutar fácilmente un número de dispositivos IoT, a merced de sus deficiencias de seguridad, para saturar un sistema usando ataques de denegación de servicios [76], no hace sino subrayar las conclusiones de Hwang [77] sobre la existente demanda de soluciones de seguridad, capaces de admitir plataformas de múltiples perfiles con diferentes niveles de seguridad.

La seguridad es una preocupación generalizada en todas las Tecnologías de la Información y las Comunicaciones que aún está ganando preocupación en el campo de IoT. Las limitaciones de diseño en términos de costes de producción y de consumo energético impulsan el continuo y acelerado desarrollo de nuevos dispositivos. Los dispositivos constreñidos, que sacrifican potencia de cálculo y complejidad en pro de un consumo energético reducido, a menudo lo hacen a expensas de la seguridad.

Numerosos estudios [78]-[82] nos muestran los abundantes riesgos, desafíos y amenazas involucrados en la seguridad de IoT, como la privacidad, los dispositivos constreñidos que carecen de potencia criptográfica y el malware dirigido a IoT. IoT está extendiendo la superficie de ataque de sus sistemas a través de la instalación de hardware en entornos no controlados, en los que el análisis de seguridad va más allá de los ámbitos habituales de las tecnologías de la información "clásica" y donde el daño físico y la manipulación de la infraestructura son una amenaza muy presente; junto con el uso de canales de comunicación e infraestructuras públicas y/o compartidas, alejando la seguridad de los escenarios tradicionales.

Es fácil ver como la combinación de mayor complejidad y la tendencia a inclinarse hacia sistemas abiertos (en los que muchas partes diferentes cooperan de manera pública, como se ve en la Figura 2.9) genera un panorama preocupante. La evaluación de seguridad clásica se basa en tener una imagen detallada de la descripción del sistema, generalmente confiando en la seguridad del perímetro y/o el control de acceso a los servidores de datos, mientras que IoT favorece los sistemas dinámicos que cambian rápidamente, moviéndose de un proveedor a otro y abrazando un gran número de tecnologías diferentes [83]. En dicho escenario es difícil tener una clara imagen de todo el sistema, para evaluar posibles vulnerabilidades.

También es de destacar que la forma en que fluyen los datos, así como la naturaleza de los mismos, también difiere de los escenarios clásicos. Los datos se pueden consumir y procesar en muchos lugares diferentes, a menudo difíciles de rastrear hasta el origen, lo que dificulta el seguimiento de las ubicaciones por las que han pasado, los cambios que han experimentado y quién ha accedido a ellos.

Por último, la preocupación pública y política se ha cristalizado en forma de normativas, como el Reglamento General de Protección de Datos (RGPD), una normativa a nivel europeo

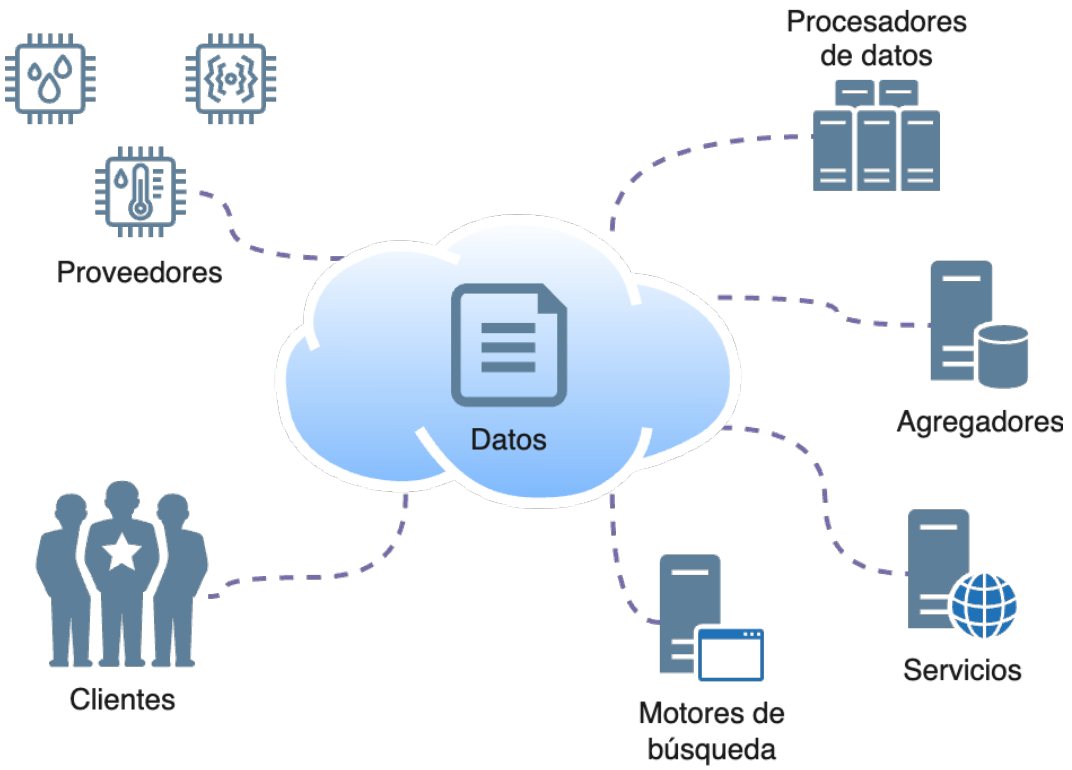


Figura 2.9: Visión general de las diferentes partes involucradas en el intercambio de datos

que impone un nuevo conjunto de dificultades y restricciones en la forma en que se manejan los datos personales y que afecta directamente al IoT [84]. Desde el punto de vista de la Ciudad Inteligente, Hernandez-Ramos, Martinez, Savarino et al. [85] abordan el problema de la seguridad y privacidad desde la perspectiva de los ciudadanos, que necesitan mecanismos para permitirles la gestión de su seguridad y privacidad por medio de sistemas de control de acceso, así como mecanismos para compartir datos de manera descentralizada. Este trabajo es respaldado por un trabajo de investigación anterior [86], donde se describe una futura sociedad, basada en datos, que requiere una visión armonizada de la ciberseguridad.

2.2.1 Protección de la privacidad

La protección de la privacidad es un tema importante que los sistemas informáticos deben tener en cuenta. En muchos países existen requisitos legales para que los sistemas de información protejan la privacidad del usuario. De manera invariable, todas estas normativas se derivan de las directrices de privacidad definidas por la Organización para la Cooperación y el Desarrollo Económicos (OCDE) [87], que establece los siguiente *8 principios de protección de datos*:

1. **Principio de Limitación Recolección**: Debe haber límites para la recopilación de datos personales y dichos datos deben obtenerse por medios legales y justos y, cuando corresponda, con el conocimiento o consentimiento del interesado.

2. **Principio de Calidad de Datos**: Los datos personales deben ser pertinentes para los fines para los que se van a utilizar y, en la medida necesaria para esos fines, deben ser exactos, completos y actualizados.

3. **Principio de Especificación del Propósito**: Los fines para los cuales se recopilan los datos personales deben especificarse a más tardar en el momento de la recopilación de datos y el uso posterior limitado al cumplimiento de esos fines u otros que no sean incompatibles con esos fines y que se especifiquen en cada ocasión de cambio de propósito.

4. **Principio de Limitación de Uso**: Los datos personales no deben divulgarse, ponerse a disposición ni utilizarse de otro modo para fines distintos a los especificados en el momento de la recopilación, excepto con el consentimiento del interesado o por la autoridad de la ley.

5. **Principio de Garantías de Seguridad**: Los datos personales deben estar protegidos por medidas de seguridad razonables contra riesgos tales como la pérdida o el acceso no autorizado, la destrucción, el uso, la modificación o la divulgación de datos.

6. **Principio de Apertura**: Debe haber una política general de apertura sobre desarrollos, prácticas y políticas con respecto a los datos personales. Debe haber medios fácilmente disponibles para establecer la existencia y naturaleza de los datos personales, y los fines principales de su uso, así como la identidad y residencia habitual del responsable del tratamiento.

7. **Principio de Participación Individual**: Un individuo debe tener derecho a:

   a) obtener de un controlador de datos, la confirmación de si el controlador de datos tiene o no datos que le conciernen

   b) haberle comunicado los datos que le conciernen en un plazo razonable, a un precio que no sea excesivo, de manera razonable y en una forma que le resulte fácilmente inteligible

   c) ser informado de las razones si se deniega una solicitud realizada conforme a los apartados (a) y (b), y poder impugnar tal denegación

   d) impugnar los datos que le conciernen y, si prospera la impugnación, hacer que se supriman, rectifiquen, completen o corrijan.

8. **Principio de Responsabilidad**: Un controlador de datos debe ser responsable de cumplir con las medidas que dar efecto a los principios enunciados anteriormente.

2.2.2 Modelos y conceptos básicos de seguridad

Los modelos de seguridad son marcos que están diseñados para proporcionar un enfoque estructurado para administrar los riesgos de seguridad asociados con los sistemas informáticos. Estos

modelos proporcionan una forma de definir y gestionar los datos necesarios para garantizar la seguridad de dichos sistemas.

Uno de los principales beneficios de usar modelos de seguridad es que brindan un enfoque coherente para administrar los riesgos de seguridad en diferentes sistemas. Al adoptar un enfoque estandarizado de la seguridad, las organizaciones pueden administrar los riesgos de seguridad de manera más eficaz, reducir el riesgo de infracciones y minimizar el impacto de los incidentes cuando ocurren.

Los modelos de seguridad de datos están diseñados para proteger estos datos del acceso no autorizado, la manipulación y el uso indebido. El propósito de la aplicación de los modelos de seguridad es ayudar a las organizaciones a identificar y administrar los riesgos de seguridad asociados con los dispositivos y datos. Estos modelos suelen definir un conjunto de controles y medidas de seguridad que deben implementarse para proteger contra varios tipos de amenazas a la seguridad.

Existen varios modelos de seguridad, cada uno con sus propias fortalezas y debilidades. Uno de estos modelos es la conocida triada CIA [88], por sus siglas en inglés *Confidentiality, Integrity* and *Availability*, comprendida por las siguientes propiedades o principios:

1. **Confidencialidad**: garantiza que los datos sean accesibles solo para las partes autorizadas. Una forma de llevarlo a cabo es utilizar técnicas de cifrado para evitar el acceso no autorizado a los datos.

2. **Integridad**: se debe garantizar que los datos no se alteren durante la transmisión o el almacenamiento. Puede ser implementado utilizando técnicas como hashing y firmas digitales para detectar cualquier cambio realizado en los datos.

3. **Disponibilidad**: los datos deben estar siempre disponibles para las partes autorizadas. Garantizar esta propiedad consiste en la aplicación de técnicas como la redundancia y la tolerancia a fallos para garantizar que los datos estén siempre accesibles.

Otro modelo similar al anterior es el promulgado por el Departamento de Defensa de los Estados Unidos de América, que además de la mencionada tríada, comprende las siguientes propiedades:

1. **Autenticidad**: permite establecer la validez de una transmisión, mensaje o dato, así como la autorización de recibir dicha información. La autenticación previene la falsificación de identidad, requiriendo al usuario confirmar su identidad antes de ser permitido acceso a datos y recursos.

2. **No repudio**: asegura que el remitente de la información recibirá prueba de entrega y que el receptor recibirá prueba de la identidad del remitente, de tal forma que ninguna de las partes pueda contradecir el envío, recepción o acceso a la información. Esta propiedad es muy similar a la de “Rendición de cuentas” que veremos a continuación.

Otro modelo popular, ampliamente aplicado en sistemas de información, es el AAA, por sus siglas en inglés **Authentication, Authorization *and* Accounting**:

1. **Autenticación**: garantiza que solo las partes autorizadas puedan acceder a los datos. Utiliza técnicas como contraseñas, biometría o la autenticación de dos factores para autenticar usuarios.

2. **Autorización**: garantiza que los usuarios puedan acceder solo a los datos a los que están autorizados a acceder. Utiliza técnicas como listas de control de acceso y control de acceso basado en roles para administrar los permisos de los usuarios.

3. **Rendición de cuentas**: garantiza que acciones y usuarios sean relacionados; los usuarios son demostrablemente responsables de sus interacciones con los datos. Utiliza técnicas como registros de auditoría y firmas digitales para rastrear las actividades de los usuarios.

Otra última propiedad que no hemos visto en los modelos anteriores es la de **Privacidad**: los datos personales no se recopilan ni procesan sin el consentimiento del usuario. Técnicas como la minimización de datos y la anonimización protegen la privacidad del usuario.

2.2.3 Autorización y control de acceso

Antes de comenzar a describir diferentes mecanismos de control de acceso a recursos, es conveniente mencionar que generalmente estos mecanismos caen bajo una (o una combinación) de dos formas: discrecionales u obligatorios.

1. El control de acceso a discreción (discretionary access control, DAC), es una forma de control de acceso en base a la identidad de un usuario, los grupos a los que pertenece y/o su necesidad de acceso a información. La discrecionalidad de este acceso se basa en que un usuario con cierto permiso de acceso a un recurso, es capaz de pasar dicho permiso (a veces de forma indirecta) a otros usuarios. Un ejemplo de esta forma de acceso serían los sistemas de ficheros en que el propietario de un fichero puede conceder acceso a otros usuarios o grupos.

2. El control de acceso obligatorio (mandatory access control, MAC), modelado y formalizado originalmente por los autores Bell y LaPadula [89], surge de la necesidad de imponer directivas de seguridad en las que el propietario de un recurso no debe tomar decisiones sobre el acceso al mismo. Este tipo de sistemas son encontrados con frecuencia en sistemas gubernamentales y de defensa y las decisiones de acceso se basan en atributos, o etiquetas, fijos asignados a los recursos y a los usuarios. En la práctica, una etiqueta de recursos se denomina “clasificación de seguridad” y en un usuario se denomina “autorización de seguridad”. Un documento con una determinada clasificación de seguridad, solo podrá ser leído por un usuario con determinada autorización de seguridad.

Dos modelos de control de acceso ampliamente utilizados en seguridad informática y que serán especialmente relevantes en secciones posteriores, son el modelo por roles y el modelo por atributos.

El control de acceso basado en roles (role-based access control, RBAC) [90], [91] es un modelo de seguridad que otorga permisos a los usuarios en función de sus papeles o responsabilidades dentro de una organización. Estos permisos contemplan diferentes acciones a realizar sobre recursos u objetos del sistema. En RBAC, a cada usuario se le asigna un rol y los permisos están asociados con dicho rol (ver Figura 2.10). Esto permite a los administradores gestionar fácilmente los permisos de los distintos usuarios mediante la gestión de roles en lugar de la asignación individual a usuarios. Por ejemplo, un usuario con el rol de "gerente" puede tener acceso a ciertos archivos o carpetas que un usuario con el rol de "empleado" no tiene.

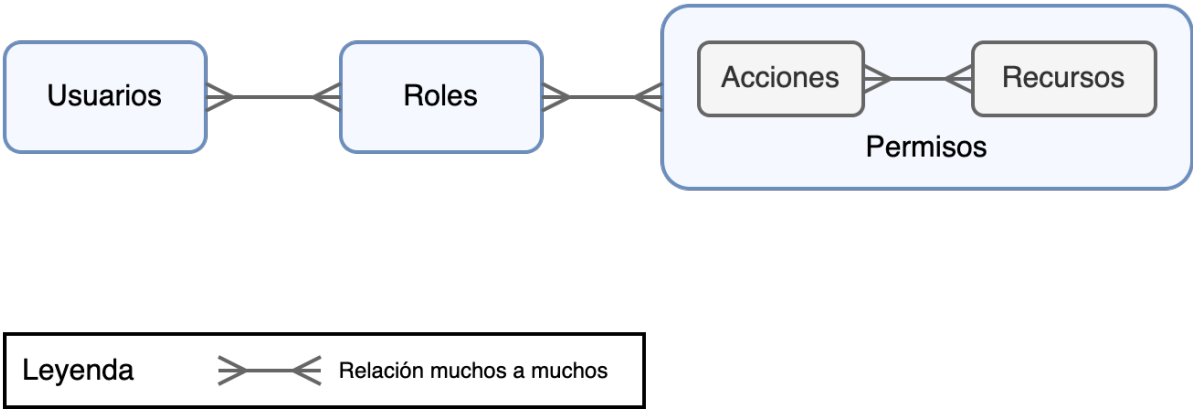


Figura 2.10: Esquema de asignación de permisos a usuarios en RBAC

La flexibilidad en la relación entre usuarios y operaciones por medio de roles se ve más aun incrementada por el mecanismo de herencia de roles: los roles pueden ser definidos en una jerarquía, de forma que los permisos son heredados de roles padres a hijos. La Figura 2.11 muestra una representación de como "Internista" y "Cirujano" son especializaciones de "Especialista", que a su vez hereda de "Médico".

RBAC es un mecanismo principalmente DAC que no está bien preparado para cubrir las necesidades de un sistema MAC; para simular este último sería necesario crear roles únicos para cada combinación distinta de etiquetas de seguridad, convirtiéndolo en un sistema difícil de gestionar.

El control de acceso basado en atributos (attribute-based access control, ABAC) [92], [93], por otro lado, es un modelo de control de acceso más flexible que utiliza atributos para tomar decisiones de acceso. Los atributos pueden representar cualquier información que describa a un usuario, como su cargo, ubicación o departamento. Las decisiones de acceso se toman en base a un conjunto de reglas que tienen en cuenta los atributos del usuario y el recurso al que se accede. Por ejemplo, una regla podría especificar que solo los usuarios con el atributo "departamento=finanzas" pueden acceder a los informes financieros.

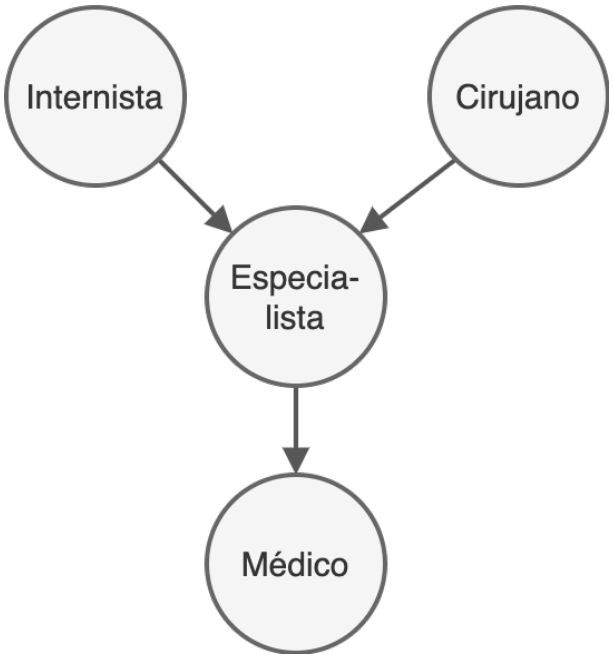


Figura 2.11: Ejemplo de jerarquía de roles en RBAC

De acuerdo a Coyne y Weil [94], en comparación con RBAC, ABAC permite un control de acceso más granular y puede tener en cuenta una gama más amplia de factores al tomar decisiones de acceso, de hecho, es fácil aplicar RBAC en ABAC viendo los roles como atributos de usuario. Sin embargo, también puede ser complejo de implementar y administrar, ya que requiere un sistema para recopilar y mantener los atributos necesarios para cada usuario.

En la Figura 2.12 podemos ver un diagrama de funcionamiento de ABAC, en el que los distintos componentes involucrados interaccionan para asegurar el cumplimiento de las políticas de seguridad definidas en el sistema. Los componentes principales que solemos encontrar en un sistema ABAC son:

1. **Punto de administración de políticas (policy administration point, PAP)**: es el punto donde la administración de las políticas tiene lugar; su definición, modificación y borrado.

2. **Punto de decisión de políticas (policy decision point, PDP)**: es el encargado de comprobar las políticas de seguridad del sistema y emitir un veredicto sobre la autorización de la acción solicitada sobre el recurso protegido. Para ello recibe del punto de aplicación de políticas (policy enforcement point, PEP) información sobre el usuario, la acción y el recurso solicitado. Posteriormente recupera información de las políticas almacenadas en el sistema, así como los atributos de seguridad asociados al usuario y emite el veredicto resultante al PEP.

3. **Punto de aplicación de políticas (policy enforcement point, PEP)**: actúa de intermediario entre el usuario y el recurso accedido y es el responsable de imponer la decisión de seguridad del sistema. En el diagrama de ejemplo, es el responsable de permitir al usuario la lectura de un documento tras la verificación de que dicho usuario tiene permiso para ello, de acuerdo a las políticas definidas en el sistema. Para cada intento de acceso a un recurso por parte de un usuario, el PEP se comunica con el PDP, indicando el recurso y la acción solicitada, así como información del usuario. El PEP esperará el veredicto del PDP para permitir o rechazar el paso de la petición del usuario.

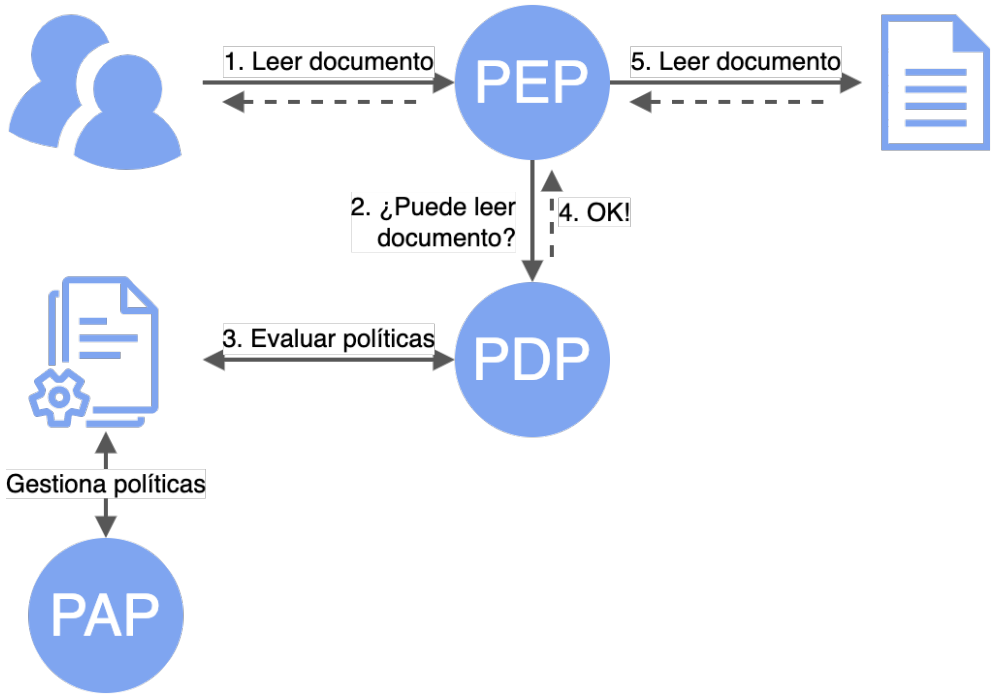


Figura 2.12: Esquema general de componentes y flujo de datos en ABAC

Para la representación de las políticas en ABAC se recomienda la utilización del estándar lenguaje extensible de marcas para control de acceso (extendible access control markup language, XACML) [95], [96] que define una arquitectura genérica para la autorización, así como un lenguaje basado en XML para la definición y el intercambio de políticas de control de acceso. También define los intercambios tipo petición/respuesta para las decisiones de autorización, facilitando la creación de mecanismos de control de acceso sobre el mismo.

2.2.4 Gestión de la identidad y autenticación de usuarios

Otra pieza clave para garantizar la confidencialidad de los datos, así como la autenticidad de datos y comunicaciones, es una correcta gestión de la identidad y la autenticación de usuarios. En la sección 2.2.3 hemos visto varios modelos para el control de acceso, que confían en la existencia

de mecanismos eficaces para garantizar la gestión de la identidad de los usuarios, así como su correcta autenticación en el sistema.

Las tecnologías actuales para la gestión de la identidad digital de usuarios siguen estrategias de federación de la identidad para simplificar la autenticación de usuarios, mejorar la seguridad, optimizar la gestión de usuarios y facilitar la colaboración entre organizaciones y sistemas.

La federación de identidades es una forma de permitir que los usuarios se autentiquen en una organización y luego usen esa autenticación para acceder a recursos en otras organizaciones, sin tener que autenticarse nuevamente. Es un sistema de sistema centralizado de gestión de identidad, que permite a los usuarios utilizar un único conjunto de credenciales, como un nombre de usuario y una contraseña, para acceder a múltiples aplicaciones o servicios que forman parte de diferentes organizaciones o dominios.

A pesar de que la federación de identidad ofrece numerosos beneficios, no está carente vulnerabilidades o riesgos, como ofrecer un punto singular de fallo, tener que lidiar con problemas de confianza, ser inherentemente complicada y ofrecer preocupaciones sobre la privacidad de los datos de usuarios.

Chadwick explica las necesidades y particularidades y los conceptos básicos relacionados con la federación de identidad en [97], así como los problemas relacionados con la misma en términos de la protección de la privacidad y niveles de garantía de confianza. Uno de los trabajos principales en que se basa Chadwick son las *7 leyes de la identidad* de Cameron [98], enumeradas a continuación:

1. **Control y Consentimiento del Usuario**: Los sistemas de identificación técnica solo deben revelar información que identifique a un usuario con el consentimiento del usuario.

2. **Divulgación Mínima Para un Uso Restringido**: La solución que revela la menor cantidad de información de identificación y limita mejor su uso es la solución más estable a largo plazo.

3. **Partes Justificables**: Los sistemas de identidad digital deben diseñarse de modo que la divulgación de información de identificación se limite a las partes que tienen un lugar necesario y justificable en una relación de identidad determinada.

4. **Identidad Dirigida**: Un sistema de identidad universal debe admitir tanto identificadores "omnidireccionales" para uso de entidades públicas como identificadores "unidireccionales" para uso de entidades privadas, lo que facilita el descubrimiento y evita la liberación innecesaria de identificadores aleatorios de correlación.

5. **Pluralismo de Operadores y Tecnologías**: Un sistema de identidad universal debe canalizar y permitir el funcionamiento entre múltiples tecnologías de identidad ejecutadas por múltiples proveedores de identidad.

6. **Integración Humana**: El metasistema de identidad universal debe definir al usuario humano como un componente del sistema distribuido integrado a través de mecanismos inequívocos de comunicación hombre-máquina que ofrecen protección contra ataques de identidad.

7. **Experiencia Consistente en Todos los Contextos**: El metasistema de identidad unificador debe garantizar a sus usuarios una experiencia simple y consistente al tiempo que permite la separación de contextos a través de múltiples operadores y tecnologías.

Para poder implementar sistemas de identidad federada que cumplan las mencionadas leyes, así como los *8 principios de protección de datos* descritos en la Sección 2.2.1, la forma más exitosa, favorecida por academia e industria, consiste en separar los proveedores de identidad (identity providers, IdP) de los proveedores de servicios, almacenando los atributos de identidad sólo en el IdP.

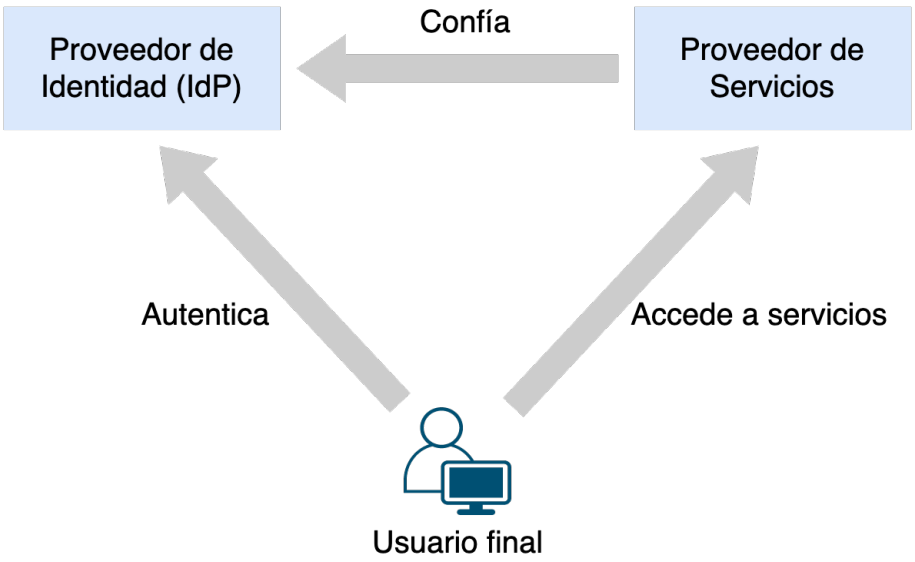


Figura 2.13: Actores y relaciones en identidad federada

La Figura 2.13 muestra los principales elementos en un escenario de identidad federada. El IdP brinda a los usuarios servicios e autenticación. Una vez han sido autenticados de manera exitosa, el IdP envía información sobre el usuario (de manera segura) al proveedor de servicios, quien utiliza posteriormente esta información para determinar si el usuario está autorizado para acceder a los recursos solicitados.

Varios estándares abiertos implementan sistemas federados de identidad, como Security Assertion Markup Language (SAML) [99], OAuth 2.0 [100] y OpenID Connect (OIDC) [101], permitiendo a diferentes aplicaciones y servicios intercambiar información de autenticación y autorización de forma segura y uniforme.

2.2.5 Gestión de identidad y control de acceso en IoT

Uno de los primeros problemas abordados tanto por la industria como por la academia ha sido la producción de estándares en forma de protocolos y frameworks, con el fin de homogeneizar y tratar de dar respuesta a muchos de los requerimientos específicos que impone IoT. Como un ejemplo digno de mención de dichos marcos, FIWARE impulsó la estandarización del modelo de datos y comunicaciones NGSI-LD, alrededor del cual se puede implementar un ecosistema de componentes de una amplia colección de "Enablers" (habilitadores) que brindan diferentes funcionalidades sobre las cuales construir plataformas seguras para IoT. Otros marcos, como oneM2M, también han generado componentes de seguridad [102].

Un aspecto que ha ganado tracción tanto en academia como industria es la combinación de autenticación y gestión de la identidad; cubierto en la bibliografía en los trabajos de Mahalle, Babar, Prasad et al. [103] y de Bernal Bernabe, Hernandez-Ramos y Skarmeta Gomez [104]. Este último ofrece una solución holística y de preservación de la privacidad, capaz de enfrentarse a escenarios heterogéneos con requisitos tradicionales de autenticación y control de acceso, junto con un enfoque orientado a IoT basado en aserciones, apropiado para las interacciones M2M. Para ello combina un sistema de autenticación basado en aserciones anónimas (Idemix [105]) junto con otros mecanismos de gestión de la identidad (identity management, IdM) tradicionales.

La gestión de la identidad, así como el control de acceso, han sido abordados en varios proyectos de investigación europeos, tales como Smartie, SocIoTal, CPaas.io e IoTCrawler, integrando tecnologías que se han mostrado como soluciones válidas para dominios como las ciudades y los edificios inteligentes. Estos proyectos proponen también el uso de mecanismos de control de acceso basados en XACML.

Debido a la naturaleza centralizada de los modelos de control de acceso ABAC, en los que cada aprobación o denegación del acceso a un recurso implica la evaluación de las políticas de seguridad, estos sistemas pueden ocasionar cuellos de botella en el sistema de validación. Este problema es aún más relevante en el caso de IoT, donde un gran volumen de solicitudes es esperado. Una de las formas en que este problema ha sido abordado ha sido mediante la aplicación de estrategias descentralizadas, como es el caso de Hernández-Ramos, Jara, Marin et al. [106], donde presentan control de acceso distribuido basado en capacidades (distributed capability-based access control, DCapBAC): una evolución de ABAC que parece extraer inspiración de Kerberos [107]. Kerberos es un protocolo de autenticación distribuido en red, que funciona por medio de "tickets". Kerberos utiliza dichos tickets para conceder acceso a un cliente directamente a un recurso. El recurso es capaz de verificar unívocamente y de forma segura, la validez del ticket, evitando el cuello de botella que supondría la verificación de identidad y validación de acceso por parte de una entidad centralizada. DCapBAC se basa, en este caso, en el uso de "tokens" (similares a los tickets de Kerberos) que permiten el acceso a recursos sin requerir una evaluación de las políticas de seguridad.

En la Figura 2.14 podemos ver un diagrama general de funcionamiento de DCapBAC, con

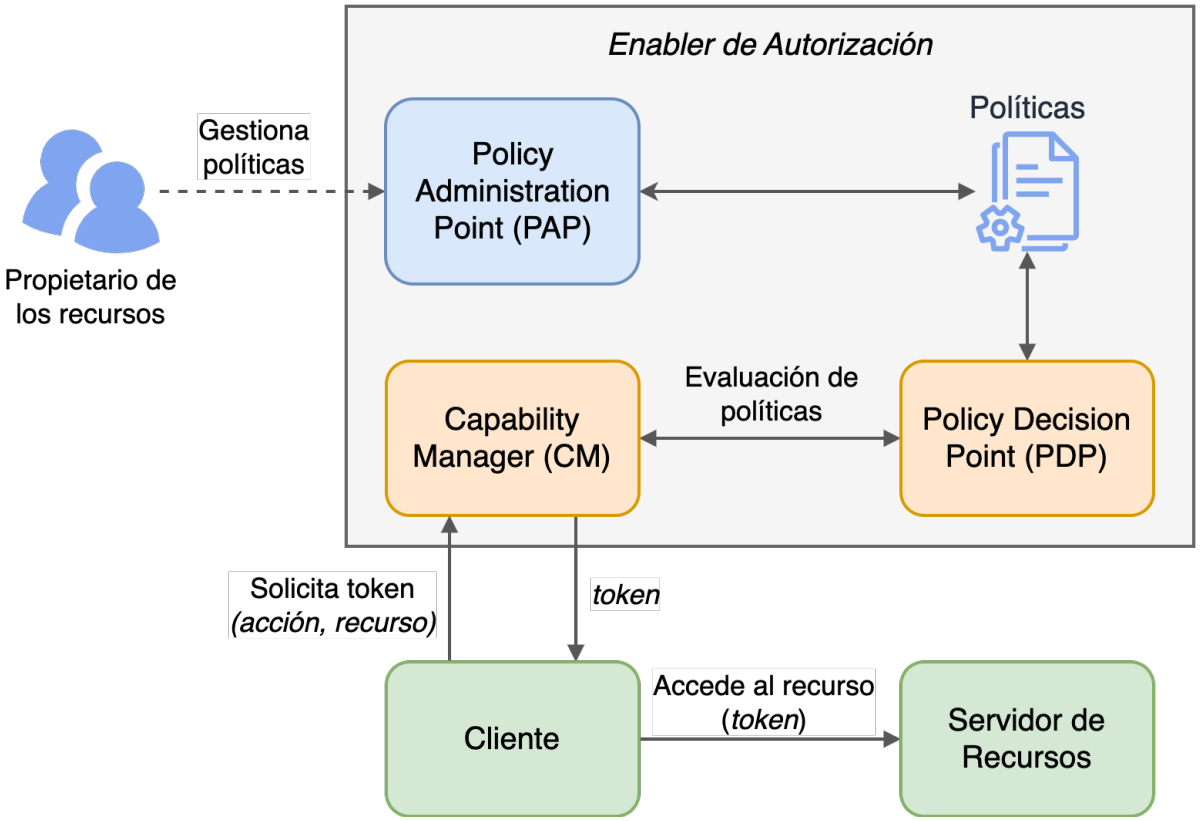


Figura 2.14: Modelo de control de acceso DCapBAC

una pieza nueva respecto al diagrama tradicional ABAC: el gestor de capacidades (capability manager, CM). En la imagen podemos ver como el Cliente obtiene un token de acceso del CM de forma previa a realizar la solicitud al Servidor de Recursos. Este token contiene información sobre el Cliente, el recurso y la acción a realizar por parte del Cliente, además de un tiempo de validez. De esta forma el Servidor de recursos, al recibir el token, es capaz de validar y verificar que, en efecto, al Cliente le ha sido concedido acceso a dicho recurso, sin ser necesaria ninguna interacción adicional entre el Servidor de Recursos y el PDP.

Posteriormente, Truong, Hernández-Ramos, Martinez et al. [108] expanden el concepto de DCapBAC a las Tecnologías de Contabilidad Distribuida (DLT por sus siglas en inglés: *Distributed Ledger Technology*), integrando un modelo de control de acceso basado en capacidades y blockchain para una evaluación totalmente distribuida de las políticas de autorización y la generación de credenciales de acceso mediante “smart contracts”.

Otra forma de abordar la privacidad de los datos en IoT, es la aplicación de Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [109], presentado por Pérez, Rotondi, Pedone et al. [110], en la que la privacidad de los datos es asegurada a través de la criptografía. Esta puede ser considerada otra forma distribuida de control de acceso, ya que los datos, aunque sean accesibles por todo el mundo, solo pueden ser descifrados por aquellos clientes que tengan la clave adecuada.

En este caso, además, aprovechan los beneficios de los esquemas de criptografía de clave simétrica; más asequibles a las capacidades de cálculo de dispositivos constreñidos comunes en IoT, para el intercambio seguro de información, preservando la privacidad de los participantes.

2.2.6 Medida, representación y evaluación de la seguridad

Uno de los nuevos escenarios de crecimiento económico presentado por IoT, es el de las plataformas de brokering[20]de datos. El objetivo de estas es hacer de intermediarias en la compraventa de datos IoT a permitiendo a los propietarios de datos, venderlos a otras empresas u organizaciones que los utilizan para mejorar sus operaciones y para la toma de decisiones.

Estas plataformas permiten a los propietarios de datos controlar y gestionar el acceso a sus datos, y a los compradores realizar búsquedas, adquirir y acceder a conjuntos de datos que puedan resultar de su interés. En este caso, las necesidades de seguridad presentan nuevos retos, algunos de los cuales pueden ser atacados con aproximaciones tradicionales y otros requerirán enfoques novedosos.

En este sentido, es interesante el caso de uso de la plataforma CityPulse [111], donde la seguridad es considerada una dimensión de la calidad de la información (quality of information, QoI), métrica que acompaña a los datos en forma de metainformación y que puede resultar de valor a consumidores de datos en grandes plataformas. Para ello diseñaron un mecanismo para anotar flujos de datos con dicha métrica, que se define como un vector numérico:

$$Q = \langle L, P, E, B, Ava, C, Acc, S \rangle$$

Donde las diferentes dimensiones representan:

L: latencia
P: precio
E: consumo de energía
B: consumo de ancho de banda
Ava: disponibilidad
C: integridad
Acc: precisión
S: seguridad

Cabe destacar que en su trabajo no proporcionan un modelo para evaluar la seguridad, ni ningún marco contextual u ontología para definirla, ni ningún contexto basado en el usuario en el que se definan tratamientos interesantes como base para la evaluación de la misma.

Una revisión bibliográfica enfocada a la búsqueda de herramientas para la representación y evaluación de la seguridad, especialmente orientada a entornos IoT, revela algunos trabajos que

[20]Un corredor (broker, en inglés) actúa como intermediario entre el comprador y el vendedor, buscando un acuerdo que satisfaga a ambas partes. Un a plataforma de brokering es, en resumen, una plataforma de compraventa.

hacen uso de tecnologías semánticas tanto para la representación de los conceptos de seguridad, como para diversas aplicaciones que hacen uso del razonamiento semántico.

Priebe, Dobmeier y Kamprath, en su trabajo [112] hacen uso del razonamiento semántico, apoyado por representación ontológica, para mejorar el emparejamiento de atributos y reglas XACML al presentar una extensión del estándar XACML en el que las políticas se simplifican al proporcionar una función de administración de atributos basada en ontologías.

Finin, Joshi, Kagal et al. presentan ROWLBAC (Role-Based Access Control in OWL) [113]. En dicho trabajo, estudian la relación entre RBAC y OWL, muestra dos enfoques diferentes para representar el modelo RBAC en OWL y luego analiza como se puede extender al modelo ABAC.

En su tesis doctoral, Costabello [114] presenta "Context-Aware Access Control and Presentation of Linked Data", un trabajo que describe los prototipos PRISSMA *y* Shi3ld, siendo el segundo un marco de control de acceso hace uso de la información contextual del cliente, habilitando la aplicación de políticas de acceso "context-aware" (conscientes del contexto) para acceder a datos enriquecidos con metainformación.

Daud, Sánchez y Viejo [115] presentan una delegación de control de acceso basada en tecnologías semánticas, como una mejora del perfil de delegación XACML.

Finalmente, como un primer intento de resolver el problema de la evaluación de la seguridad en entornos IoT, Gonzalez-Gil, Skarmeta y Martinez [7] presentamos una ontología de evaluación de seguridad en IoT, basada en contexto, que aborda la evaluación de seguridad mediante razonamiento automatizado, en base a las distintas expectativas de seguridad de diferentes observadores. Este trabajo sirvió como base e inspiración para una de las contribuciones presentadas posteriormente en esta tesis (Capítulo 3) y será brevemente descrita en la Sección 2.3.4, sobre ontologías de seguridad.

2.2.7 Conclusiones

En esta sección se han introducido los fundamentos básicos que rigen la protección de la privacidad, así como algunos de los conceptos principales de seguridad en sistemas de información con que se relacionan. También se han cubierto las principales estrategias y tecnologías de autorización y control de acceso, así como la gestión de la identidad y autenticación de usuarios. A continuación, se ha presentado las peculiaridades de la gestión de identidad y control de acceso en el caso específico de las arquitecturas para IoT y se ha terminado introduciendo la problemática de la medición, representación y evaluación de la seguridad como parámetros de QoI.

Tras este estudio del estado del arte en seguridad y privacidad en IoT, se concluye que hay un gran número de tecnologías para la seguridad y privacidad, que implementan diferentes estrategias, siendo las de la familia ABAC las más populares. Muchas de estas tecnologías de autenticación y control de acceso provienen del entorno de la web, lo que permite su rápida adaptación, reutilización y reúso en las arquitecturas para IoT que se verán más adelante en este trabajo.

2.3 Ontologías y modelos de información

Una de las grandes preocupaciones de este siglo es la clasificación y el intercambio de ingentes cantidades de información con personas y aplicaciones. Esta información debe ser clasificada de una manera controlada y significativa, de tal forma que sea comprensible tanto para humanos como para máquinas. La Informática adoptó la "ontología" de la rama del saber de la Filosofía; que utiliza para compartir conocimientos y proporcionar una visión sintáctica, semántica y conceptual de la información. Las ontologías, en la Informática, son representaciones formales del conocimiento, que describen conceptos y relaciones en un dominio.

2.3.1 Metodologías ontológicas

Las metodologías de creación de ontologías son aproximaciones sistemáticas para desarrollar ontologías. De manera general, dichas metodologías pueden clasificarse en:

1. *Top-down* (de arriba a abajo): caracterizadas por comenzar con una ontología general, de alto nivel, que va descomponiendo en ontologías específicas que representan dominios particulares.

2. *Bottom-up* (de abajo a arriba): que comienza con un conjunto de conceptos específicos y construye una jerarquía de conceptos y relaciones en base a dichos conceptos.

3. *Metodología mixta*: que combina la aproximación *top-down* con la *bottom-up* para crear ontologías que son tanto generales como específicas.

4. *Metodología iterativa*: que comienza con una ontología inicial, sobre la que realiza pruebas y mejoras iterativamente, refinándola en base a la realimentación obtenida, repitiendo el proceso hasta que el resultado es considerado satisfactorio.

Otras clasificaciones existen, como la metodología colaborativa o la de adquisición de conocimiento, que involucran a grupos de expertos de diferentes dominios, que colaboran para la creación de ontologías que representan el conocimiento compartido de un dominio particular.

De acuerdo a la revisión bibliográfica de Cristani y Cuel [116], distintas metodologías para la creación de ontologías pueden tener puntos fuertes y debilidades dependiendo de las necesidades y el ámbito de aplicación de la ontología objetivo, revelando la necesidad de clasificarlas y ofreciendo un primer análisis de algunas metodologías analizadas.

Iqbal, Murad, Mustapha et al. [117] realizan posteriormente un análisis profundo y una revisión bibliográfica de metodologías de ingeniería ontológica, resultando en la conclusión de que no existe una metodología completamente madura, ofreciéndose como una guía preliminar para llegar a metodologías de ingeniería de ontologías de última generación, que cierren las brechas y deficiencias existentes.

Ontology Development 101: A Guide to Creating Your First Ontology [118], de Noy y McGuinness, es una guía recomendada por los desarrolladores de Protégé[21] [119]. Es uno de los primeros trabajos en metodologías para la creación de ontologías y un recurso popular para aquellos que se inician en esta ciencia. Describe un proceso de desarrollo de ontologías para sistemas declarativos basados en marcos, así como algunas de las trampas y errores comunes en los que pueden caer los creadores noveles.

La metodología Mentor [120], presentada por Sarraipa, Silva, Jardim-Gonçalves et al. y definida como una "metodología para el desarrollo de ontologías de referencia empresariales", mejora el intercambio de conocimientos entre organizaciones, lo que permite a sus usuarios mantener sus propias representaciones de conocimiento y producir una ontología de referencia para el dominio de la materia.

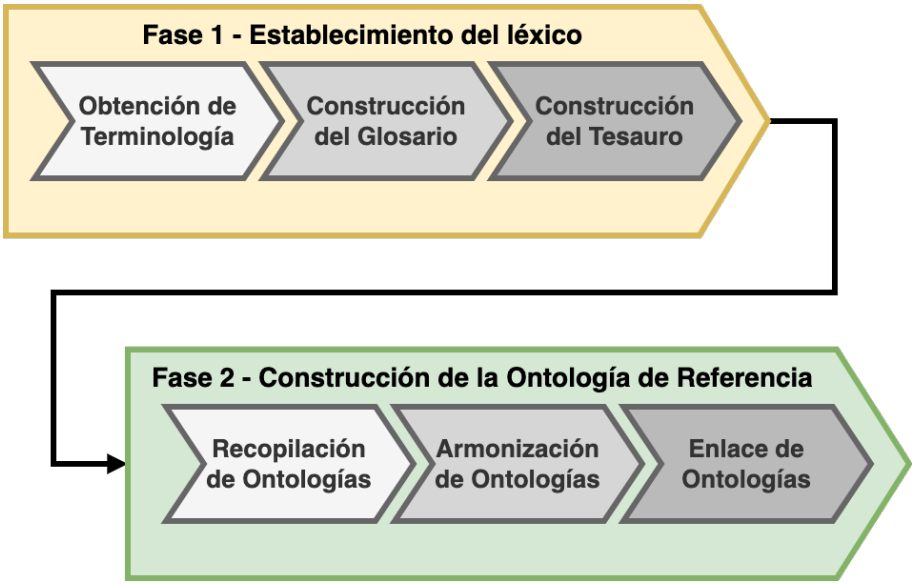


Figura 2.15: Fases y pasos en la metodología MENTOR

Esta metodología está compuesta por dos fases, representadas en la Figura 2.15: el Establecimiento del Léxico (Fase 1) y la Construcción de la Ontología de Referencia (Fase 2) con tres pasos cada una.

La primera fase representa la adquisición de conocimiento de dominio: su vocabulario incluyendo palabras y expresiones, que concluye con la construcción del tesauro. La segunda fase es donde se construye la ontología de referencia y se establecen los emparejamientos semánticos entre las ontologías organizativas y la de referencia, comenzando con la recopilación de ontologías sobre el dominio especificado, su posterior estudio y armonización para generar una ontología que cubra todos los aspectos estudiados y finalmente el enlazado de los conceptos relacionados entre estas ontologías para generar equivalencias entre ellas.

[21]Protégé es un software para el diseño y construcción de ontologías utilizando el lenguaje OWL.

Suarez de Figueroa [121] y Suárez-Figueroa, Gómez-Pérez y Fernández-López [122] nos presentan la Metodología NeOn. Su diseño orientado a una amplia audiencia y diversidad de escenarios la hace especialmente interesante en nuestro caso. Esta metodología ofrece una gama de nueve escenarios para la creación de ontologías:

1. *De la especificación a la implementación*: La red de ontologías se desarrolla desde cero (sin reutilizar los recursos existentes).

2. *Reutilización y reingeniería de recursos no ontológicos*: Los desarrolladores deben llevar a cabo el proceso de reutilización de recursos no ontológico para decidir, de acuerdo con los requisitos de la ontología, cuales de estos se pueden reutilizar para construir la red de ontología.

3. *Reutilización de recursos ontológicos*: Los desarrolladores usan recursos ontológicos, módulos de ontología y/o declaraciones de ontología para construir redes de ontologías.

4. *Reutilización y reingeniería de recursos ontológicos*: Los desarrolladores de ontologías reutilizan y rediseñan los recursos ontológicos.

5. *Reutilización y fusión de recursos ontológicos*: Este escenario surge cuando se seleccionan varios recursos ontológicos en el mismo dominio para su reutilización y los desarrolladores desean crear un nuevo recurso ontológico con los recursos seleccionados.

6. *Reutilización, fusión y reingeniería de recursos ontológicos*: Los desarrolladores de ontologías reutilizan, fusionan y rediseñan los recursos ontológicos. Similar al escenario 5, pero aquí los desarrolladores deciden rediseñar el conjunto de recursos fusionados.

7. *Reutilización de patrones de diseño de ontología*: Los desarrolladores de ontologías acceden a repositorios para reutilizar patrones de diseño de ontología.

8. *Reestructuración de recursos ontológicos*: Los desarrolladores de ontologías reestructuran (por ejemplo, modularizan, reducen, amplían y/o especializan) los recursos ontológicos para integrarlos en la red de ontologías.

9. *Localización de recursos ontológicos*: Los desarrolladores de ontologías adaptan una ontología a otros idiomas y comunidades culturales, obteniendo así una ontología multilingüe.

2.3.2 Ontologías para IoT

Numerosos trabajos se han dedicado a la representación y modelado de las relaciones entre las diversas entidades involucradas en los sistemas IoT por medio de ontologías. Estas nos proporcionan una forma estructurada y estandarizada de describir sensores, actuadores y otros

elementos que conforman los sistemas IoT, así como los datos generados y las interacciones entre ellos.

Estas ontologías están diseñadas para enfrentarse a los desafíos de la integración de datos y sistemas heterogéneos a través de diferentes industrias y dominios, ayudando a facilitar la integración entre sistemas y el intercambio de datos, jugando un papel muy importante en el desarrollo de los sistemas IoT, ayudando a garantizar la escalabilidad e interoperabilidad.

En la literatura podemos encontrar algunos trabajos de revisión bibliográfica sobre ontologías IoT [123], [124], así como trabajos que se centran en la interoperabilidad [125]. En los siguientes párrafos de esta sección, se cubren algunas de las ontologías de mayor relevancia para este trabajo.

Semantic Sensor Network (SSN) [126], presentada por Compton, Barnaghi, Bermudez et al. [127], fue desarrollada por el W3C Semantic Sensor Network Incubator Group. Esta describe sensores en términos de capacidades, procesos de medición, observaciones, implementaciones y conceptos relacionados. No describe conceptos de dominio, tiempo, ubicaciones, etc. En su lugar estos conceptos están destinados a ser importados, reutilizados desde otras ontologías.

IoT-A, de Wang, De, Toenjes et al. [128], surge a partir del modelo del proyecto IoT-Architecture[22] y extiende SSN proporcionando algunos conceptos básicos, como `Service` y `QualityOfService`, estando marcadamente orientada a servicios.

Stream Annotation Ontology [130] (SAO), de Kolozali, Bermudez-Edo, Puschmann et al. [131], es una ontología que extiende a SSN con el objetivo de incorporar elementos para anotar y razonar sobre flujos de datos: es decir, datos que se generan de forma continua y en tiempo real. Incluye conceptos tales como `StreamData`, `StreamData`, `StreamAnalysis` para representar cuan condensados y fiables son los datos en un flujo.

Sensor-Observation-Sampling-Actuator ontology [132] (SOSA), de Janowicz, Haller, Cox et al. [133], es una especificación formal de propósito general y liviana, para modelar interacciones entre entidades involucradas en IoT, en términos de observaciones, actuación y muestreo. Fue creada como una sucesora o mejora sobre SSN, incluyendo el módulo *Stimulus Sensor Observation* (SSO) de esta última. Ha sido desarrollada conjuntamente por el Open Geospatial Consortium (OGC) y el W3C.

IoT-lite, de Bermúdez-Edo, Elsaleh, Barnaghi et al. [134], parte del objetivo de describir conceptos clave de IoT tratando de reducir la complejidad y el tiempo de procesamiento; características deseables en los entornos dinámicos y de reacción rápida, comparándose favorablemente con IoT-A. IoT-Lite se define como una extensión de SSN con semántica ligera, que permite la interoperabilidad y el descubrimiento de datos generados por sensores en plataformas heterogéneas. Además proponen 10 reglas para el diseño de modelos semánticos escalables, que posteriormente utilizan para guiar el desarrollo de IoT-Lite. Entre los conceptos representados se encuentran *iot-lite:Entity*, *iot-lite:Service* y *iot-lite:Coverage*. Este último enlaza con la ontología GEO [135],

[22] "IoT-A: Internet of Things Architecture". (2023), dirección: `https://www.iot-a.eu/` (visitado 15-07-2023).

para representar dimensiones espaciales y localizaciones.

IoT-Stream [136], de Elsaleh, Enshaeifar, Rezvani et al. [137], que al igual que SAO está orientada a la anotación semántica de flujos de datos. La novedad de esta ontología es que en su diseño se ha considerado aplicaciones casi en tiempo real, que tienen requisitos estrictos en términos de latencia de procesado; por lo que en su diseño han optado por un modelo ligero, con consultas e inferencias sencillas. Al igual que otras, extiende SSN e incluye conceptos de SOSA, IoT-Lite y GEO entre otras.

2.3.3 Ontologías para SHEMSs

Para modelar la información en SHEMS, es preciso cubrir un amplio espectro de conceptos, motivo por el cual esta sección incluye un extenso estudio de diferentes ontologías relacionadas.

Los temas o áreas cubiertos incluyen la gestión de DR de la red eléctrica, gestión de DR del hogar, gestión de DER, medición de energía y evaluación del rendimiento, consejos para el ahorro de energía, infraestructura doméstica, preferencias del usuario e información sensorial ambiental y climatológica, por nombrar solo algunos.

Parece obvio decir que ninguna ontología singular cubre todas las áreas requeridas y que todas difieren en el nivel de detalle con el que se capturan los diferentes conceptos. Más allá de eso, muchas ontologías se han construido y vinculado a otras ontologías, creando diferencias en la dificultad de adopción como resultado del trabajo de integración adicional (o la falta de) con datos anotados semánticamente ya existentes.

Para ofrecer una visión general más conveniente al lector, sobre el estado del arte en ontologías relacionadas con SHEMS, la Tabla 2.2 resume los trabajos más relevantes de acuerdo con el trabajo desarrollado en esta Tesis.

DogOnt, de Bonino y Corno [138], es una ontología para el entorno inteligente domótico que utilizada con tecnologías para el razonamiento, es capaz de responder a problemas de interoperabilidad, permitiéndole describir:

- la ubicación de dispositivos domóticos;
- el conjunto de capacidades de los mismos;
- las características de las tecnologías para su interconexión;
- las posibles configuraciones que puede asumir;
- la composición del entorno del hogar;
- el tipo de elementos arquitectónicos y muebles ubicados en la vivienda.

Aunque date de 2008, DogOnt es muy utilizada incluso en la actualidad ya que es importada o extendida desde otras ontologías. En la Figura 2.16 podemos ver un diagrama mostrando algunos de los conceptos centrales representados en DogOnt.

Tabla 2.2: Resumen de ontologías para SHEMS

Ontología	**Campo de aplicación**
DogOnt [138]	Modelado del entorno inteligente domótico.
ThinkHome [59], [140]	Evaluación energética en el hogar y control de dispositivos.
BonSAI [142]	Modelado de edificio inteligente orientado a servicios.
MIRABEL [144]	Descripción de flexibilidades para respuesta de demanda.
ProSGv3 [145]	Modelado de red eléctrica inteligente orientada a prosumidores.
DNAS [146]	Eficiencia energética a través de comportamiento de ocupantes.
SAREF4EE [148], [149]	Interoperabilidad de electrodomésticos inteligentes.
MAS2TERING [53]	Soporte de la implementación USEF en red eléctrica inteligente.
EnergyUse [154]	Consejos para ahorro energético en el hogar.
DABGEO * [156]	Integración de ontologías relacionadas con energía en hogar inteligente.
SARGON [158]	Red eléctrica inteligente y automatización energética de edificios.
OSEIM [159], [160]	Razonamiento semántico para gestión energética inteligente.

* DABGEO incluye (importa) las ontologías marcadas en gris.

ThinkHome [139], presentada por Reinisch, Kofler, Iglesias et al. [59], propone una ontología que describe los recursos, la energía, el confort, las influencias exteriores, elementos arquitectónicos, actores e información de proceso. ThinkHome fue posteriormente ampliada por Kofler, Reinisch y Kastner [140] para el dominio de la energía, cubriendo la descripción de instalaciones y aparatos, con énfasis en el consumo de energía y el suministro de energía. En esta extensión de ThinkHome, también se reutiliza la ontología DogOnt.

BOnSAI [141], presentada por Stavropoulos, Vrakas, Vlachava et al. [142], presenta una ontología para incorporar inteligencia ambiental en edificios inteligentes, que se puede utilizar para la gestión y el control de la energía. Incluye conceptos sobre funcionalidad, QoS, hardware, usuarios y contexto.

MIRABEL [143], presentada por Verhoosel, Rothengatter, Rumph et al. [144], es una ontología para modelar la flexibilidad en la gestión de energía en SG. Esta permite a los actores expresar su flexibilidad energética para un dispositivo específico. También representa perfiles de energía para dispositivos, así como dispositivos de producción y almacenamiento. La flexibilidad está

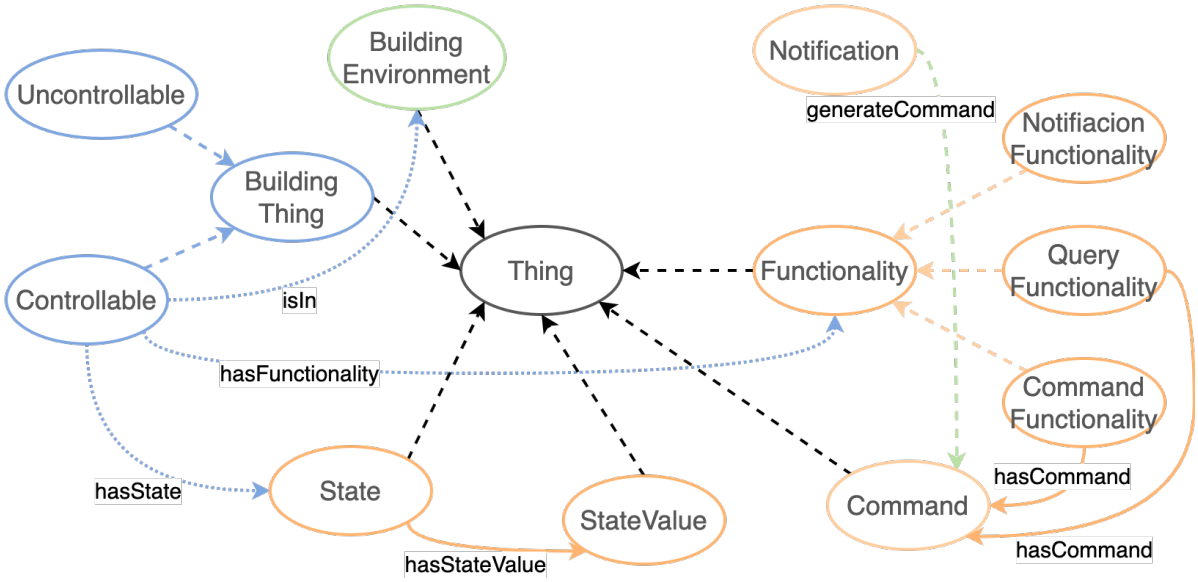


Figura 2.16: Diagrama de la ontología DogOnt, para el entorno domótico inteligente

relacionada con las preferencias de usuario en términos de energía, tiempo y precio. Cada dispositivo tiene un perfil de energía que describe la cantidad de energía consumida o producida durante un intervalo de tiempo. Una oferta de flexibilidad combina las preferencias del usuario con el perfil de energía del dispositivo correspondiente.

ProSGv3, presentada por Gillani, Laforest y Picard [145], es una ontología orientada al prosumidor para SG, diseñada para modelar información sobre productores y consumidores de energía, así como sistemas de generación y almacenamiento de energía, aparatos eléctricos, infraestructura, informes meteorológicos y eventos.

DNAS es un framework en el que se define una ontología homónima, presentada por Hong, D'Oca, Turner et al. [146], para representar el comportamiento de los ocupantes, relacionado con la energía, para comprender el consumo total energético en edificios.

SAREF [147], de Daniele, Hartog y Roes [148], es una ontología que modela dispositivos, propiedades, medidas y servicios (entre otros conceptos) facilitando la descripción de los recursos existentes y proporciona componentes básicos, utilizables en función de las necesidades específicas. Posteriormente, Daniele, Solanki, Den Hartog et al. [149], extendieron SAREF con *SAREF4EE*; creado en colaboración con EEBus[23] y Energy@home[24], para la gestión energética de electrodomésticos inteligentes. Esta busca obtener la interoperabilidad entre soluciones propietarias, en el ámbito de la casa inteligente. Los casos de uso considerados en su desarrollo están relacionados generalmente con el ámbito de DR. La información representada por esta ontología se puede categorizar como: información de configuración, información de programación, información de

[23]"EEBUS: Empowering the digitalisation of Energy transition". (), dirección: `https://www.eebus.org/` (visitado 22-04-2023).

[24]"Energy@Home". (), dirección: `http://www.energy-home.it/SitePages/Home.aspx` (visitado 22-04-2023).

monitorización y control e información sobre eventos.

MAS2TERING[25], presentada por Hippolyte, Howell, Yuce et al. [53], es una ontología desarrollada bajo el proyecto homónimo, que implementa el modelo de conocimiento de USEF a través de sistemas multiagente. El propósito de la ontología es la representación de los datos de diferentes dominios de SG y proporcionar interoperabilidad entre los agentes de SG y las partes interesadas. Se basa en Energy@Home y el *(International Electrotechnical Commission's Common Information Model (CIM)* [153].

EnergyUse, de Burel, Piccolo y Alani [154], es un framework para la creación de aplicaciones de consejos de ahorro de energía en el hogar. Enriquece PowerONT [155] (la ontología de consumo de energía) con otras ontologías y la traduce a JSON-LD.

DABGEO, presentada por Cuenca, Larrinaga y Curry [156], es una ontología semánticamente equivalente a OEMA [157], una ontología anterior de los mismos autores. Mejora OEMA al ofrecer una ontología modular que se puede importar en subconjuntos, lo que facilita su adopción en escenarios de casos de uso personalizados. Como su antecesor, vincula y amplía conceptos de otras ontologías anteriores. La base de la red de ontologías es ThinkHome, a la que se han sumado SAREF4EE, EnergyUse y ProSGV3. La base de la red de ontologías es ThinkHome, a la que se han añadido SAREF4EE, EnergyUse y ProSGV3.

Smart Energy Domain Ontology (SARGON), presentada por Haghgoo, Sychev, Monti et al. [158] extiende SAREF aportando información sobre energía. Como veremos posteriormente, SAREF4EE también tiene un objetivo similar, pero a diferencia de SAREF4EE, SARGON se centra en el control y la supervisión de las redes eléctricas de distribución, integrándolo con la automatización energética de edificios.

OSEIM y *NewOSEIM*, de Saba, Sahli y Hadidi [159] y Saba, Sahli, Abanda et al. [160], aprovechan el razonamiento semántico sobre una ontología que presenta conocimiento sobre el entorno interno y externo de una casa, para lograr una gestión inteligente de la energía.

2.3.4 Ontologías de seguridad

Muchas son las ontologías ya desarrolladas dentro del contexto de la seguridad informática. En este subapartado enumeramos algunas de los trabajos más relevantes encontrados durante el estudio del estado del arte para esta tesis. Han sido seleccionadas en base a diferentes características: debido a su interés para el dominio de aplicación de IoT, debido a que están diseñadas o desarrolladas específicamente para IoT o debido a que se utilizan como base en el desarrollo de otras ontologías de interés para este trabajo. Los trabajos revisados en esta subsección están resumidos, para mayor conveniencia, en la Tabla 2.3.

DARPA Agent Markup Language (DAML) [161] fue presentada hace más de una década por Denker, Kagal, Finin et al. [162] y posteriormente refinada por Denker, Kagal y Finin

[25]"Multi-Agent Systems and Secured coupling of Telecom and Energy gRIds for Next Generation smartgrid services (MAS2TERING)". (), dirección: `http://www.mas2tering.eu/` (visitado 23-08-2022).

Tabla 2.3: Comparativa de las diferentes ontologías de seguridad analizadas

Ontología	**Dominio**	**Propósito**
DAML [162], [163]	Srv. web semánticos	Seguridad en Srv. Web
Kim [164]	Recursos electrónicos	Anotación de recursos
Herzog [166]	Seguridad de la información	Base de conocimiento
SO [167]	Seguridad de la información	Gestión de riesgos
STAC [168]	IoT	Base de conocimiento
IoTSec [171]	IoT	Ontología de referencia
Tao [175]	Hogar inteligente	Servicio de seguridad
Choi [176]	IoT de energía	Servicio de seguridad
IoTSecEv [7]	IoT	Evaluación de seguridad
IoT-Priv [177]	IoT	Privacidad

[163], en el marco de los Servicios Web Semánticos. Esta familia de ontologías cubre numerosos aspectos de la seguridad informática. Algunos de estos conceptos, modelados con gran detalle, son la autenticación, autorización, control de acceso, integridad de datos, distribución de claves y políticas. Este trabajo, sin embargo, presenta inconvenientes cuando se aplica al escenario de IoT, como por ejemplo que algunos de los conceptos descritos están desactualizados o no son aplicables a IoT, mientras que conceptos más recientes no se han agregado a la ontología.

Kim, Luo y Kang [164] agregan un conjunto de ontologías relacionadas bajo el mismo nombre: *Security Ontology for Annotating Resources*, mejorándolas y haciéndolas extensibles al redefinir conceptos para incrementar la expresividad. Una de las ontologías referidas sobre la que pretende aportar mejoras es DAML. Según afirman los autores del trabajo, DAML solo se centra en anotar servicios web, por lo que ellos expanden su utilidad generalizándola a la anotación de recursos.

An Ontology of Information Security [165], de Herzog, Shahmehri y Duma [166], nos presenta una ontología de seguridad de la información, públicamente disponible y basada en OWL. Esta modela conceptos como: activos, amenazas, vulnerabilidades y contramedidas; así como las relaciones entre estos conceptos. La ontología puede ser utilizada como un vocabulario general y como diccionario extensible del dominio de la seguridad de la información.

Security Ontology (SO), presentada por Fenz y Ekelhart [167], describe una ontología general de seguridad que provee la estructura para el dominio de la seguridad de la información. Además, enriquece el dominio permitiendo la aportación de conocimiento específico sobre la organización considerada. Se afirma que la ontología resultante, que contiene 500 conceptos y 600 restricciones formales, organizadas en cinco subontologías, da soporte a un amplio rango de aproximación a la gestión de riesgos de seguridad de la información. De manera similar a lo que vimos en el trabajo

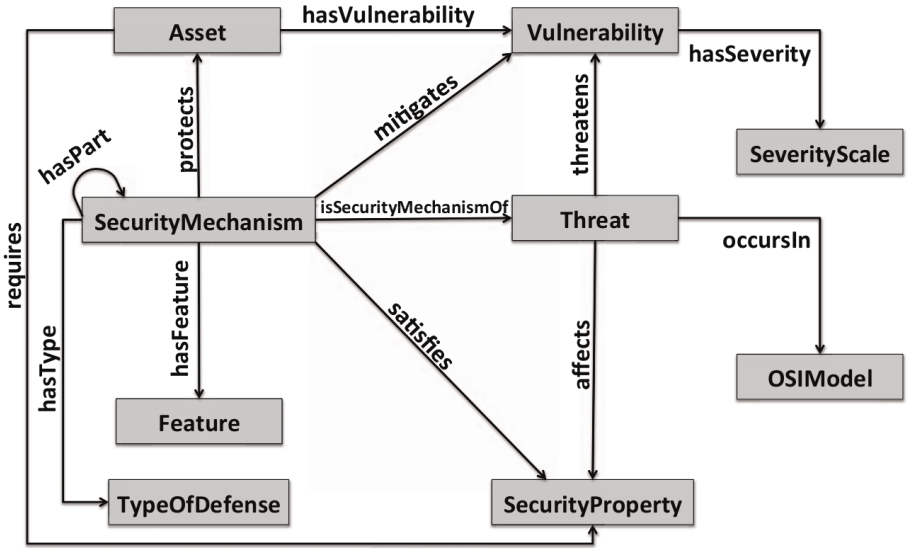


Figura 2.17: Diagrama de la ontología IoTSec, para representación de seguridad

anterior y veremos en sucesivos trabajos de este estado del arte, el vocabulario contiene términos para describir activos, amenazas, vulnerabilidades, ataques y contramedidas, concentrándose en una descripción general de seguridad del sistema, que puede ser usada como punto de partida en los procesos de evaluación de seguridad.

Gyrard, Bonnet y Boudaoud [168] presentan *Security Toolbox : Attacks and Countermeasures (STAC)*[26], otra ontología de seguridad, aunque en esta ocasión en el contexto del modelo de la arquitectura ETSI Machine 2 Machine (M2M) [170], construyendo una base de conocimiento sobre seguridad (ontología, conjunto de datos y reglas), para ayudar a los diseñadores a implementar la seguridad de aplicaciones M2M durante la fase de diseño. Una vez más, proporciona una descripción general de los sistemas generales de seguridad, centrándose en tecnologías específicas relacionadas con IoT, describiendo activos, amenazas y mecanismos de seguridad entre otros.

Mozzaquatro, Jardim-Goncalves y Agostinho [171] presentan *IoT Security Ontology (IoTSec)* [172], reuniendo y armonizando varias ontologías relacionadas (STAC una de ellas). Esta ontología (Figura 2.17) representa el conocimiento sobre seguridad de manera similar al trabajo anterior, proporcionando un conjunto de datos (o catálogo de conocimiento) extensible y amplio; así como una semántica expresiva para representar los rasgos relacionados con la seguridad. Pretende ser la ontología de referencia para la seguridad en IoT, incorporando la mayoría de las ontologías anteriormente mencionadas en un trabajo de homogeneización entre ellas.

SecAOnto [173], presentada por de Franco Rosa, Jino y Bonacin [174], es una ontología que formaliza el conocimiento sobre la evaluación de la seguridad, centrándose en sus aspectos y particularidades, abordando la relación entre la seguridad de la información y la evaluación del

[26]A. Gyrard. "STAC (Security Toolbox: Attack & Countermeasure)". Herramienta web de STAC. (2022), dirección: `http://sensormeasurement.appspot.com/?p=stac` (visitado 19-07-2023).

software. Al igual que el trabajo anterior, parte de STAC y tiene como objetivo respaldar métodos de evaluación de la seguridad basados en criterios rigurosos de evaluación.

Tao, Zuo, Liu et al. [175] presentan un framework de servicio de seguridad basado en ontologías, que respalda la preservación de la seguridad y la privacidad en las interacciones, mediante el uso de su ontología de seguridad. Esta define un vocabulario de seguridad común compartido por los proveedores de servicios y los clientes; así como razonamiento semántico basado en Semantic Web Reasoning Language (SWRL). Esta ontología permite una descripción explícita de los elementos de seguridad que intervienen en las comunicaciones entre dispositivos, centrándose en las propiedades de integridad y confidencialidad de la seguridad de la información mediante la descripción de conceptos como firma digital, encriptación y token de seguridad, relacionados con la protección de datos y control de acceso.

Choi y Choi [176] modelan una ontología de contexto de seguridad, en la que basan un framework de servicios de seguridad IoT-nube para sistemas de energía. Usando varias tecnologías de razonamiento ontológico, pueden responder a las intrusiones de seguridad de manera inteligente. Una vez más, la ontología modelada representa las diferentes amenazas, ataques y respuestas de una manera muy próxima al problema de dominio de la medición de energía, representando algunos conceptos simples como si el usuario tiene contraseña o si hay algún control de acceso a una red.

Como ya se ha mencionado en una sección anterior, Gonzalez-Gil, Skarmeta y Martinez [7], presentamos *IoTSecEv*: una ontología de evaluación de seguridad para IoT (Figura 2.18) basada en IoTSec y STAC. El objeto de esta ontología es la descripción de características y conceptos de seguridad, de interés para diferentes observadores. Mediante estos es posible realizar la evaluación de la seguridad de un sistema IoT, permitiendo la ordenación de recursos IoT ofrecidos en catálogos de agregadores, en función de requisitos personalizados de seguridad.

Arruda y Bulcão-Neto [177] presentan la ontología *IoT-Priv* como una capa de privacidad liviana que se basa en conceptos de IoT expresados en otras ontologías. Hace posible describir políticas y requisitos relacionados con la privacidad en IoT, lo que permite la evaluación de políticas utilizando enfoques ontológicos. Aunque no cubre aspectos como la autenticación o la identificación, sí cubre algunos de los temas de control de acceso de interés en este trabajo, específicamente los relacionados con el control de acceso basado en políticas, así como algunos conceptos de protección de datos y responsabilidad.

2.3.5 Conclusiones

Las ontologías consideradas en el apartado de seguridad, resumidas en la Tabla 2.3; están diseñadas desde una perspectiva “de arriba a abajo”, describiendo con gran nivel de detalle los diferentes elementos de seguridad que componen un sistema de tecnologías de la información, que abarca desde comunicaciones, redes y hardware informático, procesos de seguridad y hasta servicios y software comunes.

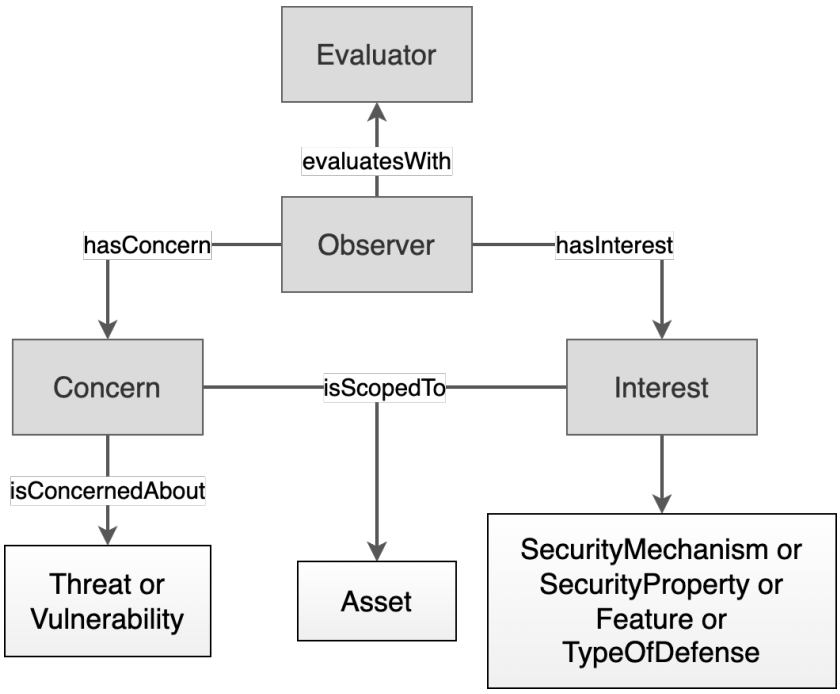


Figura 2.18: Diagrama de la ontología IoTSecEv, para evaluación de seguridad

En el estudio del arte realizado, no hemos encontrado ontologías diseñadas específicamente para expresar el tratamiento de la seguridad desde la perspectiva de los datos, que sería de gran utilidad de cara a poder describir los elementos de seguridad relevantes de cara a los consumidores de datos en plataformas de agregación de datos.

Las ontologías estudiadas tampoco capturan conceptos de seguridad de datos en profundidad, aunque algunos de ellos representan conceptos como "autenticación" o "control de acceso" como individuos en su base de conocimientos ontológicos, a menudo utilizados como base para realizar inferencias ontológicas (por ejemplo, enumerar amenazas a algún objetivo de seguridad).

Finalmente, otros conceptos relacionados con la seguridad y privacidad; como reglamentos y certificaciones que afectan a los datos en sí, tampoco han sido cubiertos por las ontologías consideradas y solo algunas de ellas son capaces de representar conceptos relacionados con la responsabilidad[27] como un elemento del conjunto más amplio definido en procedencia[28].

2.4 Objetivos de la tesis doctoral

2.4.1 Motivación

El estudio del estado del arte realizado muestra la gran cantidad de trabajo producido y el largo camino ya recorrido en el desarrollo de las tecnologías habilitadoras para el IoT y sin embargo,

[27] Del término *accountability* en inglés, referido al dominio de la seguridad informática.
[28] Del término *provenance* en inglés, en el dominio de la seguridad informática.

aún es mucho el trabajo pendiente de cara a poder cubrir las expectativas depositadas sobre este, con el objetivo de responder a los crecientes retos a los que se enfrenta nuestra sociedad, cada vez más dependiente de la tecnología.

Inspirados por los ODS mencionados en el Capítulo 1, específicamente *"11–Ciudades y comunidades sostenibles"* y *"12–Producción y consumo sostenible"*, a su vez íntimamente relacionados con otros ODS ligados a la producción de energía limpia (7) y la acción contra el cambio climático (13), se decidió enmarcar los trabajos de investigación de esta tesis doctoral en la dirección de los edificios/hogares inteligentes, la eficiencia energética y la seguridad y privacidad.

Tras un breve estudio inicial del estado del arte y un análisis preliminar de necesidades, identificamos tres ámbitos de estudio de interés, estrechamente relacionados: el diseño de arquitecturas para IoT interoperables y seguras, el modelado de datos utilizando tecnologías de la web semántica, y la necesidad de orquestar el procesado de la información entre el borde y la nube.

La justificación de esta elección radica en los siguientes puntos:

a) A pesar de que, como se ha podido ver en el estudio del arte, ya existen abundantes trabajos en arquitecturas de IoT; uno de los problemas fundamentales presentes en estas radica en la interoperabilidad: la capacidad de poder interconectar diferentes arquitecturas y compartir datos, una característica que resultará imprescindible en una economía basada en los datos.

b) No es posible atacar la interoperabilidad de las plataformas, haciendo hincapié exclusivamente en las interfaces y tecnologías seguridad. Es necesario modelar la información de tal forma que se puedan implementar estrategias automatizadas para el tratamiento de la información. Siguiendo el ejemplo de la web semántica, esto implica aplicar técnicas de modelado ontológico y la incorporación de metainformación junto con los datos. De forma novedosa en IoT, parte de la información que es necesario modelar de cara a mejorar la interoperabilidad, es la información del modelo de seguridad y privacidad, pero otros escenarios de aplicación también requieren una consideración profunda en función de sus necesidades (por ejemplo, modelado de datos para eficiencia energética en entorno de hogar/edificio inteligente).

c) El trabajo en los puntos anteriores nos permite expandir el ámbito de uso de las plataformas y arquitecturas IoT para abordar escenarios complejos, como el de las plataformas de compra/venta de datos, buscadores de información, plataformas de procesado de datos o verticales de eficiencia energética, donde los datos deben accedidos por multitud de agentes, compartidos, interpretados, procesados y retransmitidos y todo ello atendiendo a principios de seguridad y privacidad.

d) Finalmente, el punto anterior nos lleva de manera natural al siguiente reto: el de la computación en el borde y la orquestación distribuida de tareas, donde se aprovechan recursos computacionales en el borde para realizar cómputos próximos a donde serán utilizados o producidos, de forma que se optimice el uso de las infraestructuras de comunicaciones, se mejore la eficiencia global del sistema, su resiliencia e incluso su rendimiento.

Como se puede ver, estos ámbitos están estrechamente relacionados: la interoperabilidad de las arquitecturas depende en gran medida de la estandarización tanto de sus interfaces, como del propio modelo de los datos intercambiados y la facilidad con que pueda ser interpretados e integrados en sistemas existentes para su explotación. Así mismo, el procesado ubicuo de los datos depende de la arquitectura y el modelo de datos y, a raíz de las nuevas normativas relativas a la seguridad y privacidad de datos, estos requisitos deben ser factorizados igualmente en las arquitecturas resultantes.

La conclusión es que seguridad, modelo de datos, diseño de arquitectura y modelo de orquestación, deben ser considerados de forma holística de cara al diseño y aplicación de arquitecturas de datos para cualquiera de los escenarios de IoT que se planteen.

2.4.2 Objetivos

Como resultado del análisis de necesidades y el estudio preliminar del estado del arte, se plantean los siguientes objetivos generales para este trabajo doctoral:

- **O1.** Diseñar arquitecturas de datos interoperables basadas en plataformas abiertas, que permitan la integración de datos de fuentes diversas y el acceso a los datos almacenados de forma segura, garantizando la privacidad y aplicando enfoques innovadores en materia de búsqueda y acceso a la información.

- **O2.** Facilitar un modelo distribuido basado en la orquestación que permita la utilización de recursos locales y en la nube, para el reparto de tareas con necesidades especiales de latencia y potencia de cálculo, atendiendo a restricciones de ancho de banda, recursos computacionales y privacidad de datos.

- **O3.** Utilizar tecnologías semánticas en el modelado de datos, seleccionando y/o creando modelos y ontologías apropiadas para los dominios de seguridad de datos y eficiencia energética.

- **O4.** Aplicar las arquitecturas, modelos de datos y tecnologías de seguridad, en entornos reales de los ámbitos de la automatización, la eficiencia energética y la seguridad en el hogar/edificio inteligente.

2.4.3 Metodología

Esta tesis doctoral se desarrolló incrementalmente, partiendo de los objetivos mencionados en la sección anterior, simplificados en tres planos de actuación:

- Modelado de datos y estudio ontológico en los ámbitos de la seguridad, automatización del hogar, representación de datos de sensorización y eficiencia energética.

- Estudio de tecnologías para la orquestación de tareas entre el borde y la nube, aprovechando hardware en el borde y en la niebla para tratar los datos de acuerdo al principio de localidad espacial.
- Estudio y diseño de arquitecturas interoperables para el acceso a los datos de IoT aplicando tecnologías y estándares bien establecidos, atendiendo a la seguridad y la privacidad de los datos.

Las fases principales en que se desarrolló el trabajo en dichos planos fueron:

1. Estudio del estado del arte: estudio de trabajos previos y relacionados en metodologías ontológicas, ontologías existentes, arquitecturas y tecnologías de orquestación.
2. Diseño y desarrollo de arquitecturas y modelos: selección, composición y desarrollo de ontologías y diseño de arquitecturas para la interoperabilidad, atendiendo a la seguridad y la privacidad de los datos.
3. Validación de los resultados: validación por implementación de diversos casos de uso y bancos de prueba para arquitecturas, así como la aplicación de las ontologías seleccionadas, compuestas o creadas.

2.4.4 Resumen de resultados

El trabajo desarrollado durante este proyecto de doctorado tiene como resultado la producción de tres contribuciones principales, presentadas en el Capítulo 4, Capítulo 5 y Capítulo 3, así como un conjunto de publicaciones en revistas, congresos y capítulos de libro, derivadas de este trabajo, enumeradas en el Capítulo 7.

La Tabla 2.4 resume los resultados de investigación generales, obtenidos en el desarrollo de esta tesis, y los relaciona con los objetivos propuestos y las contribuciones presentadas.

Tabla 2.4: Relación entre los resultados, los objetivos y las publicaciones derivadas de la tesis

Resultado	**Obj.**	**Pub.**
R1. Creación de una ontología de seguridad de la información, desde la perspectiva de los datos; para asistir en la evaluación de la seguridad de la información, la categorización y el filtrado de conjuntos de datos en plataformas de mercado de datos IoT.	**O3**	[4]
R2. Creación de una arquitectura IoT para la gestión energética del hogar inteligente, integrable en esquemas de respuesta de demanda, basada en plataformas abiertas, atendiendo a la interoperabilidad y modularidad.	**O1-4**	[1]
R3. Desarrollo de una arquitectura distribuida para la orquestación de tareas de análisis de vídeo aprovechando dispositivos en el borde (*edge*) y aceleración por hardware de redes neurales profundas de bajo consumo.	**O1-2**, **O4**	[2]
R4. Selección de ontologías y mapeo a modelos de datos para la representación de información relacionada con IoT, gestión y eficiencia energética, SH y SC.	**O3**	[1], [3]-[5], [7]
R5. Aplicación de las arquitecturas de datos para IoT diseñadas, en diversos casos de uso específicos y bancos de prueba en diversos ámbitos	**O4**	[1]-[3], [5]

CAPÍTULO 3

ONTOLOGÍA DE SEGURIDAD DE LA INFORMACIÓN

Este capítulo presenta una ontología ligera para la representación de conceptos de seguridad de los datos en IoT, con la novedad de realizarse desde la perspectiva de los propios datos e introduciendo algunos conceptos nuevos, tales como regulaciones, certificaciones y procedencia, a los conceptos clásicos tales como control de acceso y mecanismos de autenticación.

La inspiración para la creación de esta ontología surge a partir de las necesidades del proyecto IoTCrawler[1]; un proyecto de investigación financiado por el programa de investigación e innovación Horizonte 2020 de la Unión Europea. Su objetivo principal es desarrollar un motor de búsqueda para IoT. El enfoque de IoTCrawler se centra en la integración y la interoperabilidad entre diferentes plataformas, ofreciendo soluciones dinámicas y reconfigurables para el descubrimiento e integración de datos y servicios de sistemas heredados y nuevos, de forma segura y atendiendo a la privacidad.

Toma la forma de un framework compuesto por varios componentes especializados, que comparten NGSI-LD como interfaz común para el intercambio de datos, y tiene en cuenta aspectos como la escalabilidad de todo el sistema, la indexación y clasificación de los datos registrados en la plataforma, el rastreo de datos de diferentes dominios existentes y la seguridad, entre otros. Además, el proyecto proporciona algoritmos y mecanismos seguros, adaptables y conscientes de la privacidad, los cuales permiten rastrear, indexar y clasificar sistemas IoT distribuidos.

El framework de IoTCrawler, representado en la Figura 3.1, se compone de los siguientes componentes:

- Repositorio de metadatos (metadata repository, MDR): actúa como intermediario de contexto, integrando y distribuyendo información de contexto entre los demás componentes de

[1]"IoTCrawler – a search engine for Internet of Things devices". Sitio web del proyecto IoTCrawler. (), dirección: `http://iotcrawler.eu` (visitado 21-07-2023).

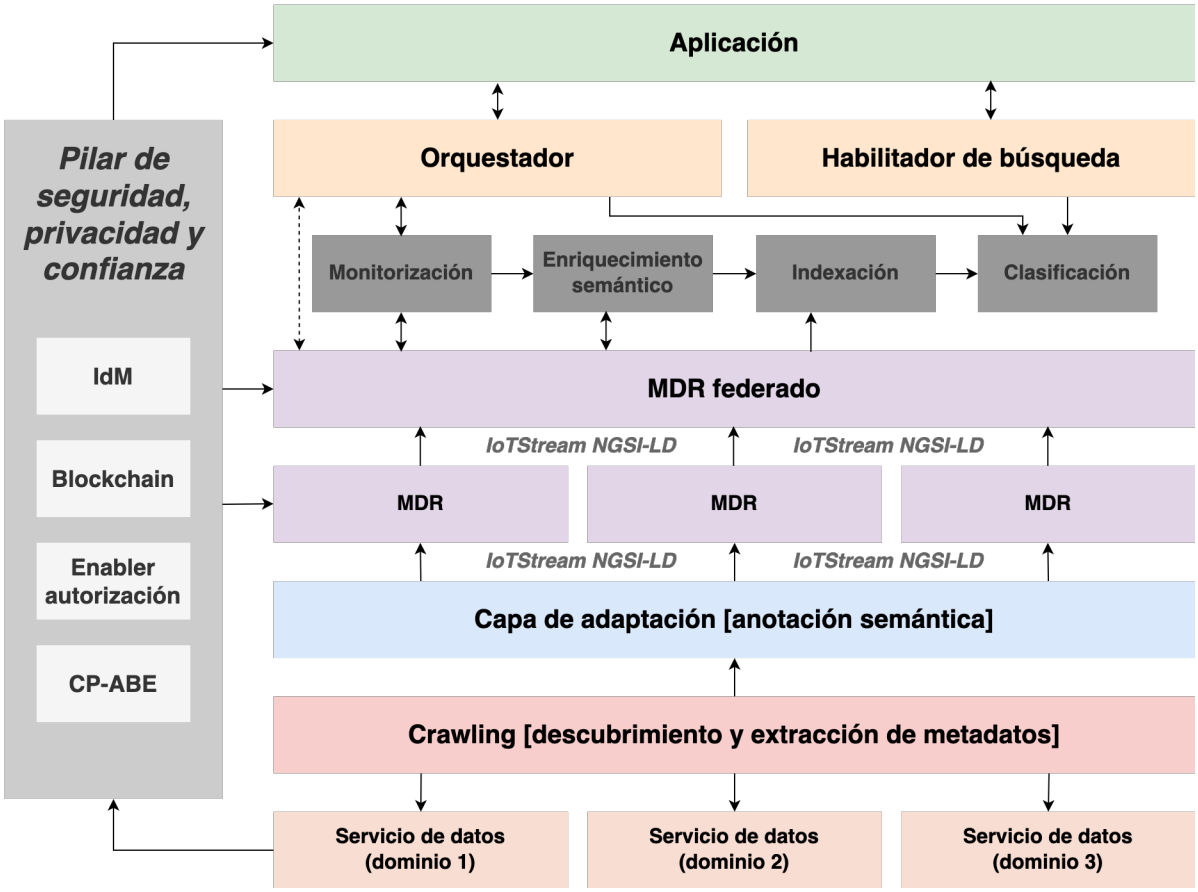


Figura 3.1: Arquitectura modular del framework del proyecto IoTCrawler y relaciones entre componentes

IoTCrawler mediante los patrones de consulta-respuesta y publicación-suscripción.

- Orquestador: permite que las aplicaciones de IoT del cliente se suscriban a flujos de datos sin necesidad de contar con un punto de conexión público, y realiza el seguimiento de las solicitudes de suscripción.

- Indexación: se utiliza para rastrear e indexar entidades mediante consultas basadas en el tipo de sensor y la ubicación.

- Clasificación: tiene como objetivo ayudar a los usuarios y aplicaciones a encontrar un conjunto de recursos relevantes para sus necesidades, y seleccionar los mejores o más apropiados dentro de dicho conjunto.

- Habilitador de búsqueda: proporciona una interfaz GraphQL para solicitar datos del MDR. El lenguaje GraphQL permite consultar y filtrar entidades de varios tipos en una sola consulta.

- Enriquecimiento semántico: es responsable de anotar los flujos de datos con calidad de información (QoI).

- Pilar de privacidad, seguridad y confianza: compuesto por PAP-PDP, PEP-Proxy, Keyrock, Security Facade, Capability Manager y Blockchain Handler.

Posteriormente en este capítulo, en la Sección 3.3, se profundiza en el modelo de datos de IoTCrawler y como se relaciona la ontología de esta propuesta en el mismo.

3.1 Descripción del problema

El problema que se pretende resolver en esta contribución es la descripción de los aspectos de seguridad de la información que afectan a los sistemas IoT. La justificación de este problema parte de un amplio conjunto de casos de uso, seleccionados con el objetivo de identificar diferentes necesidades en la descripción de la seguridad de los datos. A continuación se lista un subconjunto de muestra de los problemas, encontrados en los mencionados casos de uso, que motivan este trabajo:

- Como intercambiar requisitos de seguridad de datos en un escenario de datos federados. Los escenarios federados permiten a los solicitantes utilizar diferentes puntos de acceso para recuperar datos, que provienen de una ubicación digital diferente del punto de solicitud. Este escenario da lugar a sistemas altamente escalables, especialmente interesantes en SC, SG e Industria 4.0 entre otros. Esta federación también plantea la cuestión de como comunicar información relevante sobre la seguridad de los datos entre los elementos federados del sistema, de modo que se pueda aplicar el control de acceso.

- Como representar los aspectos de seguridad de los datos en los motores de búsqueda. Los mercados de datos y los motores de búsqueda permiten a los solicitantes buscar y clasificar datos y fuentes de datos provenientes de terceros, que cuentan con sus propios mecanismos de seguridad. En este caso, las anotaciones de seguridad de la información acerca de los datos ofrecidos por el motor de búsqueda pueden permitir no solo una forma efectiva de realizar el filtrado y la clasificación en función de los rasgos de seguridad de la fuente de datos, sino también ayudar al solicitante en la recuperación de esa información de la fuente, especialmente en el caso de M2M, donde el procesamiento semántico es crucial.

- Como expresar los requisitos de seguridad de datos de los proveedores de información. Los agregadores y las plataformas IoT que permiten a los proveedores de datos registrarse en la plataforma, necesitan una forma de expresar los requisitos de seguridad para poder transportar esa información desde el proveedor de datos a la plataforma y entre la plataforma y cualquier tercero. Los procesadores de datos y los clientes de datos se beneficiarían de

la información sobre la seguridad de los datos en esos conjuntos de datos, para facilitar el acceso y el manejo de los datos.

La revisión sistemática de ontologías de seguridad general que se ofrece en la Sección 2.3.4, revela un buen número de ontologías existentes, algunas de ellas ya dirigidas a IoT. También muestra que todas las ontologías referidas están diseñadas desde una perspectiva clásica de recursos, anotando dispositivos, servicios, bases de datos, elementos de red, tecnologías de comunicación, etc. Estas generan una malla de conocimiento que captura la estructura de seguridad global de un sistema, con el objetivo general de realizar evaluaciones de seguridad o ayudar en el desarrollo de dichos sistemas.

La evaluación inicial de las necesidades de seguridad de datos para los escenarios mencionados anteriormente muestra que es preciso anotar información funcional específica sobre seguridad de datos, cubriendo los aspectos básicos de seguridad, representados por las propiedades de la triada CIA (*confidencialidad*, *integridad* y *disponibilidad*), que no han sido previamente cubiertas por ninguna de las ontologías estudiadas.

Como se muestra en la Figura 3.2, es necesario poder anotar información sobre, al menos, los mecanismos de control de acceso, la protección de datos y la procedencia. También se deduce que, para cubrir mejor el aspecto de integridad en los datos, es posible proporcionar información adicional en forma de certificaciones, que es un campo aún no desarrollado en el ámbito de la seguridad de datos en IoT, pero que está mostrando características interesantes y tracción académica [179], ofreciendo un candidato viable para ayudar a respaldar ese aspecto de la seguridad. Finalmente, dado el impacto que están teniendo normativas como el Reglamento General de Protección de Datos (General Data Protection Regulation, GDPR) en los aspectos de seguridad de los datos [84], también se decide incorporar el concepto de “normativa”.

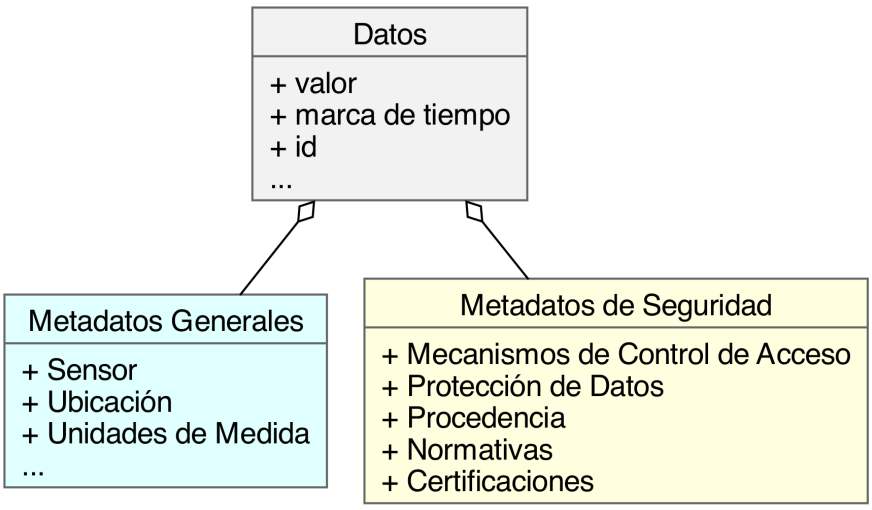


Figura 3.2: Modelo de datos simplificado de los elementos de anotación de seguridad

Por último, se establece una serie de restricciones y objetivos que debe cumplir el artefacto resultante:

- Enfoque a la perspectiva de los datos, a diferencia de las descripciones generales de sistemas. Se pretenden anotar los datos con características de seguridad para que los interesados puedan manejarlos, accederlos y comprenderlos mejor.
- Abordar problemas específicos tanto de IoT, como de M2M, agregación de datos, información personal, seguridad de extremo a extremo y procedencia.
- Favorecer el conocimiento implícito sobre el explícito, evitando describir en exceso conceptos que puedan ser extrapolados o representados mediante enlaces a información externa.
- Ofrecer una solución lo más sencilla y liviana posible, sin comprometer la funcionalidad; para facilitar la curva de aprendizaje y el impacto de desarrollo y aplicación en sistemas nuevos y existentes.
- Facilidad para su composición con otras ontologías existentes o su uso junto a ellas.

En los siguientes subapartados se muestran los diferentes elementos relacionados con la seguridad y privacidad de los datos, que deberán ser representados por la ontología resultante.

3.1.1 Mecanismos de control de acceso

Es preciso proporcionar información sobre el proceso completo requerido para acceder a algunos datos (o parte de ellos). De esta manera, los frameworks IoT, agregadores de datos, catálogos de datos y motores de búsqueda pueden proporcionar y utilizar enlaces enriquecidos que describen la información mínima necesaria para acceder a los datos.

En base a los objetivos anteriores, la información mínima requerida para poder acceder a los datos consiste en el esquema de control de acceso utilizado y el proveedor de autenticación seleccionado. En esta propuesta se han considerado esquemas como control de acceso basado en atributos (attribute-based access control, ABAC) [92], control de acceso basado en roles (role-based access control, RBAC) [91], control de acceso basado en organización (organization-based access control, OrBAC), control de acceso basado en reglas (rule-based access control, RAC) y control de acceso basado en identidad (identity-based access control, IBAC). Los nuevos esquemas de control de acceso, específicamente orientados hacia IoT tales como el control de acceso distribuido basado en capacidades (distributed capability-based access control, DCapBAC) [106] que mejoran la capacidad de escalado de ABAC, requieren más información; como la ubicación del servicio de gestor de capacidades (capability manager, CM) para poder obtener los tokens de capacidad necesarios para interactuar correctamente con los datos.

Finalmente, la mayoría de los mecanismos de control de acceso requieren alguna forma de identificación del solicitante, que será proporcionada por un proveedor de autenticación (cubierto en la Sección 3.1.4).

3.1.2 Datos ocultos

Ocultar datos es el acto de esconder o eliminar todos los rastros disponibles de la información a un observador, de modo que este no pueda saber si la información existe o no. Esto es diferente del control de acceso normal en que, si el agente que solicita los datos no tiene acceso, se emitirá un error notificando al usuario de su incapacidad para recuperarlos, proporcionando una pista sobre si esos datos realmente existen o no, mientras que los datos ocultos no ofrecen tal pista, simplemente ocultan al solicitante la información no autorizada.

Un ejemplo de esto podría ser la visualización de información personal en una página de usuario. Cuando no se muestre la información, puede ser resultado de una de dos cosas: el usuario no aportó dicha información o el usuario solicitó específicamente su ocultación, de tal forma que solo usuarios específicos pueden verla.

Una vez más, los elementos mínimos requeridos para ser anotados, para que este enfoque sea posible, son la identificación de los solicitantes, un punto de control y algún tipo de reglas, roles o políticas involucradas sobre qué y cuándo ocultar los datos sensibles.

3.1.3 Datos encriptados

Junto con los datos ocultos, los datos encriptados pertenecen a la categoría de datos secretos. Esta vez, la diferencia radica en que la información se presenta a todos los observadores, pero su contenido está encriptado, de modo que solo aquellos que tienen la clave de encriptación correcta pueden descifrarlo.

El cifrado basado en atributos de políticas de texto cifrado (ciphertext-policy attribute-based encryption, CP-ABE) [109] y algunas de sus variantes, son tecnologías de interés en el ámbito de IoT, que ofrecen la ocultación de datos a plena vista cifrándolos en una manera que sólo los destinos deseados podrán descifrarlo. Esta forma de proteger la información garantiza que los datos se almacenen y viajen de forma segura, lo que aumenta de manera efectiva la resiliencia de todo el sistema frente a una gran cantidad de amenazas.

A diferencia de los mecanismos de control de acceso habituales, estos esquemas criptográficos se basan en la distribución de las claves criptográficas entre los destinos, sin necesidad de ningún tipo de intercambio de tokens entre el titular de los datos y el lector. Para realizar dicha distribución de claves criptográficas, se debe establecer un administrador de cifrado, que además requerirá algún tipo de mecanismo de identificación.

3.1.4 Mecanismos de identificación y autenticación

Cuando se requiere alguna forma de identificación y autenticación, la forma en que se lleva a cabo el proceso depende de la tecnología seleccionada. Siendo IoT un escenario tan diverso, se beneficia especialmente de la utilización de proveedores de autenticación comunes y estandarizados. La información más necesaria y útil requerida para un acceso exitoso consiste en la referencia a la tecnología utilizada, así como el punto de acceso al servicio. Por ejemplo, Oauth2[2] usa el encabezado `x-auth-token`, de manera similar, el mismo principio se puede aplicar a otras tecnologías, como implementaciones de credenciales verificables [181] u otros mecanismos de autenticación destinados a resolver restricciones específicas de IoT [182], [183].

3.1.5 Procedencia, certificación y normativa

La procedencia y la certificación son dos requisitos desafiantes de este trabajo, ya que se han trabajado poco hasta la fecha en el ámbito de IoT. En ese sentido, la estructura y propiedades asociadas a estos conceptos son susceptibles de ser actualizadas en el futuro, a medida que se agreguen más alternativas en estos campos.

Por ahora, la información más relevante en materia de certificación (suponiendo que evolucionarán de manera similar a otras certificaciones existentes) consiste en la referencia de la certificación, identificada mediante su localizador de recurso universal (universal resource locator, URL) y algún tipo de código del certificado que permitiría al interesado consultar con la autoridad de certificación la validez del certificado, así como detalles específicos, tales como su alcance o modalidad.

La procedencia en IoT viene con su propio conjunto de desafíos también [184], aunque ya existen algunas propuestas [185], [186] que se inclinan hacia el uso de tecnologías de BlockChain. Nuevamente, y considerando un diseño minimalista, que se apoya en información implícita, la información mínima requerida para poder interactuar con los proveedores de procedencia sería conocer su tecnología (que se puede representar por medio de una URL o página web) y punto de acceso al servicio donde se realizan las consultas.

Por último, la normativa es un concepto tangente a los aspectos de seguridad. Está relacionado con la privacidad, como es el caso de la normativa del GDPR y, como tal, puede considerarse parte legítima del atributo de confidencialidad de los datos. Una vez más, los atributos requeridos que deben ser capturados por los datos, con respecto a las regulaciones que los afectan, seguramente evolucionarán; pero es seguro asumir que la referencia de la normativa (incluso sus secciones específicas) se puede vincular por medio de URL.

[2] "OAuth 2.0". Sitio web de OAuth 2.0. (), dirección: `https://oauth.net/2/` (visitado 21-07-2023).

3.2 Propuesta

En esta sección se describe la metodología seguida para el diseño y construcción de la ontología *Data Security for IoT* (DS4IoT), que resuelve los problemas descritos en la Sección 3.1 de acuerdo a las restricciones y objetivos allí establecidos. El artefacto resultante es descrito más adelante, en la Sección 3.2.2.

3.2.1 Metodología ontológica

Para el desarrollo de la ontología se siguió la metodología NeOn, descrita en la Sección 2.3.1, del estado del arte. Específicamente se siguió el escenario *"1. De la especificación a la implementación"*, que describe los pasos y procesos involucrados en la construcción de una ontología desde cero. También se consideró la posibilidad de integrar conceptos de algunas de las ontologías descritas en el estado del arte (Sección 2.3.4), creando una implementación de referencia que se vinculara con esos otros conceptos, pero finalmente se decidió en contra. La razón fue que, aunque estas ontologías describen en diferentes niveles de detalle algunos de los conceptos capturados en DS4IoT, el significado subyacente es fundamentalmente diferente y podría dar lugar a problemas de inferencia y equívocos en la descripción de conceptos.

El modelado de DS4IoT, cuyo proceso general se describe en la Figura 3.3, comenzó con la especificación del problema, estableciendo los requisitos básicos de anotación de datos para la descripción de conceptos de seguridad de datos en los escenarios actuales de IoT. Luego siguió un estudio sobre el trabajo previo sobre seguridad de datos y ontologías existentes relacionadas con IoT, descrito en el estado del arte. El resultado de esta primera fase fue el documento de especificación de requisitos ontológicos, que se se utilizó como guía principal en la siguiente fase.

Desde el documento de especificación de requisitos ontológicos, se siguió NeOn llevando a cabo el proceso de formalización de la ontología, produciendo iterativamente un vocabulario, un tesauro y gráficos de conceptos que capturan la conceptualización mínima de las características de seguridad de datos para el ámbito de IoT. Tras la fase de formalización siguió la actividad de implementación de la ontología, en la que se generó un modelo en lenguaje OWL-DL. Este desarrollo se realizó con la ayuda de la herramienta Protégé[3], que proporciona una serie de ayudantes de validación e inferencia, así como herramientas de visualización, que fueron muy útiles en las etapas de desarrollo y validación. La Figura 3.4 muestra la ontología resultante y las diferentes clases capturadas por ella.

Adicionalmente a NeOn, también se siguió las mejores prácticas de la web semántica [188] comúnmente utilizadas en la creación de ontologías; como usar definiciones de metadatos ya existentes [189] y compartir la ontología resultante al a través de la Web, así como solicitar que se haga referencia a ella en los catálogos LOV y LOV4IoT [190] y motores de búsqueda semánticos. Además, el haber utilizando la herramienta Protégé durante las fases de diseño, validación y

[3] "Protégé: a free, open-source ontology editor and framework for building intelligent systems", Stanford University. (), dirección: `https://protege.stanford.edu/` (visitado 15-07-2023).

prueba; facilita aún más su uso compartido y reutilización en otros trabajos, al tratarse de una herramienta gratuita, de código abierto y públicamente disponible.

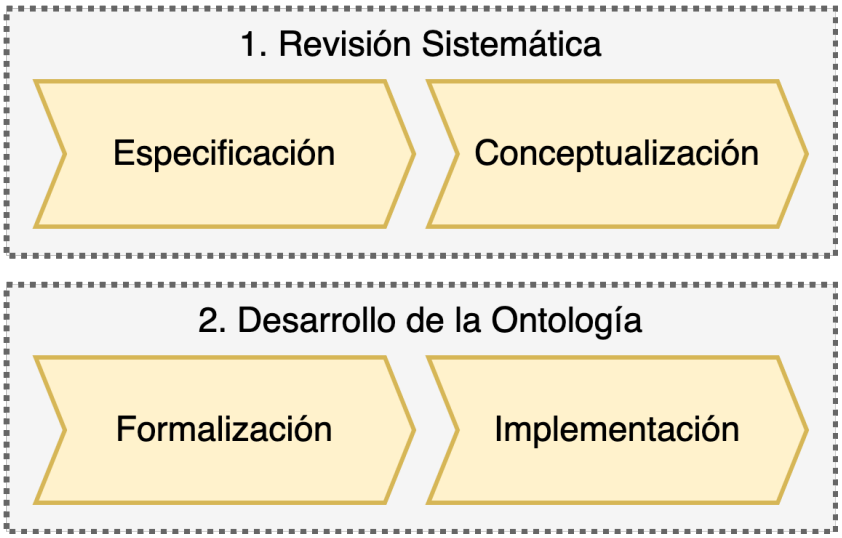


Figura 3.3: Proceso de creación de la ontología DS4IoT, siguiendo la metodología NeOn

3.2.2 Ontología DS4IoT

A partir del diccionario de sinónimos y el glosario de términos construidos, se modelan los diferentes conceptos relacionados con la ontología propuesta, en forma de 25 clases distintas, 16 propiedades de objetos y 3 propiedades de datos. Las dos jerarquías principales de clases se pueden ver en la Figura 3.5, que representa las familias de la clase `SecureData` y la clase `AccessControl`.

La clase principal de la ontología es `SecureData`, mostrada en la Figura 3.5(a). Esta representa el documento o fragmento de datos en el que estamos anotando información de seguridad de datos. Esta clase tiene una jerarquía de subclases relacionadas, como `SecretData`, que a su vez puede especializarse como `HiddenData` o `EncryptedData`. Ninguna de las clases de `SecureData` pertenece a conjuntos disjuntos, lo que significa que cualquier documento o fragmento se puede etiquetar como cualquier número de esas clases (por ejemplo, `EncryptedData` y `ProtectedData`).

La siguiente jerarquía de clases corresponde a la familia `AccessControl` (Figura 3.5(b)). Esta clase (y sus descendientes) representa mecanismos de control de acceso que imponen restricciones para una parte autenticada en el acceso a algunos datos. Algunas de las subclases son las especializaciones para mecanismos RBAC o ABAC (y consiguientemente en mayor grado de especialización por DCapBAC).

La Figura 3.6 muestra las principales relaciones. Algunos de esos enlaces representan las relaciones entre `AccessControl`, así como las clases `CryptoManager` y `CapabilityManager` y la clase `AuthenticationProvider`. La información sobre los puntos de acceso a servicios, relacio-

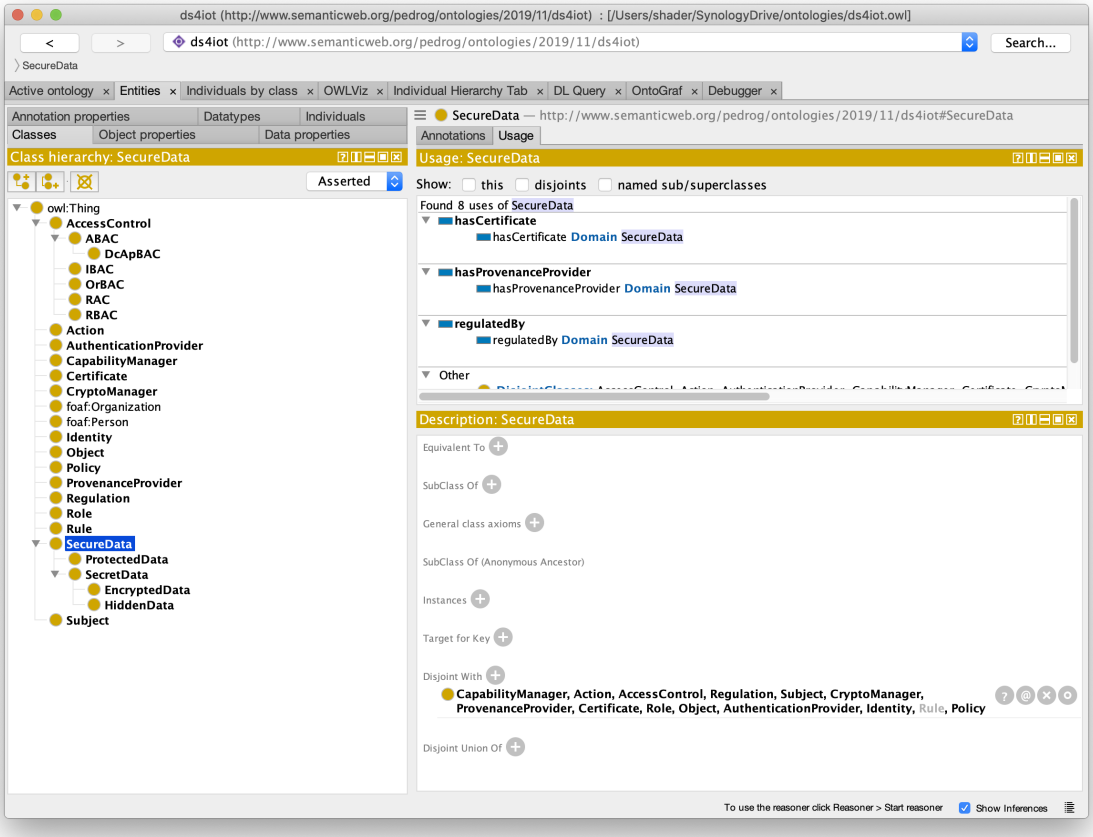


Figura 3.4: Ontología DS4IoT cargada en el programa Protégé

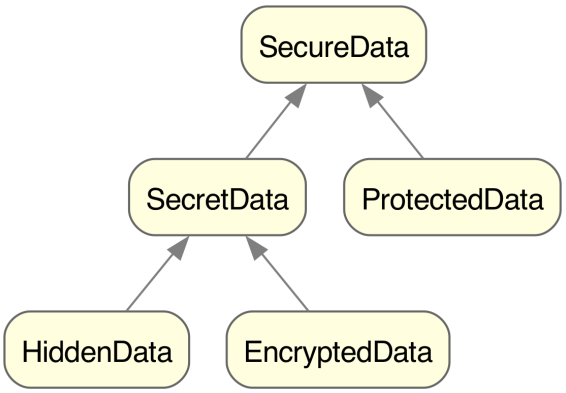
nados con cada una de las entidades asociadas a esas clases, se encuentra en una propiedad de datos de tipo `xsd:anyURI`.

La Figura 3.7 muestra los tres conceptos restantes de `SecureData`, representados por las clases `Regulation`, `Certificate` y `ProvenanceProvider`, que contienen propiedades de datos de tipo `xsd:anyURI` enlazando a las URL de las correspondientes normativas, autoridades certificadoras y proveedores de procedencia.

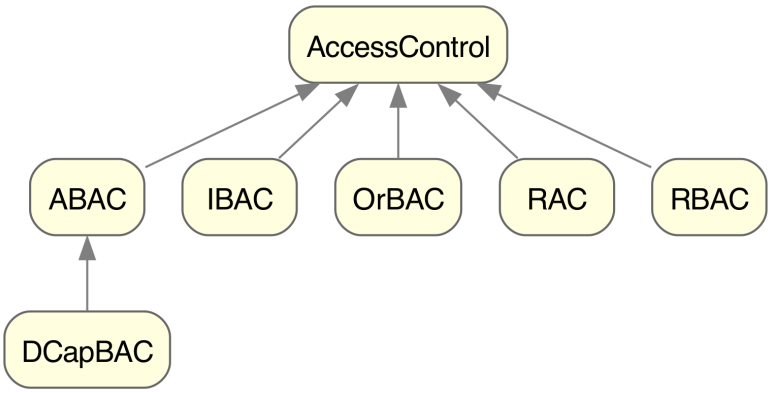
Por último, también se describen en la ontología algunas clases utilitarias para representar más información que podría usarse en casos especiales; como las anotaciones de seguridad de datos específicas del proveedor de datos, representadas por `Policy` y `Organization`.

3.3 Validación

Sirviendo tanto de muestra del potencial de la ontología DS4IoT, así como de validación de la misma; en esta sección se presenta un ejemplo de aplicación y prueba de concepto de la ontología DS4IoT. Como mencionábamos al principio de este capítulo, esta ha sido aplicada al proyecto



((a)) Jerarquía de la clase `SecureData`



((b)) Jerarquía de la clase `AccessControl`

Figura 3.5: Taxonomía de clases de DS4IoT

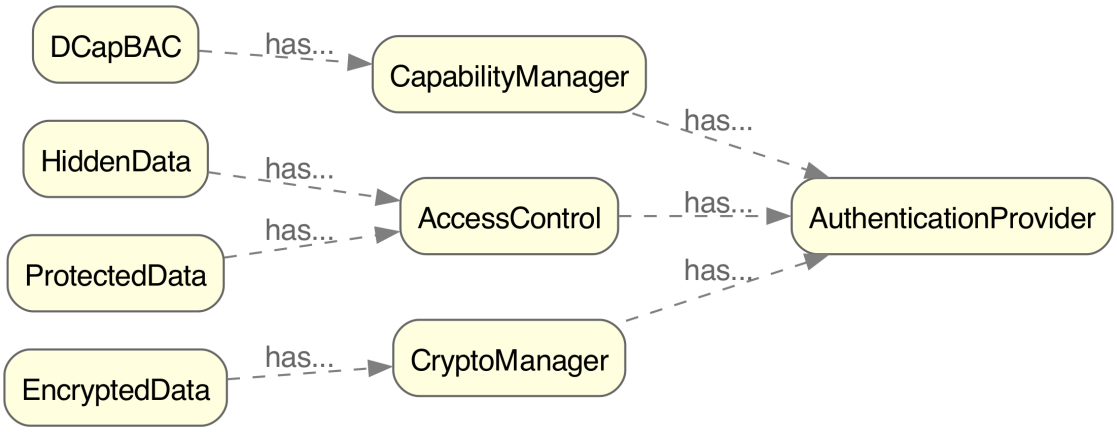


Figura 3.6: Propiedades de objeto relacionadas al control de acceso

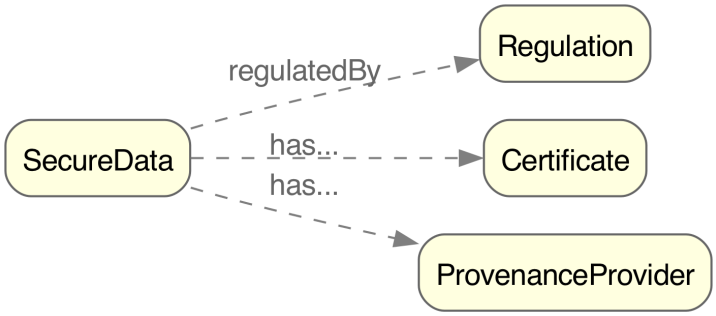


Figura 3.7: Propiedades de objeto relacionadas con regulaciones, certificados y procedencia

IoTCrawler.

En este proyecto la seguridad también se consideró como una de las dimensiones propuestas de la QoI como se muestra en la Figura 3.8, donde se puede ver que uno de los conceptos principales de la ontología de IoTCrawler es `IoTStream` que representa un flujo de datos generado por algún sensor. Para evaluar esta dimensión de la métrica de QoI para un flujo de datos de IoT, primero se necesita una descripción de los rasgos de seguridad de datos asociados a ese flujo.

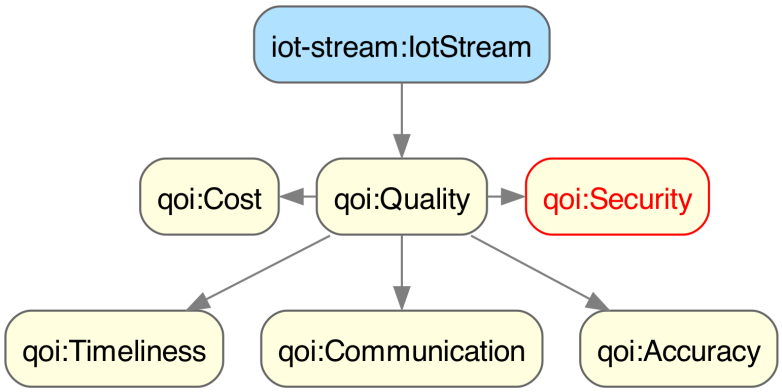


Figura 3.8: Detalle de la ontolgía de IoTCrawler, mostrando la dimensión de `qoi:Security`

Otro aspecto en el que DS4IoT beneficia a IoTCrawler es en la gestión de la seguridad de los datos como parte, tanto de la aplicación de la seguridad de los datos, como del acceso a los datos desde diferentes componentes dentro de la plataforma. La Figura 3.9 muestra como se estructuran de manera jerárquica los diferentes MDR, que contienen información sobre los diferentes IoTStreams registrados en el sistema. Esta estructura permite la integración de las plataformas existentes en el marco IoTCrawler, proporcionando al mismo tiempo, la base para la escalabilidad del sistema. Al ser la seguridad un elemento transversal, existe la necesidad de comunicar y representar los aspectos de seguridad de los datos en todos los dominios.

El primer paso para integrar DS4IoT en la ontología de IoTCrawler (Figura 3.10) fue asignar DS4IoT a la ontología NGSI-LD, que luego dirigiría el modelo de datos y la representación de la información en JSON-LD.

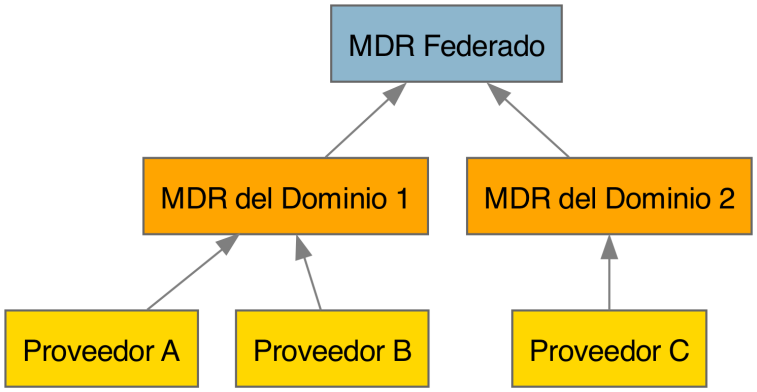


Figura 3.9: Arquitectura federada de IoTCrawler, mostrando MDR normales y federados

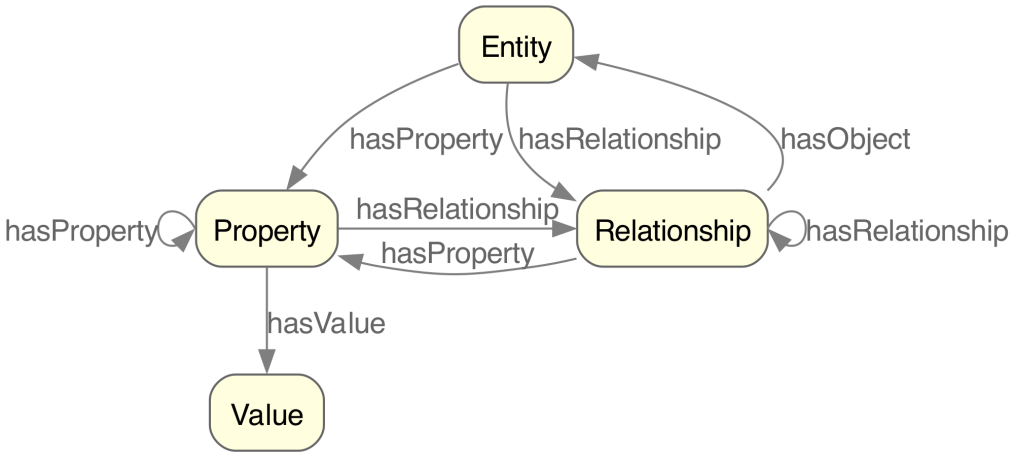


Figura 3.10: Conceptos principales de la ontología NGSI-LD

La esencia de esto es que la información en NGSI-LD está contenida en entidades que contienen relaciones apuntando a otras entidades y propiedades que contienen valores. En cierto sentido, son muy similares a las `objectProperties` y `dataProperties` de OWL. Además, tanto las propiedades como las relaciones pueden contener más propiedades y relaciones, enriqueciéndolas y agregando más información. Por ejemplo, dada una entidad que representa un sensor, una propiedad podría ser la lectura de temperatura y las propiedades de esa lectura podrían ser los valores máximo y mínimo que podría tomar.

Existen muchas alternativas válidas para el mapeo de DS4IoT a NGSI-LD, y la elegida es la que mejor se ajusta a su posterior adopción en la ontología de IoTCrawler que, como se menciona con anterioridad, integra una serie de ontologías, SOSA [132], [133] entre ellas.

La Figura 3.11 muestra el mapeo conceptual entre DS4IoT (a la izquierda) y NGSI-LD (derecha), donde las clases se representan en amarillo claro, las propiedades del objeto en verde claro y las propiedades de los datos en azul claro. En líneas discontinuas rojas y azules el dominio y rango y en líneas grises el mapeo entre conceptos. Usando este mapeo podemos

representar posteriormente la información de control de acceso requerida para acceder a un sensor, representada por una entidad de tipo `IoTStream`, de la ontología IoTCrawler.

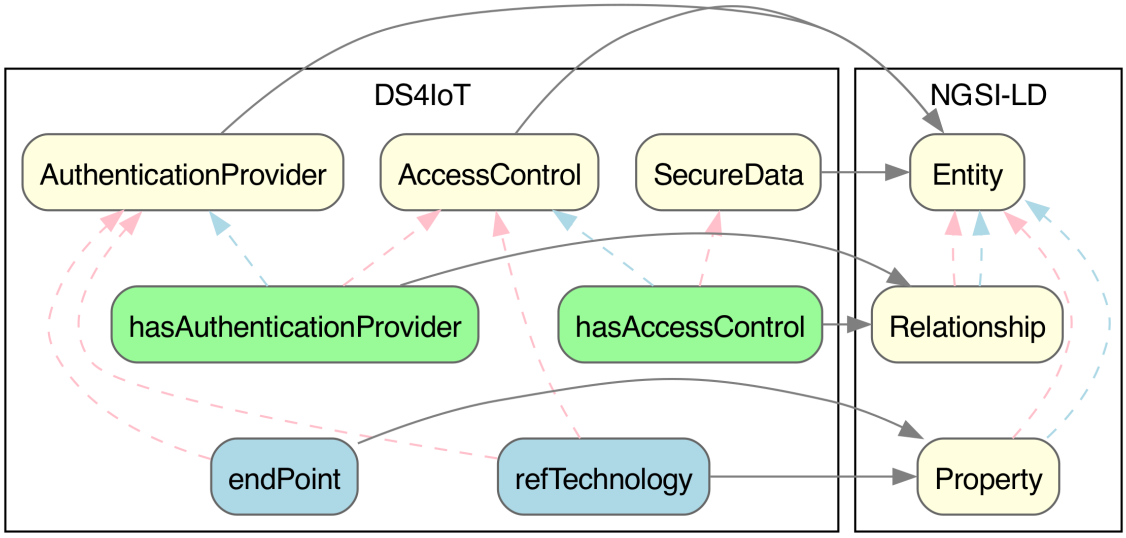


Figura 3.11: Mapeo conceptual entre DS4IoT y NGSI-LD

En el Listado 2 podemos ver una entidad NGSI-LD válida, representada en JSON-LD, que está vinculada a través de la propiedad de objeto `hasAccessControl` (representada por medio de una relación NGSI-LD), vinculando a otra entidad.

En el Listado 3 podemos ver el destino de la relación `hasAccessControl` del `IoTStream` anterior. Esta vez, una entidad de tipo ABAC (una subclase directa de `AccessControl`), de la ontología DS4IoT, representa la información necesaria para obtener acceso a la transmisión. También se podría haber representado más información aquí, como la `contactPerson` responsable del control de acceso del `IoTStream` referido, o el `CapabilityManager` donde se podría recuperar el `CapabilityToken`. Además de la relación del proveedor de autenticación, `refTechnology` (mapeado como `Property`) brinda otra información crucial, al vincular a la URL del PEP-Proxy Wilma de FIWARE, la tecnología real utilizada para ejecutar el control de acceso al stream, que es un sistema basado en políticas (de ahí el uso de la clase `ABAC`).

Finalmente, el Listado 4 muestra los detalles sobre el `AuthenticationProvider` vinculado por el mecanismo `AccessControl` previamente revisado. Esta vez, `refTechnology` enlaza con el administrador de identidad Keyrock IdM de FIWARE, basado en OAuth2.0. Además, el `endPoint`, también asignado como `Property`, se vincula con el punto de servicio real donde se debe realizar la autenticación.

```
{
  "id": "urn:ngsi-ld:Stream:stream1:Temperature",
  "type": "IoTStream",
  "observes": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:ObservableProperty:temperature"
  },
  "generatedBy": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:Sensor:sensor1"
  },
  "ds4iot:hasAccessControl": {
    "type": "Relationship",
    "object": "urn:ngsi-dl:AccessControl:ABAC:accctrl1"
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
    {
      "IoTStream": "http://purl.org/iot/ontology/iot-stream#IotStream",
      "observes": "http://www.w3.org/ns/sosa/observes",
      "generatedBy": "http://purl.org/iot/ontology/iot-stream#generatedBy",
      "hasSimpleResult": "http://www.w3.org/ns/sosa/hasSimpleResult",
      "ds4iot": "http://www.semanticweb.org/pedrog/ontologies/2019/11/ds4iot#"
    }
  ]
}
```

Listado 2: Entidad NGSI-LD representando un `IoTStream` con metadatos DS4IoT en IoTCrawler

```
{
  "id": "urn:ngsi-dl:AccessControl:ABAC:accctrl1",
  "type": "ds4iot:ABAC",
  "http://www.w3.org/ns/sosa/observes": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:ObservableProperty:temperature"
  },
  "ds4iot:hasAuthenticationProvider": {
    "type": "Relationship",
    "object": "urn:ngsi-dl:AccessControl:AuthenticationProvider:authprov1"
  },
  "ds4iot:refTechnology": {
    "type": "Property",
    "value": "https://github.com/ging/fiware-pep-proxy"
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
    {
      "ds4iot": "http://www.semanticweb.org/pedrog/ontologies/2019/11/ds4iot#"
    }
  ]
}
```

Listado 3: Entidad NGSI-LD representando un objeto de la clase `AccessControl`

3.4 Conclusiones y trabajo futuro

Aunque en la Sección 2.3.4 se estudiaron una serie de ontologías de seguridad, algunas de las cuales están dirigidas específicamente al escenario de IoT, ninguna está dirigida específicamente, o puede usarse fácilmente, para la tarea de anotar aspectos funcionales de seguridad de datos, desde el punto de vista de los datos en sí. Aún más importante, ninguna cubre problemas específicos de seguridad de datos relacionados con IoT, como regulaciones, certificaciones o procedencia. Esos resultados preliminares requieren una nueva ontología que permita que los frameworks de IoT, los agregadores de datos, los motores de búsqueda, los procesadores y los mercados de datos, compartan y consuman datos, al proporcionar un vocabulario mediante el cual se puedan realizar anotaciones de seguridad de datos.

Siguiendo la metodología NeOn, se ha desarrollado, descrito y conceptualizado los principales conceptos representados en la Sección 3.1, para luego formalizarlos e implementarlos en la

```
{
  "id": "urn:ngsi-dl:AccessControl:AuthenticationProvider:authprov1",
  "type": "ds4iot:AuthenticationProvider",
  "http://www.w3.org/ns/sosa/observes": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:ObservableProperty:temperature"
  },
  "ds4iot:endPoint": {
    "type": "Property",
    "value": "https://keyrockaddress.com:31337/"
  },
  "ds4iot:refTechnology": {
    "type": "Property",
    "value": "https://github.com/ging/fiware-idm"
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
    {
      "ds4iot": "http://www.semanticweb.org/pedrog/ontologies/2019/11/ds4iot#"
    }
  ]
}
```

Listado 4: Entidad NGSI-LD representando un objeto de la clase `AuthenticationProvider`

ontología DS4IoT resultante, posteriormente descrita en la Sección 3.2.2. El resultado es una ontología OWL DL liviana, que representa conceptos actuales y novedosos en el campo de la seguridad de datos en IoT, que favorece el uso de conocimiento implícito sobre explícito, para representar los procesos específicos y los intercambios de información necesarios en los diferentes elementos de seguridad representados. Hasta donde ha sido posible investigar, esta es la primera ontología que ofrece un vocabulario específico para la anotación de aspectos de seguridad de datos para IoT.

Para mostrar mejor el potencial de la ontología DS4IoT, así como para validar empíricamente sus afirmaciones, se ofrece una prueba de concepto en la Sección 3.3 en la que se presenta el mapeo al modelo de datos NGSI-LD, así como una adaptación a la ontología IoTCrawler, que cubre algunos aspectos básicos de la seguridad de los datos que son especialmente relevantes para el proyecto IoTCrawler.

Al tratarse de un campo activamente desarrollado y relativamente reciente cabe esperar que

pronto se crearán nuevas tecnologías, estándares, frameworks y enfoques en seguridad, y esta ontología deberá revisarse y actualizarse en consecuencia para mantenerse al día y adaptarse a los cambios por venir. Especialmente sensibles serán los conceptos sobre regulaciones y certificación, que ahora apenas han comenzado a surgir en el ámbito de IoT y seguramente estarán sujetos a revisiones, actualizaciones y debate.

Adicionalmente, otros aspectos relevantes de la seguridad de datos en el campo de IoT que se han dejado atrás en este trabajo, como el ciclo de vida de los datos, podrían ser objeto de estudio para su futura inclusión en DS4IoT.

CAPÍTULO 4

ARQUITECTURA IOT PARA GESTIÓN ENERGÉTICA DEL HOGAR INTELIGENTE

Este capítulo presenta una arquitectura para un sistema de gestión energética del hogar inteligente (smart home energy management system, SHEMS) basada en IoT, capaz de integrar un sistema de automatización del hogar (home automation system, HAS) existente e interactuar con la red eléctrica inteligente (smart grid, SG) para implementar estrategias de respuesta a la demanda (demand response, DR), orientado al prosumidor. Este sistema también aborda la seguridad y privacidad de datos desde la perspectiva energética y del control domótico, de cara a compartir información que puede ser de interés a distribuidores y productores de energía a la hora de realizar estimaciones y predicciones de consumo. Esta propuesta está basada en interfaces y modelos de datos estándares, facilitando la interoperabilidad con sistemas existentes dentro y fuera del hogar, así como la integración del mismo como parte de otras macroarquitecturas.

4.1 Propuesta

La arquitectura general o conceptual de esta propuesta (Figura 4.1) está compuesta por múltiples componentes de gestión de energía (energy management components, EMC) responsables de un dominio específico dentro de la gestión energética del hogar, que interactúan entre sí para perseguir sus objetivos, cooperando hacia un objetivo común de aumentar la eficiencia energética y reducir los costes energéticos del hogar.

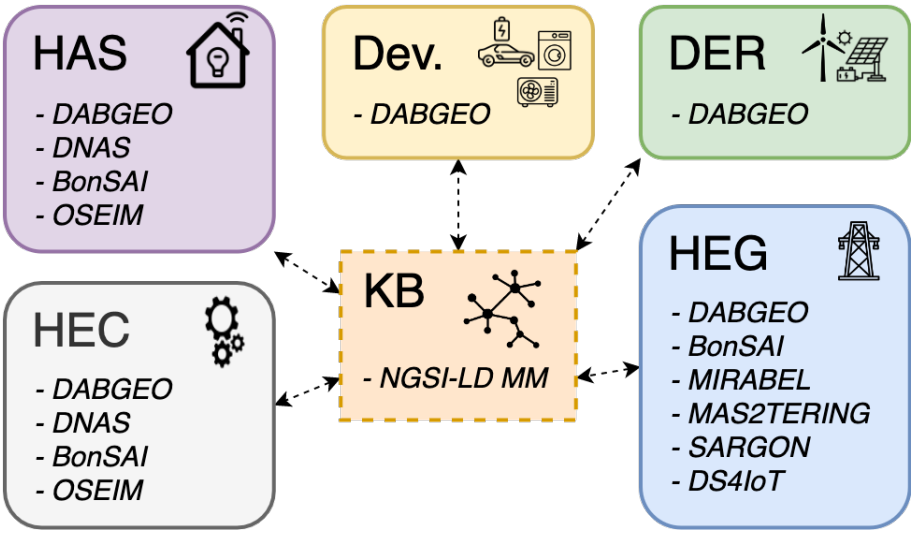


Figura 4.1: Arquitectura basada en el conocimiento del SHEMS. En esta imagen se muestran, además, las diferentes ontologías relacionadas con los diferentes EMC de la arquitectura.

4.1.1 Componentes de administración de energía

La siguiente lista describe los diferentes EMC que rigen la gestión de energía del hogar, así como algunas de las relaciones entre ellos:

1. *Sistema de automatización del hogar (home automation system, HAS)*: componente encargado de la comunicación de ida y vuelta con el sistema de automatización del hogar; lo mantiene actualizado respecto a los presupuestos de energía, estado de producción y acumulación y consumo actual. La información de alto nivel producida por otros componentes puede ser utilizada por el HAS para proporcionar a los ocupantes de la casa herramientas de apoyo para la toma de decisiones y notificaciones relacionadas con la energía. También actualiza la base de conocimiento (knowledge base, KB) con información del HAS, como inventario de dispositivos, estado y programación de dispositivos, información de presencia y ocupación, así como información procedente de fuentes externas; como información meteorológica en tiempo real y pronósticos. Esta información puede ser utilizada para construir y alimentar diferentes DER y modelos de pronóstico de consumo de energía y de esta forma permitir que los HEMS planifiquen y se adapten a las condiciones cambiantes.

2. *Recurso energético distribuido (distributed energy resource, DER)*: se encarga de tratar los recursos energéticos, tanto para la producción como para la acumulación de energía. Su propósito es: (a) producir información de alto nivel, como pronósticos de producción y almacenamiento requeridos por el componente controlador energético doméstico (home energy controller, HEC) y (b) operar los diferentes DER de manera segura, tratando de minimizar los costos de energía para el hogar y (c) cooperar con el HEC para la implementación de estrategias de DR.

3. *Pasarela energética doméstica (home energy gateway, HEG)*: actualiza el estado del enlace a la red, en función de si estamos inyectando o consumiendo energía de la red, así como información de precios y parámetros contractuales (por ejemplo, potencia máxima utilizable). También puede transmitir información relativa al consumo y producción de energía a la empresa de comercialización y/o distribución de energía; como horarios y pronósticos de uso de energía, inventario de electrodomésticos y patrones de uso y, en general, cualquier información que el propietario esté dispuesto a compartir que pueda ayudar a la empresa de servicios públicos a realizar una mejor oferta al cliente. También es responsable de comunicar las estrategias de DR, como flexibilidades, o actuar como un intermediario para el brokering en gestión paquetizada de la energía (packetized energy management, PEM) con la red (o microrredes) y realizar un seguimiento de los costos totales de energía.

4. *Dispositivo (device)*: representa dispositivos cuya gestión de energía puede ser tratada directamente por el componente HEC. Ejemplos de tales dispositivos serían grandes consumidores, electrodomésticos de funcionamiento continuo como HVAC u otros sistemas de elevado consumo como bombas de piscina, calentadores de agua y vehículos eléctricos; cuyo horario de consumo de energía puede afectar en gran medida a la gestión energética del hogar y, por lo general, pueden ser reprogramados o su consumo puede ser reducido bajo ciertas condiciones, con un impacto mínimo o nulo para los ocupantes de la casa. Por lo general, estos dispositivos serán los más críticos durante la aplicación de estrategias de DR, como la gestión de la flexibilidad.

5. *Controlador energético doméstico (home energy controller, HEC)*: encargado del control y optimización. Puede hacer uso de diferentes estrategias y técnicas, desde el razonamiento semántico hasta el uso de métodos tradicionales de optimización o inteligencia artificial. Su propósito es tratar y mediar entre los componentes, asegurando que el presupuesto de energía esté optimizado y facilitando la compartición de información útil de alto nivel con otros componentes, como el HAS y el HEG.

Vale la pena mencionar en este punto que, en su implementación, los EMC no tienen por qué ser instancias únicas o individuales. Se puede (y debe) diseñar diferentes componentes de dispositivo para una lavadora y un cargador de vehículos eléctricos. De igual manera, los componentes no tienen por qué implementarse en una sola unidad final; por ejemplo, el HEC podría dividirse en diferentes módulos de software a cargo de diferentes áreas de responsabilidad, por ejemplo: la intermediación energética, la previsión del consumo total de energía y la decisión de diferentes estrategias para DR.

4.1.2 Base de conocimiento

Como se muestra en la Figura 4.1, proponemos una arquitectura centrada en una base de conocimiento (knowledge base, KB). Los componentes son capaces de realizar sus objetivos, así

como de interactuar y cooperar entre sír mediante el uso de dicha KB, donde se almacena toda la información del sistema.

La información almacenada en la KB incluye metainformación sobre diferentes aspectos, como tipado de datos y enlaces ontológicos a los conceptos representados. Esta información "enriquecida" se llama *contexto* en nuestro sistema.

La estructura y las raíces semánticas del contexto en esta propuesta, así como los mecanismos para interactuar con él, se describen con más detalle en la Sección 4.1.5. Los componentes responsables de su gestión son presentados y descritos en la Sección 4.1.3. Finalmente, la KB también se usa como un medio para la coordinación y orquestación entre EMC en el sistema, aprovechando su naturaleza de publicación/suscripción y su modelo de datos flexible. Esto se describe con más detalle en la Sección 4.1.7.

4.1.3 Arquitectura funcional del SHEMS

La arquitectura funcional del HEMS propuesto consta de tres capas más una capa transversal de seguridad, en las que se organizan los componentes de esta arquitectura (Figura 4.2). A continuación, se enumeran las distintas capas de la arquitectura, así como su función y sus interacciones con las restantes capas:

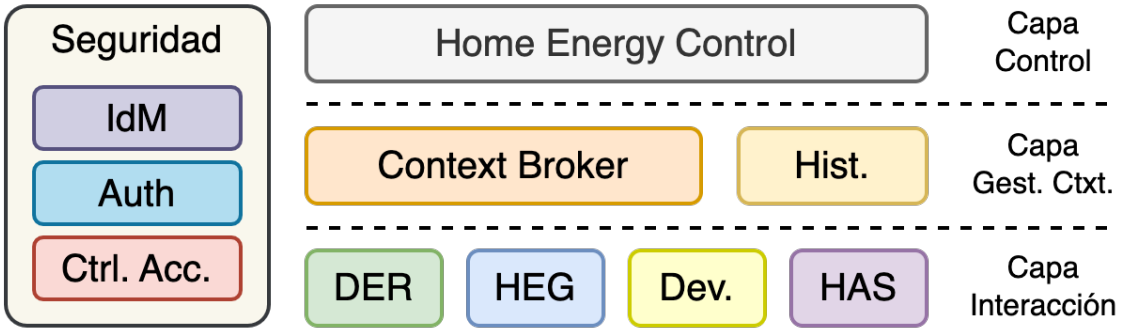


Figura 4.2: Arquitectura por capas del SHEMS

1. *Capa de control*: es el núcleo del sistema de gestión, donde conviven todos los módulos relacionados con el componente HEC. Esta capa es responsable de la estrategia y la programación energética del sistema. Intentará alcanzar los objetivos globales del sistema, trabajando con la información proporcionada por los restantes componentes. Todos los módulos de esta capa usarán el broker de contexto (context broker, CB), ubicado en la capa de gestión de contexto (2), tanto para acumular datos de la capa de interacción (3), como para enviar directivas de programación y parámetros de operación a otros componentes. En la Sección 4.1.7 se describe en más detalle la orquestación del sistema, mediante la cual se controla el resto de EMC.

2. *Capa de gestión del contexto*: esta capa actúa como la columna vertebral de la información del sistema, proporcionando varios mecanismos y características que permiten la interoperabilidad y modularidad del sistema. La principal característica de esta capa es que se basa en el estándar NGSI-LD [22] del ETSI; utilizado como estándar de comunicaciones en el marco FIWARE, previamente descrito en la Sección 2.1.1 del estado del arte. El CB es el componente responsable de hacer accesible toda la información (contexto) de la KB, así como el proveedor de servicios de búsqueda, acceso y actualización del contexto. Sus características específicas y detalles sobre la comunicación entre componentes son descritas más abajo en la Sección 4.1.5 y el fundamento ontológico y la estructura del modelo de información en la Sección 4.1.4. Esta capa también contiene el componente histórico, responsable de almacenar datos históricos necesarios (utilizados, por ejemplo, para elaboración de pronósticos y análisis de datos) y ponerla a disposición del EMC. Tanto el CB como el componente histórico se pueden instanciar a partir de las muchas implementaciones de FIWARE Generic Enablers existentes, lo que promueve la reutilización y la flexibilidad a la hora de la elección de componentes específicos para distintas instanciaciones de la arquitectura, manteniendo la interoperabilidad.

3. *Capa de interacción*: esta capa comprende todos los componentes encargados de interactuar con los dispositivos y entidades relacionadas con el HEMS. Es la capa que hace de frontera entre el SHEMS, el resto del SH y el mundo, actuando como adaptador entre las interfaces internas NGSI-LD y las diferentes interfaces externas. Es a través de esta capa que la información de los dispositivos, la red eléctrica, los DER y el HAS llegan a la KB. Es también en esta capa donde la gestión del sistema cristaliza en comandos específicos enviados a dispositivos, o comunicaciones con servicios externos. Finalmente, en esta capa se realizará toda la adaptación semántica entre el SHEMS y los dispositivos, servicios y agentes externos.

4. *Seguridad y Privacidad*: por último, esta capa es transversal a todo el sistema. Es responsable de garantizar el acceso seguro a los datos, así como asegurar la privacidad. Esto lo consigue haciendo de mediadora en el intercambio de información entre los componentes y la KB, proporcionando componentes de autenticación y autorización para el control de acceso al CB y otros componentes en la arquitectura. Los componentes que forman la capa de seguridad y privacidad y su funcionamiento se describen con más detalle en la Sección 4.1.6.

4.1.4 Modelo de Información

Las tecnologías de la web semántica presentan una solución conveniente para asegurar la interoperabilidad entre vendedores y proveedores de diferentes servicios, plataformas y dispositivos. Bajo la web semántica, la información tiene que ser anotada semánticamente (enlazada) a con-

ceptos ontológicos. La información enriquecida con etiquetas semánticas se denomina contexto en NGSI-LD, la tecnología subyacente en la que se basa la KB de esta propuesta.

De acuerdo al Modelo de Información de NGSI-LD [22], el contexto se estructura en forma de entidades. Estas entidades tienen identidad, tipo y propiedades. Además, pueden vincularse entre sí a través de relaciones. Las entidades se intercambian en forma de documentos JSON-LD que siguen un *Core Metamodel* (metamodelo núcleo o central). Junto con esas propiedades y relaciones, NGSI-LD introduce el uso de atributos JSON-LD especiales (representados por la *Ontología Cross-Domain*) que se vinculan a conceptos semánticos de otras *Ontologías de dominio específico*, creando un "modelo cebolla", por capas, representado en la Figura 4.3.

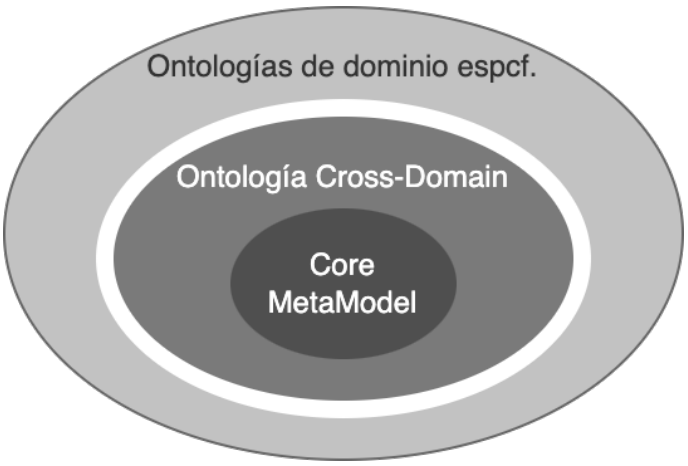


Figura 4.3: Modelo de información de NGSI-LD

Una amplia gama de ontologías existentes, estrechamente relacionadas con el dominio de los SHEMS, ya están disponibles para su uso en esta propuesta. Algunas de estas ontologías están estrechamente relacionadas con el dominio de SHEMS, cubriendo conceptos de DR y SG. Los trabajos más relevantes en esta área han sido tratados en el estado del arte (Sección 2.3.3) y se están resumidos en la Tabla 2.2.

En esta propuesta, hemos seleccionado DABGEO como la ontología principal, ya que cubre los conceptos más relevantes, relacionados con la gestión de la energía del hogar: descripción de electrodomésticos, dispositivos y consumo de energía, información relacionada con la red (desde tarifas hasta conceptos relacionados con el prosumidor), generación y acumulación de energía eléctrica, así como información relacionada con el usuario (preferencias del usuario, ocupación y otros conceptos relacionados). En la Sección 4.2.3 se proporcionan ejemplos específicos del mapeo de conceptos DABGEO a NGSI-LD.

Como ya se indicó en el estado del arte, ninguna ontología cubre todos los aspectos y escenarios posibles en el dominio de los SHEMS. La Tabla 4.1 representa el subconjunto de ontologías que pueden complementar a DABGEO en esta propuesta y los EMC para quienes pueden ser relevantes (por ejemplo, MAS2TERING podría aplicarse en el HEG aplicado al ámbito de DR, en escenarios donde se implemente USEF).

Tabla 4.1: Ontologías complementarias para SHEMS

Ontología	**HEC**	**HEG**	**HAS**
DNAS [146]	X		X
BonSAI [142]	X	X	X
MIRABEL [144]		X	
MAS2TERING [53]		X	
SARGON [158]		X	
OSEIM [159], [160]	X		X

Finalmente, hasta donde sabemos, no se ha propuesto previamente una adaptación de NGSI-LD a la ontología seleccionada. El enfoque que hemos seguido en este trabajo ha sido relacionar elementos OWL de DABGEO a NGSI-LD, de acuerdo con la Tabla 4.2. En el caso de las propiedades de objeto de OWL vinculadas a individuos, la representación como propiedades anidadas NGSI-LD puede considerarse si se cumplen los siguientes criterios: (1) los individuos vinculados están relacionados exclusivamente con una sola entidad, (2) su existencia depende de ello, (3) tienen un bajo número de propiedades y/o relaciones y (4) no serán utilizadas como claves de búsqueda en la KB.

Tabla 4.2: Correspondencia OWL a NGSI-LD

OWL	**NGSI-LD**
Individual	Entity
Class	Entity type
Object property	Relationship ó nested property
Datatype property	Property

Por último, se propone el uso de la ontología DS4IoT [4], para representación de conceptos de seguridad en IoT (presentada en el Capítulo 3), en el HEG, ya que este componente hace de pasarela de información entre el SHEMS y el resto del mundo, por lo que su aplicación facilita la distribución de la información relativa a seguridad y privacidad de los datos hacia otras plataformas IoT. La adaptación seguida en este caso, de la ontología DS4IoT a NGSI-LD, sigue el mismo patrón visto anteriormente.

4.1.5 Gestión del contexto

En NGSI-LD el contexto, representado por entidades, puede ser creado, actualizado, consultado y eliminado a través de una API basada en REST, intercambiando documentos JSON-LD[1]. Esta API también proporciona mecanismos de subscripción, que enviarán notificaciones en base a cambios realizados en el contexto (la información almacenada), formando un sistema de gestión de información de publicación/suscripción. El CB es el único punto central de la arquitectura donde se puede acceder a toda la información disponible en el sistema.

El CB ofrece mecanismos avanzados para buscar información de contexto disponible en el sistema, ofreciendo diferentes filtros y mecanismos de consulta para recuperar información. Algunos de esos filtros también se pueden usar con el mecanismo de suscripción, lo que permite que los componentes reciban actualizaciones sobre conjuntos personalizados de datos de su interés.

Finalmente, el CB también puede actuar como repetidor y directorio para otros proveedores de información previamente registrados: de esta manera, diferentes EMC pueden actuar como proveedores de contexto (context providers, CP), responsables de subconjuntos de información de la KB, capaces de responder consultas del CB y otros componentes. Este mecanismo es relevante en los casos en que la información se calcula bajo pedido o en los casos en que la información cambia constantemente en los dispositivos, pero es consultada con poca frecuencia, lo que evita actualizaciones constantes al CB.

La Figura 4.4 muestra dos casos de consulta de información, uno directo al CB y otro en el que el CB hace de intermediario, retransmitiendo la consulta al CP.

4.1.6 Seguridad y privacidad

Para proporcionar un acceso seguro y privado a la información de contexto, esta propuesta introduce el uso de DCapBAC [108], una derivación del esquema ABAC en el que la comprobación de la autorización en base a las políticas de seguridad y la aplicación de la política están desacoplados (Figura 4.5).

En la siguiente lista se describen los componentes de seguridad de la arquitectura:

1. *Gestión de la identidad (identity management, IdM)*: proporciona el servicio de autenticación a la arquitectura. Almacena información de identidad junto con atributos que luego son utilizados por el componente de autorización. En nuestra arquitectura, proponemos el uso de interfaces estándar, como OAuth 2[2] y OpenID Connect[3].

[1] "JSON for Linking Data, Data is messy and disconnected. JSON-LD organizes and connects it, creating a better Web." Sitio web de JSON-LD, World Wide Web Consortium (W3C). (), dirección: `https://JSON-LD.org` (visitado 21-07-2023).

[2] "OAuth 2.0". Sitio web de OAuth 2.0. (), dirección: `https://oauth.net/2/` (visitado 21-07-2023).

[3] "How OpenID Connect Works". Sitio web de OpenD Connect, OpenID Foundation. (), dirección: `https://openid.net/developers/how-connect-works/` (visitado 21-07-2023).

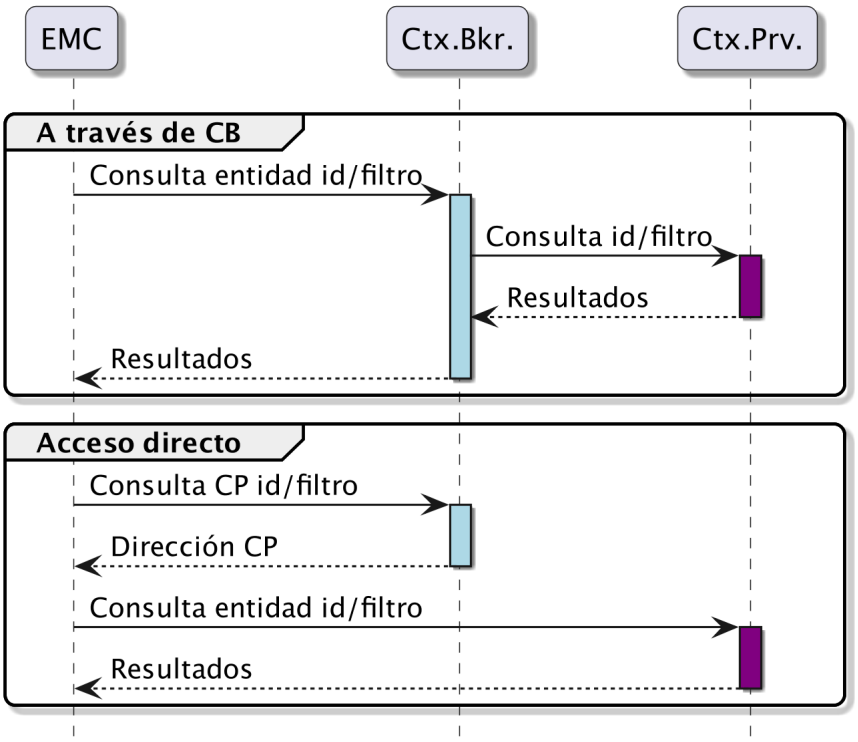


Figura 4.4: Secuencia de acceso de los EMC a la información del CB

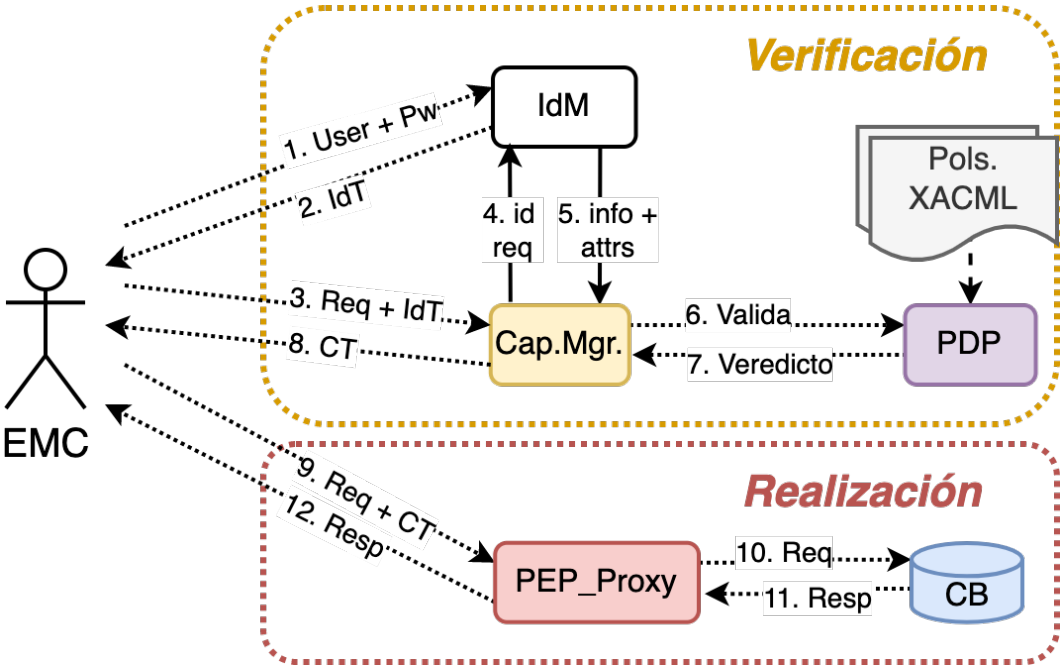


Figura 4.5: Secuencia de autorización y acceso de los EMC a la información en el CB

2. *Gestor de capacidades (capability manager, CM)*: actúa como la fachada de autorización para los componentes, otorgando o denegando el acceso al recurso solicitado por medio de tokens. Recibe solicitudes de componentes e interactúa con el IdM y el PDP para realizar su tarea.

3. *Punto de decisión de políticas (policy decision point, PDP)*: valida las solicitudes de acceso a un recurso, usando información suministrada sobre la identidad y el conjunto de políticas de seguridad del sistema. Esas políticas se administran a través del siguiente componente de la lista: el PAP.

4. *Punto de administración de políticas (policy administration point, PAP)*: ofrece un punto singular de administración y gestión para las políticas del sistema. Estas se almacenan en XACML y describen las condiciones que un sujeto debe cumplir para actuar sobre un recurso de una forma determinada. En este caso, las condiciones pueden basarse en atributos de la identidad almacenada en el IdM, las acciones realizadas sobre recursos son verbos HTTP y los recursos son recursos HTTP, representados mediante URL.

5. *PEP-Proxy*: actúa como un proxy HTTPS inverso transparente, que hace cumplir el veredicto emitido por el PDP, sin necesidad de una evaluación adicional de las políticas del sistema.

El proceso por el cual los EMC acceden a la información comienza con el proceso de autenticación. La Figura 4.6 representa una versión simplificada de una interacción típica de OpenID para la autenticación de un EMC contra el IdM. Como resultado de esta interacción, el componente obtiene un token de identidad (identity token, IdT) que se utilizará en la fase de autorización.

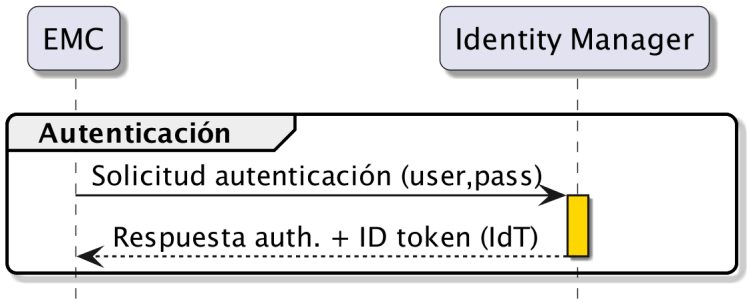


Figura 4.6: Secuencia de autenticación de los EMC en el IdM

La autorización en DCapBAC comienza con una solicitud para obtener acceso a un recurso, acompañada del IdT obtenido previamente. Esta solicitud sigue la estructura e interfaz de XACML clásicas (Figura 4.7), en la que la solicitud del EMC se compara en el PDP con las políticas XACML para verificar si la solicitud puede ser concedido o no. La diferencia es que, en lugar de acceder inmediatamente al recurso después de un veredicto positivo, esta interacción resultará en la emisión de un token de capacidad (capability token, CT).

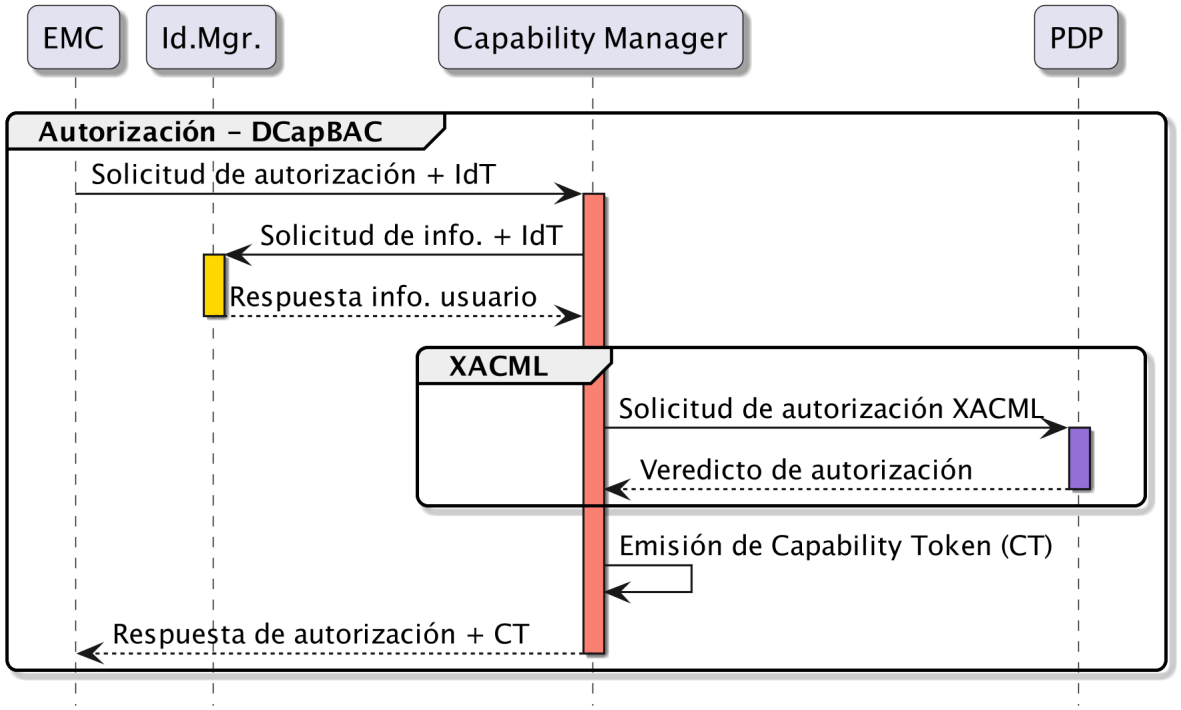


Figura 4.7: Secuencia de autorización de las peticiones de los EMC

La fase final (Figura 4.8), es el acceso real a la información de contexto. En este caso, el CB está protegido por un PEP-Proxy transparente. Este componente solo permitirá el paso de solicitudes que sean válidas de acuerdo con el CT adjunto. El acceso puede tener lugar varias veces con el mismo CT mientras este sea válido, lo que reduce de manera muy efectiva el retraso inducido por la validación de cada solicitud, ya que no es necesario volver a realizar la comprobación contra las políticas de seguridad del sistema.

Las comunicaciones entre los EMC, los componentes de seguridad y el CB se realizan a través de llamadas a una API REST y están protegidas con tecnologías web estándar, utilizando el protocolo HTTPS.

4.1.7 Orquestación del sistema

los EMC no solo comparten información a través de la KB, sino que también se comunican entre sí de forma asincrónica mediante el paso de mensajes, utilizando la funcionalidad de publicación/suscripción definida en NGSI-LD. El sistema implementa un mecanismo inspirado en la orquestación de tareas de FogFlow[4], en el que las tareas reciben información de otras tareas y del orquestador, mediante el uso de entidades intermediarias en la KB. Esas entidades contienen información de entrada/salida para sus tareas.

[4]"Fogflow 3.2.8 documentation". Sitio web de documentación de FogFlow, NEC. (2022), dirección: `https://fogflow.readthedocs.io/en/latest/` (visitado 21-07-2023).

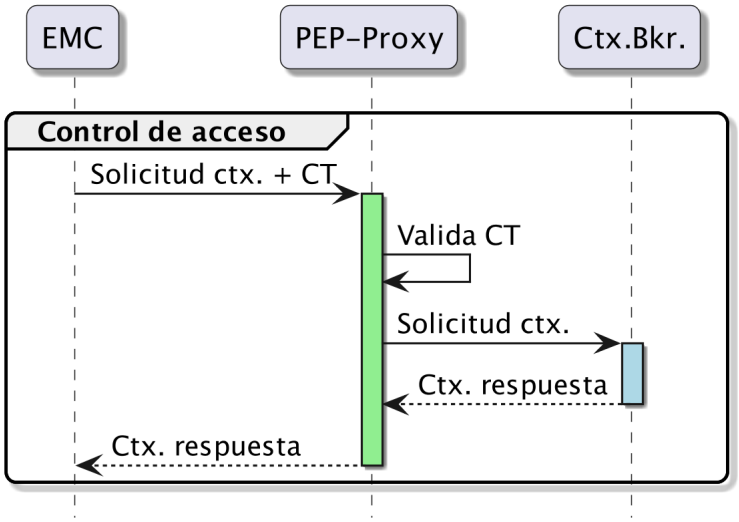


Figura 4.8: Secuencia de aplicación de la autorización de peticiones de los EMC

En esta propuesta, los EMC se suscriben a entidades específicas por medio de las cuales el HEC envía y actualiza los comandos de gestión, programación o resultados deseados que los EMC requieren como entrada para su funcionamiento (Figura 4.9). Ese mismo mecanismo es usado por el HEC para recibir salida del resto de EMC.

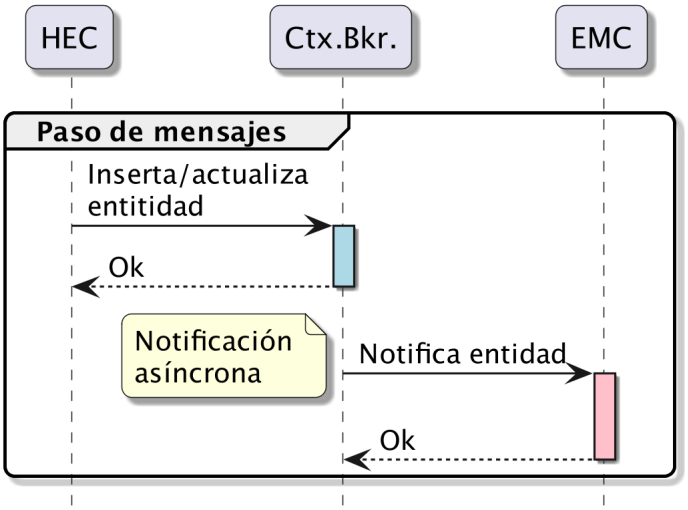


Figura 4.9: Paso de mensajes de orquestación asíncrono entre HEC y EMC

Este mecanismo de orquestación también está asegurado mediante los componentes de seguridad y privacidad propuestos, cubiertos en la Sección 4.1.6. De esta forma, se pueden implementar políticas XACML para garantizar que solo el EMC esperado pueda leer sus entidades de mensajes de entrada y actualizar las de salida.

4.2 Validación

Para validar la arquitectura propuesta, se conduce un caso de uso para el SHEMS de una vivienda unifamiliar en planta baja. Esta vivienda cuenta con una instalación de HAS ya existente, instalaciones de producción y almacenamiento de energía eléctrica y algunos electrodomésticos con elevado consumo energético.

Con respecto a los algoritmos y métodos específicos utilizados en los EMC implementados, se omite deliberadamente detalles de implementación. Las dos razones principales detrás de esta decisión son: (a) la implementación de algunos de los EMC aún está en desarrollo, y sin duda evolucionará en un futuro cercano a medida que se prueban nuevos enfoques y algoritmos y (b) la preocupación es la arquitectura mediante la cual las diferentes implementaciones de EMC pueden cooperar de manera interoperable y segura, por lo que describir completamente los algoritmos y las optimizaciones solo conduciría a una posible confusión por parte del lector y a un trabajo demasiado extenso.

No obstante, en esta sección se ofrecen algunos detalles de implementación de los EMC, así como resultados específicos que respaldan la meta propuesta de mejora de la eficiencia energética en el escenario, junto con los objetivos específicos que guían el SHEMS y los componentes de la arquitectura central seleccionados para la demostración, así como ejemplos de comunicación entre EMC.

4.2.1 Descripción del caso de prueba

El banco de pruebas (Tabla 4.3) consiste en una casa de dos plantas sobre rasante, ubicada en Murcia, en el sureste de España, con patio y garaje subterráneo. En el patio se encuentra una pequeña piscina, cuyos sistemas de filtración y cloración pueden ser controlados por el SHEMS.

El HAS existente se basa en el software Home Assistant (Figura 4.10), se ejecuta en una Raspberry Pi 4 y es capaz de controlar los sistemas de HVAC, calefacción central por suelo radiante, persianas y algunas luces, televisores inteligentes y la secadora entre otros electrodomésticos. Se integran diferentes sensores en el HAS, que monitorizan la temperatura y la humedad en diferentes habitaciones. También proporciona información meteorológica a través de servicios web externos, así como el estado de ocupación de la vivienda mediante la detección de presencia de diferentes habitantes en la vivienda.

En la terraza, una serie de paneles fotovoltaicos (photo-voltaics, PV), conectados a un inversor, proporcionan hasta 6 kW de energía eléctrica. Actualmente se encuentra en desarrollo un sistema de acumulación, con 28 kWh de capacidad de almacenamiento planificada, protegido por un sistema de gestión de baterías y controlado directamente por el inversor. La Figura 4.11 muestra una vista aérea de la localización del banco de pruebas, donde se aprecia la piscina, paneles PV y las unidades externas de HVAC y calefacción central.

Los sistemas PV y de almacenamiento proporcionan información en tiempo real sobre produc-

Tabla 4.3: Resumen del banco de prueba de gestión energética en el hogar inteligente

Elemento	**Descripción**
Sistema de automatización del hogar	Home Assistant, desplegado sobre una Raspberry Pi 4B, con 4 GB de RAM
Array fotovoltaico	14 paneles de 480 W de potencia nominal
Inversor fotovoltaico	Ingecon Sun Storage 1Play
Batería	32 celdas en serie, química LiFePo4, con capacidad nominal de 280Ah
Sistema de gestión de la batería	Batrium WatchMon-CORE, 2 Batrium CellMate K9 y ShuntMon de 500A
Medidor de consumo de red	Carlo Gavazzi EM112
Tarifa de red	5.4 kW de consumo pico, con tarifa plana. Inyección de excedentes a red posible, con rebaja de tarifa plana.
Controlador de electrodoméstico *	Enchufe inteligente TP-Link HS110 (con medidor de consumo incorporado), controlado por Home Assistant
Controlador de piscina	Basado en Node-RED, desplegado en una Raspberry Pi Zero. Medidor de consumo integrado via Modbus.

* Una cafetera expreso (ECM Synchronika), con consumo pico de 1.6 kW, está conectada al enchufe inteligente.

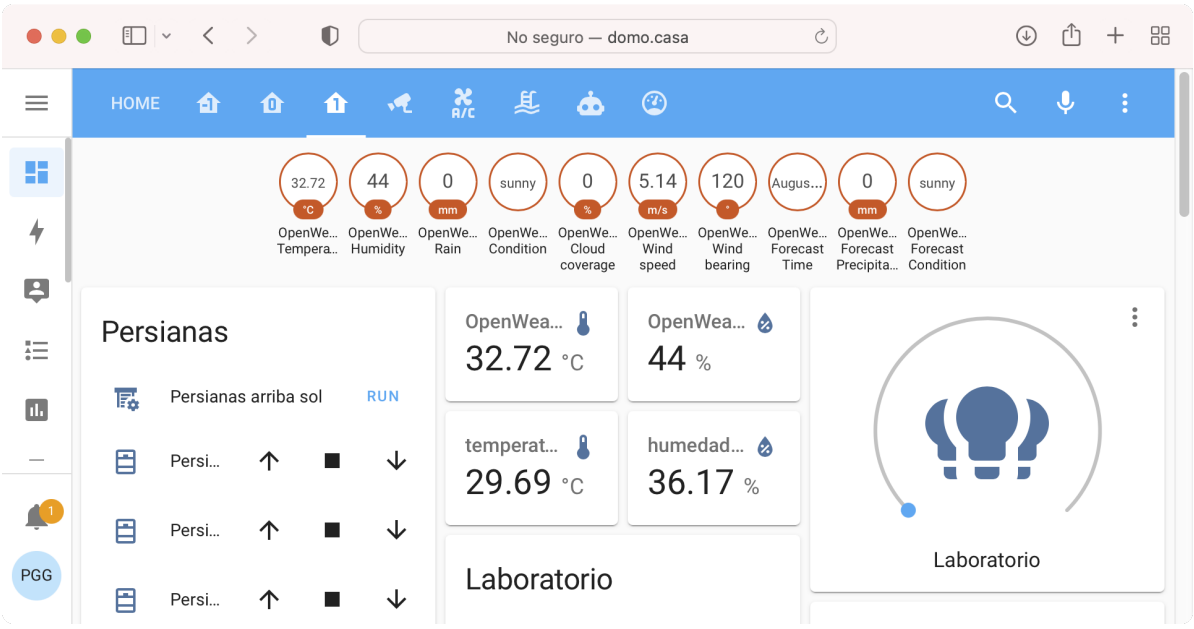


Figura 4.10: Interfaz gráfica de la instalación de Home Assistant en el banco de pruebas

Figura 4.11: Vista aérea del banco de pruebas

ción, consumo y estado de carga a través de sus controladores. Además, algunos electrodomésticos de alto consumo tienen medidores de potencia dedicados para un desglose más granular de la composición del consumo de energía; tal es el caso del sistema de filtración de la piscina y la máquina de café expreso.

Finalmente, un enlace a la red proporciona energía. El contrato con la compañía eléctrica establece una tarifa a precio constante con una potencia pico de 5,4 kW. También permite la inyección a red desde la instalación PV, con una bonificación proporcional a la energía inyectada. Para monitorizar la energía de importación/exportación, también se instaló e integró un medidor de energía conectado a la red.

4.2.2 Tareas del SHEMS

El objetivo general del banco de pruebas para el SHEMS es lograr la operación más eficiente y rentable del sistema y para hacerlo se proponen las siguientes tareas:

1. Administrar los parámetros de acumulación de la batería (estado máximo de carga y profundidad de descarga, regímenes de carga y descarga y similares), programar y administrar la carga de la batería y administrar el uso de energía almacenada.

2. Programar y gestionar el sistema de calefacción central, así como el sistema de HVAC.

3. Programar y gestionar el sistema de filtración y cloración de la piscina.

4. Reaccionar a las acciones de los habitantes que afectan y desequilibran los perfiles energéticos esperados.

5. Informar al usuario sobre los patrones de consumo de energía y el estado de la gestión de energía, generar alertas y brindar sugerencias relacionadas con la energía para mejorar la eficiencia y reducir el consumo.

Para cumplir con sus tareas, el SHEMS utilizará y generará información de y para sus diferentes EMC; como los parámetros de conexión a la red (potencia máxima utilizable, precios de energía actuales y planificados y descuentos por inyección a la red según la tarifa), consumo de energía actual y previsto, producción y almacenamiento, horarios de funcionamiento de diferentes dispositivos e incluso información de ocupación y climatológica.

4.2.3 Base de conocimiento y componentes de seguridad

Para la KB y los componentes de seguridad, se ha seleccionado algunos componentes (*Generic Enablers*[5]) del proyecto FIWARE. La KB se implementa sobre un CB *Orion-LD*[6]. Como componente de IdM se está utilizando *Keyrock Identity Management Generic Enabler*[7]. Para los componentes de DCapBAC, se ha seleccionado la implementación de código abierto de PAP-PDP, PEP-Proxy y CM proporcionada por el proyecto IoTCrawler[8,9].

La información en la KB es estructurada y almacenada en forma de entidades (Listado 5), siguiendo el *Core MetaModel* de NGSI-LD, y vinculada a la ontología DABGEO [156] (Figura 4.12).

La orquestación se realiza a través del paso de mensajes asíncronos y requiere la suscripción de los EMC a las entidades de entrada de su dominio. Un ejemplo de ello es el control del sistema de filtración y cloración de la piscina (Listado 6). En caso de exceder las restricciones de potencia del sistema (por ejemplo, si la energía importada de la red está cerca o excede la potencia máxima definida por el contrato de suministro de energía), el HEC emitirá una modificación a la entidad que representa el *controllerDesiredStatus* del sistema de filtración y cloración, solicitando al controlador del dispositivo que detenga el sistema como resultado de restricciones de energía. Esa modificación activará la suscripción adecuada en el CB, resultando en el envío de una

[5]"FIWARE Catalogue". Sitio web del catálogo de componentes FIWARE, FIWARE Foundation. (2022), dirección: `https://www.fiware.org/catalogue/` (visitado 06-03-2023).

[6]"Orion Context Broker (with Linked Data Extensions)". Repositorio de código del proyecto Orion-LD. (), dirección: `https://github.com/FIWARE/context.Orion-LD` (visitado 22-07-2023).

[7]"Identity Manager - Keyrock". Sitio web de documentación sobre Keyrock IdM, FIWARE. (), dirección: `https://fiware-idm.readthedocs.io/en/latest/` (visitado 22-07-2023).

[8]"IoTCrawler – a search engine for Internet of Things devices". Sitio web del proyecto IoTCrawler. (), dirección: `http://iotcrawler.eu` (visitado 21-07-2023).

[9]"EU H2020 IoTCrawler Project- Open Source Tools and Components". Repositorio de código del proyecto IoTCrawler, IoTCrawler. (), dirección: `https://github.com/orgs/IoTCrawler/repositories` (visitado 22-07-2023).

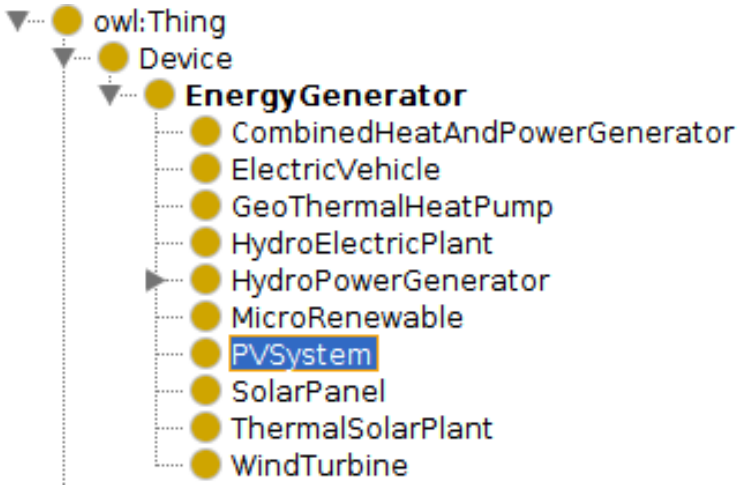


Figura 4.12: Vista en Protegé de un PVSystem (sistema fotovoltáico) en DABGEO

```
{
  "id": "urn:ngsi-ld:PVSystem:Device:0042",
  "type": "PVSystem",
  "deviceName": {
    "type": "Property",
    "value": "INGECON SUN STORAGE 1Play TL M"
  },
  "maxProducesEnergy": {
    "type": "Property",
    "value": 6,
    "unitCode": "KW"
  },
  "@context": [
    {
      "deviceName": "http://www.purl.org/oema/enaeq/deviceName",
      "maxProducesEnergy": "https://www.auto.tuwien.ac.at/downloads/thinkhome
      ↪ /ontology/EnergyResourceOntology.owl#"
    },
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
  ]
}
```

Listado 5: Entidad NGSI-LD de un inversor fotovoltaico

notificación al EMC de dispositivo de la piscina (Listado 7) y detendrá la operación hasta que el HEC restablezca el estado.

```
{
  "id": "urn:ngsi-ld:Subscription:PoolController:0001",
  "description": "Pool system dev.ctr. subscription",
  "type": "Subscription",
  "entities": [{
    "type": "PoolController",
    "id": "urn:ngsi-ld:PoolController:ID:0001"
    }],
  "watchedAttributes": ["controllerDesiredStatus"],
  "notification": {
    "attributes": ["controllerDesiredStatus"],
    "format": "normalized",
    "endpoint": {
      "uri": "http://pool:1880/notifications",
      "accept": "application/json"
    }
  }
}
```

Listado 6: Subscripción en NGSI-LD

La seguridad en el sistema comienza con la definición de diferentes identidades para los diferentes EMC, que se utilizarán para interactuar con los componentes de autorización de DCapBAC. Se establece un conjunto de políticas XACML que rigen qué EMC puede actuar sobre las diferentes entidades almacenadas en la KB. En NGSI-LD, los diferentes verbos HTTP se asignan a las funcionalidades de creación, recuperación, actualización y eliminación, lo que ofrece una buena granularidad sobre las diferentes acciones que se pueden realizar sobre las entidades. Las entidades afectadas por la política se pueden definir en términos de identificador (literal o patrón) y diferentes filtros de consulta según el tipo de entidad u otras propiedades.

Cuando un EMC quiere interactuar con información en la KB, primero necesita autenticarse con el IdM, obteniendo un IdT (Figura 4.13). Luego obtiene un CT del CM que le habilita para una determinada acción sobre determinada información alojada en la KB. Finalmente, realizará esa interacción contra el PEP-Proxy, adjuntando el CT en la solicitud, quien permitirá (o denegará) el acceso. El proceso de seguridad se explica en detalle en la Sección 4.1.6.

El beneficio adicional de DCapBAC es que las solicitudes posteriores no necesitarán repetir las fases de autenticación y autorización, lo que añadido al hecho de que el control de acceso

```
{
  "subscriptionId": "urn:ngsi-ld:Subscription:PoolController:0001",
  "data": [{
    "id": "urn:ngsi-ld:PoolController:ID:0001",
    "type": "PoolController",
    "controllerDesiredStatus": {
      "type": "Property",
      "value": "OFF",
        "observedAt": "2022-06-10T10:11:57.000Z"
      }
  }]
}
```

Listado 7: Notificación en NGSI-LD

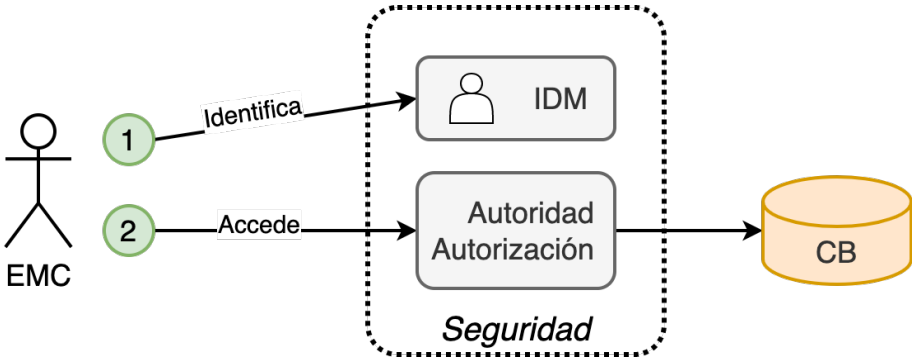


Figura 4.13: Acceso seguro de EMC al CB

realizado por PEP-Proxy no necesita interactuar con PDP, reduce considerablemente la latencia de la ejecución de la seguridad.

4.2.4 Componentes de gestión energética

Los componentes de gestión de energía (energy management components, EMC) de nuestro banco de pruebas (Figura 4.14) se encuentran en diferentes etapas de desarrollo. Se ha utilizado Node-RED[10] como la herramienta de desarrollo preferida, por su rapidez y simplicidad a la hora de generar y evolucionar soluciones para el control, además de su orientación al trabajo con flujos de datos. La enumeración que sigue a este texto resume los detalles y el estado de implementación de cada uno de los componentes:

[10]"Node-RED, Low-code programming for event-driven applications". Sitio web de Node-RED, OpenJS Foundation. (), dirección: `https://nodered.org` (visitado 22-07-2023).

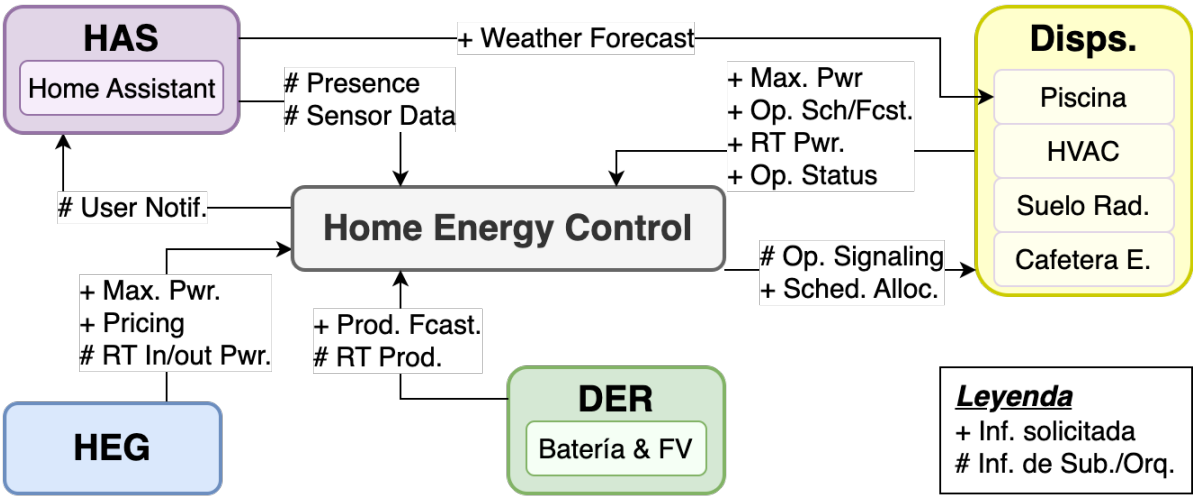


Figura 4.14: Componentes e interacciones de los EMC del banco de pruebas

1. *HAS*: se comunica con la instancia de Home Assistant del banco de pruebas. Extrae información de él relativa a pronósticos meteorológicos, datos de sensores y ocupación del hogar y la actualiza en la KB para ser usada por otros EMC. También recibe mensajes del HEC, para notificar a los usuarios sobre diferentes situaciones. La interacción entre el componente HAS y Home Assistant se lleva a cabo a través de una API REST, aprovechando los mecanismos de notificación al usuario de Home Assistant (Figura 4.16).

2. *Dispositivos HVAC y de calefacción*: estos dispositivos ya estaban integrados en Home Assistant, utilizando microcontroladores ESP8266 personalizados y la integración de ESPHome[11] en Home Assistant. Estos dispositivos ofrecen otra API REST con la que interactuar. Se ha aprovechado esta interfaz por su sencillez y para evitar utilizar Home Assistant como intermediario. Actualmente, sus componentes solo envían actualizaciones de estado a la KB y son capaces de detener temporalmente la operación, a petición del HEC (a través del mecanismo de orquestación) para reducir el consumo eléctrico. En el futuro, se podría implementar la programación de la operación del sistema de calefacción e integrarse en la estrategia de gestión de la energía. También sería posible aplicar modelos de predicción de operación del HVAC, para así obtener los patrones de uso de los habitantes.

3. *Dispositivo de cafetera expreso*: este dispositivo se controla a través de un enchufe inteligente (consulte la Tabla 4.3), lo que permite a los usuarios encender o apagar la máquina de forma remota. Este dispositivo (también) está integrado en Home Assistant[12]. En la implementación de este componente, esta vez se ha optado por interactuar con la API REST de Home Assistant, en lugar de hacerlo directamente con el enchufe inteligente. La razón

[11] "ESPHome". Sitio web de ESPHome. (), dirección: `https://esphome.io` (visitado 22-07-2023).

[12] "TP-Link Kasa Smart". Sitio web de la integración TP-Link en Home Assistant. (), dirección: `https://www.home-assistant.io/integrations/tplink` (visitado 22-07-2023).

detrás de esta decisión es simplemente ahorrar esfuerzo, reutilizando el API de Home Assistant. En la implementación actual, solo actualiza el consumo de energía en tiempo real y el estado del electrodoméstico (encendido/apagado) en la KB, pero en el futuro, se desea realizar una mejor integración en el SHEMS implementando otras funciones, como la predicción y la programación de operaciones.

4. *Dispositivo piscina*: el EMC de filtración de la piscina se ha incorporado directamente en el controlador de la misma (que también está basado en Node-RED). Actualiza el estado de funcionamiento y el consumo de energía en tiempo real en la KB. También solicita un horario de operación al HEC por la cantidad de horas de filtración que estime conveniente, en base a las previsiones meteorológicas recuperadas de la KB (las cuales son actualizadas por el componente HAS). Este componente también reacciona a las entradas del HEC (a través del mecanismo de orquestación) para detener/reanudar temporalmente la operación y evitar así superar el consumo máximo doméstico establecido.

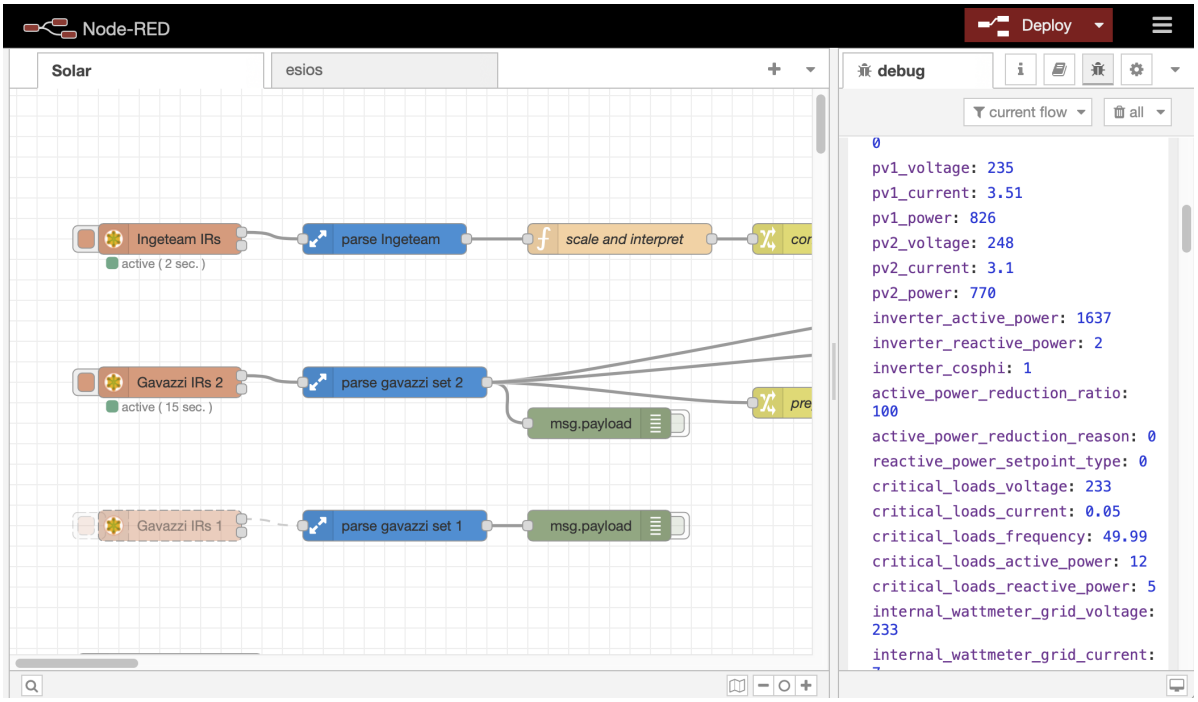


Figura 4.15: Desarrollo en Node-RED del componente DER para PV

5. *DER de PV y batería*: el componente responsable del grupo de paneles PV y la batería, interactúa con el inversor a través de Modbus-TCP (Figura 4.15). La implementación actual del controlador actualiza en la KB la producción en tiempo real y el estado de carga de las baterías. En el futuro, se pretende desarrollar el mecanismo para recibir mensajes de HEC para actualizar su estado máximo de carga y profundidad de descarga, con el objetivo de optimizar el uso de la batería.

6. *HEG*: este componente actualiza en la KB la entrada/salida de energía en tiempo real, medida con el medidor de energía conectado a la red eléctrica en el punto de suministro de la vivienda. Este medidor está conectado al componente utilizando Modbus. También actualiza la potencia máxima que puede entregar el suministro de red eléctrica ofrecido por la empresa de suministradora, así como la máxima que puede ser producida desde la instalación PV. Este componente también recupera los precios horarios de la electricidad, tanto consumida de la red como inyectada a esta, de ESIOS[13] a través de una API REST, según la tarifa PVPC (precio voluntario para el pequeño consumidor) española, aunque el actual contrato de suministro eléctrico, en el banco de pruebas, es de tarifa plana.

7. *HEC*: la implementación actual de este componente es capaz de producir horarios para el consumo de electricidad bajo pedido, tratando de maximizar el uso de electricidad generada por PV. Actualmente lo hace mediante el uso de información de predicción meteorológica sobre cobertura de nubes (proveniente del componente HAS), así como las horas previstas de salida y puesta del sol. También reacciona a las notificaciones relacionadas con dispositivos de alto consumo y toma decisiones basadas en el presupuesto de energía actual (PV, batería y red) para indicar que dispositivos pueden ser detenidos temporalmente para reducir el consumo eléctrico. Finalmente, también notifica a los usuarios a través del componente HAS (Figura 4.16) cuando: (a) la importación de energía a la red alcanza el 80 %, (b) cuando los dispositivos de alto consumo están activos y el hogar no está ocupado y (c) cuando el sistema ha tenido que activar los mecanismos de contingencia (como detener temporalmente el funcionamiento de un dispositivo para prevenir que salten las protecciones de sobrecarga eléctrica doméstica).

4.2.5 Resultados

Entre los distintos subsistemas del banco de pruebas, hemos decidido mostrar los resultados obtenidos en el sistema de cloración, filtración y electrólisis de la piscina; ya que presentan un buen equilibrio entre sencillez, interacción con otros EMC y resultados en reducción de costo energético.

Antes de la implementación del SHEMS, este subsistema se controlaba con un interruptor temporizador eléctrico enchufable. Su programación debía configurarse manualmente varias veces al año, para adaptarse a las diferencias de temperatura, así como a la incidencia solar. La Figura 4.17 muestra la demanda y producción máxima de energía, por hora, para un día laboral promedio de primavera/verano.

De él se puede deducir el horario del sistema de filtración y cloración de la piscina, en el que se aprecian dos huecos (de 6 a 9 y de 14 a 17 horas). Esos intervalos corresponden a períodos

[13] "esios - red eléctrica, PVPC". Sitio web de esios-PVPC, Red Eléctrica Española. (), dirección: `https://www.esios.ree.es/es/pvpc` (visitado 22-07-2023).

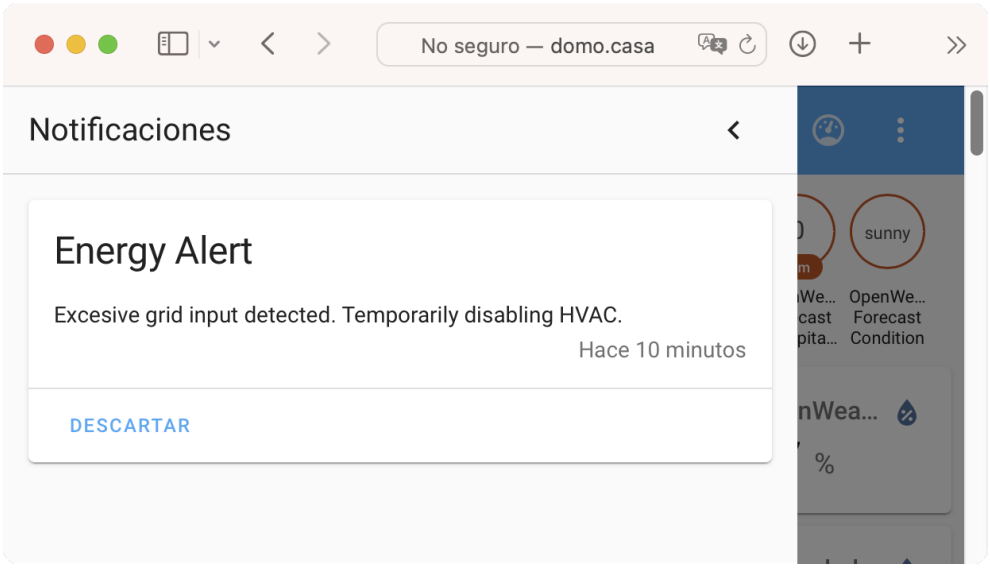


Figura 4.16: Alertas y notificaciones a través de Home Assistant

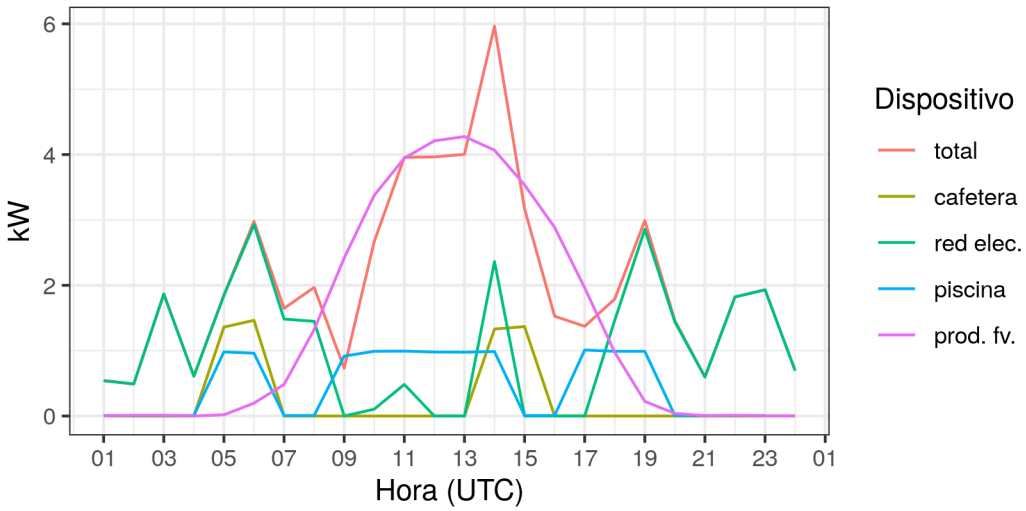


Figura 4.17: Picos de consumo energético previo al SHEMS

específicos del día en los que los usuarios tienen un consumo energético especialmente alto durante días laborables: la hora del desayuno y la comida. En esos momentos, los usuarios utilizan dispositivos de alto consumo como la máquina de café expreso, el fogón vitrocerámico o el microondas. En el pasado, estos dispositivos han demostrado la posibilidad de hacer saltar el magnetotérmico de entrada, cuando se usan en conjunto con el sistema de filtración de la piscina. Para evitar los disparos del elemento de protección, se adoptó un enfoque conservador, evitando ese horario. Vale la pena notar que, aunque el segundo periodo coincide con los momentos de alta producción de energía por los paneles PV, la cobertura de nubes ocasionalmente ha producido caídas transitorias en la producción, ocasionando disparos del interruptor de energía.

La Figura 4.18 representa las mismas variables del ejemplo anterior y condiciones similares en nuestro banco de pruebas (día laboral promedio de primavera/verano), pero esta vez la piscina tiene un controlador de dispositivo integrado en el SHEMS. La implementación del EMC de la piscina decide el tiempo de filtración y cloración en función de la información y los pronósticos meteorológicos locales (actualizados en KB por el módulo HAS) y solicita un horario de operación al HEC, que lo proporciona en función del costo de la energía, lo que resulta en una asignación durante la franja de máxima producción de PV.

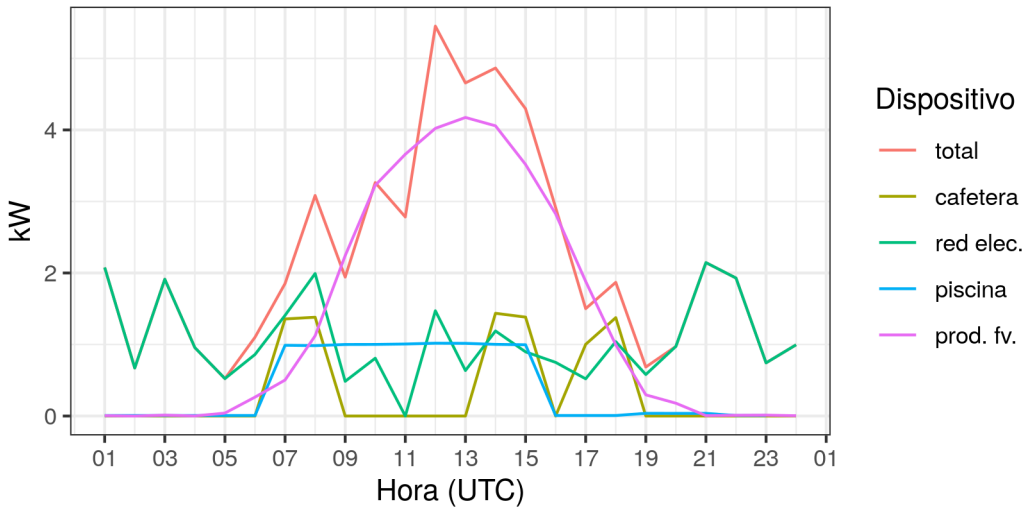


Figura 4.18: Picos de consumo energético con SHEMS

En el escenario anterior, este horario de operación del sistema de filtración habría incurrido situaciones de disparo de las protecciones de sobrecarga. En este caso, el controlador de la piscina integrado en SHEMS recibe comandos del HEC para detener temporalmente la operación cuando existe el riesgo de exceder la carga máxima del punto de suministro de red eléctrica, evitando la necesidad de programar fuera de las horas de producción PV. El HEC envía comandos de control al sistema de filtración de la piscina para detener/reanudar la operación, según el estado de energía de todo el sistema, como se describe en la Sección 4.2.4. La orquestación de los comandos de control también se ha mostrado en la Sección 4.2.3 y finalmente, el mensaje de suscripción

utilizado ha sido mostrado en el Listado 6 y el Listado 7 muestra una notificación correspondiente.

En último lugar, la Figura 4.19 compara la energía total acumulada por hora, importada de la red, para los dos ejemplos anteriores, mostrando que la estrategia conservadora seguida por la implementación del temporizador podría estar asociada con un uso menos eficiente de la producción PV debido a la asignación de tiempo de filtración fuera de las horas de producción, lo que a su vez produce un aumento en la importación de energía de la red, en comparación con el control SHEMS.

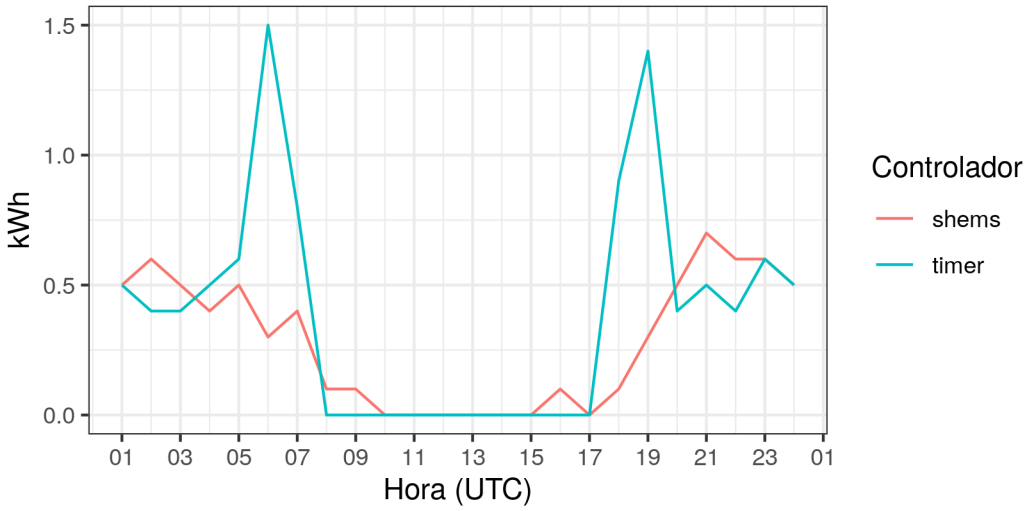


Figura 4.19: Comparativa de importación de energía de la red eléctrica en ambos escenarios

4.3 Discusión

Es preciso abrir la discusión afirmando que los resultados mostrados en la Sección 4.2 no pueden ser tomados como prueba de que el banco de pruebas para el SHEMS logra obtener una mayor eficiencia energética que el escenario anterior, ni del grado en que tal beneficio se obtiene. Solo se han ofrecido para respaldar la afirmación de que el sistema es capaz de integrar con éxito a los diferentes actores presentes en la gestión energética de una casa inteligente para tomar medidas en función de la información procedente de diversas fuentes.

Con la demostración, se pone de manifiesto que es posible la integración un sistema existente de automatización del hogar (Home Assistant), poniendo toda su información a disposición del SHEMS y aprovechándola como una forma conveniente de llegar a los usuarios a través de notificaciones.

Se ha creado una implementación de prueba de concepto de HEG, capaz de integrar el precio de la electricidad en el SHEMS, estableciendo una buena base para la integración de futuras implementaciones de DR. Este HEG puede actuar como intermediario entre un SHEMS y la red eléctrica (o microrredes), abriendo la posibilidad de compartir de forma segura y privada

información seleccionada de la KB del sistema con la red, lo que podría ser beneficioso en DR y escenarios de flexibilidad de la demanda.

Se ha integrado los DER en forma de instalación PV y un sistema de baterías de acumulación adicional, así como diferentes electrodomésticos de alto consumo, con diferentes niveles de funcionalidad, utilizando diferentes tecnologías de comunicación e integrado su información para otros componentes del SHEMS propuesto.

Por último, se muestra un ejemplo sencillo de implementación de administración energética, capaz de programar el consumo para la optimización de energía de PV y reaccionar a la alta demanda de electricidad notificando a los usuarios y deshabilitando dispositivos de alto consumo no críticos.

Esta arquitectura trae al campo un framework para SHEMS modulares donde diferentes partes pueden construir diferentes EMC y donde la DR, automatización del hogar, los DER, dispositivos y usuarios, han sido considerados. Como beneficio adicional, los elementos centrales de la arquitectura se pueden implementar fácilmente desde los componentes comerciales, listos para usar; muchos de los cuales provienen del proyecto FIWARE, como la pila de componentes de seguridad y el CB.

En la propuesta, toda la información del sistema puede ser accedida de forma segura y privada por cualquiera de los EMC del sistema. Además, se accede a esta información mediante protocolos de comunicación estándar diseñados específicamente para la interoperabilidad. Además de eso, la información se formatea y estructura siguiendo los principios de la Web Semántica, aprovechando las ontologías existentes que representan todos los conceptos necesarios, logrando la interoperabilidad de los datos. Por último, utilizando el estándar de comunicaciones de NGSI-LD, implementamos un mecanismo de orquestación para la administración de energía aprovechando la funcionalidad de publicación/suscripción de NGSI-LD. Estos cuatro aspectos son la principal contribución de este trabajo.

4.4 Conclusiones y trabajo futuro

En este capítulo se ha propuesto una arquitectura modular, interoperable y segura para la construcción de SHEMS, presentando un conjunto de EMC para la gestión de la energía de una casa inteligente. La arquitectura presentada también considera las interacciones del prosumidor con la red, la generación local y la optimización de la acumulación, la gestión de dispositivos de alto consumo y la integración con HAS existentes, al tiempo que considera la seguridad y privacidad de los datos a través de mecanismos de control de acceso.

La propuesta se basa en el estándar NGSI-LD, que se utiliza tanto como KB semántica, como para paso de mensajes asíncronos para la orquestación. El uso de este estándar abre la posibilidad de reutilizar implementaciones existentes de diferentes componentes FIWARE en la arquitectura, como el CB, utilizado para contener y acceder a la KB, así como algunos

componentes de seguridad, como el IdM y el PEP-Proxy. Por último, también se han reutilizado las implementaciones existentes de otros proyectos que se encuentran dentro del ecosistema FIWARE, como los componentes DCapBAC del proyecto IoTCrawler.

La base ontológica del modelo de información se ha establecido a partir de una selección de ontologías existentes, analizadas en la Sección 2.3.3, de las cuales se ha seleccionado DABGEO como ontología base, junto con una serie de ontologías complementarias para casos de uso específicos.

La propuesta ha sido validada a través de la presentación de un banco de pruebas que consiste en un hogar con un HAS existente, e instalaciones de producción y almacenamiento PV, que han sido integradas en el SHEMS. Los componentes centrales de la arquitectura, a cargo de la gestión del contexto y la seguridad, se han instanciado a partir de implementaciones existentes. También se han mostrado ejemplos de la representación de información y orquestación de diferentes tareas entre componentes.

Finalmente, se han mostrado resultados en forma de un SHEMS capaz de: (a) programar dispositivos de alto consumo para una mejor utilización de la energía provista por la instalación PV, (b) reaccionar ante un alto consumo de energía notificando a los usuarios y deshabilitando cargas no críticas. Estos resultados demuestran la viabilidad de la solución, pudiendo programar con éxito el sistema de cloración de la piscina en función de la producción de PV e integrando la información proveniente del HAS para reaccionar a los cambios en el consumo de los habitantes. Esto ha sido posible gracias a la capacidad del sistema para integrar diversas fuentes de información y su habilidad para proporcionar mecanismos seguros para acceder a la información desde los diferentes componentes utilizados.

Este trabajo abre la puerta a futuras propuestas sobre SHEMS multifacéticos en los que optimizar sistemas complejos controlando la acumulación y la generación, las estrategias de DR, la interacción e intercambio de datos con el SG y, al mismo tiempo, aprovechando la instalación HAS existente para recuperar información e incluso interactuar con el sistema y los usuarios.

También abre la puerta para el desarrollo y la implementación de EMC específicos que se pueden conectar fácilmente a cualquier SHEMS siguiendo nuestra propuesta de arquitectura. Un ejemplo de ello podría ser el de implementaciones específicas de HEG por parte de diferentes proveedores de energía que permitirían la comunicación entre los hogares y el SG para implementar estrategias elaboradas de DR.

Por último, presenta la posibilidad de crear frameworks e implementaciones de SHEMS específicos, listos para ser implementados e integrados con otras soluciones existentes en un solo clic, que permitirían la fácil implantación de un SHEMS por parte de usuarios legos en la materia que, a su vez, se convertiría en un factor fundamental para el despliegue generalizado de estrategias avanzadas de DR.

CAPÍTULO 5

ARQUITECTURA PARA ORQUESTAR TAREAS DE PROCESADO DE IMÁGENES

Este capítulo presenta una arquitectura para la orquestación de tareas de procesado de imágenes entre el borde y la nube, capaz de aprovechar hardware de aceleración de procesado de imágenes embarcado en dispositivos embebidos de IoT, ubicados en el borde. Esta arquitectura ofrece una mejora en latencia, eficiencia energética, ancho de banda y seguridad respecto a aproximaciones tradicionales. La arquitectura, al igual que la contribución presentada en el Capítulo 4, está basada en interfaces estándar, permitiendo la interoperabilidad y su fácil integración con otras arquitecturas existentes, así como la reutilización de componentes. Finalmente, el uso del estandar NGSI-LD habilita, una vez más, la aplicación de aproximaciones semánticas al modelado de datos.

5.1 Propuesta

En esta sección, se describe una propuesta de arquitectura para la orquestación de tareas de procesamiento distribuido de imágenes, utilizando red neuronal profunda (deep neural network, DNN), capaz de distribuir el procesado entre nodos de computación en el borde y en la nube.

La arquitectura tiene como objetivo aprovechar el nuevo hardware IoT capaz de acelerar la ejecución de modelos modernos basados en DNN para el procesamiento de imágenes, mediante el uso de unidades de procesamiento de gráficos (graphics processing units, GPU). En el proceso de creación de este artefacto, también se ha realizado un ejercicio de reutilización de frameworks existentes y adhesión a estándares existentes para el desarrollo de API y modelos de datos, en la medida de lo posible.

Para la orquestación de las tareas de procesamiento de imágenes, se propone el uso de

FogFlow [200], un marco para la computación en la niebla basado en la familia del estándar de NGSI para datos enlazados (NGSI for linked data, NGSI-LD), la evolución de la anterior interfaz de servicio de nueva generación (next generation service interface, NGSI), ambos utilizados en la iniciativa FIWARE. Al presentar FogFlow, se asegura una amplia compatibilidad con varias soluciones para ciudad inteligente (smart city, SC) y edificios inteligentes ya existentes, también basadas en NGSI-LD.

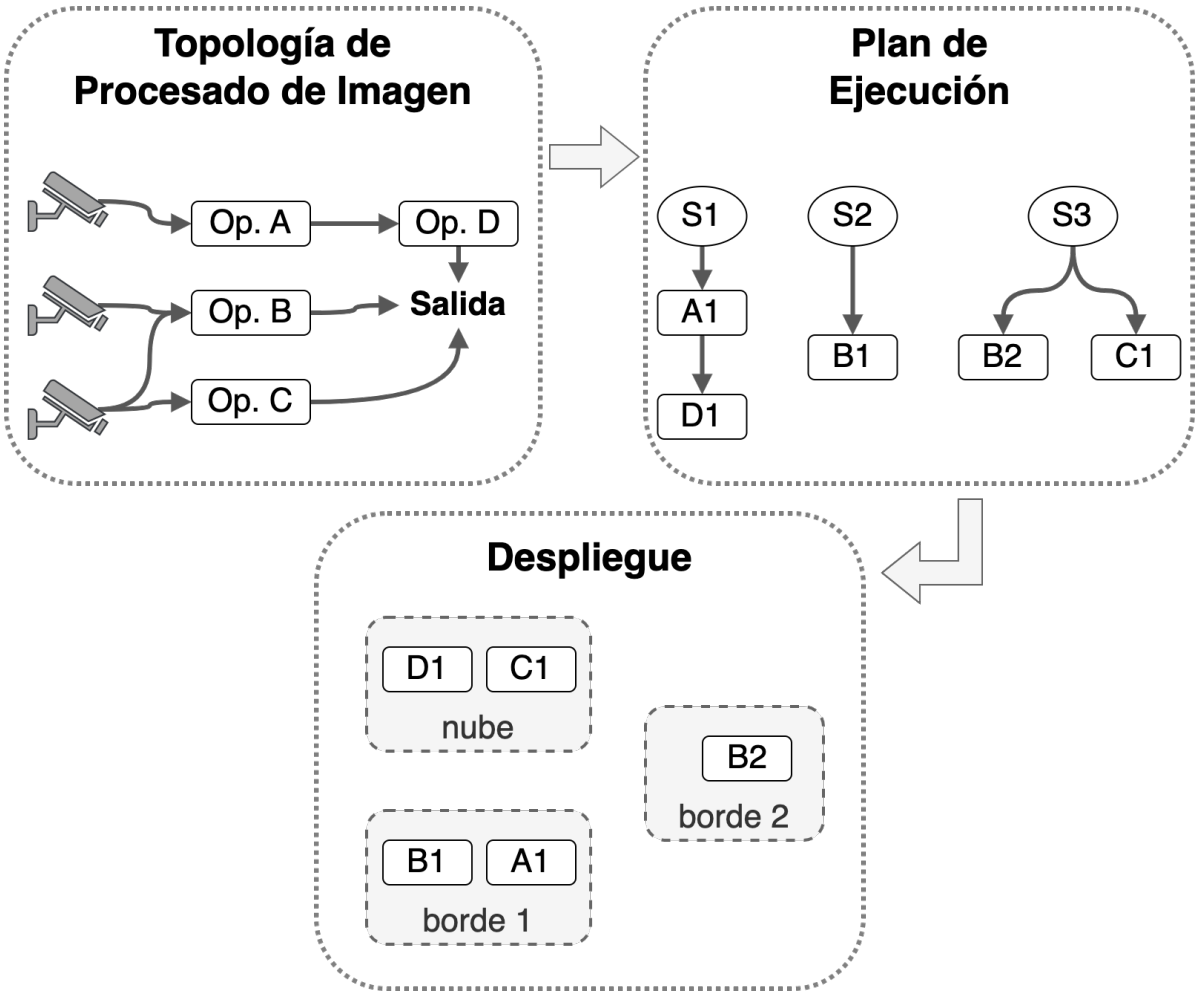


Figura 5.1: Modelo de procesado de imágenes propuesto

La Figura 5.1 muestra el modelo de procesamiento propuesto en el que los operadores procesan datos que son continuamente generados, provenientes de dispositivos en el borde (cámaras), para producir el resultado esperado. El proceso se divide en tres pasos: primero, definimos los operadores que se ejecutarán sobre los datos provenientes de las diferentes fuentes de imagen y/o vídeo. Esos operadores se pueden crear e integrar fácilmente en la arquitectura general a través del Repositorio de operadores o se pueden elegir del catálogo existente de operadores proporcionado en este trabajo. En segundo lugar, en función de la topología de procesamiento de imágenes, FogFlow crea un plan de ejecución para determinar las instancias específicas de

workers (trabajadores) que se implementarán y los flujos de información entre ellos. Finalmente, FogFlow despliega automáticamente las tareas resultantes a los workers más adecuados, donde los operadores finalmente serán ejecutados y alimentados con datos.

5.1.1 Vista general de la arquitectura

Para soportar el modelo de procesamiento, se propone una arquitectura que consta de tres subsistemas lógicos, mostrados en la Figura 5.2, que permite la definición de topologías de procesamiento, la orquestación y la ejecución de tareas de procesamiento de imágenes en el borde y la nube, el sistema de gestión de la información (contexto) y el control de todo el sistema.

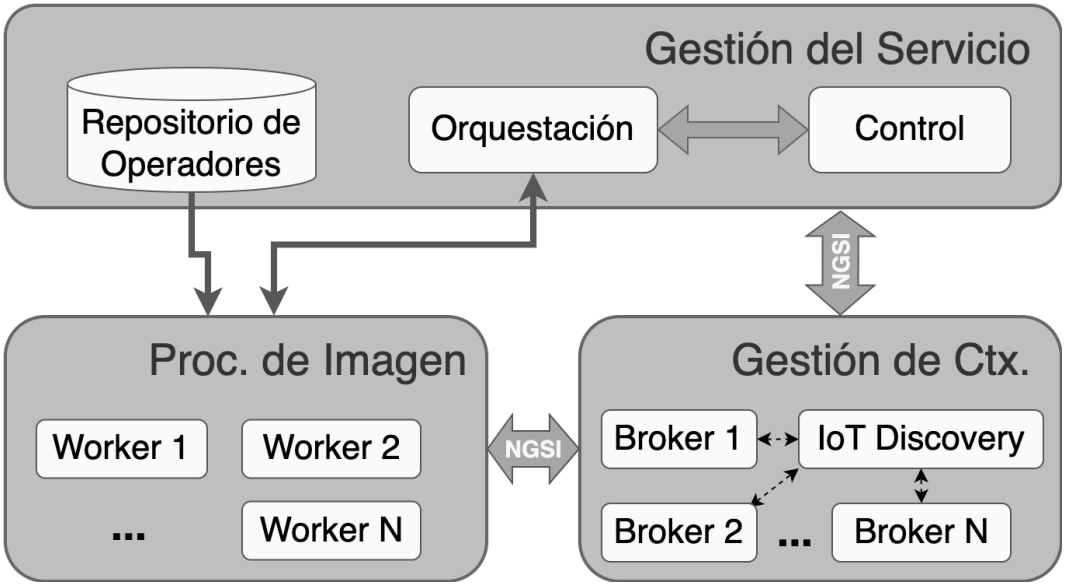


Figura 5.2: Vista general de la arquitectura para orquestación de procesado de imagen

La primera capa representa el subsistema de gestión del servicio, que incluye el módulo de control, el módulo de orquestación y el repositorio de imágenes Docker[1], que almacena las implementaciones de todos los operadores disponibles en el sistema.

La capa de Procesamiento de Imágenes, donde se realiza el cómputo, está compuesta por un conjunto de nodos de procesamiento de imagen (image processing nodes, IPN) que ejecutan la tarea asignada por el módulo de orquestación y realizan el procesamiento de imágenes. Los IPN son dispositivos implementados en el borde o en la nube que pueden ejecutar múltiples tareas según sus recursos de cómputo y son descritos en profundidad en la Sección 5.1.4.

Finalmente, la capa de administración de contexto es una capa distribuida que administra toda la información de contexto y proporciona interfaces basadas en el estándar NGSI-LD, para consultas, suscripciones y actualizaciones de entidades de contexto.

[1]"Docker. Accelerated, Containerized Application Development". Sitio web de Docker, Docker Inc. (), dirección: `https://www.docker.com` (visitado 22-07-2023).

5.1.2 Gestión del servicio

FogFlow nos permite implementar servicios de procesamiento de imágenes mediante un modelo de programación basado en la intención[2] y una arquitectura de orquestación basada en el contexto. Para hacerlo, ofrece dos formas de realizar la implementación del servicio para admitir diferentes tipos de patrones de carga de trabajo: topologías de servicio y fog-functions.

Las topologías de servicio definen conjuntos de operadores vinculados que procesan los datos de manera progresiva, conforme se hacen disponibles, para producir algún resultado. Estas topologías se activan en función de la demanda de los consumidores, de manera automática: cuando un dato es requerido, se despliega una topología que generará el resultado. Por otro lado, las fog-functions son tareas simples que se activan y despliegan durante su activación y se ejecutan continuamente cuando se generan datos de entrada, o se actualizan las entidades de entrada asociadas a algún operador y pueden crear nuevas entidades o modificar entidades existentes. En ambos casos, el framework utiliza workers para implementar y ejecutar tareas de procesamiento de datos, también llamadas operadores, que pueden ser desplegadas en el borde y en la nube.

Se han utilizado fog-functions para implementar las tareas de procesamiento de imagen, porque se adaptan de manera más sencilla y flexible a la arquitectura propuesta que la topología de servicio. Utilizando las fog-function como base para la arquitectura de procesado de imagen, es necesario definir entidades para activar las funciones, operadores que procesan los datos y entidades de resultado.

El módulo de control proporciona una interfaz para que los usuarios del sistema diseñen, administren y monitoreen topologías de procesamiento de imágenes y flujos de datos. También se encarga de dar de alta nuevos nodos de procesamiento en el sistema y realizar un seguimiento de los mismos. Por último, el módulo de Control se encarga de la configuración de entidades iniciales, que impulsarán la ejecución de algunas tareas de procesamiento que se inician a partir de notificaciones, como flujos de protocolo de transporte en tiempo real (real-time transfer protocol, RTP) o cámaras integradas.

El módulo de Orquestación controla dinámicamente el proceso de implementación de tareas, en función de los trabajadores y los datos disponibles, y decide dónde implementar las tareas en la nube y los dispositivos perimetrales.

El Repositorio de Operadores consiste en un repositorio Docker en el que se almacenan todos los operadores, haciéndolos disponibles para su despliegue en forma de tareas en los IPN. La estructura del repositorio de operadores se discutirá más adelante en la Sección 5.1.6.

[2]Del inglés *intent-based programming*, un paradigma de programación desarrollado por Charles Simonyi, en el que el código se desarrolla describiendo la intención del mismo, según el usuario o el programador

5.1.3 Gestión del contexto

El subsistema de administración de contexto proporciona una vista global del sistema distribuido de contexto, que incluye un conjunto de brokers IoT y un componente IoT discovery centralizado, que actúa como directorio central donde consultar la información contenida en el sistema. La Figura 5.3 ilustra estos componentes.

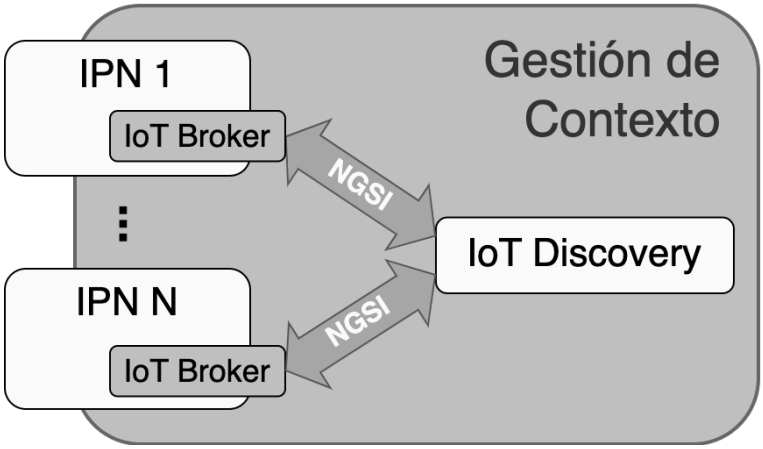


Figura 5.3: Gestión del contexto en la arquitectura

Cada broker contiene un subconjunto de la información de todo el sistema, en forma de entidades. El IoT discovery, por otro lado, sirve como un punto centralizado donde se puede consultar toda la información disponible en el sistema (en poder de los intermediarios). Tanto el discovery como el broker siguen el estándar NGSI-LD, que establece tanto el API REST como el formato de datos para las comunicaciones. Con estos intercambian datos entre tareas y gestionan los cambios en el contexto.

Los brokers proporcionan servicios de consulta local con los que se puede recuperar la información contenida en este. También proporcionan servicios para la creación y modificación de la información. Finalmente, los IoT brokers proporcionan un mecanismo de publicación/suscripción en el que se basa todo el sistema de orquestación de tareas, para realizar la entrega de datos a las tareas en cuestión, de manera asíncrona mediante notificaciones que son disparadas cuando se generan cambios en las entidades existentes o se generan nuevas entidades.

El IoT discovery realiza un seguimiento de los diferentes brokers, manteniendo un registro de todas las entidades disponibles en el sistema y su ubicación. Cuando se le pregunta por alguna información específica, el componente discovery hace de intermediario, siendo capaz de encontrar la ubicación de esa entidad y retransmitir la respuesta.

5.1.4 Procesado de imagen

Los nodos de procesamiento de imagen (image processing nodes, IPN) son dispositivos heterogéneos, cada uno con diferentes prestaciones y recursos de cómputo, ubicados en diferentes sitios,

que pueden procesar múltiples tareas dependiendo de su potencia de cómputo y características. Por lo tanto, los IPN se definen como unidades de procesamiento de datos que se pueden implementar tanto en la nube como en el perímetro.

La Figura 5.4 muestra la estructura del nodo de procesamiento de imagen, en la que vemos dos elementos principales: el broker, responsable de contener y hacer disponible toda la información de contexto relacionada con las tareas locales y el worker, que es la pieza encargada de ocuparse de la gestión del ciclo de vida de las tareas en el nodo local. Esta es una pieza importante en el proceso global de orquestación de procesado de imagen, encargada del despliegue local de tareas y su monitorización.

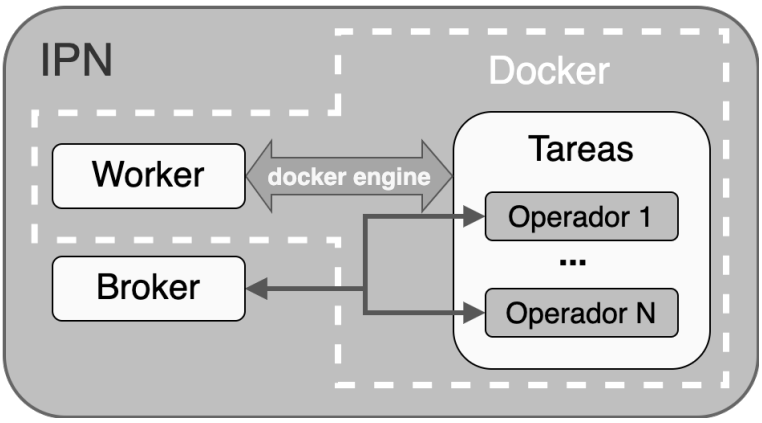


Figura 5.4: Estructura del nodo de procesado de imágenes (IPN)

Cuando se crea una instancia de una tarea y se destina a un IPN determinado, el módulo de orquestación envía la configuración de la tarea (flujos de entrada y salida y el operador de la tarea) al worker del IPN, que a su vez ejecuta la tarea en forma de un contenedor Docker, utilizando la imagen Docker correspondiente al operador de la tarea, que puede ser descargada del repositorio de imágenes Docker y finalmente materializa la configuración requerida por la tarea para que esta pueda comenzar a trabajar.

Para poder aprovechar la aceleración de DNN que ofrece la GPU en dispositivos embebidos para el borde de última generación, como la Jetson Nano[3], fue necesario realizar algunas modificaciones en el código fuente de FogFlow, para poder ejecutar contenedores Docker capaces de utilizar el entorno de ejecución de NVIDIA capaz de aprovechar el hardware de aceleración con arquitectura de computación unificada de dispositivos (Compute Unified Device Architecture, CUDA). Estas modificaciones consistían principalmente en la forma de invocar la ejecución de contenedores, para que estos pudieran acceder al hardware de aceleración, así como en las imágenes Docker base utilizadas para la creación de nuevos operadores, que fueron sustituidas por otras que disponían de las librerías CUDA, proporcionadas por NVIDIA [203].

[3]"Meet Jetson, the Platform for AI at the Edge". Sitio web de la plataforma Jetson de NVIDIA, Nvidia Corporation. (), dirección: `https://developer.nvidia.com/embedded-computing` (visitado 22-07-2023).

5.1.5 Tareas de procesado de imagen

Las tareas pueden considerarse instancias de un operador para un conjunto específico de datos. Si bien consideramos a los operadores como los principales motores de procesamiento de imágenes y vídeos, las Tareas pueden considerarse el vínculo entre el sistema de orquestación y el sistema de procesamiento de imágenes. Son contenedores Docker que ejecutan un operador específico en un IPN y usan NGSI-LD para recibir notificaciones y enviar actualizaciones al broker. La estructura básica y el modelo funcional de una tarea se pueden ver en la Figura 5.5.

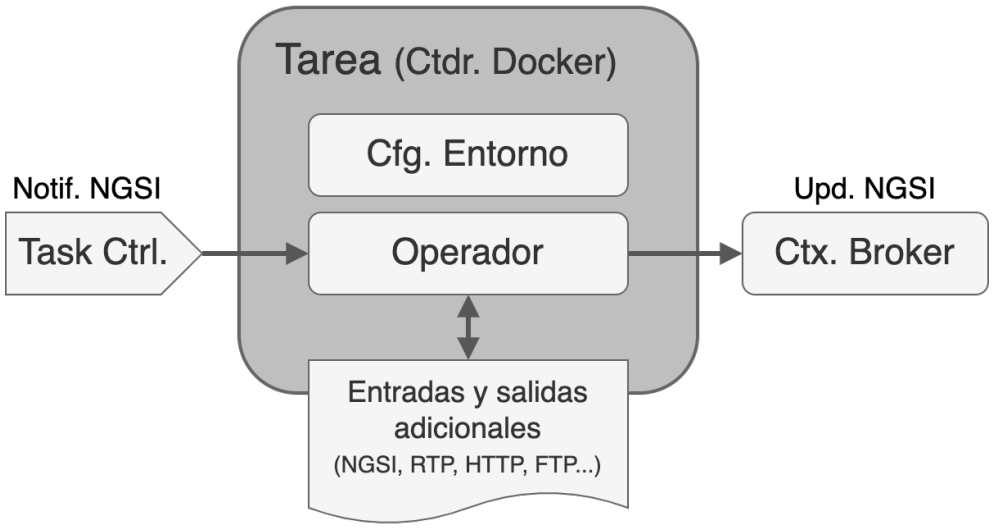


Figura 5.5: Estructura de la tarea de procesado de imagen

Las tareas son creadas por el worker, que establece una suscripción en el broker para que este envíe, conforme se produzca, la información que alimentará la tarea. Además, pasa la configuración necesaria a la tarea; como el broker local al que tiene que enviar sus resultados y otros parámetros necesarios. Esas configuraciones las pasa el worker al contenedor Docker en forma de variables de entorno.

La entrada principal para una tarea proviene de la notificación que el worker establece para ella, en el momento de la creación. A través de esta suscripción, la tarea recibirá la información necesaria para realizar el procesamiento de imagen/vídeo. Además, las tareas pueden usar otras interfaces de entrada y salida, como RTP, HTTP o FTP (file transfer protocol, protocolo de transferencia de ficheros), para intercambiar datos adicionales. En base al conjunto de operadores ideados inicialmente para el sistema, identificamos cuatro casos diferentes para los datos de entrada de tarea:

- Las tareas se conectarán a una fuente de vídeo proveniente de una cámara conectada directamente al dispositivo (una cámara interna). Las cámaras MIPI (mobile industry processor interface, interfaz de proceso móvil industrial) y bus de serie universal (universal

serial bus, USB) son ejemplos comunes. Esto se representará como una cámara interna con una URL o ruta específica para acceder a la transmisión de vídeo.

- La tarea utiliza un flujo de vídeo en línea, representado con una URL RTP, como fuente de vídeo. El operador se conectará así al flujo de vídeo y procesará los datos recibidos directamente de la fuente.

- La tarea recibe una imagen incrustada. La tarea de procesamiento de imágenes se realizará solo en esa imagen.

- La entidad de solicitud tiene una dirección URL. Los datos pueden ser un vídeo o una imagen y deben extraerse del recurso (por ejemplo, un servidor de archivos al que puede acceder la instancia de la tarea).

De manera similar, las tareas pueden tener diferentes salidas dependiendo del operador ejecutado. A continuación, se ofrece una lista de algunos ejemplos de elementos que se pueden encontrar en una respuesta típica:

- Un valor numérico con el número y tipo de los elementos detectados; como el número de peatones o animales detectados.

- Conjuntos de elementos que contienen las coordenadas dentro de la imagen y el tipo de detección; como las coordenadas de los objetos detectados en la imagen, y opcionalmente el clasificador de tipo (animal, peatón, coche. . .).

- Una imagen original incrustada que representa la imagen completa a la que se refiere el elemento enumerado anteriormente.

- Una imagen modificada incrustada con áreas delineadas donde se ha detectado alguna característica. Este tipo de imagen podría ser útil para enviar notificaciones a usuarios humanos.

- Una matriz de imágenes con los elementos detectados: fragmentos de la imagen original, que representan los objetos detectados.

- Una URL de transmisión de vídeo RTP que ofrece el flujo modificado en tiempo real, de manera similar a un punto anterior en el que se dibujaron contornos sobre la imagen original, delineando las áreas de detección. Esto podría ser útil para tableros de mando y monitorización en tiempo real por parte de operadores humanos.

- Una dirección URL a un archivo que representa algún fragmento de vídeo o imagen, resultante del proceso de detección, que la tarea ha subido a un servidor de archivos local u otra facilidad de almacenamiento.

En base a la descripción de entradas y salidas detallada, es posible usar los datos resultantes de una tarea como entrada para otra, como se representó previamente en la Figura 5.1.

5.1.6 Operadores

Los operadores son implementaciones software concretas de una tarea de procesamiento de imágenes. Se almacenan en forma de imágenes Docker que contienen todas las aplicaciones y/o modelos DNN necesarios para procesar los datos de entrada. Definimos varios procesadores de imagen que hacen uso de los *Jetson Pre-Trained Models*[4] (Tabla 5.1). Cada operador implementará típicamente un modelo de procesador de imagen dependiendo del caso de uso del operador.

Tabla 5.1: Modelos preentrenados disponibles para procesado de imagen

Modelo	**Tipo**	**Descripción**
AlexNet [204]	Reconocimiento de Imagen	Clasificación de imagen perceptual. Capaz de identificar 1000 clases diferentes (ILSVRC2010)
GoogleNet [205]	Reconocimiento de Imagen	Variante de Inception Network. Capaz de identificar 1000 clases diferentes (ILSVRC 2014). Dos versiones disponibles: la original y una versión reducida entrenada en el subconjunto de 12 clases de ILSVRC 2014.
ResNet [206]	Reconocimiento de Imagen	Deep Residual Network. Capaz de identificar 1000 clases diferentes (ILSVRC2015). Cuatro versiones disponibles según el número de capas (18, 50, 101 y 152).
VGG [207]	Reconocimiento de Imagen	Red neuronal convolucional. Capaz de identificar 1000 clases diferentes (ILSVRC 2014). Versiones de 16 y 19 capas disponibles.
Inception-v4 [208]	Reconocimiento de Imagen	Última versión de Inception CNN. Capaz de identificar 1000 clases diferentes (ILSVRC2015, entrenado en el conjunto de datos ILSVRC2012).

[4]D. Franklin. "Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson". Repositorio de código de modelos preentrandos para Jetson Nano, NVIDIA Corporation. (), dirección: `https://github.com/dusty-nv/jetson-inference` (visitado 22-07-2023).

Continuación de Tabla 5.1

Modelo	**Tipo**	**Descripción**
SSD-Mobilenet [209]-[211]	Detección de Objetos	Capaz de identificar 91 clases de objetos del catálogo MS-COCO [212]. Hay dos versiones disponibles.
SSD-Inception V2 [213], [214]	Detección de Objetos	Capaz de identificar 91 clases de objetos del catálogo MS-COCO
DetectNet-COCO	Detección de Objetos	DNN NVIDIA para detectar solo una clase MS-COCO. Hay disponibles modelos preentrenados para las clases: Perro, Botella, Silla y Avión del catálogo COCO.
Modelo Ped-100	Detección de Objetos	DetectNet para detectar un peatón o persona
Modelo Multiped-500	Detección de Objetos	DetectNet para detección de múltiples peatones y equipaje
Facenet-120 [215]	Detección de Objetos	Detector de caras humanas, entrenado en el conjunto de datos FDDB
Cityscapes	Segmentación Semántica	Modelo preentrenado FCN-Resnet18, utilizando el conjunto de datos Cityscapes (versiones 512x256, 1024x512 y 2048x1024)
DeepScene	Segmentación Semántica	Modelo FCN-Resnet18 preentrenado utilizando el conjunto de datos DeepScene (versiones 576x320 y 864x480)
Multi-Human	Segmentación Semántica	Modelo FCN-Resnet18 preentrenado utilizando el conjunto de datos Multi-Human (versiones 512x320 y 640x360)
Pascal VOC	Segmentación Semántica	Modelo FCN-Resnet18 preentrenado utilizando el conjunto de datos Pascal VOC (versiones 320x320 y 512x320)
SUN RGB-D [216]	Segmentación Semántica	Modelo FCN-Resnet18 preentrenado usando el conjunto de datos SUN RGB-D (versiones 512x400 y 640x512)

En algunos casos de uso, los operadores pueden alimentarse desde la salida de otro operador. Ese podría ser el caso de un flujo en el que un primer operador genera un conjunto de imágenes de detección de rostros a partir de una transmisión de vídeo o una imagen, el segundo utiliza estas imágenes para crear una lista de firmas biométricas (*embeddings*) que representan las características más importantes del rostro, que a su vez serán utilizadas en un tercer operador, donde se identifican esos rostros buscando rostros desconocidos para el sistema.

Usando el ejemplo anterior, identificamos una jerarquía entre operadores, algunos de ellos usan datos crudos (sin procesar) provenientes de sensores y otros usan datos procesados para generar información adicional útil, como notificaciones o alertas. La tabla 5.2 muestra algunos ejemplos de operadores.

Los operadores que trabajan con flujos RTP generan salidas continuamente, por lo que necesitamos limitar el rendimiento para evitar la posible saturación del sistema. Para este fin, se almacena temporalmente en un búfer las detecciones, de acuerdo con un parámetro de "limitación" (*throttling*), que define la tasa de salida a la que el operador enviará notificaciones de acuerdo con las detecciones de entrada y que es pasado a través de la configuración en la entrada de la tarea. Para evitar superar el límite de detecciones por unidad de tiempo marcado, se descartarán notificaciones intermedias o se realizará alguna operación de agregación. Cuando la salida es otro flujo de vídeo RTP, inicialmente no hay necesidad de limitar la tasa de salida de datos, que generalmente tendrá la misma velocidad de fotogramas que la entrada, pero aun así podría ser interesante remuestrear el flujo de vídeo para reducir la velocidad de fotogramas y/ o la resolución, como etapa previa a la ejecución de un modelo más complejo que se beneficiaría de la reducción de complejidad en la entrada.

Los operadores se almacenan en forma de imágenes Docker, que pueden ser distribuidas a través del repositorio de operadores y se pueden crear fácilmente, extendiendo plantillas base de operador, disponibles para los lenguajes Python y NodeJS. Estas plantillas están basadas en las plantillas originales ofrecidas por FogFlow, e incluyen las bibliotecas de visión NVIDIA Jetson para aceleración hardware de DNN. Una vez compilada, la imagen Docker del Operador debe almacenarse en el repositorio de operadores para que los workers puedan utilizarla para iniciar tareas en base a dichos operadores.

El repositorio de operadores (presentado en la Figura 5.6) que, como se mencionó anteriormente, es un repositorio privado de Docker; permite el almacenamiento de imágenes Docker para múltiples arquitecturas. Gracias a esta característica es posible iniciar operadores de manera agnóstica a la arquitectura subyacente y por tanto definir las topologías de procesamiento de igual manera.

Se puede usar esta característica para construir imágenes Docker con una implementación diferente según la arquitectura de destino. Por ejemplo, es posible definir un operador de reconocimiento facial con tres arquitecturas de destino: i86, ARM (advanced RISC machine, máquina RISC avanzada) con aceleración CUDA y x64 con una TPU (tensor processing unit, unidad de

Tabla 5.2: Ejemplos de operadores para procesado de imagen y vídeo

Operador	**Descripción**
PedestrianDetector	Genera un conjunto de imágenes con el peatón detectado partiendo de una imagen o flujo RTP
FacesDetector	Genera un conjunto de imágenes de caras detectadas partiendo de una imagen o flujo RTP
BioHashGenerator	Genera un conjunto de signaturas (hashes) biométricas a partir de un conjunto de imágenes de caras
CountUnknownPeople	Cuenta personas desconocidas a partir de un conjunto de signaturas biométricas
CountAnimals	Cuenta animales detectados a partir de una imagen o flujo RTP
PlateIdentifier	Genera un conjunto de imágenes con las matrículas de vehículo detectadas a partir de una imagen o un flujo RTP

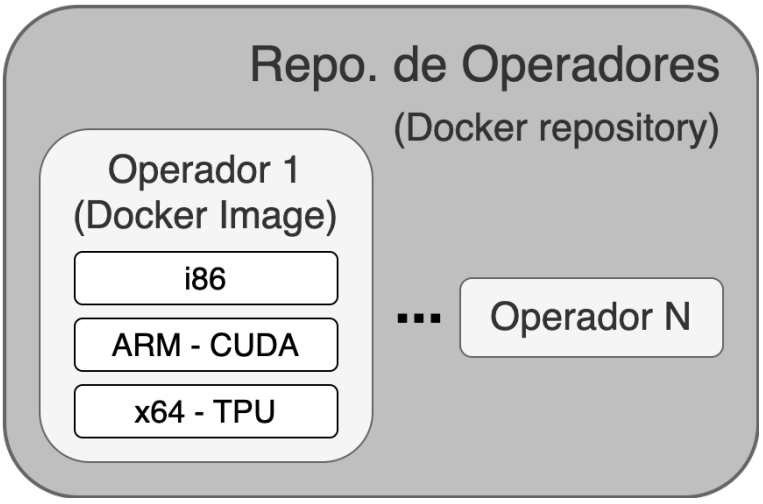


Figura 5.6: Repositorio de operadores y arquitecturas disponibles

procesamiento de tensores). En el primer caso, el operador se implementa sin ninguna aceleración de hardware y el proceso de reconocimiento facial solo usa un procesador compatible con i86. En el segundo caso, el operador está implementado para usar la aceleración CUDA para acelerar el proceso y reducir el consumo de energía. Finalmente, el tercer caso es similar al segundo, pero requiere otra implementación del operador que usa una TPU hipotética, para acelerar el procesamiento. Las tres versiones se almacenan en el repositorio de Docker con el nombre del operador, de modo que cuando los dispositivos de borde crean una instancia de esos operadores, el motor de Docker es responsable de extraer automáticamente la imagen de Docker adecuada a la arquitectura del dispositivo.

5.1.7 Modelo de datos

Como ya se mencionó, la entrada y salida de los operadores son entidades NGSI-LD representadas mediante JSON-LD. Cada caso de uso tendrá un conjunto específico de atributos de entrada y salida NGSI-LD basado en los requisitos, capacidades y resultados del operador. En la Figura 5.7 podemos encontrar la abstracción de los atributos requeridos para el conjunto de operadores que hemos presentado como propuesta inicial.

Los atributos que se encontrarán principalmente en cada entrada son los que representan la imagen de origen o el contenido de vídeo para el procesamiento, representados por *internalCamera*, *videoStream*, *embeddedImage* y *externalFile*. Además, se requiere el atributo *throttling* para aquellas tareas que pueden operar en flujos continuos, para evitar la saturación de los pasos posteriores en la cadena de procesamiento.

Los atributos de salida son más diversos, ya que están estrechamente relacionados con el procesamiento específico realizado por el operador, pero en general representan la característica detectada o un conjunto de ellas (*objectDetection*, *numberDetections* y *ordinateSet*), la imagen original y el fragmento o colección de fragmentos dentro de la imagen donde se ha detectado algo (*embeddedImage* and *pictureSet*). Opcionalmente, para monitorización y tableros de mando, el operador también puede generar un vídeo modificado, con marcadores superpuestos sobre las áreas detectadas (*videoStream* and *externalFile*).

Como ejemplo de representación de las entradas recibidas por nuestro operador de detección de objetos, el Listado 8 muestra una posible configuración de este operador para ingerir datos de un flujo RTP.

En este listado, podemos ver que el *type* y el *id* de la entidad NGSI-LD que actúa como configuración de entrada para la tarea de detección de objetos, representan identificadores del operador en uso, así como la instancia de tarea específica implementada en un dispositivo perimetral. Esto es necesario para que el sistema de orquestación FogFlow pueda identificar y dirigirse específicamente a la tarea correcta y realice la entrega de datos.

Finalmente, el Listado 9 muestra una entidad NGSI-LD, representada en JSON-LD, de la salida del mismo operador. En esta podemos ver algunas de las regiones donde se han detectado

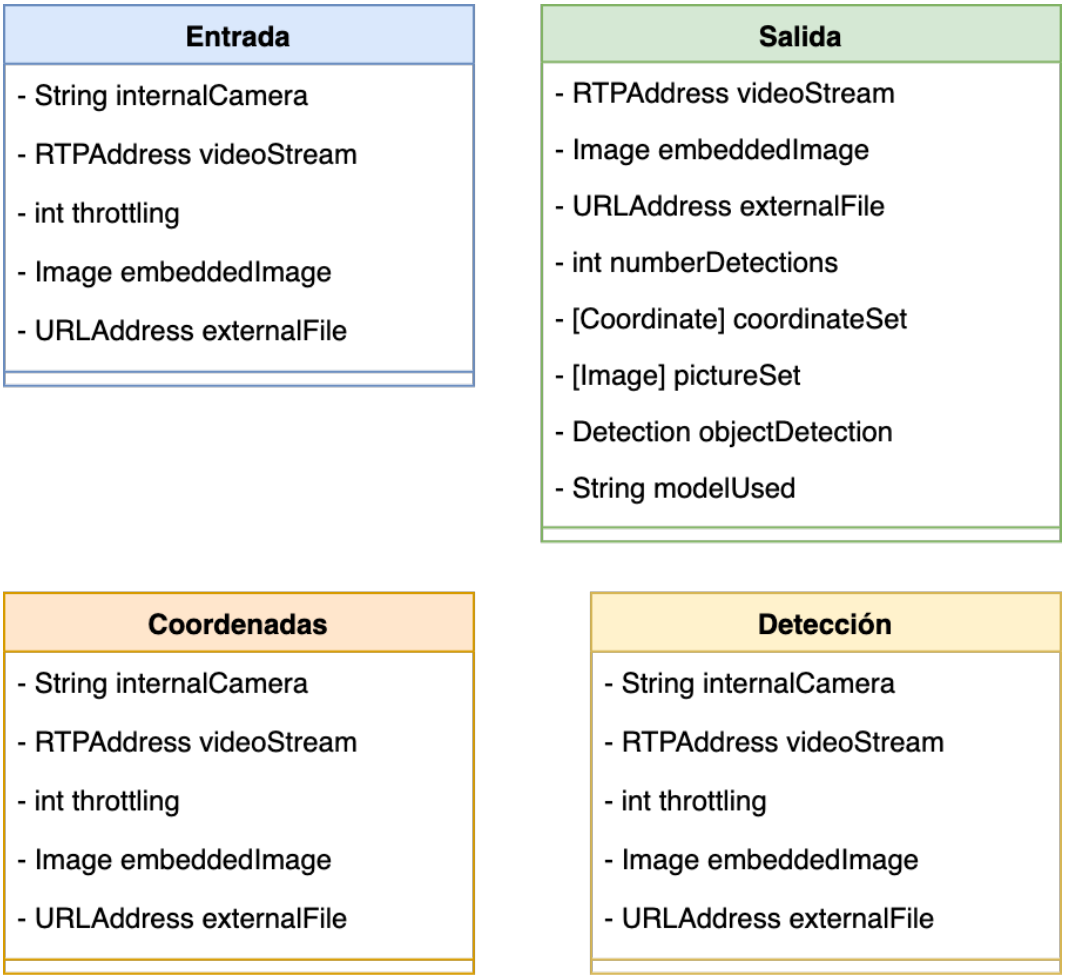


Figura 5.7: Modelos de datos generales de entrada y salida de tareas

```
{
  "id": "object_detection:1_in",
  "type": "cam1_taskA1_input",
  "videoStream": {
    "value": "rtsp://192.168.1.140:88/videoMain",
    "type": "URL"
  },
  "throttling": {
    "value": 60,
    "type": "Integer"
  }
}
```

Listado 8: Entidad NGSI-LD correspondiente a la entrada de un operador de tarea

diferentes objetos, así como una representación codificada en Base64 del fotograma donde se han detectado y el identificador de tipo del objeto detectado.

5.2 Validación

Siguiendo las pautas metodológicas de *Design Science Research*[217], se realiza la evaluación del artefacto mediante demostración. Para este trabajo, se ha seleccionado un ejemplo basado en un sistema de vigilancia para hogar inteligente en un escenario de casa rural. También se han considerado otros ejemplos que han sido descartados para no extender en exceso esta sección, como el conteo de personas que podría aplicarse en diferentes áreas públicas (como pasillos y bibliotecas) en un campus universitario y la monitorización de presencia en interiores por reconocimiento facial, entre otros.

5.2.1 Sistema de videovigilancia en hogar inteligente

Para la evaluación de este sistema, se presenta un caso de uso demostrativo de mejora del sistema de seguridad y vigilancia de una casa rural hipotética. Esta casa está ubicada en emplazamiento remoto con opciones limitadas de conectividad a Internet y ancho de banda reducido. La vivienda ya cuenta con un sistema NVR (network video recorder, grabador de vídeo en red), compuesto por varias cámaras IP y un grabador central, que es capaz de detectar y grabar eventos de movimiento y enviar notificaciones a través de Internet. Las cámaras están ubicadas en el exterior de la casa, apuntando al perímetro interior de la propiedad. Al ser un área rural, hay algo de vida silvestre presente, lo que activa las alarmas de detección de movimiento de manera ocasional. Además, otros eventos, como el sombreado de las nubes, el movimiento de ramas debido al viento e incluso el cambio entre imágenes en color y en blanco y negro en las horas del crepúsculo (cuando el sistema de imágenes infrarrojas cambia de filtro) generan numerosos falsos positivos.

Limitado por la conectividad disponible, el envío de flujos de vídeo a un servicio de procesamiento en la nube es una opción inviable. Como resultado, la solución tiene que hacer uso de procesamiento local. En este punto, se identifican dos propuestas alternativas adicionales, una de las cuales es instalar una estación central de procesamiento de imágenes equipada con hardware acelerado, capaz de realizar un procesamiento avanzado de imágenes en las secuencias de vídeo y detectar con éxito eventos de interés para el propietario (como intrusión humana, presencia de animales salvajes o vehículos) mientras se rechazan tantos falsos positivos como sea posible.

Otra alternativa es implementar dispositivos de borde, más cerca de las cámaras o embebidos en ellas, para realizar el procesamiento de imágenes distribuidas en todo el borde, aprovechando las capacidades avanzadas de los nuevos dispositivos equipados con GPU. La Tabla 5.3 resume algunas de las ventajas y desventajas de cada una de las alternativas presentadas: computación en la nube, instalación de un servidor de procesamiento de imágenes local y utilización de la solución propuesta en este trabajo, descrita a continuación.

```
{
  "id": "object_detection:1_out",
  "type": "cam1_taskA1_output",
  "embeddedImage": {
    "value": "TWFuIGlzIGRpc...",
    "type": "Base64"
  },
  "numberDetections": {
    "value": 2,
    "type": "Integer"
  },
  "coordinateSet": [{
    "type": "Array",
    "value": {
      "x1": 37,
      "y1": 75,
      "x2": 83,
      "y2": 151,
      "detection": {
        "class": "CAT",
        "catalogue": "COCO",
        "confidence": 0.875
      }
    }
  },
  {
    "type": "Array",
    "value": {
      "x1": 47,
      "y1": 150,
      "x2": 195,
      "y2": 151,
      "detection": {
        "class": "PERSON",
        "catalogue": "COCO",
        "confidence": 0.986
      }
    }
  }]
}
```


Listado 9: Entidad NGSI-LD correspondiente a la salida de un operador de tarea

Tabla 5.3: Fortalezas y debilidades de diferentes estrategias computación

Caso	**Fortalezas**	**Debilidades**
Computación en la nube	Elevada potencia de procesado AI y potencia de cómputo. Máxima capacidad de almacenamiento.	Elevada latencia. Elevado consumo de ancho de banda y dependencia de la conexión de red.
Servidor local	Baja latencia. Ahorra ancho de banda externo.	Centralizado. Elevado consumo de ancho de banda en red local.
Computación en el borde	Mínima latencia. Ahorra ancho de banda. Descentralizado.	Limitaciones de procesado y almacenamiento

La solución propuesta es utilizar microcomputadoras Jetson Nano, integradas en el sistema de orquestación propuesto. Podría decirse que las microcomputadoras son menos intrusivas que la estación de procesamiento, ya que esta última requiere un entorno protegido, ocupa espacio de la vivienda (que podría ser escaso o de difícil acceso), suelen generar calor y ruido y en definitiva debe ser una instalación debidamente planificada. Por otro lado, las Jetson Nano se pueden desplegar más cerca de las cámaras (incluso en pequeñas carcasas ubicadas junto a la cámara) y brinda varios beneficios, como:

- Uso mejorado de la red local, ya que las secuencias de vídeo no inundarán la red local.
- Escalabilidad mejorada, ya que se pueden agregar más microcomputadoras junto con nuevas cámaras, lo que genera costos más granulares en comparación con el aumento de GPU centralizado.
- Tolerancia a fallas mejorada, en comparación con la solución centralizada en que existe un punto único de falla de hardware, capaz de interrumpir totalmente el servicio.

Un punto final en el que esta solución podría mostrar beneficios sobre el enfoque centralizado, es el de la eficiencia. El reducido consumo de energía de las Jetson Nano, podría brindarles una ventaja en algunas situaciones en las que se requiere una mayor eficiencia energética (como se muestra en [218], [219]). Sin embargo, el consenso es que mejorar la eficiencia afecta la latencia del cómputo y, por lo tanto, puede ser perjudicial sino se aborda adecuadamente en el caso de uso específico [220].

5.2.2 Selección de hardware

Hay muchas opciones comerciales disponibles de dispositivos perimetrales, como Raspberry Pi, que son útiles para la computación perimetral. Por otro lado, para cumplir con los requisitos de

latencia y potencia informática de esta propuesta, la aceleración del procesado de imagen y vídeo utilizando GPU es necesaria, sino obligatoria, en el presente caso de uso; por lo tanto, Jetson Nano se presenta como la mejor candidata para el procesamiento de imágenes en el borde. La Tabla 5.4 compara las características del hardware de la Jetson Nano, con la popular Raspberry Pi 4 Model B.

Tabla 5.4: Comparación de hardware entre Raspberry Pi 4B y Jetson Nano

Spcf.	**Raspberry Pi 4B**	**Jetson Nano**
Fabricante	Raspberry Pi Foundation	NVIDIA Corporation
OS	Raspberry Pi OS (Debian)	Linux4Tegra (Ubuntu 18.04)
CPU	ARM A72 (Quad-core 1.5 GHz)	ARM A57 (Quad-core 1.4 GHz)
GPU	Broadcom VideoCore VI	NVIDIA Maxwell (128 CUDA cores)
RAM	4 GB LPDDR4	4 GB LPDDR4
Almacenamiento	microSD	microSD
Ethernet	✓	✓
Wireless	✓	✗
Bluetooth	✓	✗
AI-Enabled	✗	✓
USB	2x USB 3.0, 2x USB 2.0	4x USB 3.0, 1x USB 2.0
Consumo	2 - 5 W	5 - 10 W
Dimensiones	85 mm x 56 mm	70 mm x 45 mm
Coste	aprox. 55 €	aprox. 100 €

También cabe mencionar que existen dispositivos de bajo consumo para acelerar la ejecución de DNN, diseñados para funcionar con Raspberry Pi, como el Intel Movidius Neural Compute Stick[5]. Esta opción tiene menos capacidad de cómputo que la alternativa Jetson y ofrece menos ancho de banda de información (al ser un dispositivo USB), teniendo un precio combinado similar al de la Jetson Nano. Sin embargo, en los experimentos realizados, el principal inconveniente ha sido que: (a) las bibliotecas para el dispositivo de Intel no ofrecen el nivel de integración que NVIDIA ofrece a través de CUDA para los modelos de DNN más populares y (b) que tiene una peor integración con Docker, lo que limita las posibilidades de despliegue. Por el contrario, las bibliotecas JetPack de CUDA son capaces de acceder al hardware GPU desde contenedores Docker, integrándose sin problemas y sin esfuerzo en la arquitectura propuesta. Más que eso,

[5]"Intel Neural Compute Stick 2 (Intel NCS2)", Intel Corporation. (), dirección: `https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html`.

JetPack ofrece implementaciones preentrenadas de muchos modelos DNN populares, de nuevo facilitando el desarrollo de operadores para esta arquitectura.

5.2.3 Medidas de rendimiento

En esta sección, se evalúa experimentalmente el rendimiento de la microcomputadora, cuando se ejecutan diferentes modelos de detección en diversas situaciones: varios escenarios de ejecución de modelos simultáneos, que van desde 1 a 4 instancias de modelo que se ejecutan en el mismo dispositivo Jetson Nano y ejecución independiente de diferentes modelos en un solo dispositivo (con acceso exclusivo y completo a los recursos de hardware). Los modelos seleccionados han sido elegidos teniendo en cuenta su interés para el caso de uso en cuestión: detección y clasificación de objetos y reconocimiento facial.

Para el primer escenario de pruebas, la Figura 5.8 representa el tiempo de detección promedio para 1000 detecciones consecutivas, cuando se ejecutan simultáneamente en el mismo dispositivo. Se han ejecutado entre una y cuatro instancias de tareas simultáneas, siempre sobre la misma secuencia de vídeo. La detección se realizó usando el modelo DNN de *inception v2* sobre una simulación de transmisión RTP desde una cámara capaz de capturar vídeo de 1280x720p de resolución, que transmitía a 30 fps[6] con el códec H264 a una tasa de bits de 1500 kbps.

El experimento alcanzó un máximo de 4 tareas simultáneas, siguiendo las limitaciones marcadas según las especificaciones del fabricante. La observación de la utilización de memoria RAM (random access memory, memoria de acceso aleatorio) durante los experimentos, revela que se alcanza un máximo de 3,9 de los 4 GB disponibles en la Jetson Nano, al ejecutar 4 tareas simultáneas, explicando el límite impuesto por el fabricante.

Para el segundo escenario de pruebas, se compara el tiempo de detección promedio para 1000 detecciones consecutivas de diferentes modelos DNN, ejecutados en las mismas condiciones experimentales que antes (Jetson Nano y flujo de cámara). Esta vez, solo se ejecutó un único operador a la vez, lo que garantiza el acceso total a los recursos del dispositivo. La figura 5.9 representa el promedio de los tiempos de detección de *inception v2*, *mobilenet v1* y *facenet 120*.

Como se aprecia, la mayoría de ellos están por debajo de los 55 ms, lo que proporcionaría una tasa de detección de alrededor de 18 fps para nuestra transmisión y hasta 28 fps para el caso de *mobilenet v1*, que promedió menos de 35 ms de tiempo de detección. Solo dos de los modelos de detección que probamos, mostrados en la Figura 5.10, tardaron en promedio más de 100 ms: *multiped-500* y *pednet-100*. Incluso en este caso, eso significaría una tasa de detección efectiva de 10 fps, o aproximadamente realizar la detección en uno de cada tres fotogramas provenientes de la cámara.

[6]Fotogramas por segundo

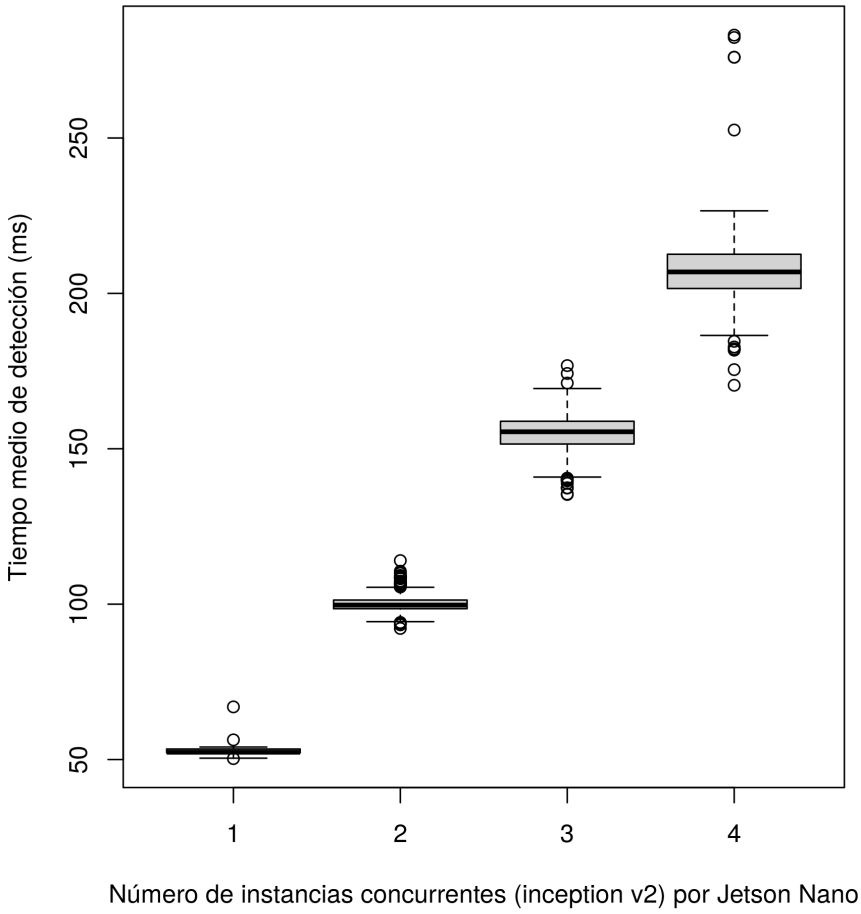


Figura 5.8: Tiempo de detección medio en cargas multitarea en Jetson Nano

5.3 Conclusiones y trabajo futuro

Como se muestra en la Sección 5.2, el trabajo presentado en este documento aborda con éxito la mayoría de los problemas presentados al comienzo de este capítulo. Al utilizar dispositivos IoT habilitados con aceleración hardware del procesado de imágenes, podemos trasladar exitosamente el trabajo otrora realizado en la nube, al borde y con ello mejorar la seguridad, reducir la latencia y hacer un uso más eficiente de los recursos de la red, al mismo tiempo que ganamos flexibilidad y reducimos las preocupaciones relacionadas con las regulaciones de privacidad [222], [223].

La arquitectura propuesta no solo aprovecha el estándar NGSI-LD ampliamente aceptado, sino que también se basa en implementaciones existentes, como el framework FogFlow, proporcionando una solución confiable e interoperable que se puede integrar fácilmente en muchas soluciones existentes.

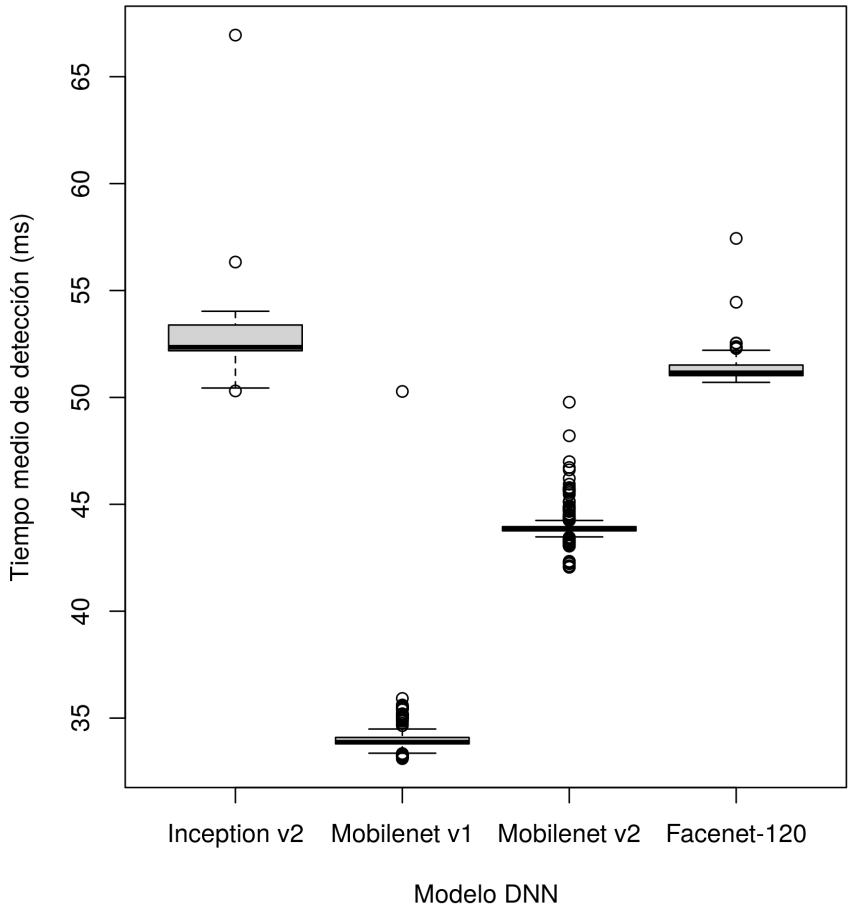


Figura 5.9: Tiempo de detección promedio de inception, mobilenet y facenet en Jetson Nano

A partir de aquí se puede continuar el desarrollo del sistema, implementando varios operadores que permitirán realizar tareas habilitadas para DNN como detección y clasificación de imágenes y recolectando datos experimentales sobre su desempeño. También se pueden explorar más ejemplos de casos de uso, como escenarios y aplicaciones relacionados con edificios inteligentes y ciudad inteligente.

Finalmente, hay un tema especialmente interesante, omitido en este trabajo: la exploración de las implicaciones de privacidad de la implementación de algoritmos especialmente sensibles como los de reconocimiento facial, para los cuales ya se ha considerado la aplicación de enfoques criptográficos avanzados como el cifrado homomórfico, con el fin de proteger las firmas biométricas que representan las caras identificadas, al tiempo que permite realizar cálculos de distancia sobre ellas.

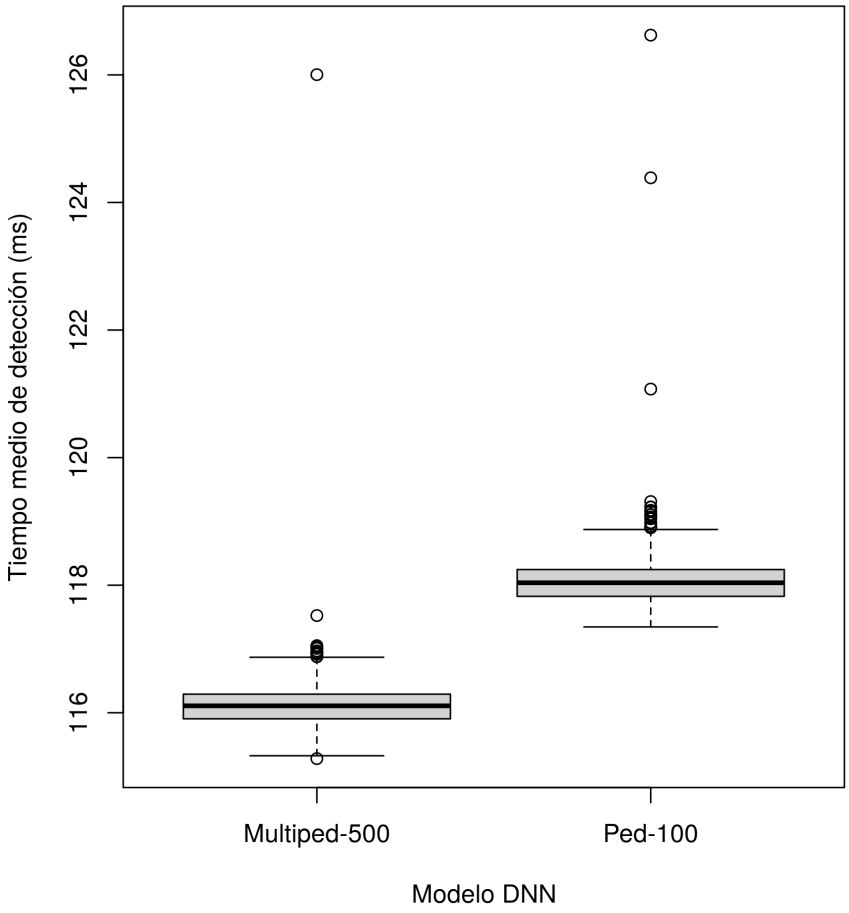


Figura 5.10: Tiempo medio de detección en Jetson Nano de Pednet y multiped

CAPÍTULO 6

CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

El trabajo en esta tesis se ha enfocado en varias áreas o aspectos estratégicos para el desarrollo de las arquitecturas de datos para IoT: la necesidad de interoperabilidad en interfaces y modelos de datos, la seguridad y privacidad en los datos compartidos y el procesado ubicuo de la información a través de la distribución/orquestación de tareas entre el borde y la nube.

Estos aspectos afectan directamente al desarrollo potencial del IoT, respondiendo a las necesidades actuales de integrar información y sistemas diversos a todos los niveles, desde el dispositivo hasta la plataforma de datos, buscando el beneficio en sinergias e interacciones innovadoras resultantes de poder compartir datos entre diferentes ámbitos.

La economía basada en los datos presenta oportunidades de desarrollo en plataformas para compraventa de información producida en IoT, así como nuevas estrategias para implementar buscadores de datos, también beneficiadas por la implementación de arquitecturas interoperables tanto en interfaces como en los datos, así como por el uso de tecnologías de la web semántica.

La seguridad y privacidad: aspectos bien conocidos en la informática y ampliamente trabajados históricamente, presentan nuevos retos cuando tratamos de gestionar las nuevas formas de compartir y gestionar la información, que toma un papel central convirtiéndose en la nueva protagonista. Hasta ahora hemos definido implícitamente los elementos de seguridad de los datos, en función de su punto de acceso. Con la llegada de IoT, los datos fluyen y son consultados a través de diferentes sistemas, perdiendo dicha información implícita y revelando la necesidad de nuevas soluciones.

Por último, la cantidad de información producida por dispositivos IoT impone una elevada

carga en las redes de comunicaciones de cara a su transporte hacia los centros de datos en la nube, donde históricamente ha sucedido el procesado y almacenado de los datos. Esta estrategia queda manifiestamente obsoleta ante el creciente desarrollo de nuevos dispositivos, capaces de realizar tareas computacionalmente complejas, de manera eficiente, en el borde; reduciendo de esta forma la latencia impuesta por el viaje de ida y vuelta a través de las redes de comunicaciones, reduciendo la carga del sistema de comunicaciones e incluso ofreciendo nuevas posibilidades de cara al tratamiento de información privada, evitando su envío a entidades de procesado de datos en la nube.

a Aunque en el estado del arte se han estudiado ontologías de seguridad en el contexto de IoT, ninguna de ellas aborda específicamente la anotación de aspectos funcionales de seguridad de datos desde el punto de vista de los propios datos. A resultas de la necesidad de una nueva ontología que permita compartir y consumir datos en el contexto de IoT, en el Capítulo 3 se describe el desarrollo de la ontología DS4IoT utilizando la metodología NeOn. DS4IoT es una ontología liviana que representa conceptos actuales y novedosos en seguridad de datos en IoT, favoreciendo el uso de conocimiento implícito sobre explícito. También se presenta una prueba de concepto que muestra el mapeo al modelo de datos de NGSI-LD y una aplicación de la ontología en el proyecto IoTCrawler, que define un framework modular para la creación de buscadores de datos para IoT y donde la anotación de la información con conceptos de seguridad de los datos y privacidad resulta de gran importancia de cara al cumplimiento de normativas como el Reglamento General de Protección de Datos (General Data Protection Regulation, GDPR).

b En el Capítulo 4 se propone una arquitectura modular, interoperable y segura para la construcción de SHEMS en el contexto de una casa inteligente. La arquitectura considera la gestión de la energía, las interacciones del prosumidor con la red, la generación local, la optimización de la acumulación y la seguridad de los datos. Se utiliza el estándar NGSI-LD como base semántica y para la orquestación, lo que permite la reutilización de implementaciones existentes de componentes FIWARE. La base ontológica del modelo de información se establece a partir de ontologías existentes, con DABGEO como ontología base. Se valida la propuesta mediante un banco de pruebas en un hogar real, integrando instalaciones de producción y almacenamiento de energía solar en el SHEMS. Finalmente se muestran resultados exitosos en la programación de dispositivos y la reacción a altos consumos de energía, demostrando la viabilidad de la arquitectura propuesta.

c Finalmente, el Capítulo 5 aborda con éxito la orquestación de tareas utilizando recursos en el borde y la nube. Mediante el uso de dispositivos IoT con aceleración de hardware, se logra trasladar el procesamiento de imágenes de la nube al borde, mejorando la seguridad, reduciendo la latencia y optimizando los recursos de la red. Además, se gana flexibilidad y se abordan preocupaciones relacionadas con la privacidad. La arquitectura propuesta aprovecha

el estándar NGSI y se basa en implementaciones existentes, como el framework FogFlow, lo que garantiza una solución confiable e interoperable que puede integrarse fácilmente en diversas soluciones existentes.

Para terminar podemos concluir, que en base a lo mostrado en capítulos anteriores, el uso de aproximaciones inspiradas o directamente heredadas de la web semántica, ha resultado una decisión clave que ha permitido satisfacer de manera holística la interoperabilidad de interfaces y de modelos de datos, la seguridad y privacidad de la información y el desarrollo de estrategias para la orquestación de tareas entre el borde y la nube; habilitando la reutilización de tecnologías existentes y probadas, permitiendo el rápido desarrollo de soluciones y su rápida y fácil integración en sistemas existentes.

También queda de manifiesto, no obstante, la dificultad de atacar de manera individual estos problemas, debido a la fuerte interdependencia de los aspectos mencionados; así como la necesidad de seguir trabajando en esta línea, desarrollando nuevas soluciones capaces de ofrecer mejoras en el procesado ubicuo de la información, atendiendo a la privacidad y seguridad de los datos así como a la necesidad de poder interconectar sistemas y explotar los datos contenidos.

6.2 Trabajo futuro

Como nota final a este trabajo, se proponen las siguientes líneas de trabajo futuro en función de las contribuciones presentadas:

a En la ontología de DS4IoT una de las líneas de trabajo futuras es su revisión y actualización como respuesta al desarrollo de nuevas tecnologías y estándares, en especial los conceptos sobre regulaciones y certificación. Adicionalmente se puede trabajar en otros aspectos de la seguridad de datos en el campo de IoT, como el ciclo de vida de los datos.

b Sobre la contribución de la arquitectura para sistemas de gestión energética en el hogar inteligente, se puede trabajar en nuevas propuestas de SHEMS multifacéticos que optimicen sistemas complejos al controlar la acumulación, la generación de energía, las estrategias de demanda y respuesta y el intercambio de datos con la SG. También proporciona la oportunidad de desarrollar e implementar componentes específicos, como por ejemplo pasarelas energéticas para diferentes proveedores de energía. Además, se plantea la posibilidad de crear frameworks e implementaciones específicas de SHEMS que se integren fácilmente con otras soluciones existentes, facilitando su implementación por parte de usuarios no expertos y promoviendo la adopción generalizada de estrategias avanzadas de DR.

c Por último, partiendo de la solución propuesta para la orquestación de tareas de procesado de imagen entre el borde y la nube, el desarrollo del sistema puede continuar implementando diversos operadores para realizar tareas basadas en DNN, como la detección y clasificación de

imágenes, además de recolectar datos experimentales para evaluar su rendimiento. También se pueden explorar más casos de uso, como aplicaciones relacionadas con edificios inteligentes y ciudades inteligentes. Por último, un tema de gran interés que no se abordó en este trabajo es la exploración de las implicaciones de privacidad en la implementación de algoritmos sensibles, como el reconocimiento facial. Para proteger las firmas biométricas que representan las caras identificadas, se ha considerado el uso de enfoques criptográficos avanzados como el cifrado homomórfico, que permite realizar cálculos de distancia sin comprometer la privacidad de los datos.

CAPÍTULO 7

PUBLICACIONES DERIVADAS DE LA TESIS DOCTORAL

Artículos de revista

[1] P. Gonzalez-Gil, J. A. Martinez y A. Skarmeta, "A Prosumer-Oriented, Interoperable, Modular and Secure Smart Home Energy Management System Architecture", *Smart Cities*, vol. 5, n.º 3, págs. 1054-1078, ago. de 2022, ISSN: 2624-6511. DOI: 10.3390/smartcities5030053. dirección: https://www.mdpi.com/2624-6511/5/3/53.

[2] P. Gonzalez-Gil, A. Robles-Enciso, J. A. Martínez y A. F. Skarmeta, "Architecture for Orchestrating Dynamic DNN-Powered Image Processing Tasks in Edge and Cloud Devices", *IEEE Access*, vol. 9, págs. 107 137-107 148, jul. de 2021, ISSN: 2169-3536. DOI: 10.1109/access.2021.3101306. dirección: https://ieeexplore.ieee.org/document/9502087/.

[3] T. Iggena, E. Bin Ilyas, M. Fischer et al., "IoTCrawler: Challenges and Solutions for Searching the Internet of Things", *Sensors*, vol. 21, n.º 5, feb. de 2021, ISSN: 1424-8220. DOI: 10.3390/s21051559. dirección: https://www.mdpi.com/1424-8220/21/5/1559.

[4] P. Gonzalez-Gil, J. A. Martinez y A. F. Skarmeta, "Lightweight Data-Security Ontology for IoT", *Sensors*, vol. 20, n.º 3, feb. de 2020, ISSN: 1424-8220. DOI: 10.3390/s20030801. dirección: https://www.mdpi.com/1424-8220/20/3/801.

Congresos

[5] P. Gonzalez-Gil, R. Marin-Perez, A. González-Vidal, A. P. Ramallo-González y A. F. Skarmeta, "Interoperable and Intelligent Architecture for Smart Buildings", en *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, IEEE, ago. de

2021, págs. 359-364. DOI: 10.1109/SmartIoT52359.2021.00067. dirección: https://ieeexplore.ieee.org/document/9555845.

[6] P. Gonzalez-Gil, A. F. Skarmeta y J. A. Martinez, "The Security Framework of Fed4IoT", en *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*, ép. CCIoT '20, Virtual Event, Japan: Association for Computing Machinery (ACM), nov. de 2020, págs. 1-6, ISBN: 9781450381314. DOI: 10.1145/3417310.3431396.

[7] P. Gonzalez-Gil, A. F. Skarmeta y J. A. Martinez, "Towards an Ontology for IoT Context-Based Security Evaluation", en *2019 Global IoT Summit (GIoTS)*, IEEE, jun. de 2019, págs. 1-6. DOI: 10.1109/GIOTS.2019.8766400. dirección: https://ieeexplore.ieee.org/document/8766400.

Capítulos de libro

[8] P. Gonzalez-Gil, J. A. Martinez, H. T. T. Truong, A. Sforzin y A. F. Skarmeta, "IoTCrawler. Managing Security and Privacy for IoT", en *Security and Privacy in the Internet of Things: Challenges and Solutions* (Ambient Intelligence and Smart Environments), Ambient Intelligence and Smart Environments. IOS Press, 2020, vol. 27, págs. 167-181, ISBN: 978-1-64368-052-1. DOI: 10.3233/AISE200011. dirección: https://ebooks.iospress.nl/volumearticle/53860.

Referencias

[9] "Global Sustainable Development Report 2023, Advance", Unedited Version, ONU, 14 de jun. de 2023. dirección: https://sdgs.un.org/sites/default/files/2023-06/Advance%20unedited%20GSDR%2014June2023.pdf (visitado 27-06-2023).

[10] P. Kasinathan, R. Pugazhendhi, R. M. Elavarasan et al., "Realization of Sustainable Development Goals with Disruptive Technologies by Integrating Industry 5.0, Society 5.0, Smart Cities and Villages", *Sustainability*, vol. 14, n.º 22, pág. 15 258, 17 de nov. de 2022. DOI: 10.3390/su142215258.

[11] M. Andronie, G. Lăzăroiu, M. Iatagan, I. Hurloiu e I. Dijmărescu, "Sustainable Cyber-Physical Production Systems in Big Data-Driven Smart Urban Economy: A Systematic Literature Review", *Sustainability*, vol. 13, n.º 2, pág. 751, 14 de ene. de 2021. DOI: 10.3390/su13020751.

[12] P. Sethi y S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications", *Journal of Electrical and Computer Engineering*, vol. 2017, págs. 1-25, 2017. DOI: 10.1155/2017/9324035.

[13] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun y H.-Y. Du, "Research on the architecture of Internet of Things", en *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, IEEE, ago. de 2010. DOI: 10.1109/icacte.2010.5579493.

[14] M. Noura, M. Atiquzzaman y M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges", *Mobile Networks and Applications*, vol. 24, n.º 3, págs. 796-809, 21 de jul. de 2018. DOI: 10.1007/s11036-018-1089-9.

[15] "FIWARE - Open APIs for Open Minds". Sitio web de FIWARE, FIWARE Foundation. (2022), dirección: https://www.fiware.org/ (visitado 18-02-2023).

[16] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng y E. Kovacs, "A Standard-Based Open Source IoT Platform: FIWARE", *IEEE Internet of Things Magazine*, vol. 2, n.º 3, págs. 12-18, sep. de 2019. DOI: 10.1109/iotm.0001.1800022.

[17] "oneM2M The IoT Standard, Published Specifications". Sitio web de oneM2M, oneM2M. (), dirección: https://onem2m.org/technical/published-specifications (visitado 18-02-2023).

[18] S. K. Datta, A. Gyrard, C. Bonnet y K. Boudaoud, "oneM2M architecture based user centric IoT application development", en *2015 3rd International Conference on Future Internet of Things and Cloud*, IEEE, IEEE, ago. de 2015. DOI: 10.1109/ficloud.2015.7.

[19] H. Park, H. Kim, H. Joo y J. Song, "Recent advancements in the Internet-of-Things related standards: A oneM2M perspective", *ICT Express*, vol. 2, n.º 3, págs. 126-129, sep. de 2016. DOI: 10.1016/j.icte.2016.08.009.

[20] J. Swetina, G. Lu, P. Jacobs, F. Ennesser y J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M", *IEEE Wireless Communications*, vol. 21, n.º 3, págs. 20-26, jun. de 2014. DOI: 10.1109/mwc.2014.6845045.

[21] "FIWARE Catalogue". Sitio web del catálogo de componentes FIWARE, FIWARE Foundation. (2022), dirección: https://www.fiware.org/catalogue/ (visitado 06-03-2023).

[22] J. M. Cantera Fonseca, P. Guillemin, M. Bauer et al., "NGSI-LD Information Model, ETSI GS CIM 009 V1.1.1", ETSI ISG CIM, inf. téc., ene. de 2019. dirección: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf (visitado 18-02-2023).

[23] T. Kindberg, J. Barton, J. Morgan et al., "People, places, things: Web presence for the real world", IEEE Comput. Soc, 2000, págs. 19-28. DOI: 10.1109/mcsa.2000.895378.

[24] D. Guinard, V. Trifa y E. Wilde, "A resource oriented architecture for the Web of Things", en *2010 Internet of Things (IOT)*, IEEE, nov. de 2010. DOI: 10.1109/iot.2010.5678452.

[25] D. Guinard, V. Trifa, F. Mattern y E. Wilde, "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices", en *Architecting the Internet of Things*, Springer Berlin Heidelberg, 2011, págs. 97-129. DOI: 10.1007/978-3-642-19157-2_5.

[26] D. Zeng, S. Guo y Z. Cheng, "The Web of Things: A Survey (Invited Paper)", *Journal of Communications*, vol. 6, n.º 6, sep. de 2011. DOI: 10.4304/jcm.6.6.424-438.

[27] D. Raggett, "The Web of Things: Challenges and Opportunities", *Computer*, vol. 48, n.º 5, págs. 26-32, mayo de 2015. DOI: 10.1109/mc.2015.149.

[28] R. Fielding, R. Taylor, M. Ackerman y D. Rosenblum, "Architectural Styles and the Design of Network-based Software Architectures", Tesis doct., University of California, Irvine, 2000. dirección: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (visitado 26-02-2023).

[29] R. T. Fielding y R. N. Taylor, "Principled design of the modern Web architecture", *ACM Transactions on Internet Technology*, vol. 2, n.º 2, págs. 115-150, mayo de 2002. DOI: 10.1145/514183.514185.

[30] G. Steve Harris, A. Seaborne y E. Prud'hommeaux, "SPARQL 1.1 Query Language", W3C Recommendation, World Wide Web Consortium (W3C), inf. téc., 21 de mar. de 2013. dirección: https://www.w3.org/TR/sparql11-query/ (visitado 26-02-2023).

[31] G. Schreiber e Y. Raimond, "RDF 1.1 Primer", W3C Working Group Note, World Wide Web Consortium (W3C), inf. téc., 24 de jun. de 2014. dirección: https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/ (visitado 26-02-2023).

[32] B. Mcbride, "The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS", en *Handbook on Ontologies*, Springer Berlin Heidelberg, 2004, págs. 51-65. DOI: 10.1007/978-3-540-24750-0_3.

[33] D. Brickley y R. V. Guha, "RDF Schema 1.1", W3C Recommendation, World Wide Web Consortium (W3C), inf. téc., 25 de feb. de 2014. dirección: https://www.w3.org/TR/rdf-schema/ (visitado 26-02-2023).

[34] J. Bao, D. Calvanese, B. Cuenca Grau et al., "OWL 2 Web Ontology Language Document Overview (Second Edition)", W3C Recommendation, World Wide Web Consortium (W3C), inf. téc., 11 de dic. de 2012. dirección: https://www.w3.org/TR/owl2-overview/ (visitado 26-02-2023).

[35] "World Wide Web Consortium (W3C) - making the Web work". Sitio web de W3C, World Wide Web Consortium (W3C). (2023), dirección: https://www.w3.org (visitado 06-03-2023).

[36] M. Lagally, R. Matsukura, M. McCool et al., "Web of Things (WoT) Architecture 1.1. W3C Candidate Recommendation Snapshot 19 January 2023.", W3C Candidate Recommendation Snapshot, World Wide Web Consortium (W3C), inf. téc., 19 de ene. de 2023. dirección: https://www.w3.org/TR/wot-architecture/ (visitado 19-02-2023).

[37] "W3C Web of Things". Sitio web de W3C para Web of Things, World Wide Web Consortium (W3C). (2023), dirección: https://www.w3.org/WoT/ (visitado 06-03-2023).

[38] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, P.-A. Champin y N. Lindström, "JSON-LD 1.1, A JSON-based Serialization for Linked Data", W3C Recommendation 16 July 2020, World Wide Web Consortium (W3C), inf. téc., 16 de jul. de 2020. dirección: https://www.w3.org/TR/json-ld/ (visitado 21-07-2023).

[39] B. Mahapatra y A. Nayyar, "Home energy management system (HEMS): concept, architecture, infrastructure, challenges and energy management schemes", *Energy Systems*, vol. 13, n.º 3, págs. 643-669, nov. de 2019, ISSN: 1868-3967. DOI: 10.1007/s12667-019-00364-w.

[40] M. Beaudin y H. Zareipour, "Home energy management systems: A review of modelling and complexity", *Renewable and Sustainable Energy Reviews*, vol. 45, págs. 318-335, mayo de 2015. DOI: 10.1016/j.rser.2015.01.046.

[41] A. Q. H. Badar y A. Anvari-Moghaddam, "Smart home energy management system – a review", *Advances in Building Energy Research*, vol. 16, n.º 1, págs. 118-143, ago. de 2020, ISSN: 1756-2201. DOI: 10.1080/17512549.2020.1806925.

[42] B. Lashkari, Y. Chen y P. Musilek, "Energy Management for Smart Homes—State of the Art", *Applied Sciences*, vol. 9, n.º 17, pág. 3459, ago. de 2019, ISSN: 2076-3417. DOI: 10.3390/app9173459. dirección: https://www.mdpi.com/2076-3417/9/17/3459.

[43] B. Zhou, W. Li, K. W. Chan et al., "Smart home energy management systems: Concept, configurations, and scheduling strategies", *Renewable and Sustainable Energy Reviews*, vol. 61, págs. 30-40, ago. de 2016. DOI: 10.1016/j.rser.2016.03.047.

[44] J. Mongay Batalla y F. Gonciarz, "Deployment of smart home management system at the edge: mechanisms and protocols", *Neural Computing and Applications*, vol. 31, n.º 5, págs. 1301-1315, 23 de mayo de 2018. DOI: 10.1007/s00521-018-3545-7.

[45] "Home Assistant - awaken your home". Sitio web de Home Assistant. (), dirección: https://www.home-assistant.io (visitado 18-03-2023).

[46] "Domoticz - control at your fingertips". Sitio web de Domoticz. (), dirección: https://www.domoticz.com (visitado 18-03-2023).

[47] "openHAB - empowering the smart home". Sitio web de openHAB. (), dirección: https://www.openhab.org (visitado 18-03-2023).

[48] P. Singh, M. Masud, M. S. Hossain y A. Kaur, "Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid", *Computers & Electrical Engineering*, vol. 93, pág. 107 209, jul. de 2021, ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2021.107209.

[49] C. Milchram, G. van de Kaa, N. Doorn y R. Künneke, "Moral Values as Factors for Social Acceptance of Smart Grid Technologies", *Sustainability*, vol. 10, n.º 8, pág. 2703, ago. de 2018, ISSN: 2071-1050. DOI: 10.3390/su10082703. dirección: http://www.mdpi.com/2071-1050/10/8/2703.

[50] I. Machorro-Cano, G. Alor-Hernández, M. A. Paredes-Valverde, L. Rodríguez-Mazahua, J. L. Sánchez-Cervantes y J. O. Olmedo-Aguirre, "HEMS-IoT: A Big Data and Machine Learning-Based Smart Home System for Energy Saving", *Energies*, vol. 13, n.º 5, pág. 1097, mar. de 2020, ISSN: 1996-1073. DOI: 10.3390/en13051097. dirección: https://www.mdpi.com/1996-1073/13/5/1097.

[51] H. Elshaafi, M. Vinyals, I. Grimaldi y S. Davy, "Secure Automated Home Energy Management in Multi-Agent Smart Grid Architecture", *Technology and Economics of Smart Grids and Sustainable Energy*, vol. 3, n.º 1, abr. de 2018. DOI: 10.1007/s40866-018-0042-0.

[52] "OSGi Working Group, The Dynamic Module System for Java". Sitio web de OSGi, Eclipse Foundation. (), dirección: https://www.osgi.org (visitado 11-03-2023).

[53] J. L. Hippolyte, S. Howell, B. Yuce et al., "Ontology-based demand-side flexibility management in smart grids using a multi-agent system", en *2016 IEEE International Smart Cities Conference (ISC2)*, IEEE, sep. de 2016, ISBN: 9781509018451. DOI: 10.1109/ISC2.2016.7580828. dirección: http://ieeexplore.ieee.org/document/7580828/.

[54] "Universal Smart Energy Framweork (USEF)". (jun. de 2021), dirección: https://www.usef.energy (visitado 22-04-2023).

[55] F. Bliek, A. Backers, M. Broekmans et al., *An introduction to the Universal Smart Energy Framework*, en. USEF Foundation, 2014. DOI: 10.13140/2.1.2275.1046.

[56] J. Zhang, Q. Li y E. M. Schooler, "iHEMS: An information-centric approach to secure home energy management", en *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, nov. de 2012, págs. 217-222, ISBN: 9781467309110. DOI: 10.1109/smartgridcomm.2012.6485986.

[57] N. Shah, K. M. Chao, T. Zlamaniec y A. Matei, "Ontology for Home Energy Management Domain", en *Communications in Computer and Information Science*, PART 2, vol. 167 CCIS, Springer Berlin Heidelberg, 2011, págs. 337-347, ISBN: 9783642220265. DOI: 10.1007/978-3-642-22027-2_28.

[58] A. Rossello-Busquet, J. Soler y L. Dittmann, "A Novel Home Energy Management System Architecture", en *Proceedings - 2011 UKSim 13th International Conference on Modelling and Simulation, UKSim 2011*, IEEE, mar. de 2011, págs. 387-392, ISBN: 9780769543765. DOI: 10.1109/uksim.2011.80. dirección: http://ieeexplore.ieee.org/document/5754251/.

[59] C. Reinisch, M. J. Kofler, F. Iglesias y W. Kastner, "Thinkhome energy efficiency in future smart homes", *Eurasip Journal on Embedded Systems*, vol. 2011, n.º 1, pág. 104 617, 2011, ISSN: 1687-3955. DOI: 10.1155/2011/104617. dirección: http://jes.eurasipjournals.com/content/2011/1/104617.

[60] Y. Wu, "Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing", *IEEE Internet of Things Journal*, vol. 8, n.º 16, págs. 12 792-12 805, ago. de 2021, ISSN: 2327-4662. DOI: 10.1109/jiot.2020.3014845.

[61] F. Bonomi, "Connected vehicles, the internet of things, and fog computing", en *The eighth ACM international workshop on vehicular inter-networking (VANET), Las Vegas, USA*, sn, 2011, págs. 13-15.

[62] F. Bonomi, R. Milito, J. Zhu y S. Addepalli, "Fog computing and its role in the internet of things", en *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, ACM, ago. de 2012. DOI: 10.1145/2342509.2342513.

[63] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu y M. Rovatsos, "Fog Orchestration for Internet of Things Services", *IEEE Internet Computing*, vol. 21, n.º 2, págs. 16-24, mar. de 2017. DOI: 10.1109/mic.2017.36.

[64] S. Hoque, M. S. D. Brito, A. Willner, O. Keil y T. Magedanz, "Towards Container Orchestration in Fog Computing Infrastructures", *Proceedings - International Computer Software and Applications Conference*, vol. 2, págs. 294-299, jul. de 2017, ISSN: 0730-3157. DOI: 10.1109/compsac.2017.248.

[65] A. Willner, "OpenIoTFog: Eine anbieterunabhängige Verwaltungsschale für Industrie-4.0-Komponenten", en *Tagung Industrie 4.0 - "Safety und Security - Mit Sicherheit gut vernetzt"2017*, mayo de 2017. dirección: https://publica.fraunhofer.de/handle/publica/404652 (visitado 21-04-2023).

[66] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa y A. Kitazawa, "FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities", *IEEE Internet of Things Journal*, vol. 5, n.º 2, págs. 696-707, abr. de 2018, ISSN: 2327-4662. DOI: 10.1109/jiot.2017.2747214.

[67] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall y M. Zhao, "Standards-Based Worldwide Semantic Interoperability for IoT", *IEEE Communications Magazine*, vol. 54, n.º 12, págs. 40-46, dic. de 2016, ISSN: 0163-6804. DOI: 10.1109/mcom.2016.1600460cm.

[68] T. Akidau, R. Bradshaw, C. Chambers et al., "The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing", *Proceedings of the VLDB Endowment*, vol. 8, págs. 1792-1803, 2015. dirección: https://research.google/pubs/pub43864/ (visitado 21-04-2023).

[69] A. Rocha Neto, T. P. Silva, T. V. Batista, F. C. Delicato, P. F. Pires y F. Lopes, "An Architecture for Distributed Video Stream Processing in IoMT Systems", *Open Journal of Internet Of Things (OJIOT)*, vol. 6, n.º 1, págs. 89-104, 2020, ISSN: 2364-7108. dirección: https://www.ronpub.com/ojiot/OJIOT_2020v6i1n09_Neto.html.

[70] S. Abdel Magid, F. Petrini y B. Dezfouli, "Image classification on IoT edge devices: profiling and modeling", *Cluster Computing*, vol. 23, n.º 2, págs. 1025-1043, ago. de 2019, ISSN: 1573-7543. DOI: 10.1007/s10586-019-02971-9.

[71] E. S. Lage, R. L. Santos, S. M. T. Junior y F. Andreotti, "Low-Cost IoT Surveillance System Using Hardware-Acceleration and Convolutional Neural Networks", en *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, IEEE, abr. de 2019, págs. 931-936, ISBN: 9781538649800. DOI: 10.1109/wf-iot.2019.8767325.

[72] "Cisco Annual Internet Report (2018–2023), White paper", CISCO, inf. téc., 9 de mar. de 2020. dirección: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf (visitado 19-03-2023).

[73] "Ericsson Mobility Report", Ericsson, inf. téc., nov. de 2022. dirección: https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf (visitado 19-03-2023).

[74] E. Fazeldehkordi y T.-M. Grønli, "A Survey of Security Architectures for Edge Computing-Based IoT", *IoT*, vol. 3, n.º 3, págs. 332-365, jun. de 2022. DOI: 10.3390/iot3030019.

[75] C. Kolias, G. Kambourakis, A. Stavrou y J. Voas, "DDoS in the IoT: Mirai and Other Botnets", *Computer*, vol. 50, n.º 7, págs. 80-84, 2017, ISSN: 0018-9162. DOI: 10.1109/mc.2017.201.

[76] N. Garcia, T. Alcaniz, A. González-Vidal, J. B. Bernabe, D. Rivera y A. Skarmeta, "Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial Intelligence", *Journal of Network and Computer Applications*, vol. 173, pág. 102 871, ene. de 2021. DOI: 10.1016/j.jnca.2020.102871.

[77] Y. H. Hwang, "IoT Security & Privacy: Threats and Challenges", en *Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security*, ACM, abr. de 2015. DOI: 10.1145/2732209.2732216.

[78] F. Meneghello, M. Calore, D. Zucchetto, M. Polese y A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices", *IEEE Internet of Things Journal*, vol. 6, n.º 5, págs. 8182-8201, oct. de 2019, ISSN: 2327-4662. DOI: 10.1109/jiot.2019.2935189.

[79] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal y B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures", *IEEE Access*, vol. 7, págs. 82 721-82 743, 2019, ISSN: 2169-3536. DOI: 10.1109/access.2019.2924045.

[80] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum y N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations", *IEEE Communications Surveys & Tutorials*, vol. 21, n.º 3, págs. 2702-2733, 2019, ISSN: 1553-877X. DOI: 10.1109/comst.2019.2910750.

[81] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen y S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities", en *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, IEEE, nov. de 2014, ISBN: 9781479968336. DOI: 10.1109/soca.2014.58.

[82] M. Abomhara y G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues", en *2014 international conference on privacy and security in mobile systems (PRISMS)*, IEEE, mayo de 2014, págs. 1-8. DOI: 10.1109/prisms.2014.6970594.

[83] F. A. Alaba, M. Othman, I. A. T. Hashem y F. Alotaibi, "Internet of Things security: A survey", *Journal of Network and Computer Applications*, vol. 88, págs. 10-28, jun. de 2017. DOI: 10.1016/j.jnca.2017.04.002.

[84] S. Wachter, "Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR", *Computer Law and Security Review*, vol. 34, n.º 3, págs. 436-449, jun. de 2018, ISSN: 0267-3649. DOI: 10.1016/j.clsr.2018.02.002.

[85] J. L. Hernandez-Ramos, J. A. Martinez, V. Savarino et al., "Security and Privacy in Internet of Things-Enabled Smart Cities: Challenges and Future Directions", *IEEE Security & Privacy*, vol. 19, n.º 1, págs. 12-23, ene. de 2021. DOI: 10.1109/msec.2020.3012353.

[86] J. L. Hernandez-Ramos, D. Geneiatakis, I. Kounelis, G. Steri e I. N. Fovino, "Toward a Data-Driven Society: A Technological Perspective on the Development of Cybersecurity and Data-Protection Policies", *IEEE Security & Privacy*, vol. 18, n.º 1, págs. 28-38, ene. de 2020. DOI: 10.1109/msec.2019.2939728.

[87] OECD, *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. OECD, feb. de 2002. DOI: 10.1787/9789264196391-en.

[88] S. Samonas, "The CIA strikes back: redefining confidentiality, integrity and availability in security", *Journal of Information System Security*, vol. 10, n.º 1, 2001, ISSN: 1551-0123. dirección: https://www.proso.com/dl/Samonas.pdf (visitado 25-03-2023).

[89] D. E. Bell y L. J. LaPadula, "Secure computer systems: Mathematical foundations", MITRE CORP BEDFORD MA, inf. téc., 1 de nov. de 1973. dirección: https://apps.dtic.mil/sti/citations/AD0770768 (visitado 06-04-2023).

[90] R. Sandhu, E. Cope, H. Feinstein y C. Youman, "Role-based access control models", *Computer*, vol. 29, n.º 2, págs. 38-47, 1996. DOI: 10.1109/2.485845.

[91] D. Ferraiolo, J. Cugini y D. R. Kuhn, "Role-Based Access Control (RBAC): Features and Motivations", en *Proceedings of 11th annual computer security application conference*, 1995, págs. 241-48. dirección: https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1995/12/15/role-based-access-control-rbac-features-and-motivations/documents/ferraiolo-cugini-kuhn-95.pdf (visitado 04-04-2023).

[92] E. Yuan y J. Tong, "Attributed based access control (ABAC) for Web services", en *IEEE International Conference on Web Services (ICWS'05)*, IEEE, 2005. DOI: 10.1109/icws.2005.25.

[93] V. C. Hu, D. Ferraiolo, R. Kuhn et al., "Guide to Attribute Based Access Control (ABAC) Definition and Considerations", inf. téc., ene. de 2014. DOI: 10.6028/nist.sp.800-162.

[94] E. Coyne y T. R. Weil, "ABAC and RBAC: Scalable, Flexible, and Auditable Access Management", *IT Professional*, vol. 15, n.º 3, págs. 14-16, mayo de 2013. DOI: 10.1109/mitp.2013.37.

[95] B. Parducci y H. Lockhart, "eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard", OASIS, inf. téc., 22 de ene. de 2013. dirección: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (visitado 11-03-2023).

[96] P. Mazzoleni, B. Crispo, S. Sivasubramanian y E. Bertino, "XACML Policy Integration Algorithms", *ACM Transactions on Information and System Security*, vol. 11, n.º 1, págs. 1-29, feb. de 2008. DOI: 10.1145/1330295.1330299.

[97] D. W. Chadwick, "Federated Identity Management", en *Foundations of Security Analysis and Design V*, Springer Berlin Heidelberg, 2009, págs. 96-120. DOI: 10.1007/978-3-642-03829-7_3.

[98] K. Cameron. "The Laws of Identity". (2005), dirección: https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf (visitado 12-04-2023).

[99] J. Hughes, S. Cantor, J. Hodges et al., "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard", OASIS, inf. téc., 15 de mar. de 2005. dirección: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf (visitado 15-04-2023).

[100] D. Hardt, "The OAuth 2.0 Authorization Framework", RFC Editor, RFC 6749, oct. de 2012. dirección: http://www.rfc-editor.org/rfc/rfc6749.txt (visitado 15-04-2023).

[101] "How OpenID Connect Works". Sitio web de OpenD Connect, OpenID Foundation. (), dirección: https://openid.net/developers/how-connect-works/ (visitado 21-07-2023).

[102] S. R. Oh e Y. G. Kim, "Development of IoT security component for interoperability", en *2017 13th International Computer Engineering Conference (ICENCO)*, IEEE, dic. de 2017, ISBN: 9781538642665. DOI: 10.1109/icenco.2017.8289760.

[103] P. Mahalle, S. Babar, N. Prasad y R. Prasad, "Identity Management Framework towards Internet of Things (IoT): Roadmap and Key Challenges", en *Recent Trends in Network Security and Applications*, Springer Berlin Heidelberg, 2010, págs. 430-439. DOI: 10.1007/978-3-642-14478-3_43.

[104] J. Bernal Bernabe, J. L. Hernandez-Ramos y A. F. Skarmeta Gomez, "Holistic privacy-preserving identity management system for the internet of things", *Mobile Information Systems*, vol. 2017, págs. 1-20, 2017. DOI: 10.1155/2017/6384186.

[105] J. Camenisch y E. Van Herreweghen, "Design and Implementation of the idemix Anonymous Credential System", en *Proceedings of the 9th ACM conference on Computer and communications security*, ACM, nov. de 2002. DOI: 10.1145/586110.586114.

[106] J. L. Hernández-Ramos, A. J. Jara, L. Marin y A. F. Skarmeta, "Distributed capability-based access control for the internet of things", *Journal of Internet Services and Information Security (JISIS)*, vol. 3, n.º 3/4, págs. 1-16, 2013. DOI: 10.22667/JISIS.2013.11.31.001. dirección: https://isyou.info/jisis/vol3/no34/jisis-2013-vol3-no34-01.pdf.

[107] B. C. Neuman y T. Ts'o, "Kerberos: an authentication service for computer networks", *IEEE Communications Magazine*, vol. 32, n.º 9, págs. 33-38, sep. de 1994. DOI: 10.1109/35.312841.

[108] H. Truong, J. L. Hernández-Ramos, J. A. Martinez et al., "Enabling Decentralized and Auditable Access Control for IoT through Blockchain and Smart Contracts", *Security and Communication Networks*, vol. 2022, B. Bhushan, ed., págs. 1-14, jun. de 2022, ISSN: 1939-0122. DOI: 10.1155/2022/1828747. dirección: https://www.hindawi.com/journals/scn/2022/1828747/.

[109] J. Bethencourt, A. Sahai y B. Waters, "Ciphertext-Policy Attribute-Based Encryption", en *2007 IEEE Symposium on Security and Privacy (SP '07)*, IEEE, mayo de 2007. DOI: 10.1109/sp.2007.11.

[110] S. Pérez, D. Rotondi, D. Pedone, L. Straniero, M. J. Núñez y F. Gigante, "Towards the CP-ABE Application for Privacy-Preserving Secure Data Sharing in IoT Contexts", en *International conference on innovative mobile and internet services in ubiquitous computing*, Springer International Publishing, jul. de 2017, págs. 917-926. DOI: 10.1007/978-3-319-61542-4_93.

[111] D. Puiu, P. Barnaghi, R. Tonjes et al., "CityPulse: Large Scale Data Analytics Framework for Smart Cities", *IEEE Access*, 2016, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2541999.

[112] T. Priebe, W. Dobmeier y N. Kamprath, "Supporting attribute-based access control with ontologies", *Proceedings - First International Conference on Availability, Reliability and Security, ARES 2006*, págs. 465-472, 2006. DOI: 10.1109/ares.2006.127.

[113] T. Finin, A. Joshi, L. Kagal et al., "ROWLBAC - representing role based access control in OWL", *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT*, págs. 73-82, 2008. DOI: 10.1145/1377836.1377849.

[114] L. Costabello, "Docteur en Sciences Context-Aware Access Control and Presentation of Linked Data", Tesis doct., Université de Nice-Sophia Antipolis, 2013.

[115] M. I. Daud, D. Sánchez y A. Viejo, "Ontology-Based Delegation of Access Control: An Enhancement to the XACML Delegation Profile", en *Trust, Privacy and Security in Digital Business*, Springer International Publishing, 2015, págs. 18-29, ISBN: 9783319229058. DOI: 10.1007/978-3-319-22906-5_2.

[116] M. Cristani y R. Cuel, "A Survey on Ontology Creation Methodologies", *International Journal on Semantic Web and Information Systems*, vol. 1, n.º 2, págs. 49-69, abr. de 2005. DOI: 10.4018/jswis.2005040103.

[117] R. Iqbal, M. A. A. Murad, A. Mustapha y N. M. Sharef, "An Analysis of Ontology Engineering Methodologies: A Literature Review", *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, n.º 16, págs. 2993-3000, sep. de 2013. DOI: 10.19026/rjaset.6.3684.

[118] N. F. Noy y D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880", Stanford Knowledge Systems Laboratory, inf. téc., ene. de 2001. dirección: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html (visitado 21-04-2023).

[119] M. A. Musen, "The protégé project", *AI Matters*, vol. 1, n.º 4, págs. 4-12, jun. de 2015. DOI: 10.1145/2757001.2757003.

[120] J. Sarraipa, J. P. M. A. Silva, R. Jardim-Gonçalves y A. A. C. Monteiro, "MENTOR - A methodology for enterprise reference ontology development", en *2008 4th International IEEE Conference Intelligent Systems*, IEEE, sep. de 2008, ISBN: 9781424417391. DOI: 10.1109/is.2008.4670436.

[121] M. Suarez de Figueroa, "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse", Tesis doct., Universidad Politécnica de Madrid. FAcultad de Informática. Departamento de Inteligencia Artificial., jul. de 2010. DOI: 10.20868/upm.thesis.3879.

[122] M. C. Suárez-Figueroa, A. Gómez-Pérez y M. Fernández-López, "The NeOn Methodology for Ontology Engineering", en *Ontology Engineering in a Networked World*, Springer Berlin Heidelberg, dic. de 2011, págs. 9-34, ISBN: 9783642247941. DOI: 10.1007/978-3-642-24794-1_2.

[123] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas y V. Issarny, *A study of existing Ontologies in the IoT-domain*, 2017. DOI: 10.48550/ARXIV.1707.00112.

[124] I. Szilagyi y P. Wira, "Ontologies and Semantic Web for the Internet of Things - a survey", en *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, oct. de 2016. DOI: 10.1109/iecon.2016.7793744.

[125] R. Agarwal, D. G. Fernandez, T. Elsaleh et al., "Unified IoT ontology to enable interoperability and federation of testbeds", en *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, IEEE, dic. de 2016. DOI: 10.1109/wf-iot.2016.7845470.

[126] "Semantic Sensor Network Ontology (SSN)", W3C Semantic Sensor Network Incubator Group. (2005), dirección: https://www.w3.org/2005/Incubator/ssn/ssnx/ssn (visitado 23-04-2023).

[127] M. Compton, P. Barnaghi, L. Bermudez et al., "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group", *SSRN Electronic Journal*, vol. 17, págs. 25-32, 2012. DOI: 10.2139/ssrn.3198991.

[128] W. Wang, S. De, R. Toenjes, E. Reetz y K. Moessner, "A Comprehensive Ontology for Knowledge Representation in the Internet of Things", en *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, IEEE, jun. de 2012. DOI: 10.1109/trustcom.2012.20.

[129] "IoT-A: Internet of Things Architecture". (2023), dirección: https://www.iot-a.eu/ (visitado 15-07-2023).

[130] "Stream Annotation Ontology, Working Draft", University of Surrey. Institute for Communication Systems. (12 de mayo de 2016), dirección: http://iot.ee.surrey.ac.uk/citypulse/ontologies/sao/sao (visitado 15-07-2023).

[131] S. Kolozali, M. Bermudez-Edo, D. Puschmann, F. Ganz y P. Barnaghi, "A Knowledge-Based Approach for Real-Time IoT Data Stream Annotation and Processing", en *2014 IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, IEEE, sep. de 2014, págs. 215-222. DOI: 10.1109/ithings.2014.39.

[132] "SOSA Ontology", Spatial Data on the Web Working Group. (16 de nov. de 2016), dirección: https://www.w3.org/2015/spatial/wiki/SOSA_Ontology (visitado 15-07-2023).

[133] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc y M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators", *Journal of Web Semantics*, vol. 56, págs. 1-10, mayo de 2018. DOI: 10.1016/j.websem.2018.06.003.

[134] M. Bermúdez-Edo, T. Elsaleh, P. Barnaghi y K. Taylor, "IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics", *Personal and Ubiquitous Computing*, vol. 21, n.º 3, págs. 475-487, feb. de 2017. DOI: 10.1007/s00779-017-1010-8.

[135] "W3C Semantic Web Interest Group. Basic Geo (WGS84 lat/long) Vocabulary.", W3C. (2003), dirección: https://www.w3.org/2003/01/geo/ (visitado 15-07-2023).

[136] "IoT-Stream: A Lightweight Ontology for IoT Data Streams", Centre for Vision, Speech and Signal Processing. University of Surrey. (1 de dic. de 2019), dirección: http://iot.ee.surrey.ac.uk/iot-crawler/ontology/iot-stream/ (visitado 15-07-2023).

[137] T. Elsaleh, S. Enshaeifar, R. Rezvani, S. T. Acton, V. Janeiko y M. Bermudez-Edo, "IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams and Its Use with Data Analytics and Event Detection Services", *Sensors*, vol. 20, n.º 4, pág. 953, feb. de 2020. DOI: 10.3390/s20040953.

[138] D. Bonino y F. Corno, "DogOnt - Ontology Modeling for Intelligent Domotic Environments", en *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, págs. 790-803. DOI: 10.1007/978-3-540-88564-1_51.

[139] "ThinkHome ontology". (), dirección: https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/ (visitado 22-04-2023).

[140] M. J. Kofler, C. Reinisch y W. Kastner, "A semantic representation of energy-related information in future smart homes", *Energy and Buildings*, vol. 47, págs. 169-179, abr. de 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[141] "BONSAI: Smart Building Ontology for Ambient Intelligence". Sitio web de la ontología BONSAI, Intelligent Systems laboratory. Aristotle University of Thessaloniki. (), dirección: http://lpis.csd.auth.gr/ontologies/ontolist.html#bonsai (visitado 22-04-2023).

[142] T. G. Stavropoulos, D. Vrakas, D. Vlachava y N. Bassiliades, "BOnSAI: A smart building ontology for ambient intelligence", en *ACM International Conference Proceeding Series*, New York, New York, USA: ACM Press, 2012, págs. 1-12, ISBN: 9781450309158. DOI: 10.1145/2254129.2254166. dirección: http://dl.acm.org/citation.cfm?doid=2254129.2254166.

[143] "MIRABEL ontology". (), dirección: https://sites.google.com/site/smartappliancesproject/ontologies/mirabel-ontology (visitado 22-04-2023).

[144] J. Verhoosel, D. Rothengatter, F. J. Rumph y M. Konsman, "An ontology for modeling flexibility in smart grid energy management", en *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, jul. de 2012, págs. 931-938, ISBN: 9780415621281. DOI: 10.1201/b12516-146.

[145] S. Gillani, F. Laforest y G. Picard, "A generic ontology for prosumer-oriented smart grid", en *CEUR Workshop Proceedings*, vol. 1133, 2014, págs. 134-139.

[146] T. Hong, S. D'Oca, W. J. N. Turner y S. C. Taylor-Lange, "An ontology to represent energy-related occupant behavior in buildings. Part I: Introduction to the DNAs framework", *Building and Environment*, vol. 92, págs. 764-777, oct. de 2015, ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2015.02.019. dirección: https://linkinghub.elsevier.com/retrieve/pii/S0360132315000761.

[147] "Smart Applications REFerence Ontology, and extensions (SAREF).", ETSI. (21 de abr. de 2023), dirección: https://saref.etsi.org/ (visitado 18-02-2023).

[148] L. Daniele, F. den Hartog y J. Roes, "Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology", en *Lecture Notes in Business Information Processing*, vol. 225, Springer International Publishing, 2015, págs. 100-112, ISBN: 9783319215440. DOI: 10.1007/978-3-319-21545-7_9.

[149] L. Daniele, M. Solanki, F. Den Hartog y J. Roes, "Interoperability for Smart Appliances in the IoT World", en *Lecture Notes in Computer Science*, vol. 9982, Springer International Publishing, 2016, págs. 21-29, ISBN: 9783319465463. DOI: 10.1007/978-3-319-46547-0_3.

[150] "EEBUS: Empowering the digitalisation of Energy transition". (), dirección: https://www.eebus.org/ (visitado 22-04-2023).

[151] "Energy@Home". (), dirección: http://www.energy-home.it/SitePages/Home.aspx (visitado 22-04-2023).

[152] "Multi-Agent Systems and Secured coupling of Telecom and Energy gRIds for Next Generation smartgrid services (MAS2TERING)". (), dirección: http://www.mas2tering.eu/ (visitado 23-08-2022).

[153] "Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base, International Standard", International Electrotechnical Commission, inf. téc., 2 de feb. de 2022. dirección: https://webstore.iec.ch/publication/74467 (visitado 22-04-2023).

[154] G. Burel, L. S. G. Piccolo y H. Alani, "Energyuse - A collective semantic platform for monitoring and discussing energy consumption", en *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9982 LNCS, 2016, págs. 257-272, ISBN: 9783319465463. DOI: 10.1007/978-3-319-46547-0_26.

[155] D. Bonino, F. Corno y L. De Russis, "PowerOnt: An Ontology-Based Approach for Power Consumption Estimation in Smart Homes", en *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 150, Springer International Publishing, 2015, págs. 3-8, ISBN: 9783319196558. DOI: 10.1007/978-3-319-19656-5_1.

[156] J. Cuenca, F. Larrinaga y E. Curry, "DABGEO: A reusable and usable global energy ontology for the energy domain", *Journal of Web Semantics*, vol. 61-62, pág. 100 550, mar. de 2020, ISSN: 1570-8268. DOI: 10.1016/j.websem.2020.100550.

[157] J. Cuenca, F. Larrinaga y E. Curry, "A unified semantic ontology for energy management applications", en *CEUR Workshop Proceedings*, vol. 1936, mar. de 2017, págs. 86-97. dirección: https://www.edwardcurry.org/publications/WOMoCoE_17.pdf (visitado 22-04-2023).

[158] M. Haghgoo, I. Sychev, A. Monti y F. H. P. Fitzek, "SARGON – Smart energy domain ontology", *IET Smart Cities*, vol. 2, n.º 4, págs. 191-198, dic. de 2020, ISSN: 2631-7680. DOI: 10.1049/iet-smc.2020.0049.

[159] D. Saba, Y. Sahli y A. Hadidi, "An ontology based energy management for smart home", *Sustainable Computing: Informatics and Systems*, vol. 31, pág. 100 591, sep. de 2021, ISSN: 2210-5379. DOI: 10.1016/j.suscom.2021.100591. dirección: https://linkinghub.elsevier.com/retrieve/pii/S2210537921000809.

[160] D. Saba, Y. Sahli, F. H. Abanda, R. Maouedj y B. Tidjar, "Development of new ontological solution for an energy intelligent management in Adrar city", *Sustainable Computing: Informatics and Systems*, vol. 21, págs. 189-203, mar. de 2019, ISSN: 2210-5379. DOI: 10.1016/j.suscom.2019.01.009. dirección: https://linkinghub.elsevier.com/retrieve/pii/S221053791830341X.

[161] "DAML Ontology Library". (30 de abr. de 2004), dirección: http://www.daml.org/ontologies/ (visitado 19-07-2023).

[162] G. Denker, L. Kagal, T. Finin, M. Paolucci y K. Sycara, "Security for DAML Web Services: Annotation and Matchmaking", en *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003, págs. 335-350. DOI: 10.1007/978-3-540-39718-2_22. dirección: https://ebiquity.umbc.edu/_file_directory_/papers/59.pdf.

[163] G. Denker, L. Kagal y T. Finin, "Security in the Semantic Web using OWL", *Information Security Technical Report*, vol. 10, n.º 1, págs. 51-58, ene. de 2005. DOI: 10.1016/j.istr.2004.11.002.

[164] A. Kim, J. Luo y M. Kang, "Security Ontology for Annotating Resources", en *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2005, págs. 1483-1499, ISBN: 3540297383. DOI: 10.1007/11575801_34.

[165] "Security Ontology". Ontología descrita en .An Ontology of Information Security"de Herzog et al., Linköping University. (), dirección: http://www.ida.liu.se/~iislab/projects/secont (visitado 19-07-2023).

[166] A. Herzog, N. Shahmehri y C. Duma, "An Ontology of Information Security", *International Journal of Information Security and Privacy (IJISP)*, vol. 1, n.º 4, págs. 1-23, oct. de 2007, ISSN: 1930-1669. DOI: 10.4018/jisp.2007100101.

[167] S. Fenz y A. Ekelhart, "Formalizing information security knowledge", en *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ACM, mar. de 2009, ISBN: 9781605583945. DOI: 10.1145/1533057.1533084.

[168] A. Gyrard, C. Bonnet y K. Boudaoud, "An Ontology-Based Approach for Helping to Secure the ETSI Machine-to-Machine Architecture", en *2014 IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, IEEE, sep. de 2014, págs. 109-116, ISBN: 978-1-4799-5967-9. DOI: 10.1109/ithings.2014.25. dirección: http://ieeexplore.ieee.org/document/7059650/.

[169] A. Gyrard. "STAC (Security Toolbox: Attack & Countermeasure)". Herramienta web de STAC. (2022), dirección: http://sensormeasurement.appspot.com/?p=stac (visitado 19-07-2023).

[170] "Machine-to-Machine communications (M2M); Functional architecture", ETSI, inf. téc., oct. de 2013. dirección: https://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v020101p.pdf (visitado 30-04-2023).

[171] B. A. Mozzaquatro, R. Jardim-Goncalves y C. Agostinho, "Towards a reference ontology for security in the Internet of Things", en *2015 IEEE International Workshop on Measurements and Networking (M&N)*, IEEE, oct. de 2015, págs. 117-122. DOI: 10.1109/iwmn.2015.7322984.

[172] "Fichero OWL de la ontología IoTSec". (), dirección: https://raw.githubusercontent.com/brunomozza/IoTSecurityOntology/master/iotsec.owl (visitado 19-07-2023).

[173] "Fichero OWL de la ontología SecAOnto". (), dirección: https://raw.githubusercontent.com/ferruciof/Files/master/SecAOnto/SecAOnto_V4.owl (visitado 19-07-2023).

[174] F. de Franco Rosa, M. Jino y R. Bonacin, "Towards an Ontology of Security Assessment: A Core Model Proposal", en *Advances in Intelligent Systems and Computing*, vol. 738, Springer International Publishing, 2018, págs. 75-80, ISBN: 9783319770277. DOI: 10.1007/978-3-319-77028-4_12.

[175] M. Tao, J. Zuo, Z. Liu, A. Castiglione y F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes", *Future Generation Computer Systems*, vol. 78, págs. 1040-1051, ene. de 2018, ISSN: 0167-739X. DOI: 10.1016/j.future.2016.11.011.

[176] C. Choi y J. Choi, "Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service", *IEEE Access*, vol. 7, págs. 110 510-110 517, 2019. DOI: 10.1109/access.2019.2933859.

[177] M. F. Arruda y R. F. Bulcão-Neto, "Toward a lightweight ontology for privacy protection in IoT", *Proceedings of the ACM Symposium on Applied Computing*, págs. 880-888, abr. de 2019. DOI: 10.1145/3297280.3297367.

[178] "IoTCrawler – a search engine for Internet of Things devices". Sitio web del proyecto IoTCrawler. (), dirección: http://iotcrawler.eu (visitado 21-07-2023).

[179] G. Baldini, A. Skarmeta, E. Fourneret, R. Neisse, B. Legeard y F. L. Gall, "Security certification and labelling in Internet of Things", en *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, IEEE, dic. de 2016, págs. 627-632, ISBN: 9781509041305. DOI: 10.1109/wf-iot.2016.7845514.

[180] "OAuth 2.0". Sitio web de OAuth 2.0. (), dirección: https://oauth.net/2/ (visitado 21-07-2023).

[181] M. Sporny, G. Noble, D. Longley, D. C. Burnett, B. Zundel y K. D. Hartog, "Verifiable Credentials Data Model", W3C Recommendation, World Wide Web Consortium (W3C), inf. téc., 3 de mar. de 2022. dirección: https://www.w3.org/TR/vc-data-model/ (visitado 09-05-2023).

[182] S. Garg, K. Kaur, G. Kaddoum y K.-K. R. Choo, "Toward Secure and Provable Authentication for Internet of Things: Realizing Industry 4.0", *IEEE Internet of Things Journal*, vol. 7, n.º 5, págs. 4598-4606, mayo de 2020. DOI: 10.1109/jiot.2019.2942271.

[183] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues y M. Guizani, "Secure and Lightweight Authentication Scheme for Smart Metering Infrastructure in Smart Grid", *IEEE Transactions on Industrial Informatics*, vol. 16, n.º 5, págs. 3548-3557, mayo de 2020, ISSN: 1551-3203. DOI: 10.1109/tii.2019.2944880. dirección: https://ieeexplore.ieee.org/document/8854144/.

[184] A. Alkhalil y R. A. Ramadan, "IoT Data Provenance Implementation Challenges", *Procedia Computer Science*, vol. 109, págs. 1134-1139, 2017, ISSN: 1877-0509. DOI: 10.1016/j.procs.2017.05.436.

[185] N. Baracaldo, L. A. D. Bathen, R. O. Ozugha, R. Engel, S. Tata y H. Ludwig, "Securing Data Provenance in Internet of Things (IoT) Systems", en *Service-Oriented Computing – ICSOC 2016 Workshops*, Springer International Publishing, 2017, págs. 92-98, ISBN: 978-3-319-68136-8. DOI: 10.1007/978-3-319-68136-8_9.

[186] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat y L. Njilla, "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability", en *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, IEEE, mayo de 2017, ISBN: 9781509066100. DOI: 10.1109/ccgrid.2017.8.

[187] "Protégé: a free, open-source ontology editor and framework for building intelligent systems", Stanford University. (), dirección: https://protege.stanford.edu/ (visitado 15-07-2023).

[188] A. Gyrard, M. Serrano y G. A. Atemezing, "Semantic web methodologies, best practices and ontology engineering applied to Internet of Things", en *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, IEEE, dic. de 2015, ISBN: 9781509003655. DOI: 10.1109/wf-iot.2015.7389090.

[189] P.-Y. Vandenbussche y B. Vatant, "Metadata Recommendations For LinkedOpen Data Vocabularies", Ontologies Engineering Group - UPM, inf. téc., 19 de ago. de 2012. dirección: https://lov.linkeddata.es/Recommendations_Vocabulary_Design.pdf (visitado 09-05-2023).

[190] A. Gyrard, G. Atemezing, C. Bonnet, K. Boudaoud y M. Serrano, "Reusing and Unifying Background Knowledge for Internet of Things with LOV4IoT", en *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, ago. de 2016, ISBN: 9781509040520. DOI: 10.1109/ficloud.2016.45.

[191] "JSON for Linking Data, Data is messy and disconnected. JSON-LD organizes and connects it, creating a better Web." Sitio web de JSON-LD, World Wide Web Consortium (W3C). (), dirección: https://JSON-LD.org (visitado 21-07-2023).

[192] "Fogflow 3.2.8 documentation". Sitio web de documentación de FogFlow, NEC. (2022), dirección: https://fogflow.readthedocs.io/en/latest/ (visitado 21-07-2023).

[193] "Orion Context Broker (with Linked Data Extensions)". Repositorio de código del proyecto Orion-LD. (), dirección: https://github.com/FIWARE/context.Orion-LD (visitado 22-07-2023).

[194] "Identity Manager - Keyrock". Sitio web de documentación sobre Keyrock IdM, FIWARE. (), dirección: https://fiware-idm.readthedocs.io/en/latest/ (visitado 22-07-2023).

[195] "EU H2020 IoTCrawler Project- Open Source Tools and Components". Repositorio de código del proyecto IoTCrawler, IoTCrawler. (), dirección: https://github.com/orgs/IoTCrawler/repositories (visitado 22-07-2023).

[196] "Node-RED, Low-code programming for event-driven applications". Sitio web de Node-RED, OpenJS Foundation. (), dirección: https://nodered.org (visitado 22-07-2023).

[197] "ESPHome". Sitio web de ESPHome. (), dirección: https://esphome.io (visitado 22-07-2023).

[198] "TP-Link Kasa Smart". Sitio web de la integración TP-Link en Home Assistant. (), dirección: https://www.home-assistant.io/integrations/tplink (visitado 22-07-2023).

[199] "esios - red eléctrica, PVPC". Sitio web de esios-PVPC, Red Eléctrica Española. (), dirección: https://www.esios.ree.es/es/pvpc (visitado 22-07-2023).

[200] B. Cheng, "FogFlow tutorial, Release v3.2.8", inf. téc., 29 de nov. de 2022. dirección: https://fogflow.readthedocs.io/_/downloads/en/latest/pdf/ (visitado 20-05-2023).

[201] "Docker. Accelerated, Containerized Application Development". Sitio web de Docker, Docker Inc. (), dirección: https://www.docker.com (visitado 22-07-2023).

[202] "Meet Jetson, the Platform for AI at the Edge". Sitio web de la plataforma Jetson de NVIDIA, Nvidia Corporation. (), dirección: https://developer.nvidia.com/embedded-computing (visitado 22-07-2023).

[203] D. Franklin. "Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson". Repositorio de código de modelos preentrandos para Jetson Nano, NVIDIA Corporation. (), dirección: https://github.com/dusty-nv/jetson-inference (visitado 22-07-2023).

[204] A. Krizhevsky, I. Sutskever y G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Neural Information Processing Systems*, vol. 25, ene. de 2012. DOI: 10.1145/3065386.

[205] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions", en *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun. de 2015, págs. 1-9. DOI: 10.1109/cvpr.2015.7298594.

[206] K. He, X. Zhang, S. Ren y J. Sun, "Deep Residual Learning for Image Recognition", en *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun. de 2016, págs. 770-778. DOI: 10.1109/CVPR.2016.90.

[207] K. Simonyan y A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2014. DOI: 10.48550/ARXIV.1409.1556.

[208] C. Szegedy, S. Ioffe, V. Vanhoucke y A. Alemi, *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, 2016. DOI: 10.48550/ARXIV.1602.07261.

[209] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single Shot MultiBox Detector", 2015. DOI: 10.48550/ARXIV.1512.02325.

[210] A. G. Howard, M. Zhu, B. Chen et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, 2017. DOI: 10.48550/ARXIV.1704.04861.

[211] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, jun. de 2018. DOI: 10.1109/cvpr.2018.00474.

[212] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: Common Objects in Context", en *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele y T. Tuytelaars, eds., Cham: Springer International Publishing, 2014, págs. 740-755, ISBN: 978-3-319-10602-1. DOI: 10.1007/978-3-319-10602-1_48.

[213] S. Ioffe y C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015. DOI: 10.48550/ARXIV.1502.03167.

[214] J. Huang, V. Rathod, C. Sun et al., "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors", en *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jul. de 2017. DOI: 10.1109/cvpr.2017.351.

[215] F. Schroff, D. Kalenichenko y J. Philbin, "FaceNet: A unified embedding for face recognition and clustering", en *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun. de 2015. DOI: 10.1109/cvpr.2015.7298682.

[216] S. Song, S. P. Lichtenberg y J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite", en *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun. de 2015. DOI: 10.1109/cvpr.2015.7298655.

[217] K. Peffers, M. Rothenberger, T. Tuunanen y R. Vaezi, "Design Science Research Evaluation", en *Lecture Notes in Computer Science*, vol. 7286 LNCS, Springer Berlin Heidelberg, 2012, págs. 398-410, ISBN: 9783642298622. DOI: 10.1007/978-3-642-29863-9_29.

[218] K. Adamek, J. Novotny, J. Thiyagalingam y W. Armour, "Efficiency Near the Edge: Increasing the Energy Efficiency of FFTs on GPUs for Real-Time Edge Computing", *IEEE Access*, vol. 9, págs. 18 167-18 182, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3053409. arXiv: 2009.06009.

[219] F. Mantovani y E. Calore, "Performance and Power Analysis of HPC Workloads on Heterogenous Multi-Node Clusters", *Journal of Low Power Electronics and Applications*, vol. 8, n.º 2, pág. 13, mayo de 2018, ISSN: 2079-9268. DOI: 10.3390/jlpea8020013.

[220] A. A. Suzen, B. Duman y B. Sen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN", *HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, págs. 3-7, jun. de 2020. DOI: 10.1109/HORA49412.2020.9152915.

[221] "Intel Neural Compute Stick 2 (Intel NCS2)", Intel Corporation. (), dirección: https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html.

[222] V. Naresh Boddeti, "Secure Face Matching Using Fully Homomorphic Encryption", en *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, oct. de 2018, págs. 1-10, ISBN: 978-1-5386-7180-1. DOI: 10.1109/BTAS.2018.8698601. dirección: https://ieeexplore.ieee.org/document/8698601/.

[223] A. Hassan, F. Liu, F. Wang e Y. Wang, "Secure image classification with deep neural networks for IoT applications", *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, n.º 8, págs. 8319-8337, oct. de 2020, ISSN: 1868-5145. DOI: 10.1007/s12652-020-02565-z.