

UCLV
Universidad Central
"Marta Abreu" de Las Villas



FIE
Facultad de
Ingeniería Eléctrica

Departamento

TRABAJO DE DIPLOMA

Título: Plataformas de inteligencia artificial vinculadas a la computación en los borde para IoT.

Autor: Osvaldo Morera Aguila

Tutor: MSc. Arelys Ramos Fleites

MSc. Daniel Iglesias De La Torre

Santa Clara, noviembre 2021
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419

DEDICATORIA

A mis padres, quienes han sido siempre mi motor impulsor, sin ellos no hubiese sido posible lograr esta meta.

A mi novia, por siempre confiar en mí.

A mis amigos, que han hecho de este recorrido lo mejor de mi vida.

A mi familia en general, por siempre estar a mi lado.

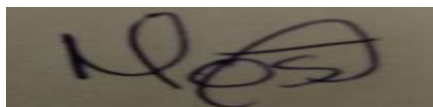
A todos los que de una u otra forma estuvieron ahí para hacer más ameno el camino.

AGRADECIMIENTOS

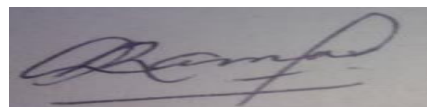
Quiero agradecer a toda mi familia, en especial a mis padres y a mi novia por su apoyo incondicional. Por supuesto agradezco a mis amigos, muchas gracias de corazón por esos momentos inolvidables que han dado a este período la alegría y la energía necesarias para llegar hasta aquí. Doy las gracias también a mis tutores Arelys Ramos Fleites y Daniel Iglesias De La Torre, quienes me han guiado profesionalmente y me han brindado sus conocimientos para salir adelante. Agradecer, además, de forma general a la Universidad Central Marta Abreu de Las Villas, a la Facultad de Ingeniería Eléctrica y a todos los profesores que a lo largo de la carrera contribuyeron a mi formación, especialmente a Félix Florentino Álvarez Paliza, por la ayuda brindada en este proyecto. A todos los que estuvieron al tanto y se preocuparon por mí, gracias por la confianza y el apoyo.

TAREA TÉCNICA

1. Revisión bibliográfica de temas relacionados con la computación de borde, la Internet de las cosas y las plataformas de inteligencia artificial.
2. Descripción de las particularidades de la computación en los bordes como tecnología para enfrentar los problemas de IoT.
3. Análisis de las plataformas de IA vinculadas a la computación en los bordes y a IoT.
4. Evaluación según sus ventajas de las mejores plataformas de borde descritas para procesar datos en tiempo real.
5. Verificación del funcionamiento de una de las plataformas abordadas.



Firma del Autor



Firma del Tutor

RESUMEN

El aumento sustancial de los datos generados por el Internet de las cosas y las solicitudes de información, en tiempo real, por parte de los dispositivos, hacen que la capacidad de la computación en la nube quede rezagada. Se hace necesario la búsqueda de soluciones para evitar que el ancho de banda de la red y la latencia de las comunicaciones se conviertan en serios cuellos de botella, lo que ha impulsado un nuevo paradigma informático: la computación de borde. Esta tecnología tiene como objetivo proporcionar servicios basados en Internet en las proximidades de los usuarios al colocar las infraestructuras de tecnologías de la información en el borde de la red en formas de pequeños centros de datos, permitiendo de esta manera aliviar la carga de procesamiento en la nube. Con este trabajo se analizan distintas plataformas de inteligencia artificial y se verifica su funcionalidad para satisfacer los objetivos de la computación en los bordes.

TABLA DE CONTENIDOS

DEDICATORIA	iii
AGRADECIMIENTOS	iv
TAREA TÉCNICA.....	v
RESUMEN	vi
INTRODUCCIÓN.....	1
CAPÍTULO 1. Marco teórico.....	4
1.1 Internet de las cosas	4
1.1.1 Historia y definición	5
1.1.2 Modelo de referencia	8
1.2 Edge Computing.....	10
1.2.1 Historia y concepto	11
1.2.2 Arquitectura	12
1.2.3 Fog Computing vs. Edge Computing	14
1.2.4 Seguridad	15
1.2.5 Ventajas	16
1.3 Inteligencia artificial	17
1.3.1 IA en IoT y en Edge Computing.....	20
1.4 Conclusiones del capítulo	22

CAPÍTULO 2. Plataformas de IA vinculadas a la computación en el borde y a IoT	24
2.1 Plataformas de IoT con IA	25
2.1.1 Ejemplos	26
2.2 Plataformas vinculadas a la computación en el borde	33
2.2.1 Ejemplos	34
2.3 Conclusiones del capítulo	37
CAPÍTULO 3. Resultados y evaluación de la plataforma de borde EdgeX Foundry	38
3.1 EdgeX Foundry	38
3.1.1 Objetivos	39
3.1.2 Organización	40
3.1.3 Arquitectura	41
3.2 Caso de estudio	44
3.2.1 Instalación de Edgex Foundry	44
3.2.2 Ejecución de Edgex Foundry	44
3.3 Control de la funcionalidad del proyecto EdgeX Foundry	48
3.4 Conclusiones del capítulo	52
CONCLUSIONES Y RECOMENDACIONES	53
Conclusiones	53
Recomendaciones	54
REFERENCIAS BIBLIOGRÁFICAS	55
ANEXOS	61
Anexo I Servicios de EdgeX Foundry iniciados de manera automática tras la instalación [74].	61
Anexo II Servicios deshabilitados de forma predeterminada tras la instalación de EdgeX Foundry [74].	62

INTRODUCCIÓN

La evolución de las comunicaciones causó la proliferación de los dispositivos móviles, por lo que el concepto de *computación en la nube* (*cloud computing*) está saturándose, ya que los dispositivos están haciendo solicitudes de información en tiempo real a una capacidad superior a la que la nube podría responder, una mayor iteración de datos supondría, además, un aumento dramático en los costos de operación. Tecnologías 4G, 4.5G y 5G están imponiéndose en el mercado como una evolución en la transmisión de datos a dispositivos móviles; la computación de borde (EC, *edge computing*) se vuelve una solución vital para el desarrollo de dichas tecnologías, con el fin de precautelar la cantidad de información entregada al usuario. Se va haciendo necesario buscar soluciones que relacionen a nivel de ubicación física del cliente o de la fuente de datos para lograr servicios y procesamiento de datos críticos más rápidos y confiables al usuario final, esto puede ser posible si se emplea la computación en los bordes que vinculado con la inteligencia artificial (IA) para su uso en los sistemas de Internet de las cosas (IoT, *Internet of things*) podría dar lugar a grandes avances en el desarrollo de la humanidad [1].

En una arquitectura tradicional IoT, la información se captura desde dispositivos y sensores que se encuentran dispersos a través de la red. Esta información se envía a servidores centrales donde se procesa conjuntamente y en algunos casos se almacena. Para procesar estos datos de forma eficiente, escalable y de forma rápida, se emplea la computación en la nube. En líneas generales, se trata de servidores publicados en Internet que se encargan de atender peticiones en cualquier momento y se caracterizan por su alto grado de automatización, capacidad para distribuir recursos, y la facilidad de acceder a ellos con solo una conexión a Internet. Desafortunadamente, las arquitecturas basadas en un procesamiento en la nube centralizado, no confluyen con lo que se espera de IoT y se

enfrentan a muchas dificultades. Por ejemplo, las aplicaciones móviles generan tal cantidad de datos que muchas veces no pueden ser atendidos por los servidores y producen tiempos de espera muy altos. De las aplicaciones IoT se espera un tiempo de respuesta muy bajo, prácticamente inmediato, en tiempo real, y de forma ininterrumpida, por lo tanto, en este tipo de arquitecturas lo que se pretende es reducir la latencia, así como los costes de ancho de banda y almacenamiento [2].

Actualmente, los dispositivos IoT que se encuentran pueden ser de muchos tipos, desde sensores, teléfonos inteligentes, relojes inteligentes, cámaras, hasta asistentes domésticos. Todos ellos tienen en común que son objetos físicos, que se comunican y que lo hacen a través de la nube. Pero cada uno de ellos tiene una capacidad de cómputo limitada y su propósito es transmitir los datos producidos por sus sensores a servidores de procesamiento en la nube a través de comunicaciones en red. Y todo ello procesando lo mínimo imprescindible en el propio dispositivo. Pero ¿y si gran parte de este procesamiento se puede realizar en el propio dispositivo evitando saturar de datos a los servidores de la nube? En la medida que la aparición de estos dispositivos IoT se incrementa, la forma de procesar la información que producen toma un papel cada vez más relevante. De esta forma surge el concepto de Edge Computing, un modelo de computación que sirve para optimizar el rendimiento de las aplicaciones IoT permitiendo realizar el cómputo de estas en el propio dispositivo, de ahí su nombre traducido como “computación de borde”. Esta nueva forma de procesar los datos se basa en realizar los cálculos lo más cerca posible de donde se generan, esto es, en los propios dispositivos o en el borde de la red antes de ser enviados a través de la red a los servidores centrales [3].

Luego, el problema radica en la necesidad de evaluar alternativas partiendo de la búsqueda de información sobre plataformas abiertas de borde que den solución a la situación anterior. La situación descrita anteriormente lleva a plantear la siguiente interrogante científica:

¿Cuáles serían las posibles plataformas de IA de código abierto que puedan implementarse a nivel de borde para sistemas IoT?

Para dar solución al problema anterior se plantea como **objetivo general** de esta investigación:

- Evaluar posibles plataformas de computación en los bordes que puedan ser implementadas en sistemas IoT.

Objetivos Específicos:

- Analizar el estado de avance de la aplicación de la inteligencia artificial en IoT.
- Describir las tecnologías de IA que se vinculan a sistemas IoT.
- Determinar las aplicaciones de IoT que admiten convergencia de la IA y computación en los bordes.
- Seleccionar las plataformas de IA más adecuadas para en sistemas IoT.
- Evaluar las plataformas de IoT con IA que garantizan mejores prestaciones en cuanto a rapidez y confiabilidad.

Interrogantes Científicas:

- ¿Cuál es el estado de avance de la IA?
- ¿Cuáles son las tecnologías de IA más vinculadas a las aplicaciones de IoT?
- ¿Cuáles son las plataformas de IA que se desarrollan en la actualidad para el perímetro de la red en sistemas IoT?
- ¿Cuál de las plataformas analizadas tiene más aplicabilidad en la actualidad?

Estructura del Trabajo:

El informe de la investigación se estructura en: introducción, capitulario, conclusiones, bibliografía y anexos. A continuación, se presenta brevemente el contenido de los capítulos:

Capítulo 1: Marco Teórico.

Se dedicará a definir todo lo relacionado con los conceptos claves de la investigación: computación de borde, Internet de las cosas e inteligencia artificial.

Capítulo 2: Plataformas de IA vinculadas a la computación en el borde y a IoT.

En este capítulo se dan a conocer una serie de plataformas de IA vinculadas a la EC y a la IoT y se hace un análisis de algunas de ellas.

Capítulo 3: Resultados y evaluación de la plataforma de borde EdgeX Foundry.

Se especifican las características de la plataforma seleccionada, se realiza una simulación con un sensor virtual y se analizan los resultados obtenidos.

CAPÍTULO 1. Marco teórico

La cantidad de datos generados por sensores, actuadores y otros dispositivos ha aumentado, sustancialmente, en las últimas décadas. Actualmente, los datos son procesados en la nube, en consecuencia, el ancho de banda de la red y la latencia de las comunicaciones se convierten en serios cuellos de botella, lo que ha impulsado un nuevo paradigma informático: computación de borde. EC tiene como objetivo proporcionar servicios basados en Internet en las proximidades de los usuarios al colocar las infraestructuras de tecnologías de la información en el borde de la red en formas de pequeños centros de datos, permitiendo de esta manera aliviar la carga de procesamiento en la nube [4].

En este capítulo se pretende analizar la Internet de las cosas de forma general y este nuevo paradigma de computación conocido como *computación de borde*, que acerca parte de las capacidades de análisis y procesamiento que actualmente ofrecen los sistemas en la nube al lugar en el que se generan los datos, disminuyendo de esta forma el volumen de información que debe enviarse a la red y mejorando el tiempo de respuesta de los sistemas. Asimismo, se detallan aspectos de inteligencia artificial que más adelante se vincularán con algunas de las posibles plataformas de computación en los bordes que puedan ser implementadas en sistemas IoT.

1.1 Internet de las cosas

El tamaño y variedad de los datos que circulan por las redes actuales está aumentando exponencialmente e IoT contribuye significativamente a este aumento de volumen (figura 1.1), esto implica un incremento sustancial del tráfico que llega a los centros de datos para su análisis. Este aumento de tráfico afecta de forma directa al tiempo de respuesta de los sistemas de IoT, lo que en entornos críticos puede resultar inaceptable.



Figura 1.1 Escenarios de aplicación de IoT [5].

El Internet de las cosas es una de las tecnologías disruptivas que mayor interés ha generado en los últimos años, no solo en el ámbito de la investigación científica, sino también en el sector gubernamental y empresarial. Su incorporación a todos los entornos de actividad humana y social como los servicios públicos, agricultura, industria, economía, educación o salud, permite obtener datos desde cualquier dispositivo conectado a una red. El principal problema de la acelerada integración de Internet de las cosas es la gestión del gran volumen de datos generados constantemente por los dispositivos conectados, es decir, su procesamiento, distribución y almacenamiento, para que posteriormente sean utilizados a modo de información en los procesos de inteligencia de negocios [6].

1.1.1 Historia y definición

En 1926 Nikola Tesla preparó las bases de las comunicaciones, después en el año 1990 Berners-Lee creó el Protocolo de Transferencia de Hipertexto (HTTP, Hypertext Transfer Protocol); estas son las bases donde posteriormente en el año 1999 Kevin Ashton fue la persona que utilizó por primera vez la expresión de IoT en una conferencia, desde entonces comenzó a ser normal referirse al sistema de conexión de cosas a internet. Kevin Ashton trabajaba en Procter & Gamble (P&G), tenía 28 años en ese momento y estaba en problemas porque los productos que manejaba no estaban disponibles en las tiendas, se dio

cuenta de cuál era el problema de información, por lo tanto se le ocurrió una idea de colocar sensores a los productos para saber cuándo dejarán de estar en *stock*, él trató de convencer a P&G para poder implementar la idea que tenía. Asthon asegura que «entendió que la palabra *internet* podría atraer la atención de esta compañía porque en 1998 los gerentes pensaban que la red era lo más importante y buscaban nuevos proyectos». Después la palabra *cosas* se comenzó a usar por la idea de empotrar las computadoras en las mesas y cada vez los equipos llegaban más económicos y más pequeños, la idea era confusa pero era lo suficiente para que comenzara a investigar sobre IoT [7].

En 2005 la Unión Internacional de Telecomunicaciones (International Telecommunications Union, ITU) realizó el primer estudio sobre el tema, ellos afirmaron lo siguiente: «una nueva dimensión se ha agregado al mundo de las tecnologías de información y la comunicación (TIC): a cualquier hora, en cualquier lugar, ahora vamos a tener conectividad para cualquier cosa. Las conexiones se multiplican y crearán una nueva red dinámica de redes con redes, Internet de las Cosas». En el año 2008 un grupo de empresas crearon una alianza para promover el uso de protocolos de internet de objetos inteligentes, comenzaron a trabajar en ello para que se hiciera realidad esta idea. La Alliance IPSO tiene empresas involucradas actualmente como Google, Bosch, Motorola, Toshiba, etc. Primero, comenzaron con el proyecto de desarrollo del protocolo IPV6. En el año 2009 comenzó a ser más escuchada esa palabra: *IoT*; en 2011 fue lanzado como tal el protocolo IPV6 y otros fabricantes anunciaron sus proyectos, después se inició la adopción de estándares para IoT a escala global [7].

Hoy en día, se puede ver en nuestra vida cotidiana gran cantidad de objetos conectados que forman parte del Internet de las cosas, una realidad muy presente que está evolucionando a pasos agigantados. Estos millones de dispositivos conectados generan una gran cantidad de datos que llegan a una plataforma IoT que recolecta, procesa y analiza dichos datos. Esta información se hace relevante al usuario porque gracias a ella se pueden sacar conclusiones de los hábitos y preferencias del mismo, facilitándole la vida. Por sí solos, estos dispositivos pueden realizar tareas sencillas, mediante pequeños sensores, desde regular la temperatura o la luminosidad de una habitación hasta medir el tráfico de taxis en una ciudad, además pueden avisarte, por ejemplo, si hay algún fallo en tu coche para que lo lleves al taller.

A diario, cámaras de seguridad velan por la seguridad en los edificios y los paneles del metro nos indican el tiempo que falta hasta la llegada del próximo tren. Incluso en las multas del tráfico existe poca intervención humana. Cada vez más objetos están siendo integrados con sensores, ganando capacidad de comunicación y con ello las barreras que separan al mundo real del virtual se difuminan [8].

El término IoT ha sido definido por diferentes autores de diversas formas, Vermesan et al. [9] definen Internet de las cosas, simplemente como una interacción entre lo físico y lo digital. El mundo digital interactúa con el mundo físico usando una plétora de sensores y actuadores. Otra definición [10] plantea Internet de las cosas como un paradigma en el que la capacidad de cómputo y comunicación están incrustadas en cualquier tipo de objeto concebible.

El eje central de IoT son los datos recogidos por sensores y dispositivos, estos datos se envían a servidores remotos o a la nube para su tratamiento, una vez que se ha extraído de ellos la información que se considera importante, los dispositivos de IoT pueden recibir del servidor o de la nube una serie de instrucciones para realizar una determinada acción. En términos empresariales, el valor para las organizaciones está en la información que se puede extraer de dichos datos porque permite automatizar procesos, optimizar recursos y tomar mejores decisiones, lo que conlleva una mayor eficiencia operativa [11].

Todo este crecimiento del tráfico procedente de IoT supone un desafío para la computación en la nube, tecnología que ha impulsado el uso de estos sistemas IoT, ofreciendo las capacidades de almacenamiento y de cómputo necesarias para tratar y extraer información de los datos generados, ya que si toda la información tiene que viajar hasta los centros de datos en la nube, ser procesada y hay que devolver una acción a ejecutar, es más probable que esta llegue tarde, afectada por el continuo incremento del tráfico en la red; por esto, se hace necesario acortar estas distancias y la solución sería procesar la información y tomar la decisión en el extremo de la red, disminuyendo sustancialmente la latencia y pudiéndose ejecutar la acción de forma casi inmediata; esto último tan necesario para muchas de las aplicaciones de IoT como los vehículos autónomos, que no pueden permitirse un retraso entre una posible colisión y el actuar en consecuencia, dependiendo de sistemas que funcionen en tiempo real, con muy baja latencia; asimismo, necesario para las comunicaciones con control táctil en diversos

sectores, alarmas contra incendios, detectores de humo o sensores conectados a pacientes que permite a los médicos el seguimiento de sus condiciones fuera del hospital y en tiempo real.

1.1.2 Modelo de referencia

Las arquitecturas de gestión de red, computación, aplicaciones y datos que están listas para IoT requieren un modelo de comunicación y procesamiento. Cisco establece un modelo de referencia de IoT, cuyo propósito es proporcionar definiciones y descripciones claras que puedan aplicarse con precisión a los elementos y funciones de los sistemas y aplicaciones de IoT. Este modelo simplifica, aclara, identifica, organiza y estandariza, proporcionando un primer paso para permitir a los proveedores crear productos de IoT que funcionen entre sí. El modelo de referencia de IoT propuesto se compone de siete niveles. Cada nivel se define con terminología que se puede estandarizar para crear un marco globalmente aceptado de referencia. El modelo de referencia de IoT no restringe el alcance o la localidad de sus componentes. También permite que el procesamiento que se produce en cada nivel varíe de trivial a complejo, dependiendo de la situación. El modelo describe cómo deben manejarse las tareas en cada nivel para mantener la simplicidad, permite una alta escalabilidad y garantiza la compatibilidad. Finalmente, el modelo define las funciones requeridas para que un sistema IoT esté completo [12].

A continuación (figura 1.2) se muestra dicho modelo de referencia con sus niveles, propuesto por Cisco en el Fórum Mundial de IoT de 2014. Es importante señalar que en IoT los datos fluyen en ambas direcciones. En un patrón de control, la información de control fluye desde la parte superior del modelo (nivel 7) hacia la parte inferior (nivel 1). En un patrón de seguimiento, el flujo de información es inverso. En la mayoría de los sistemas, el flujo será bidireccional.

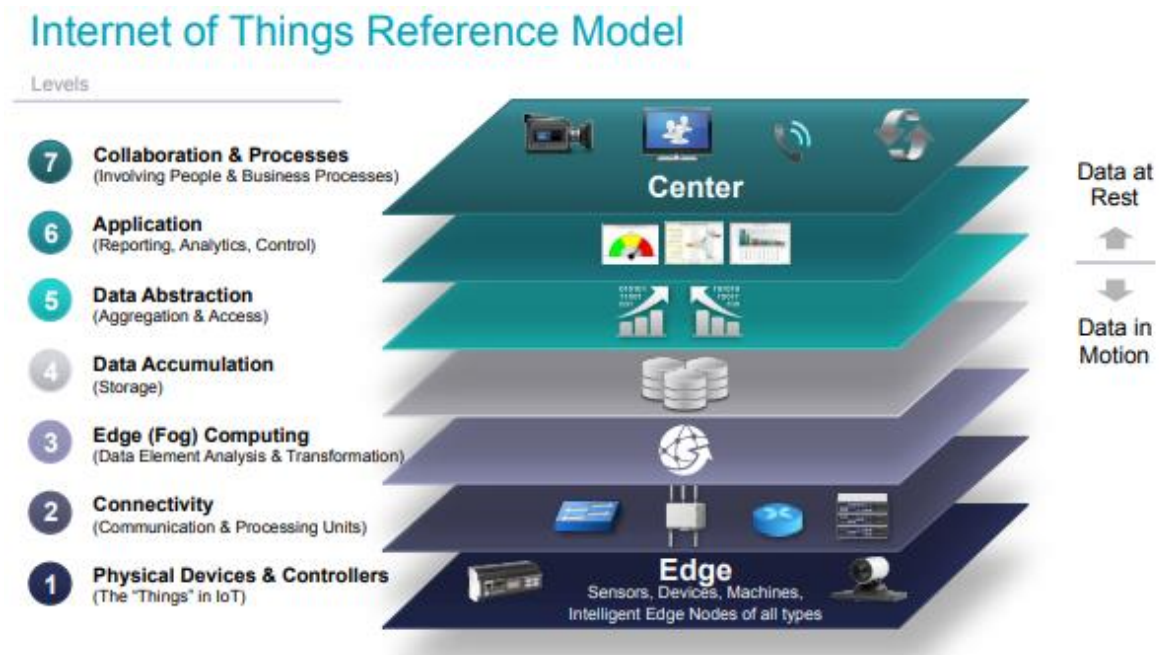


Figura 1.2 Modelo de Referencia para IoT propuesto por Cisco [12].

Nivel 1: Controladores y dispositivos físicos.

Se encuentran en este nivel las *cosas* del IoT: los sensores, actuadores y dispositivos físicos que pueden generar, enviar y recibir datos a través de un protocolo de red.

Nivel 2: Conectividad.

Se corresponde con los equipos de red que permiten que los dispositivos del nivel 1 se puedan comunicar entre ellos y con niveles superiores. Generalmente, el nivel 1 y el 2 se encuentra en las instalaciones del cliente.

Nivel 3: Computación de borde (niebla).

Tiene como objetivo realizar un análisis inicial del flujo de datos, puede verse como una capa intermedia entre el *hardware* y las infraestructuras remotas en la nube, se debe ubicar lo más cerca posible del origen de los datos, en el perímetro de la red. En este nivel se realiza una evaluación inicial de los datos para determinar si deben ser procesados por un nivel superior o por el nivel en el que se encuentran, se les da formato, se elimina datos erróneos, se comprimen..., se evita de esta forma sobrecargar la red con información poco útil, optimizando el uso de recursos de red y ancho de banda disponible.

Del nivel 4 al 7: Acumulación de datos, Abstracción de datos, Aplicación, y Colaboración y proceso.

Se encuentran las infraestructuras remotas, las aplicaciones y los servicios que ofrece Cloud Computing, se convierten los datos en información y conocimiento, es donde encontramos el mayor potencial de IoT.

El valor de IoT viene dado de una combinación de Edge y Cloud Computing. Procesar los flujos de datos en el menor tiempo posible implica disminuir el alto grado de centralización que los servicios en la nube tienen en la actualidad, para ello hay que propiciar el análisis y la generación de conocimiento cerca del lugar en el que estos se generan. En este aspecto, la capa de Edge (Fog) Computing tiene un papel fundamental.

1.2 Edge Computing

Edge Computing representa una solución para enfrentar y aliviar la carga de procesamiento y almacenamiento en la nube, en aplicaciones y tecnologías que requieren de un ancho de banda exponencial y una baja o nula latencia. Esta es la esencia del paradigma Edge Computing, los datos son procesados en el borde de la red, lo más cerca de los usuarios finales [4].

Relacionados con Edge Computing existen otros paradigmas informáticos como Cloud Computing, Fog Computing, etc. En este epígrafe se tratan algunas particularidades generales que incluyen estas tecnologías complementarias que impulsan IoT (figura 1.3), pero el estudio realizado más a fondo tiene como base la computación de borde,.

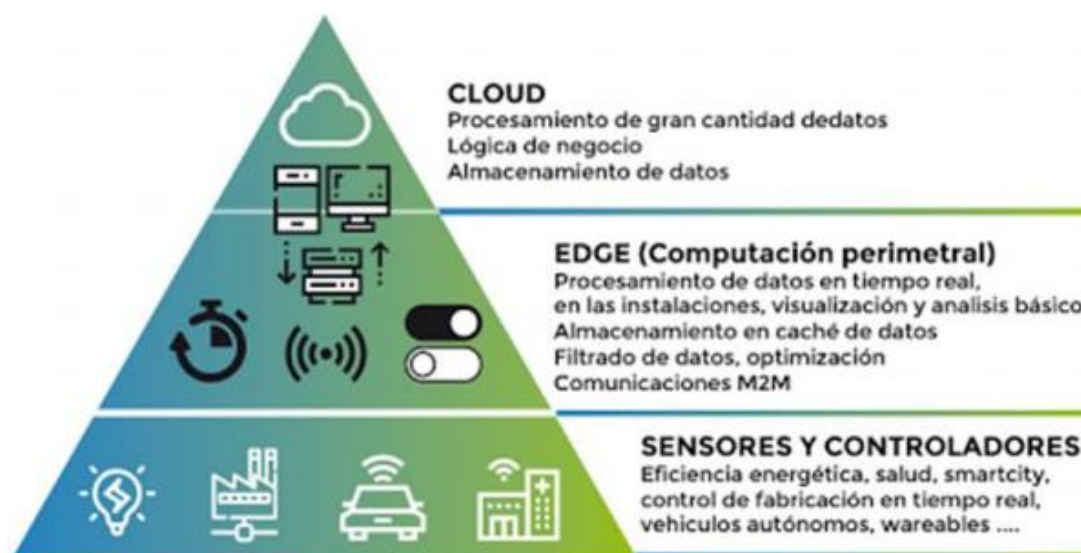


Figura 1.3 Tecnologías complementarias que impulsan IoT [13].

1.2.1 Historia y concepto

Los datos son el elemento vital de las empresas modernas. Proporcionan información comercial valiosa y respaldan el control en tiempo real de los procesos y operaciones comerciales críticas. Hoy en día, las empresas están inundadas con un océano de datos y los sensores y dispositivos de IoT pueden capturar de forma rutinaria grandes cantidades de datos que operan en tiempo real desde ubicaciones remotas y entornos operativos hostiles en casi cualquier parte del mundo. Pero este flujo virtual de datos también cambia la forma en que las empresas manejan las computadoras. El paradigma de datos tradicional basado en un centro de datos centralizado e Internet cotidiano no es adecuado para mover volúmenes de datos reales cada vez mayores. Las limitaciones de ancho de banda, los problemas de latencia y las interrupciones impredecibles de la red pueden conspirar para socavar dichos esfuerzos. La plataforma deberá responder a estos desafíos de datos mediante el uso de una arquitectura de Edge Computing [13].

La computación de borde o Edge Computing apareció alrededor de 2002 y se refiere al paradigma que permite ubicar los recursos de computación en el borde de la red, de modo que la computación ocurra cerca de las fuentes de datos [14]. Ya en 2003, IBM y Akamai propusieron la computación de borde en un proyecto de investigación interno llamado *Desarrollo de aplicaciones de computación perimetral*, y proporcionaron servicios basados en el borde en su Web Sphere. En 2004, Pang propuso por primera vez la computación de borde en la literatura abierta de la 20.^a Conferencia Internacional de la IEEE, en la que se menciona que «la computación de borde es lograr servicios web escalables y de alta disponibilidad que impulsarán la lógica empresarial y los centros de procesamiento de datos. La ventaja del lado perimetral del servicio proxy es que la operación de la aplicación en el lado del borde reduce la latencia de la red y da como resultado una respuesta más rápida del servicio web» [15].

Por lo tanto, la computación de borde es un nuevo modelo de computación que proporciona aplicaciones como computación convergente, almacenamiento, y redes en el borde de la red cerca de la fuente o fuente de datos. Al mismo tiempo, la informática de borde es una tecnología que aborda las necesidades críticas de la industria con una conectividad de forma ágil, servicios en tiempo real, optimización de datos, inteligencia de aplicaciones,

seguridad, y protección de la privacidad proporcionando estos recursos en el borde de la red [16].

1.2.2 Arquitectura

Investigaciones existentes [17-19] generalmente dividen la arquitectura de EC desde la red central hasta el borde de la red en tres capas: la capa de computación en la nube, la capa de computación de borde y la capa final (figura 1.4). Las diferentes capas generalmente se dividen de acuerdo con sus capacidades informáticas y de almacenamiento. Las capacidades de computación y almacenamiento de la capa final, la capa de computación de borde, y la capa de computación en la nube son incrementadas secuencialmente. Para lograr comunicación *intra-layer* y *cross-layer* cada tecnología de comunicación se puede conectar mediante diversas tecnologías de comunicación, incluida la comunicación por cable (como Ethernet, fibra óptica...), comunicación inalámbrica (como Bluetooth, LTE, ZigBee, NFC, enlace por satélite...) o una combinación de dos tecnologías [20].

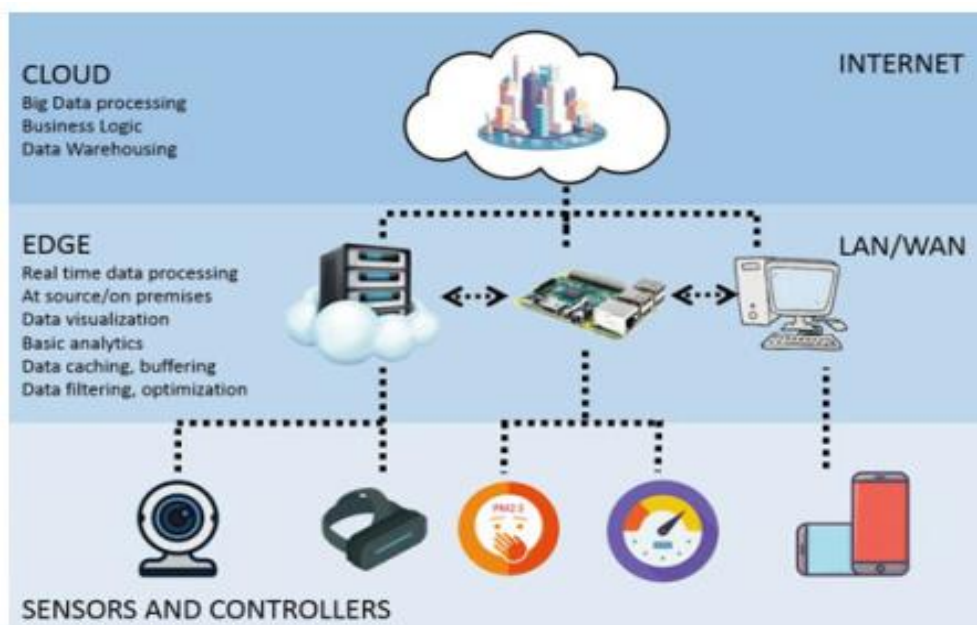


Figura 1.4 Arquitectura de EC [21].

A continuación se describe brevemente la conformación y función de cada una de las capas que componen la arquitectura de EC descrita anteriormente:

- 1) **Capa final:** es la capa más cercana al usuario final y consta de varios dispositivos IoT como sensores, teléfonos inteligentes, vehículos inteligentes, tarjetas

inteligentes, lectores de tarjetas y más. Para extender el tiempo de servicio del equipo terminal, debe evitarse ejecutar tareas de cálculo complejas en el equipo terminal. Por lo tanto, solo se recopilan los datos brutos del dispositivo terminal y se cargan en la capa superior para el cálculo y almacenamiento. La capa superior de la capa final está conectada principalmente a través de la red celular [19].

- 2) **Capa de computación de borde:** está ubicada en el borde de la red y consta de una gran cantidad de nodos de borde, incluidos dispositivos de red (enrutadores, puertas de enlace, puntos de acceso, estaciones base, etc.) y servidores de borde específicos. Estos nodos de borde están ampliamente distribuidos entre dispositivos terminales y la nube, como cafés, centros comerciales, autobuses, terminales, calles, parques, etc. Los nodos de borde se pueden implementar en cualquier lugar de la conexión de red, como en un teléfono inteligente, en una fábrica, en una unidad de carretera, en un vehículo, o en la parte superior de un poste de servicios públicos. Son capaces de calcular, transmitir y almacenar temporalmente datos sensoriales recibidos. Dado que estos nodos de borde solo tienen uno o dos saltos desde el dispositivo, el análisis en tiempo real y las aplicaciones sensibles al retraso se pueden realizar en la capa de computación de borde [22]. Además, los nodos de borde pueden conectarse al centro de datos en la nube a través de la red principal IP y son responsables de interactuar con la nube para colaborar para una informática más potente y funciones de almacenamiento. Esta capa está diseñada para extender la computación en la nube al borde de la red con algunas capacidades informáticas y de almacenamiento. La capa de la computación de borde es responsable de enviar periódicamente datos a la capa de la nube, proporcionando tres funciones principales de almacenamiento en caché de datos, computación de localización y acceso inalámbrica para satisfacer las demandas de baja latencia y alto tráfico de dispositivos móviles [12].
- 3) **Capa de la nube:** consta de múltiples servidores y dispositivos de almacenamiento de alto rendimiento con potentes capacidades informáticas y de almacenamiento para realizar tareas informáticas complejas. El módulo en la nube puede gestionar y programar de forma eficaz los nodos perimetrales y los centros de computación en la nube a través de políticas de control para proporcionar mejores servicios para los usuarios [15].

1.2.3 Fog Computing vs. Edge Computing

Uno de los términos asociados a Edge Computing fue introducido por Cisco en 2011: Fog Computing (computación en la niebla) [15]. Según [23] la computación en la niebla es una arquitectura de computación distribuida geográficamente con un grupo de recursos que consta de uno o más dispositivos heterogéneos conectados ubicuamente (incluidos los dispositivos de borde) en el borde de la red y no exclusivamente respaldados sin problemas por los servicios en la nube, para proporcionar de forma colaborativa computación elástica, almacenamiento y comunicación (y muchos otros nuevos servicios y tareas) en entornos aislados a una gran escala de clientes en proximidad.

La idea detrás de este concepto es la misma que la de EC: acercar el procesamiento a la fuente de datos. Son dos conceptos que se confunden a menudo pero aunque compartan la idea no son lo mismo.

En otras de las fuentes consultadas [24] se define Fog Computing como una arquitectura horizontal que abarca a Edge Computing y a la red requerida para llevar los datos preprocesados a su destino final. Con frecuencia, se sitúa la diferencia entre ambas arquitecturas en el lugar en el que se ubica la capacidad de procesamiento. En Fog Computing se establece en las cercanías de la red local donde el flujo de datos es enviado hacia nodos de niebla, *gateway* de IoT o pequeños servidores que procesan y enrutan el tráfico hacia donde se necesite.

Sin embargo, en Edge Computing esa capacidad de procesamiento se sitúa en la fuente de datos o lo más cerca posible a esta, en sensores conectados a controladores programables, en la maquinaria de una fábrica, en vehículos autónomos..., así un nodo de computación Edge se define como un hardware con capacidad de cómputo, situado físicamente cerca de los dispositivos o equipos que hacen uso de sus recursos.

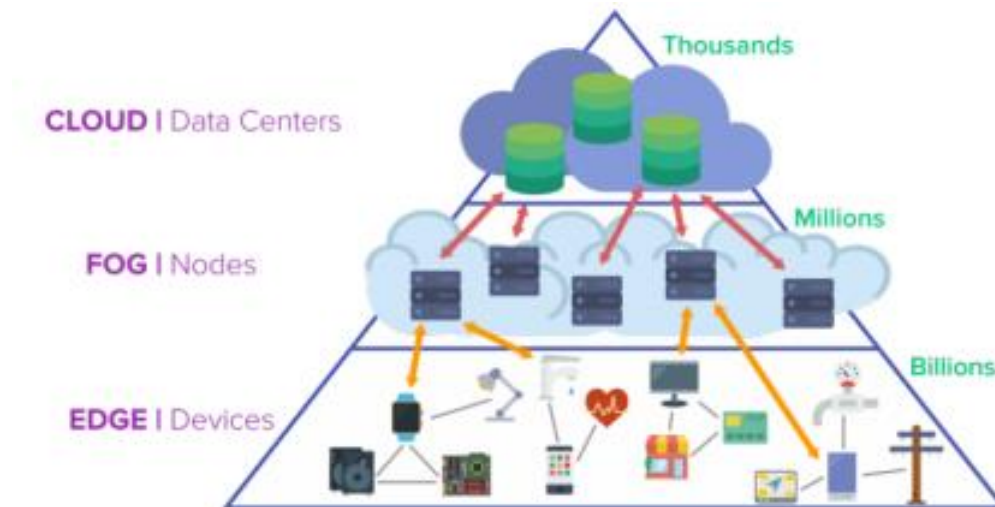


Figura 1.5 Edge, Fog y Cloud Computing [25].

En [26] se presentan los resultados de una encuesta, denominada *Computación en el borde y en la niebla, tendencias e inmersión*, realizada por la Asociación Colombiana de Ingenieros en Sistemas (Acis) dirigida a ingenieros de sistemas y computación, con el fin de conocer su opinión, conocimiento y grado de aplicación de estas dos tecnologías. Como resultado se manifiesta que:

- Se requieren mayores esfuerzos para que los directivos y expertos en Tecnologías de la Información (TI) conozcan y entiendan estas tecnologías.
- Si bien las alternativas de procesamiento en el borde en un entorno IoT, ofrecen respuestas en cuanto a la velocidad de respuesta, la seguridad de los datos y la eficiencia del procesamiento, persiste escepticismo en los usuarios atinentes a la confidencialidad, la disponibilidad y la integridad de los datos.
- La velocidad de cambio de las tendencias tecnológicas, desafía los ciclos de evolución y maduración de los sistemas de información y plantea retos a directivos y tomadores de decisiones, con tecnologías disruptivas que permean los paradigmas organizacionales, con alternativas, a veces, no consolidadas o suficientemente probadas

1.2.4 Seguridad

Muchas de las tecnologías que apoyan la virtualización, que son empleadas dentro de las arquitecturas de redes de computación en los bordes (contenedores y módulos centrales de sistemas operativos [unikernels]), son evaluadas y analizadas por los investigadores en

cuanto a seguridad. En [27] se muestra una tabla con un análisis de los escenarios de posibles ataques y también identifican las acciones maliciosas que un atacante puede acometer acorde con las tecnologías adoptadas, teniendo como meta desestabilizar el sistema y comprometer su privacidad.

Debido a que la mayoría del accionar suministrado por IoT es dependiente de la comunicación máquina a máquina en la red local o con la nube, durante la transferencia de datos pueden ocurrir ataques que están fuera del control humano. Luego el empleo de detectores de error (suma chequeo o códigos cíclicos redundantes CRC) aseguran la integridad de los datos transmitidos y recibidos. La característica más importante de los usuarios de IoT está en la disponibilidad de los datos en cualquier instante y lugar. Debido a que los dispositivos finales de IoT pueden estar muy limitados en su nivel de hardware los métodos de defensa utilizados en la computación en la nube no pueden ser siempre aplicados en este caso.

Algunos autores dirigen sus trabajos hacia la confidencialidad, integridad y disponibilidad de los datos mediante una arquitectura de seguridad construida en la cima de la arquitectura de un dispositivo IoT; con la implementación de algoritmos de encriptado para la autenticación y privacidad de los datos, en la detección de intrusos es también implementada. Mientras que otros autores ofrecen una implementación de soluciones concatenadas de bloques (*blockchain*) que permiten el registro digital de transacciones que son duplicadas y distribuidas a través de la red [28].

1.2.5 Ventajas

Muchos sectores (industrial, salud, telecomunicaciones, finanzas, agricultura y otros) pueden verse beneficiados de las ventajas que ofrece acercar el procesamiento y el análisis a la fuente que genera los datos.

Dado que el cálculo del borde está más cerca de la fuente de datos, los datos se pueden adquirir a la primera vez, y los datos se pueden analizar y procesar de forma inteligente en tiempo real, que es más eficiente y seguro que la computación en la nube pura. Comparado con la computación en la nube, la computación de borde incluye principalmente las siguientes ventajas según [29]:

- **Baja latencia distribuida:** Dado que el cálculo del borde está cerca de la fuente de recepción de datos, los datos se pueden adquirir y analizar en tiempo real, lo cual puede soportar mejor el procesamiento inteligente en tiempo real y la ejecución del servicio local.
- **Alta Eficiencia:** El cálculo del borde está más cerca del usuario y los datos se pueden filtrar y analizar en el nodo de borde sin esperar el tiempo de transmisión de datos, por lo que la eficiencia será mayor.
- **Alivia la presión de flujo:** La informática de borde realiza un procesamiento de datos simple a través de nodos de borde al realizar la transmisión en la nube. Cuando se enfrenta a una gran cantidad de datos, cierto algoritmo de compresión se puede utilizar para extraer información útil y luego transmitir, lo que puede ahorrar una gran cantidad de consumo de ancho de banda de datos.
- **Mayor seguridad y procesamiento de datos:** Después de recibir los datos, el cálculo de borde puede cifrar los datos y luego transmitirlos, lo que mejora la seguridad de los datos. La computación de borde requiere procesamiento de datos independiente después de recibir datos. Si el rendimiento del chip es insuficiente no se pueden procesar algoritmos de procesamiento de datos complejos.

De esta forma, EC se presenta como una solución para muchos servicios de IoT emergentes que demandan procesamiento en tiempo real y mejorar la eficiencia en la recopilación y análisis de grandes volúmenes de datos.

1.3 Inteligencia artificial

En 1950, el matemático inglés Alan Turing, publicó un artículo titulado *Computing Machinery and Intelligence*, este trabajo abrió las puertas al campo que años adelante se denominaría *inteligencia artificial*. El trabajo de Turing inicia con la pregunta «¿Pueden las máquinas pensar?», posteriormente propone un método para evaluar el nivel de inteligencia de una máquina, que se conoce como la prueba de Turing. John McCarthy, conocido como uno de los pioneros de la Inteligencia Artificial, empleó el término por primera vez, en un trabajo para la conferencia de *Dartmouth Summer Research Project on Artificial Intelligence* en 1956, la primera conferencia de inteligencia artificial, y la definió como «la ciencia y la ingeniería de hacer máquinas inteligentes» [30].

Con la inteligencia artificial los datos son el combustible que impulsa la IoT y la capacidad de la organización para derivar el significado de ellos es lo que definirá su éxito a largo plazo. La inteligencia artificial es un tema complejo. Por esa razón, se pueden encontrar distintas definiciones acerca de ella.

En [31] definen IA de una forma simple como «la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana». Pero, para brindar una definición más detallada, definen luego la IA como «la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. Sin embargo, a diferencia de las personas, los dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez. Asimismo, la proporción de errores es significativamente menor en las máquinas que realizan las mismas tareas que sus contrapartes humanas».

Otras de las bibliografías consultadas [32] plantea que «la inteligencia artificial es cualquier técnica que tiene como meta posibilitar a las computadoras imitar el comportamiento humano, incluido el aprendizaje de máquinas (ML), el procesamiento natural del lenguaje (NLP), la síntesis del lenguaje, la visión por computadora, la robótica, el análisis sensorial, la optimización y la simulación».

Dentro de la inteligencia artificial están los campos de Aprendizaje de máquinas (ML, Machine Learning), Visión por computadora (CV, Computer Vision), Lógica difusa (FL, Fuzzy Logic) y Procesamiento de lenguaje natural (NLP, Natural Language Processing), entre otros (figura 1.6). Estos cuatro subcampos están muy relacionados entre sí; principalmente, por los algoritmos que se utilizan para procesar la información o las técnicas que se utilizan para enseñar la información necesaria para crear el modelo que necesitan las diferentes técnicas. Además, estos subcampos se utilizan para crear máquinas o programas más inteligentes que ayudan en la vida diaria, como intenta hacer Internet de las cosas (IoT).

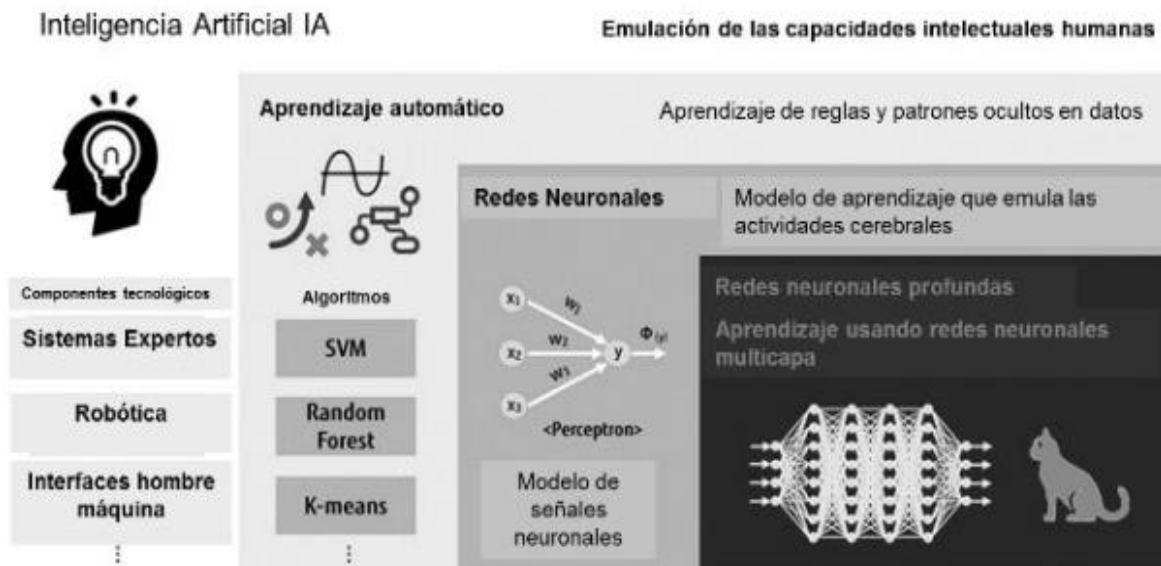


Figura 1.6 Inteligencia artificial y subcampos [30].

La IA en la actualidad está siendo aplicada para plantear y resolver un gran número de problemas en áreas de investigación y de la industria. La inteligencia artificial tiene aplicaciones (figura 1.7) en casi todas las ocupaciones hoy en día. Su alta precisión combinada con la baja latencia que ofrece la computación de borde convierte a la IA en una tecnología de vanguardia. Muchas empresas ya asumen trabajos a nivel de taller en grandes industrias, colocando así a los humanos con un papel más de supervisor. Otras empresas están utilizando IA para analizar datos, hacer análisis y comprar o vender acciones sin la interferencia de ningún ser humano. Algunas de las aplicaciones más habituales de IA son las siguientes:

- Industria de juegos.
- Industrias pesadas.
- Predicción del tiempo.
- Sistemas expertos.
- Minería de datos o extracción de conocimiento.
- Representación del conocimiento.

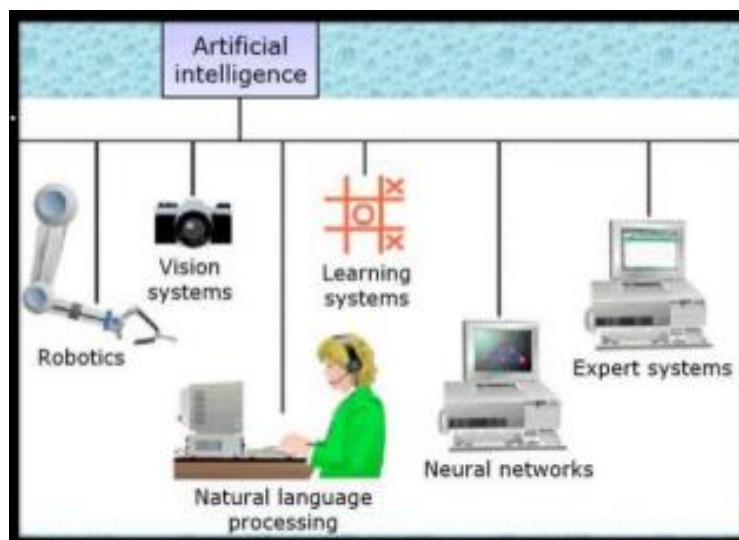


Figura 1.7 Aplicaciones de IA [33].

1.3.1 IA en IoT y en Edge Computing

El concepto de Internet de las cosas actualmente comprende una interrelación de varias disciplinas. Si bien en sus primeras etapas se limitaba al «conexión de dispositivos electrónicos dotados de sensores y conectados a Internet», actualmente involucra el conocimiento de dominios tales como la física newtoniana, la física electrónica, la minería de datos, el *big data* y sobre todo la inteligencia artificial [34].

Internet de las cosas (IoT) está impulsando la integración de los mundos físico y virtual. Los dispositivos con sensores proporcionan rica información sensorial a servicios conscientes del contexto. Los actuadores se implementan en la aplicación de las necesidades del mundo físico. El papel del software está aumentando a medida que más y más recursos se virtualizan y se definen por software. El entorno inteligente es un concepto emergente que tiene el potencial de abordar muchos de los problemas de la humanidad.

Prácticamente, cualquier espacio físico puede convertirse en inteligente: lo que se llama un entorno inteligente. Un entorno inteligente es un espacio físico que emplea tecnología para conectar cosas con el mundo virtual, con el objetivo de aumentar el nivel de conciencia de lo que sucede en los entornos físicos y con las personas (utilizando información aleatoria). Ofrece todo como servicio (XaaS, Anything as a Service), a través de gemelos digitales, permitiendo la analítica de datos para determinar tendencias y revertir los cambios del mundo virtual al físico mediante actuadores IoT. Integra IoT, plataformas de

software/middleware/firmware, servicios, IA, ML, *big data*, computación en la nube, EC, conectividad heterogénea, realidades virtuales/mixtas, y cientos de otras tecnologías para mejorar la calidad de vida de las personas, reduciendo el impacto ambiental y optimizando el uso de los recursos físicos. Los entornos inteligentes añaden valor para las personas, las empresas, la mejora de la sostenibilidad, haciendo eficiente y eficaz el uso de los recursos [35].

Con los avances en Internet de las cosas (IoT), la inteligencia artificial y las tecnologías de la comunicación, la computación de borde ofrece una nueva solución al problema de procesar todos o parte de los datos localmente en el borde de un sistema. Existe una amplia bibliografía que vincula estas tecnologías; se puede citar como ejemplo práctico el diseño, desarrollo, implementación y evaluación de un sistema de vigilancia de estacionamiento inteligente, eficiente y confiable, con inteligencia artificial de borde en dispositivos IoT, propuesto en [36]. En este se hizo una búsqueda exhaustiva de la literatura sobre vigilancia de estacionamiento inteligente y el sistema propuesto fue uno de los primeros esfuerzos en la aplicación de técnicas de computación de borde a la vigilancia en estacionamiento del mundo real. El sistema de procesamiento de canalizaciones y algoritmos fue cuidadosamente diseñado con el propósito de cambiar razonablemente la carga de trabajo informática hasta el límite, reduciendo así significativamente el volumen de transmisión de datos y permitir la detección de estacionamiento en línea eficientemente. El sistema alcanzó el 95,6 % de precisión de detección en general, en diferentes escenarios, incluidos interiores, al aire libre, nublado, lluvioso, soleado, con niebla, oclusión, durante el día y situaciones nocturnas. El diseño tiene múltiples ventajas sobre los sistemas de vigilancia de estacionamiento de última generación y tiene una perspectiva brillante en las aplicaciones de ciudad inteligente y sistemas de transporte inteligente.

Mediante la consideración de los bordes inteligentes o de la inteligencia en los bordes se han desarrollado diferentes escenarios de aplicaciones e implementaciones prácticas de tecnologías establecidas de inteligencia artificial, aprendizaje de máquinas y aprendizaje profundo (DL) en el marco de la computación en los bordes. El DL es adoptado en un gran número de aplicaciones y corrobora su desempeño superior. Actualmente un gran número de unidades procesadoras gráficas (GPU, *graphics processing unit*), unidades de procesamiento tensoriales (TPU, *tensor processing unit*) y matrices de puertas

programables (FPGA, Field Programmable Gate Arrays) son desarrolladas de forma acelerada para ser desplegadas en los bordes de la nube para procesar solicitudes de servicio de aprendizaje profundo. Las aplicaciones de aprendizaje profundo son más valiosas cuando se implementan sobre dispositivos móviles ligeros, tales como teléfonos móviles, dispositivos que pueden ser llevados puestos, cámaras de vigilancia y otros que se pueden colocar cercanos a los eventos. Los dispositivos IoT de borde de baja potencia pueden ser utilizados para encargarse del cómputo ligero del aprendizaje profundo y por ende evitar la comunicación con la nube, pero esto, aun así, necesita aún afrontar limitados recursos de cómputo, un tamaño de memoria y consumo de energía. Por lo que a través de este cuello de botella muchos autores se han centrado en microcontroladores del tipo ARM Cortex-M y CMSIS-NN, los cuales son una colección de módulos eficientes de Redes Neuronales (NN, Neural Networks) [37].

Otros trabajos de investigación han desarrollado métodos en los cuáles descomponen el procesamiento de los datos dividiendo estos de forma inteligente entre los dispositivos y los nodos. Aplicando aprendizaje activo sobre los dispositivos de borde y aprendizaje federado sobre los nodos en la neblina (*fog node*) lo que reduce la muestra de datos a entrenar y reduce los costos de la comunicación [38].

Vivimos en una era en la que la Inteligencia Artificial (IA) se está convirtiendo en una plataforma para el cálculo y la interacción entre máquinas y objetos inteligentes en aplicaciones en tiempo real y muchos aspectos críticos de la vida diaria. Con áreas de aplicación como ciudades inteligentes, redes inteligentes, gestión inteligente del agua, salud inteligente, cadena de suministro inteligente y hogares inteligentes en IoT; se puede considerar la IA como un paquete complementario de los objetos inteligentes en red. Desde esta perspectiva, es fundamental comprender que el papel de la IA va a proporcionar una columna vertebral global para el intercambio y procesamiento de información mundial en el futuro cercano [39].

1.4 Conclusiones del capítulo

En este capítulo se han aclarado conceptos claves para la realización de este proyecto: Internet de las cosas, computación de borde e inteligencia artificial. Tres conceptos muy relacionados entre sí y a partir del dominio de cada uno se empieza a dar solución al

objetivo general de este trabajo: evaluar posibles plataformas de IA vinculadas a la computación en los bordes que puedan ser implementadas en sistemas IoT. En el siguiente capítulo se evalúan varias de estas plataformas.

CAPÍTULO 2. Plataformas de IA vinculadas a la computación en el borde y a IoT

Las plataformas de IA facilitan el desarrollo de aplicaciones inteligentes y de asesoría, incluyendo asistentes inteligentes que pueden imitar las habilidades del conocimiento humano. En conjunto se considera que las plataformas de inteligencia artificial están convirtiéndose en poderosas estructuras para la actividad económica y social en las próximas décadas [40].

El papel del software está aumentando a medida que más y más recursos se virtualizan y se definen por software. El entorno inteligente es un concepto emergente que tiene el potencial de abordar muchos de los problemas de la humanidad. En este contexto, ¿cuáles son las herramientas prometedoras para construir los entornos inteligentes del futuro? En este capítulo se proporciona una revisión de plataformas que pueden ayudar en este gran desafío; se muestran algunos ejemplos de plataformas de IoT y de plataformas vinculadas a la computación en el borde, y se presentan sus arquitecturas, ciclo de vida del servicio, virtualización, seguridad, privacidad, modelo de comunicación, soporte para IA y aprendizaje automático, entre otros aspectos.

Uno de los grandes retos a los que se enfrentan las soluciones de IoT es la interoperabilidad, la capacidad que tienen los sistemas o sus componentes de comunicarse entre sí. Solo hay que pensar en la diversidad de dispositivos que forman este tipo de soluciones donde se combinan diferentes requisitos y protocolos de comunicación. En este aspecto es necesario una colaboración entre fabricantes que permita alcanzar un conjunto de estándares comunes para que la comunicación entre dispositivos no se vea afectada por limitaciones técnicas o comerciales. De la misma forma, las plataformas de desarrollo

también tienen que propiciar esa interoperabilidad entre dispositivos para impulsar y facilitar el despliegue de estas soluciones.

La no interoperabilidad da lugar a la duplicación del trabajo, a soluciones incompatibles y uso de recursos no optimizados. Una plataforma de software novedosa y completa podría proporcionar dicho servicio, incluidas las instalaciones para el desarrollo de aplicaciones, la integración, la implementación, y gestión para facilitar la construcción de aplicaciones sofisticadas, por ejemplo, para ciudades inteligentes (Smart Cities) [41].

2.1 Plataformas de IoT con IA

Hoy en día, Internet de las cosas (IoT) se asocia principalmente con sistemas integrados verticalmente que a menudo son cerrados y fragmentados en su aplicabilidad. Para construir un mejor ecosistema de IoT, la plataforma abierta de IoT se ha convertido en un término popular en los últimos años. En [42] se caracterizan los tipos de apertura de las plataformas de IoT y se investiga qué las hace abiertas, asimismo se presentan 46 plataformas de IoT que se han caracterizado como abiertas; mediante la revisión de más de 200 artículos se llega a la conclusión de que las plataformas de IoT abiertas más ampliamente aceptadas y utilizadas son NodeMCU y ThingSpeak, que juntas tienen una participación de más del 70 % de las plataformas de IoT abiertas declaradas en los artículos seleccionados.

Las plataformas de IoT juegan un papel importante en la ayuda a la integración y el desarrollo de sistemas de IoT. Nuevas plataformas de IoT emergen constantemente y estas plataformas ya ofrecen formas heterogéneas de acceder a ellas y a sus datos [43]. Según Mineraud et al. [44] una «plataforma de IoT se define como el *middleware* y la infraestructura que permite a los usuarios finales interactuar con objetos inteligentes».

Hay diferentes tipos de plataformas disponibles que a menudo se denominan plataformas de IoT, como las de dispositivo a dispositivo (*device-to-device*), plataformas basadas en la nube (*cloud-based*) y plataformas de dispositivo a nube (*device-to-cloud*) [45-47].

Los desarrolladores suelen tener dificultades para alcanzar una masa crítica e incluso los usuarios finales necesitan navegar a través de diferentes marcas y comprender qué dispositivos son compatibles en relación con qué plataformas de IoT. Las plataformas de

IoT comerciales o patentadas llevan un modelo de precios y a menudo promueven la dependencia del proveedor. Por lo tanto, a menudo los proveedores de plataformas de IoT carecen de soporte para nuevos protocolos, herramientas y formatos de datos a tiempo, debido a un panorama de IoT en constante cambio [48].

2.1.1 Ejemplos

En el estudio realizado en [42] se definen dos cuestiones importantes: qué plataformas de IoT han sido caracterizadas como *abiertas* y cuáles son los tipos de apertura de las plataformas de IoT. Luego de definir una serie de características específicas muestran como resultado que hay 25 plataformas de IoT caracterizadas como *abiertas*, las cuales fueron categorizadas como *usadas* e *indicadas* (tabla 2.1). Además, 21 plataformas de IoT, que fueron categorizadas en *propuestas*, como se presenta en la tabla 2.2, han sido caracterizadas como *abiertas*, lo que suma un total de 46 plataformas abiertas de IoT identificadas en dicho estudio. Estas plataformas se utilizan en diferentes dominios e implementaciones, y se observa que hay siete plataformas de IoT abiertas que se utilizan principalmente dentro de la comunidad de investigación, mientras que las otras plataformas se utilizan solo una o dos veces. Por ejemplo, la tabla 2.3 muestra las plataformas de IoT abiertas más utilizadas: NodeMCU, ThingSpeak, FIWARE, Mobius, Kaa, OpenIoT y ThingsBoard. Dicho estudio también sugiere que el tipo de apertura más común es el de código abierto, pero en la literatura también se utilizan estándares abiertos, Interfaz de Programación de Aplicaciones (API, Application Programming Interface) abiertas y capas abiertas.

Tabla 2.1 Plataformas de IoT abiertas identificadas en las categorías *usadas* e *indicadas* [42].

Plataformas <i>usadas/indicadas</i>				
1. NodeMCU	6. Mobius	11. Arduino Raspberry Pi	16. Emon	21. FLIP
2. ThingSpeak	7. ThingsBoard	12. Arduino Uno ESP8266	17. MediaSense	22. IoTmakers
3. OpenIoT	8. IoTivity	13. Mobius	18. RIOT	23. Tridium

4. FIWARE	9. OIiot	14. Contiki OS	19. Tacit	24. VITAL-OS
5. KAA	10. OM2M	15. Cosm OS	20. Heroku	25. Californium

Tabla 2.2 Plataformas de IoT abiertas identificadas en la categoría de *propuestas* [42].

Plataformas
propuestas

1. ComfortBox	8. OPEL	15. Waziup
2. bIoTope	9. OpenIoT	16. Liu and Nielsen [49]
3. EverySense	10. SensorCentral	17. Jinbo et al. [50]
4. HANDYPIA	11. IoT Manager	18. Jeon et al. [51]
5. IoTEP	12. Snap4City	19. Park et al. [52]
6. KIBAN	13. SWAMP	20. Kianoush et al. [53]
7. MONICA	14. viota	21. Andreev [54]

Tabla 2.3 Comparación de la definición real de las plataformas abiertas de IoT más usadas [42].

Plataformas	Descripción original
NodeMCU	<i>Firmware</i> de código abierto y kit de desarrollo que ayuda a crear un prototipo de un producto de IoT en unas pocas líneas de comandos de Lua (https://www.nodemcu.com).
ThingSpeak	Plataforma IoT abierta con análisis de MATLAB (https://thingspeak.com).
Mobius	Plataforma de servidor IoT de código abierto basada en el estándar oneM2M. Plataforma de servidor IoT basada en Node.js (http://developers.iotocean.org/archives/module/mobius).

FIWARE	Plataforma de código abierto para nuestro futuro digital inteligente (https://www.fiware.org).
KAA	La plataforma IoT más flexible para su negocio (https://www.kaaproject.org).
OpenIoT	<i>Middleware</i> de código abierto para obtener información de los sensores nubes, sin tener que preocuparse por los sensores exactos usados (http://www.openiot.eu).
ThingsBoard	Plataforma IoT de código abierto para recopilación, procesamiento y visualización de datos, y gestión de dispositivos (https://thingsboard.io).

A continuación se especifican varias características importantes como arquitectura, modelo de comunicación, resolución de nombres, ciclo de vida del servicio, representación de entidades, seguridad, etc., de algunas de las plataformas antes mencionadas:

➤ **FIWARE**

FIWARE [55, 56] es la plataforma tecnológica de la iniciativa europea FI-PPP. Es un ecosistema abierto que utiliza una plataforma de software estandarizada para facilitar el desarrollo de aplicaciones inteligentes en varios sectores, incluido el IoT y ciudades inteligentes. La plataforma FIWARE integra servicios utilizando interfaces de servicio de próxima generación (NGSI, New Generation Service Interface) como pegamento. La arquitectura permite agregar nuevos servicios como habilitadores genéricos (GE). Los GE ofrecen diversas funcionalidades, implementando protocolos e interfaces para planos de control y datos.

1) Arquitectura y Cloud Computing:

La figura 2.1 ilustra una descripción general simplificada de los componentes arquitectónicos de FIWARE para IoT, que incluyen: (i) agente de IoT (IoT broker); (ii) gestión de dispositivos de *back-end* (BDM); (iii) intermediario de contexto (CB); (iv) análisis de *big data* (BDA); y (v) procesamiento de eventos complejos (CEP). Antes de describirlos, es importante observar que se admiten tres tipos de dispositivos: (a) dispositivos que son compatibles con la interfaz de servicio de próxima generación (NGSI) versión 9/10; (b) dispositivos que no son compatibles con NGSI 9/10, sin embargo, las

puertas de enlace sí; y (c) dispositivos y pasarelas que no son compatibles con NGSI-9/10. El agente de IoT recupera, recopila y procesa información de cosas que exponen los dispositivos como recursos RESTful [57]. La BDA expone tecnologías heredadas (estandarizadas o propias) como recursos para el CB a través de NGSI-9/10. Los agentes de IoT instan a manejar, configurar y monitorear dispositivos y puertas de enlace no NGSI. El CB proporciona un servicio de intermediario de contexto de publicación/suscripción a través de la interfaz NGSI-9/10. Los contextos pueden ser registrados, actualizados, consultados, notificados, suscritos, etc. Por ejemplo, un dispositivo nativo NGSI-9/10 puede crear un contexto que lleve el valor actual de la temperatura en una determinada habitación. BDA es una versión extendida de *hadoop* de Telefónica (llamado Cosmos). Finalmente, CEP es un habilitador genérico de IBM para correlacionar eventos en tiempo real según las reglas programadas. Los datos generados por CEP o BDA se publican en CB. BDA es alimentado por CB y procesa datos de CEP. Por lo tanto, FIWARE permite casi en tiempo real mapear/reducir operaciones sobre grandes cantidades de datos de IoT [35].

2) Modelo de comunicación:

El agente de contexto FIWARE Orion admite dos modos *pub/sub*: (i) método empujar (*push*) para enviar información al corredor; (ii) tirar (*pull*) para enviar alguna información solicitada por el corredor. El método de empuje es para el caso en el que el corredor no solicite la información. En este caso, los clientes suscritos reciben la información de forma continua, cada vez que se actualiza. El método de empuje es similar a la primitiva de publicación empleada en los agente de mensajes MQTT (Message Queing Telemetry Transport) [35].

3) Interoperabilidad de nomenclatura, resolución de nombres y semánticas:

FIWARE retransmite capacidades de resolución de nombres y nomenclaturas DNS. Los nombres de los recursos se exponen a través de NGSI. Ramparany et al. [55] han discutido la importancia del modelado semántico y la orquestación basada en semántica en plataformas de IoT, presentando cómo FIWARE soporta componentes basados en semántica para el desarrollo de aplicaciones de IoT. El caso de uso de una aplicación de farolas se ha implementado, desplegado y evaluado utilizando FIWARE. An et al. [58] han proporcionado una solución para interconectar FIWARE a la plataforma oneM2M. Se ha

desarrollado un enfoque de integración basado en semántica para permitir mapeo estático de datos de detección entre plataformas, así como interoperabilidad semántica dinámica a través de un proxy semántico. Se ha evaluado una prueba de concepto en la ciudad de Santander, España.

4) Ciclo de vida del servicio y soporte para IA:

FIWARE engloba un catálogo que apoya de forma transparente adición de nuevos servicios como habilitadores genéricos. El ciclo de vida del servicio es compatible con las funciones de OpenStack. FIWARE incluye soporte nativo para *big data* y análisis. También proporciona inteligencia empresarial a través del habilitador genérico de Knowage [35].

5) Representación de entidades:

La gestión de dispositivos FIWARE contiene una representación de software (llamada agente de IoT) para cada dispositivo *back-end* conectado. Las interfaces NGSI-9/10 se proponen para puertas de enlace, incluso se admiten otras interfaces. Aparentemente, los temas de MQTT empleados en muchos *middleware* de IoT son equivalentes a los mecanismos *pub/sub* de FIWARE CB que utilizan la interfaz NGSI-9/10 [35].

6) Seguridad, privacidad y confianza:

La seguridad de FIWARE depende del secreto de las interfaces NGSI-9/10 y autenticación. Además, FIWARE ofrece varios habilitadores de seguridad: (i) autenticación que preserva la privacidad, que emite credenciales, políticas de presentación y verificadores para calcular *tokens* de acceso; (ii) gestión de identidad de KeyRock para manejar varios problemas relacionados con el acceso de los usuarios a la plataforma; (iii) AuthZForce proporciona una API para obtener decisiones de autorización basadas en políticas de autorización y solicitudes de autorización de los puntos de aplicación de políticas (PEP). La formación de confianza se limita a estos habilitadores. Alonso et al. [59] han propuesto un modelo para conectar los servicios FIWARE autenticados con OAuth 2.0 a la identificación electrónica y los servicios de confianza (eIDAS), por lo que cumplen con las recientes normativas de protección de datos/privacidad.

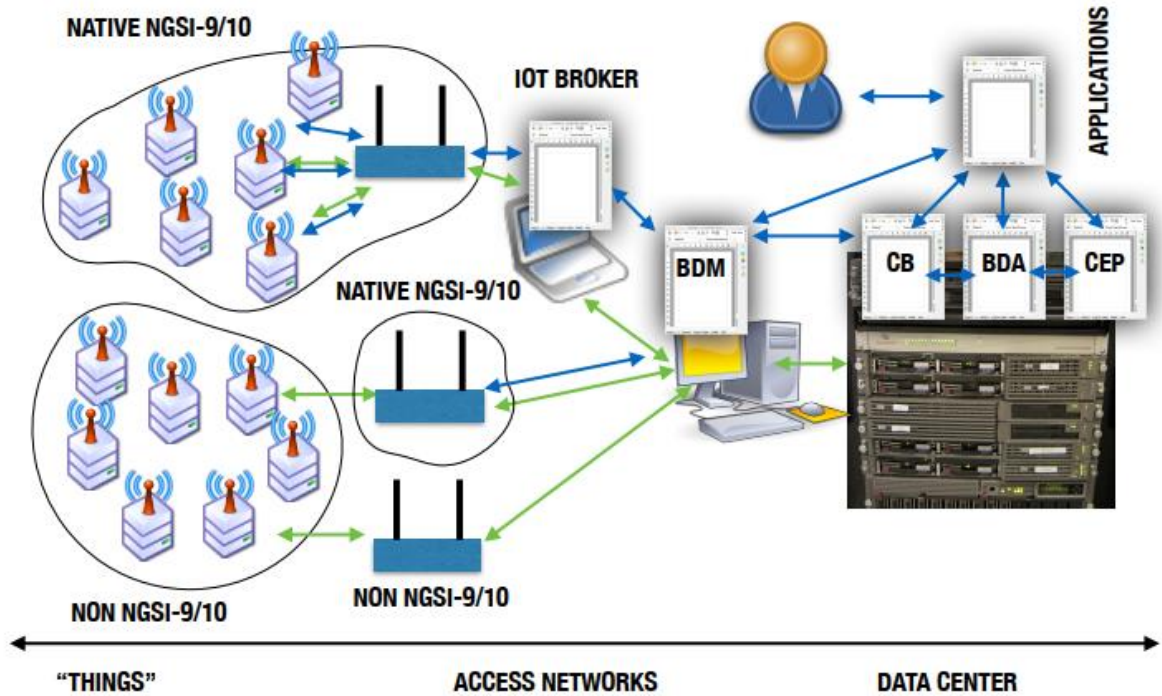


Figura 2.1 Escenario FIWARE para cubrir Internet futuro (FI, Future Internet), IoT, *big data*, y Cloud Computing [35].

➤ OpenIoT

OpenIoT es un proyecto europeo financiado por el FP7 con el objetivo de proporcionar una plataforma de *middleware* para la interoperabilidad semántica de escenarios heterogéneos de IoT integrados en la nube informática. OpenIoT aborda el problema de combinar flujos de datos y servicios de diferentes escenarios de IoT. Por ejemplo, flujos de datos con diferentes unidades, valores de sensor sin procesar, etc. Su objetivo es recopilar, filtrar, contextualizar y seleccionar datos de dispositivos de detección heterogéneos. Los conjuntos de datos están vinculados mediante el concepto de datos vinculados. También incluye herramientas de visualización para facilitar el desarrollo de aplicaciones IoT basadas en la nube.

1) Arquitectura y Cloud Computing:

La arquitectura OpenIoT comprende siete componentes: (i) un *middleware* de sensores llamado red global de sensores extendida (XGSN) para recopilar, filtrar y agregar flujos de datos de sensores físicos o virtuales que emplean un agente móvil *pub/sub*; (ii) un almacenamiento de base de datos en la nube llamado *middleware* de flujo vinculado ligero

(LSM-*light*) para almacenar flujos de datos y metadatos de *middleware* de sensores; (iii) un programador para la implementación de servicios bajo demanda y darles acceso a flujos de datos; (iv) un servicio de entrega y un gerente de servicios públicos (SD&UM) para la combinación y gestión de flujos de datos impulsados por servicios; (v) un componente para la especificación de solicitudes de servicio (con interfaz de usuario), que permite la programación de nuevos servicios en el programador; (vi) un componente para seleccionar los *scripts* apropiados mientras se visualizan los resultados de los servicios; y (vii) un componente para el funcionamiento de los dispositivos. Como ejemplo de aplicación de OpenIoT, Medvedev et al. [60] han desarrollado una arquitectura para informar de incidentes en las calles de la ciudad inteligente. Ubicación, marca de tiempo, notas de voz, y las grabaciones de video se envían por teléfonos inteligentes al *middleware* de OpenIoT que se ejecuta en un centro de datos local. Los informes se almacenan en una base de datos. El reconocimiento de voz y el razonamiento se aplican a entregar informes para su procesamiento manual cuando sea necesario.

2) Modelo de comunicación:

OpenIoT se retransmite en un *middleware* de publicación/suscripción llamado *middleware de publicación/suscripción basado en la nube para IoT* (CUPUS). CUPUS tiene dos componentes: (i) un corredor móvil que ejecuta en dispositivos móviles para recopilar datos de sensores; (ii) una nube servicio de procesamiento de datos. CUPUS ofrece varios mecanismos de suscripciones, que evitan la transferencia de datos irrelevantes a la nube. CUPUS ofrece procesamiento *pub/sub* elástico en la nube [35].

3) Interoperabilidad de nomenclatura, resolución de nombres y semánticas:

OpenIoT mejora las ontologías existentes para tratar la anotación semántica de unidades de medida, datos brutos del sensor y puntos de interés para ellos. El software LSM implementa la ontología de la arquitectura, transformando los datos de los sensores virtuales (objetos inteligentes) en datos vinculados, que se almacenan en formato de descripción de recursos (RDF). Estos datos semánticos anotados son consultados utilizando SPARQL. Las lecturas del sensor son accesibles por varias API, que se pueden ampliar para hacer frente a nuevos tipos de sensores. La nomenclatura y la resolución de nombres siguen las actuales tecnologías de Internet [35].

4) Ciclo de vida del servicio y soporte para IA:

Soldatos et al. [61] no comentan sobre la composición dinámica de servicios. Aparentemente, los servicios de la plataforma son fijos y se introducen manualmente. La plataforma no incluye componentes para *AI/ML/Big data*.

5) Representación de entidades:

OpenIoT admite redes de sensores semánticos (SSN) W3C estándar para representar sensores del mundo físico y virtual. Una representación de sensor combina la visión de sus medidas, características, funcionalidades y cómo procesa los datos. La actualización de sensores virtuales (objetos inteligentes) también se basa en envoltorios, como comunicación en serie, UDP, HTTP, etc. [35].

6) Seguridad, privacidad y confianza:

OpenIoT proporciona un módulo de privacidad y seguridad para la gestión, autenticación y autorización del usuario. Los usuarios autenticados están representados por objetos *token* con tiempo de vencimiento. El token se reenvía de un servicio a otro según la cadena de servicios. La implementación está hecha utilizando OAuth2.0. Se implementa la autorización por servicio utilizando Apache Shiro [35].

2.2 Plataformas vinculadas a la computación en el borde

La computación en el borde proporciona a organizaciones una plataforma perfecta para desplegar aplicaciones y servicios de nueva generación donde se concentra la demanda, tanto propia como de sus clientes. A su vez, gracias a la tecnología de la información (TI) en el borde pueden contar con nuevas tecnologías como sensores, cámaras y todo tipo de dispositivos IoT que generan gran cantidad de datos. Ante la gran avalancha de datos que recogen en estas localizaciones, la estrategia más inteligente es adoptar soluciones de computación en el borde que puedan no solo recoger los datos y enviarlos, sino procesarlos y alimentar directamente las aplicaciones en el borde, proporcionando servicios directamente a los usuarios finales. Esto implica contar con una plataforma capaz de ejecutar aplicaciones como la inteligencia artificial, que tiene unos requisitos de rendimiento elevados, en muchos casos con un coste importante. Para reducirlos, los fabricantes de tecnología están desarrollando nuevas plataformas de computación y

almacenamiento más asequibles y con capacidades avanzadas, que proporcionen soluciones de IA en el borde para aplicaciones de cualquier escala [62].

2.2.1 Ejemplos

Un ejemplo excelente es la alianza que han formado el proveedor de soluciones de almacenamiento NetApp y el fabricante de equipos informáticos Lenovo, que han sumado sus fuerzas para construir una plataforma de inteligencia artificial *edge* potente, sólida, asequible y escalable, adaptada a las necesidades de clientes de todos los tamaños. Para ello han diseñado un sistema que combina las matrices de almacenamiento NetApp AFF C190 con los servidores Lenovo ThinkSystem SE350, conectados mediante un conmutador Ethernet 10 GbE. Según [62], NetApp y Lenovo explican que su solución es una arquitectura de escalamiento horizontal flexible, ideal para implementaciones de inferencia de IA empresarial. Y destacan que el almacenamiento de NetApp ofrece el mismo rendimiento o más que el almacenamiento SSD local, proporcionando una serie de beneficios específicos para los científicos de datos, los ingenieros de datos y los tomadores de decisiones de TI:

- Fácil intercambio de datos entre sistemas de IA y análisis.
- Computación y almacenamiento escalables por separado.
- Reducción de los costes del almacenamiento.
- Satisfacción de las necesidades comerciales, cada vez más exigentes.

En el caso de [63] se ve un ejemplo de una plataforma industrial. La implementación de la automatización industrial basada en la IA en el borde de red es crucial en las aplicaciones de misión crítica que requieren el procesamiento de datos en tiempo real y una respuesta de baja latencia. Por ejemplo, el sistema de visión industrial basado en la IA en el borde de red de la fábrica puede procesar el análisis de datos de visión, la inferencia de aprendizaje profundo y el reaprendizaje, lo que resulta en la mejora inmediata de la precisión de la detección de defectos, sin transmitir grandes volúmenes de datos a la nube, lo que conlleva mayores gastos generales en el ancho de banda de la red y el consumo de energía. Sin embargo, el despliegue de un sistema de visión artificial basado en la inteligencia artificial requiere un viaje largo, que requiere una integración total de la cámara de alta velocidad, el hardware de computación de la inteligencia artificial con el acelerador y los bloques de

construcción de software adaptados a la aplicación. Para acelerar el tiempo de implementación de la IA en el borde de red industrial, Lanner proporciona el kit de IA consolidado para el borde de red industrial que incluye la plataforma de computación de borde de red de Lanner LEC-2290 impulsada por CPU Intel® Coffee Lake y VPU Movidius Myriad X, Vision & Analytics Suite consiste en Intel® Edge Insights para SW Industrial, aplicaciones de referencia IoT y kit de herramientas OpenVINO, y cámara de red.

Otra de las plataformas que se pueden citar por su uso para el cómputo de borde es EdgeX Foundry [64]. Esta es un proyecto de código abierto dentro de Linux Foundation que surgió en abril del 2017 con el apoyo de 50 organizaciones miembros, y que actualmente cuenta con más de 70. Para ponerlo en marcha Dell aportó más de una docena de microservicios y 125 000 líneas de código en Apache 2.0. El proyecto tiene como objetivo contribuir a la estandarización de IoT creando un marco de referencia común para Edge Computing que permita la interoperabilidad entre las aplicaciones y los estándares de conectividad existentes.

La tendencia emergente de la informática de borde ha llevado a varios proveedores de “nube” a lanzar sus propias plataformas para realizar computación en el borde de la red. Tal es el caso de estas dos plataformas tratadas en [65]: Amazon AWS Greengrass y Microsoft Azure IoT Edge. En dicho artículo se presenta EdgeBench, un paquete de referencia para plataformas informáticas de borde sin servidor. Con este, se estudian dos plataformas de computación de borde administradas, Greengrass y Azure Edge. Además, se compara el rendimiento de estas plataformas con la de las implementaciones solo en la nube de los mismos puntos de referencia. Se obtienen resultados que muestran que el rendimiento de Greengrass y Azure Edge son comparables, con la excepción de que Azure Edge exhibe un mayor nivel de latencia de extremo a extremo debido a su enfoque de procesamiento por lotes. Más lejos, otros resultados muestran que para las canalizaciones de imágenes y escalares, el rendimiento de Greengrass es comparable al de las canalizaciones de AWS solo en la nube, al tiempo que reduce el uso de ancho de banda de la red. Estos resultados indican que la computación de borde es una alternativa prometedora a la computación en la nube para cargas de trabajo livianas de la CPU.

Otro ejemplo se manifiesta por Red Hat Inc., el proveedor líder mundial de soluciones de código abierto, que ha presentado nuevas funcionalidades y mejoras en la plataforma empresarial de Linux líder a nivel mundial, que sitúan a Red Hat Enterprise Linux como una potente plataforma para la nube híbrida abierta, desde el centro de datos hasta el *edge*. Red Hat Enterprise Linux 8.4 simplifica el papel del sistema operativo como una plataforma ligera para los despliegues en producción en el *edge*, añadiendo nuevas funcionalidades de gestión y despliegue de contenedores Linux escalables para cubrir las necesidades del EC. Red Hat Enterprise Linux es la base para la iniciativa Red Hat Edge, cuyo objetivo es ampliar las funcionalidades del portfolio de nube híbrida abierta de Red Hat al *edge* en todos los sectores de telecomunicaciones y transporte hasta los coches inteligentes y los dispositivos empresariales. Red Hat Enterprise Linux 8.4 es más que una base para el *edge*, incorpora nuevas y mejoradas funcionalidades para ayudar a las organizaciones a crear, desplegar y mantener estrategias de nube híbrida de una forma más eficiente, desde el centro de datos hasta la nube. «La nube híbrida abierta no se limita a un centro de datos empresarial o a entornos de nube pública; incluye servidores remotos, equipos avanzados y otros dispositivos que existen en los extremos de la red empresarial. La naturaleza dispar de estos entornos significa que la consistencia es fundamental para el éxito: Red Hat Enterprise Linux, como columna vertebral de la iniciativa Red Hat Edge, proporciona esta plataforma consistente, nativa del *edge* e inteligente para satisfacer las demandas dinámicas de la nube híbrida, desde los servidores básicos hasta la nube y el *edge*», ha declarado Stefanie Chiras, vicepresidenta mayor y directora general de Red Hat Enterprise Linux Business Unit [66].

Por último, en [67] se analiza la plataforma MEC (Mobile Edge Computing/Multi-access Edge Computing) que brinda procesamiento cooperativo entre un servidor de borde y un dispositivo móvil. El enfoque se centra en el requisito de una comunicación continua y transparente, así como en la reducción de la latencia para mejorar el procesamiento de descarga entre el servidor de borde y el dispositivo móvil. La plataforma propuesta utiliza una puerta de enlace de soporte de movilidad IP en cada dominio de red para lograr este requisito y establecer una relación de comunicación entre estas puertas de enlace controlando a través de orquestadores.

2.3 Conclusiones del capítulo

A lo largo del capítulo se ha brindado información y bibliografías citando varias plataformas que pueden servir para estudios de otros proyectos: plataforma de **NetApp y Lenovo**, la de **Lanner**, **EdgeX Foundry**, **Amazon AWS Greengrass**, **Microsoft Azure IoT Edge**, **Red Hat Enterprise Linux** y **Multi-access Edge Computing**. Como se ha podido comprobar dichas plataformas se presentan como una alternativa muy interesante para aquellas organizaciones que quieran implementar un sistema de IoT siguiendo el modelo de computación propuesto por la computación de borde.

CAPÍTULO 3. Resultados y evaluación de la plataforma de borde EdgeX Foundry

En el capítulo anterior se menciona muy brevemente la plataforma EdgeX Foundry. En este capítulo se amplía más la información de dicha plataforma, seleccionada por ser altamente flexible y neutral para el proveedor que permite a los desarrolladores crear aplicaciones que se ejecutan en el borde, actuando como un *software* intermedio entre las cosas y la nube.

3.1 EdgeX Foundry

EdgeX Foundry pone el foco en los sistemas de IoT industriales donde la complejidad, la fragmentación del mercado y la falta de un *framework* común está dificultando su adopción. Esta iniciativa se centra en componentes de capacidades limitadas distribuidos geográficamente, cualquier elemento capaz de recopilar y procesar datos podrá hacer de enlace entre los dispositivos de IoT y los sistemas en la nube. No llega al nivel del sensor, pero el proyecto asegura que los dispositivos (*routers*, *switches*, servidores, etc.) que los conectan, hablen todos el mismo idioma (figura 3.1).

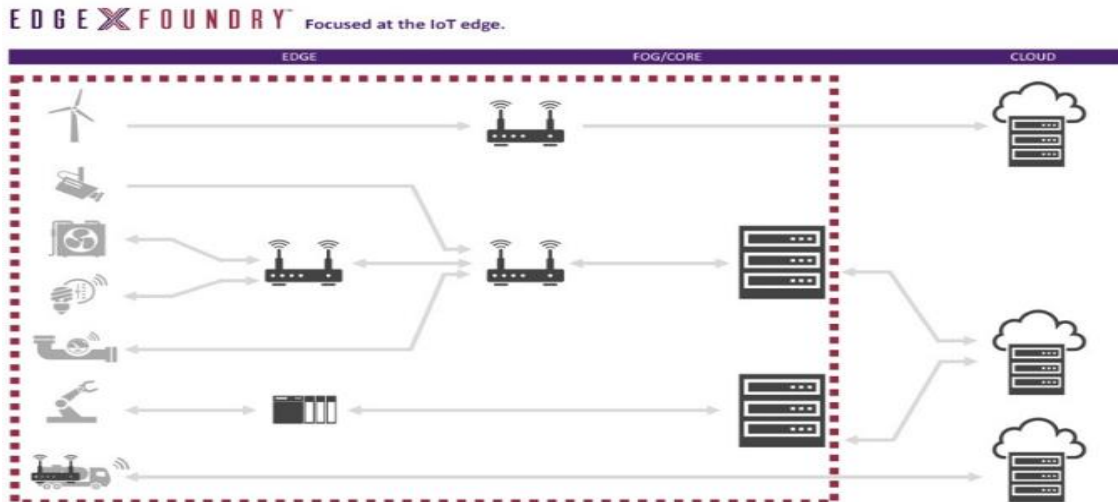


Figura 3.1 EdgeX Foundry centrado en dispositivos IoT Edge [64].

De esta forma, mediante un ecosistema de componentes interoperables definidos por *software*, independientes del *hardware* y del sistema operativo, las organizaciones podrán integrar con facilidad los estándares seleccionados para los dispositivos de IoT con sus soluciones de *backend* (locales o en la nube).

3.1.1 Objetivos

Los objetivos del proyecto EdgeX Foundry [64] pasan por:

- Fomentar el crecimiento de las soluciones de IoT industriales a través de una comunidad que crea y mantiene servicios y API comunes.
- Promocionar EdgeX como una arquitectura abierta y común para el uso de Edge Computing.
- Alentar a la comunidad de proveedores de soluciones de IoT industriales a colaborar con la creación de un ecosistema de componentes *plug-and-play* interoperables.
- Certificar los componente de EdgeX para garantizar su interoperabilidad y compatibilidad.
- Proporcionar herramientas para crear de forma rápida soluciones *edge* para el mercado de IoT industrial basadas en componentes de EdgeX que se puedan adaptar fácilmente a las necesidades cambiantes de los negocios.

- Colaborar con proyectos de código abierto, grupos de estándares, y crear alianzas con la industria para garantizar la coherencia y la interoperabilidad en todos los niveles de IoT.

3.1.2 Organización

Se distinguen tres categorías de miembros dentro de la comunidad del proyecto: Platinum, Silver y Associate [64]. Sin embargo, como proyecto de código abierto se da la bienvenida a cualquiera que contribuya y se involucre en la comunidad de desarrollo [68]. Solo los miembros Platinum y Silver podrán formar parte del Consejo Directivo del proyecto. Los miembros Associate se limitan a organizaciones sin ánimo de lucro, proyectos de código abierto y entidades gubernamentales, que requieren la aprobación del Consejo Directivo. Los tres tipos de miembros tienen derecho a participar en reuniones generales del proyecto, iniciativas y cualquier otra actividad, de la misma forma podrán identificarse como miembros o participantes del proyecto EdgeX Foundry.

Tal como se muestra en la figura 3.2 el Comité de Dirección Técnica (Technical Steering Committee - TSC) está compuesto por grupos de trabajo, cada uno de estos grupos es una unidad organizativa centrada en un dominio técnico específico (Seguridad, DevOps, SDK, QA y Test...). A su vez, cada grupo de trabajo se encarga de uno o más proyectos, y cada proyecto entregará un elemento de trabajo, como puede ser un nuevo desarrollo o la modificación de un desarrollo existente. Todos los proyectos tienen una página en EdgeX Foundry Wiki [69]. Compañías miembros, como Dell, tienen equipos de desarrollo trabajando exclusivamente para el proyecto.

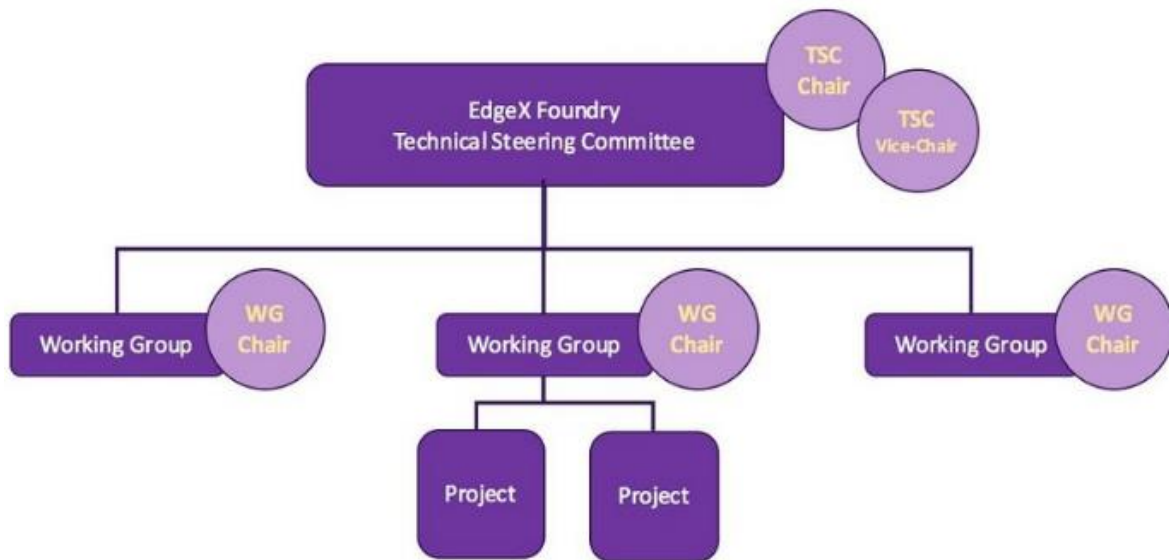


Figura 3.2 Organización del proyecto EdgeX Foundry [64].

3.1.3 Arquitectura

EdgeX Foundry se define a través de una arquitectura modular basada en microservicios [70] que se comunican a través de API, lo que permite convertir aplicaciones complejas en procesos simples, interoperables y reutilizables, que se pueden modificar e implementar de forma independiente, y que pueden ser desplegados con facilidad. Las arquitecturas basadas en microservicios permiten un escalado eficiente y horizontal en función de la demanda.

El código del proyecto se puede obtener desde los repositorios de GitHub [71], los microservicios que ofrece EdgeX se pueden desplegar desde varios puntos de vista:

- Visión de los contribuyentes: enfoque dirigido a desarrolladores, estos pueden descargar el código del proyecto y generar e implementar servicios que pueden ser desplegados en las plataformas elegidas.
- Visión de los usuarios: está dirigido a usuarios que no tienen la necesidad de construir sus propios servicios, así que se pueden descargar las imágenes de contenedores de EdgeX y desplegarlas y ejecutarlas en plataformas que tengan Docker [72] instalado lo que les proporciona las capacidades mínimas de una plataforma edge.
- Visión híbrida: es una mezcla de las dos anteriores, algunos servicios pueden ser implementados y desplegados utilizando el código del proyecto, y otros servicios se

desplegan y ejecutan usando las imágenes de contenedores de Docker que ofrece EdgeX.

En la figura 3.3 se muestra la organización de estos microservicios que definen la arquitectura, el sistema se distribuye en cuatro capas o niveles (Servicios Principales, Servicios de Apoyo, Servicios de Exportación y Servicios de Dispositivos), y en dos servicios subyacentes a todo el sistema (Seguridad y Administración).

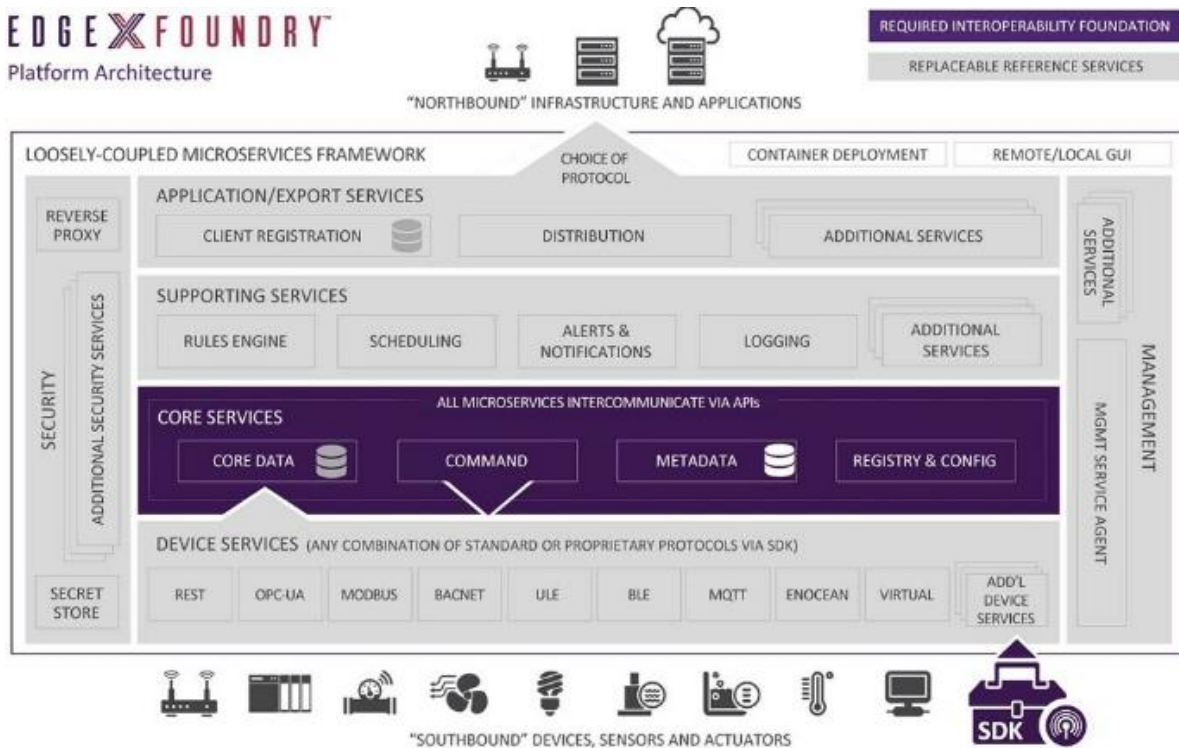


Figura 3.3 Arquitectura de EdgeX Foundry [64].

En el “Southbound” o “lado sur” se encuentran todos los objetos de IoT, la parte física, que engloba a todo elemento que se comunica con sensores, actuadores y otros dispositivos de IoT para recopilar información de ellos. En el extremo superior, “Northbound” o “lado norte”, se encuentra las infraestructuras remotas, este lado incluye la parte de red que se comunica con la nube, donde se almacenan y analizan los datos para convertirlos en información. Las capas de la arquitectura proporcionan un motor de transformación, los datos provenientes de sensores y otros dispositivos se traducen y se les da formato, posteriormente son entregados a aplicaciones y sistemas en la nube, teniendo en cuenta los protocolos y estructuras seleccionadas por el cliente.

Cada uno de los niveles que forma la arquitectura de EdgeX Foundry [64] proporcionan una serie de microservicios y funciones:

- **Servicios de Dispositivos:** los microservicios de este nivel comunican y recopilan datos de sensores, actuadores y otros dispositivos de IoT a través de los protocolos nativos de cada elemento, una vez recopilados son convertidos a una estructura de datos común de EdgeX Foundry para ser enviados al nivel superior o a otros microservicios. La plataforma proporciona un kit de desarrollo SDK para facilitar la creación de servicios de dispositivos propios.
- **Servicios Principales:** este nivel separa la capa del lado norte de la capa del lado sur. Entre sus funciones podemos destacar que proporciona almacenamiento persistente temporal y un servicio de administración para los datos recopilados por sensores y dispositivos de IoT en el nivel inferior, y además, ofrece un servicio que facilita y controla las solicitudes y peticiones que se envían del lado norte al lado sur.
- **Servicios de Apoyo:** este nivel abarca un gran número de microservicios que proporcionan “inteligencia” y análisis cerca de los dispositivos *edge*. Si se piensa en la protección de maquinaria industrial por sobrecalentamiento, capas inferiores recolectan y transforman a una estructura común los niveles de temperatura, en este nivel se analizan “localmente”, si se sobrepasa un umbral establecido se emite una orden que pasa al nivel inferior en forma de comando para “apagar” el equipo. Todo ello, sin tener que enviar la información a infraestructuras remotas en la nube para su análisis.
- **Servicios de Exportación:** una de las características de EdgeX Foundry es la capacidad de operar independientemente de otros sistemas cuando sea necesario, a través de *gateways* que a menudo puedan actuar en entornos aislados o desconectados del “lado norte” durante largos periodos de tiempo. Los microservicios de este nivel realizan actividades como: permitir a los clientes registrar los datos relevantes para ser analizados en local, enrutar parte de los datos y de la “inteligencia” recolectada por niveles inferiores hacia la nube o sistemas remotos, etc.

Los servicios subyacentes a todo el sistema, Seguridad y Administración del Sistema [64], proporcionan:

- Mecanismos de protección para datos, sensores y otros objetos de IoT administrados por EdgeX Foundry.
- Administración central para iniciar/detener/reiniciar los servicios de EdgeX y métricas que permiten monitorizar dichos servicios.

3.2 Caso de estudio

EdgeX Foundry es un colección de microservicios que pueden ser desplegados para proporcionar las capacidades de una plataforma *edge* mínima. El código fuente de estos microservicios puede ser descargado para que los desarrolladores interesados puedan construir y desplegar sus propios componentes [64].

En este caso de estudio se pretende validar desde un punto de vista práctico la documentación del proyecto EdgeX Foundry [73] analizando la funcionalidad de los microservicios que el proyecto pone a disposición de cualquier interesado.

3.2.1 Instalación de Edgex Foundry

La instalación de dicha plataforma se lleva a cabo en una computadora con sistema operativo Ubuntu Desktop 18.04 con 4 Gb de RAM y se siguen las instrucciones del tutorial [74]. Se realiza, primero, la instalación de complementos (*snaps* en inglés) según las instrucciones de [75]; el *snap* se puede instalar usando *snap install*. Luego para instalar la última versión estable se utiliza: **\$ sudo snap install edgexfoundry**. También puede especificar lanzamientos específicos usando la opción **--channel**. Por ejemplo, para instalar la versión de Irlanda (2.0) del *snap*: **\$ sudo snap install edgexfoundry --channel=2.0**.

Con el propósito de entender cómo funciona EdgeX se muestra en el anexo 1 un breve resumen de los servicios que se inician de forma automática e inmediata tras la instalación, y en el anexo 2 los que de manera predeterminada están deshabilitados.

3.2.2 Ejecución de Edgex Foundry

EdgeX proporciona una serie de comandos y utilidades que permite verificar el estado y funcionamiento de los servicios que ofrece la plataforma, una vez que estos servicios han sido desplegados e iniciados.

Algunos de los comandos disponibles, que se pueden ejecutar localmente, se muestran en [76]. A través de ellos se pueden ver todos los dispositivos creados, las variables y los datos recibidos. Para este propósito, se puede utilizar la consola o, alternativamente, el programa *Postman*. Las URL se pueden ingresar en el navegador. Para una llamada externa, el *localhost* debe reemplazarse con la dirección IP de la puerta de enlace EdgeX.

En la figura 3.4 mediante el terminal de Ubuntu se observan los servicios habilitados y deshabilitados, asimismo los activos e inactivos, todos ellos de manera predeterminada; luego en dependencia de lo que se desee realizar, se puede cambiar. También se pueden ver estos servicios a través de un navegador como se muestra en la figura 3.5.

```
tele1@tele1-IPMSB-VH1-HDMI-ODM:~$ snap services
Service                               Startup   Current   Notes
edgexfoundry.app-service-configurable disabled inactive -
edgexfoundry.consul                   enabled  active    -
edgexfoundry.core-command              enabled  active    -
edgexfoundry.core-data                 enabled  active    -
edgexfoundry.core-metadata             enabled  active    -
edgexfoundry.device-virtual            enabled  active    -
edgexfoundry.kong-daemon               enabled  active    -
edgexfoundry.kuiper                    disabled inactive  -
edgexfoundry.postgres                  enabled  active    -
edgexfoundry.redis                     enabled  active    -
edgexfoundry.security-bootstrap-redis  enabled  inactive  -
edgexfoundry.security-proxy-setup      enabled  inactive  -
edgexfoundry.security-secrets-setup    enabled  inactive  -
edgexfoundry.security-secretstore-setup enabled  inactive  -
edgexfoundry.support-notifications     disabled inactive  -
edgexfoundry.support-scheduler         disabled inactive  -
edgexfoundry.sys-mgmt-agent            disabled inactive  -
edgexfoundry.vault                     enabled  active    -
```

Figura 3.4 Lista de servicios mediante el terminal de Ubuntu.

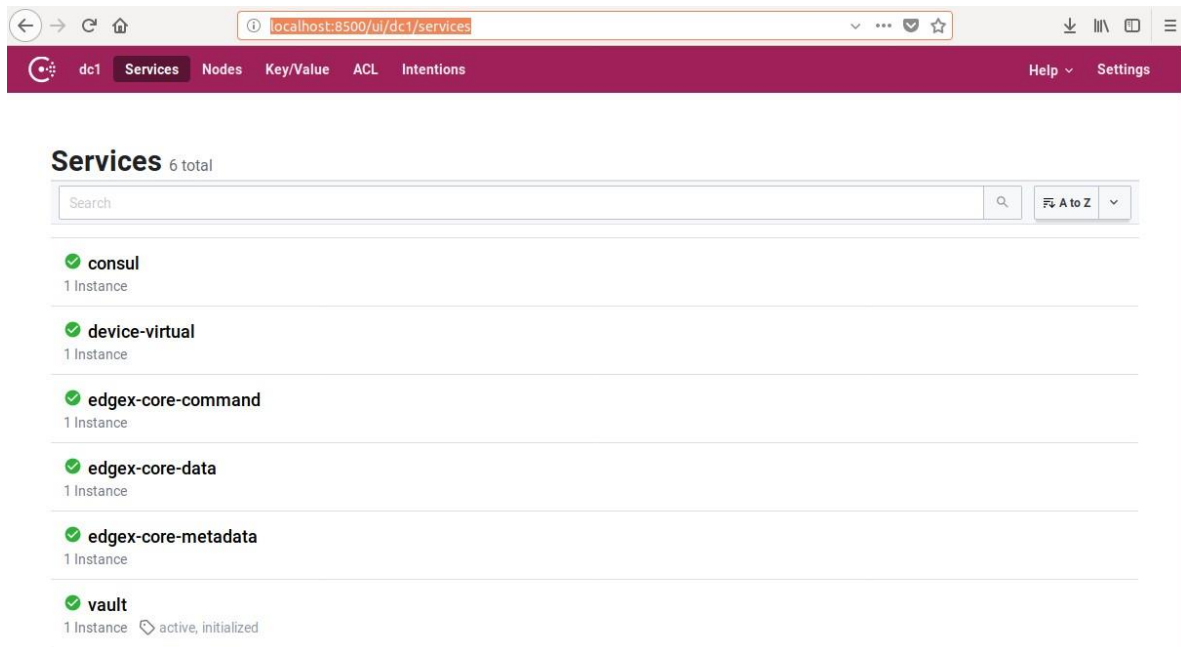


Figura 3.5 Lista de servicios mediante el navegador.

Como se observa en la Figura 3.6, la herramienta muestra el estado de todos los servicios desplegados, seleccionando cualquier servicio, en este caso *device-virtual*, se observa con más detalles su estado de salud. Así, con esta utilidad se ha podido verificar que todos los servicios se han registrado correctamente y se encuentran en ejecución, lo que constituye una forma rápida de comprobar el estado del entorno EdgeX.

The screenshot displays the EdgeX Foundry interface for a service named 'device-virtual'. The top navigation bar includes 'dc1', 'Services', 'Nodes', 'Key/Value', 'ACL', 'Intentions', 'Help', and 'Settings'. Below the navigation, the breadcrumb path is '< All Services / Service (device-virtual)'. The main heading is 'device-virtual' with a 'localhost' icon. A table shows the service name 'device-virtual' and node name 'tele1-IPMSB-VH1-HDMI-ODM'. Two tabs are visible: 'Health Checks' (active) and 'Tags & Meta'. The 'Health Checks' section contains two entries:

ServiceName	CheckID	Type	Notes
device-virtual	device-virtual	http	Check the health of the API

The output for this check is: HTTP GET http://localhost:49990/api/v1/ping: 200 OK Output: pong

NodeName	CheckID	Type	Notes
tele1-IPMSB-VH1-HDMI-ODM	serfHealth		-

The output for this check is: Agent alive and reachable

Figura 3.6 Disponibilidad de los servicios.

Cada uno de los microservicios de EdgeX ha sido diseñado para responder a una solicitud HTTP de *ping*, lo que permite chequear la disponibilidad de dichos microservicios a través de peticiones HTTP al recurso *ping* de la API. Esta funcionalidad se puede probar a través de cualquier cliente que permita hacer este tipo de peticiones, para este caso de estudio se ha usado el cliente *Postman*, verificando que los microservicios responden correctamente a peticiones *ping* (figura 3.7).

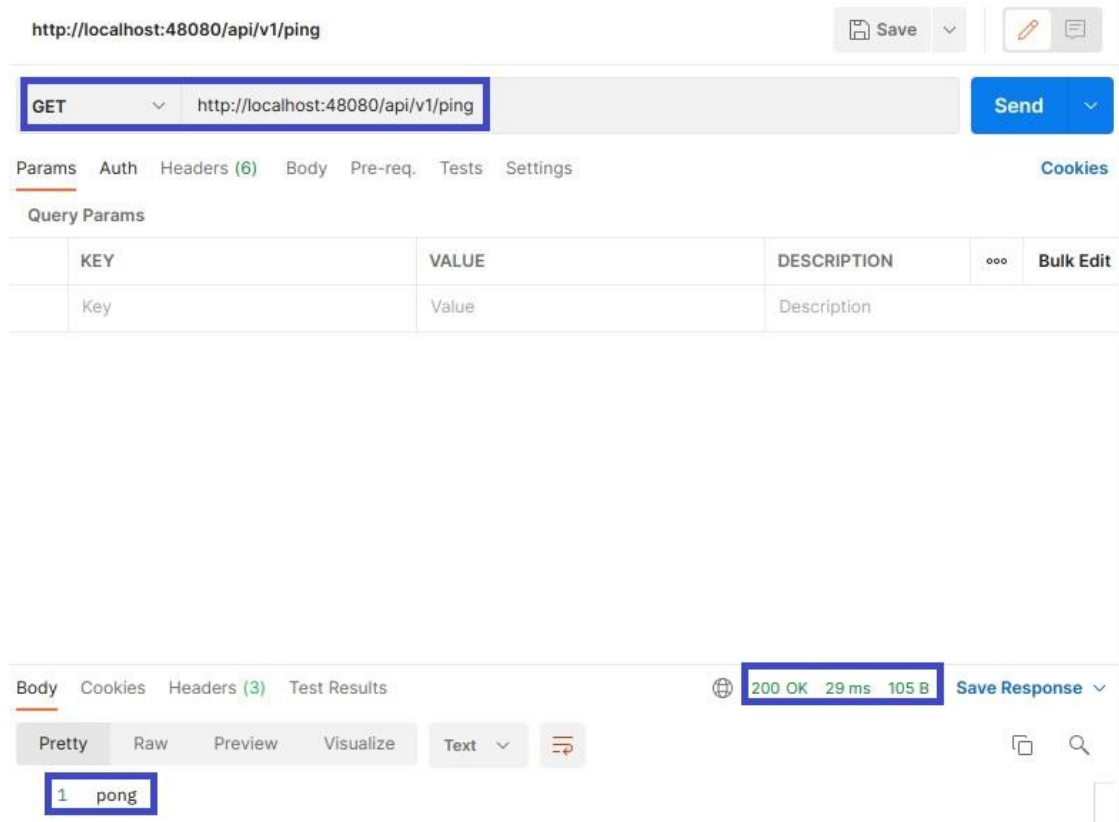


Figura 3.7 Chequeo de la disponibilidad de servicios a través de *Postman*.

3.3 Control de la funcionalidad del proyecto EdgeX Foundry

El proyecto EdgeX Foundry proporciona un servicio de dispositivo virtual (figura 3.8). Este simula diferentes tipos de dispositivos para generar eventos y lecturas en los microservicios de datos centrales, y los usuarios envían comandos y obtienen respuestas a través del microservicio de comando y control. Estas características de los servicios de dispositivos virtuales son útiles cuando se ejecutan pruebas funcionales o de rendimiento sin tener ningún dispositivo real [77].

Virtual Device

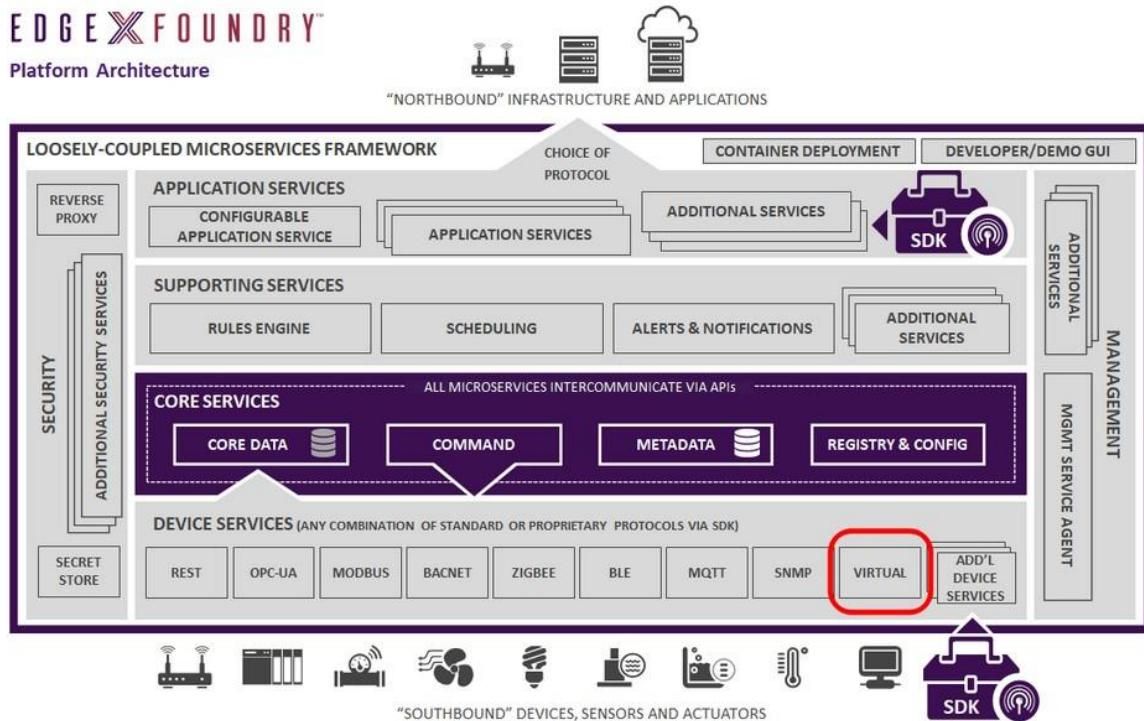
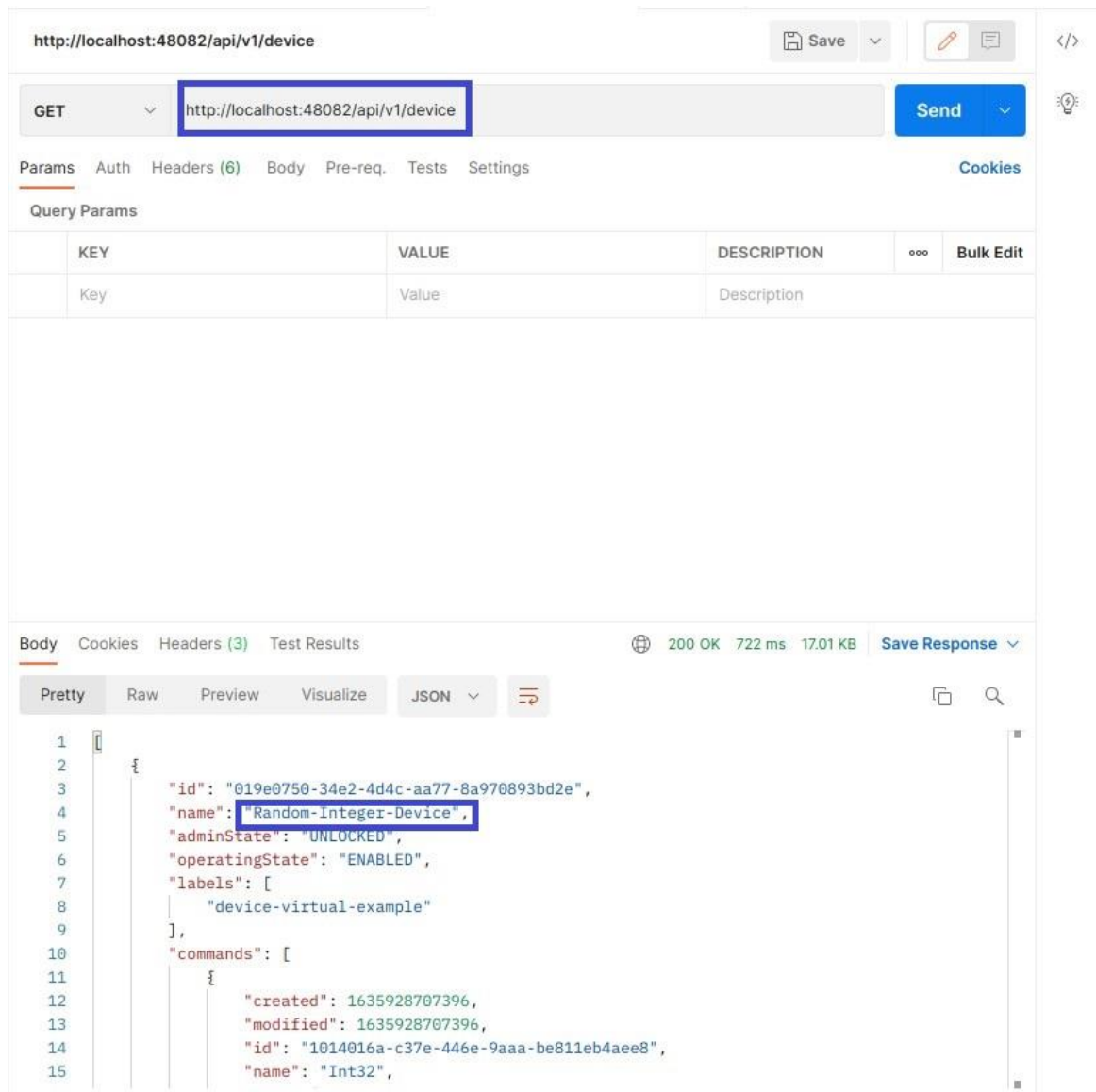


Figura 3.8 Servicio de dispositivo virtual [77].

Según [77], el servicio de dispositivo virtual contiene 5 dispositivos predefinidos como generadores de valores aleatorios:

- Dispositivo booleano aleatorio (Random-Boolean-Device)
- Dispositivo de número entero aleatorio (Random-Integer-Device)
- Dispositivo entero sin firmar aleatorio (Random-UnsignedInteger-Device)
- Dispositivo flotante aleatorio (Random-Float-Device)
- Dispositivo binario aleatorio (Random-Binary-Device)

El servicio *core-metadata* conoce los sensores y dispositivos conectados al sistema, así que una llamada conectada a la API correspondiente indica que el dispositivo, en este caso Random-Integer-Device (figura 3.9), está conectado al entorno EdgeX.



The screenshot shows a Postman interface for a REST client. The URL bar contains `http://localhost:48082/api/v1/device`. The method is set to `GET`. The response status is `200 OK` with a response time of `722 ms` and a size of `17.01 KB`. The response body is displayed in JSON format, showing a list of virtual devices. The first device is highlighted with a blue box:

```
1 [
2   {
3     "id": "019e0750-34e2-4d4c-aa77-8a970893bd2e",
4     "name": "Random-Integer-Device",
5     "adminState": "UNLOCKED",
6     "operatingState": "ENABLED",
7     "labels": [
8       "device-virtual-example"
9     ],
10    "commands": [
11      {
12        "created": 1635928707396,
13        "modified": 1635928707396,
14        "id": "1014016a-c37e-446e-9aaa-be811eb4aee8",
15        "name": "Int32",
```

Figura 3.9 Tipo de dispositivo virtual.

Mientras el dispositivo `Random-Integer-Device` esté operativo, todos los eventos y lecturas que genera se envían al `core-data` para su almacenamiento temporal. Con el cliente Postman se puede hacer una llamada a la API correspondiente del servicio `core-data` y solicitarle una de las lecturas del dispositivo (la figura 3.10 muestra los paquetes de datos a través del navegador y en la figura 3.11, utilizando Postman, se ve más claramente que se están recibiendo datos), de esta forma se puede concluir que las lecturas generadas por el dispositivo virtual se están registrando correctamente en la plataforma EdgeX Foundry desplegada; se observa también cómo estas lecturas se encapsulan/transforman en una estructura de datos común de EdgeX.

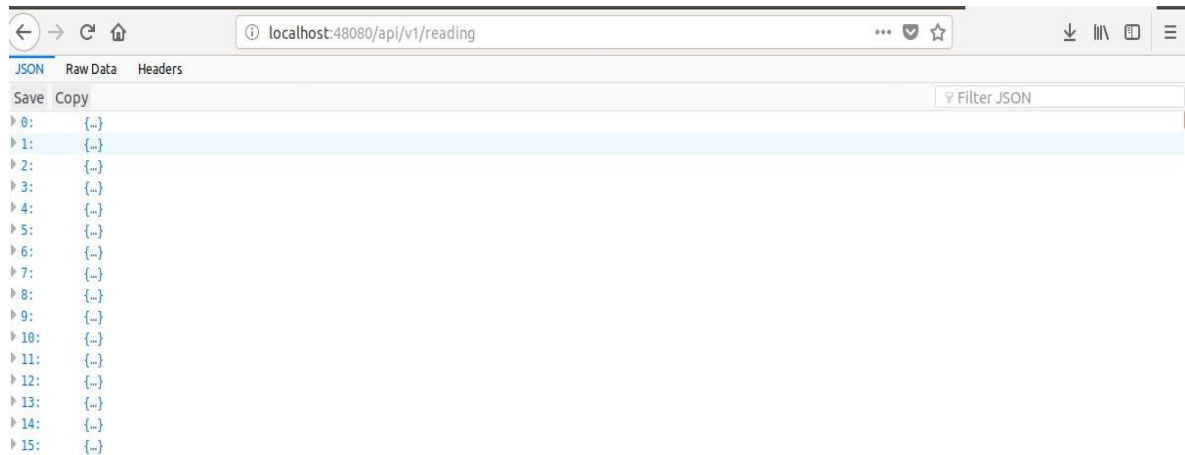


Figura 3.10 Lectura de los datos.

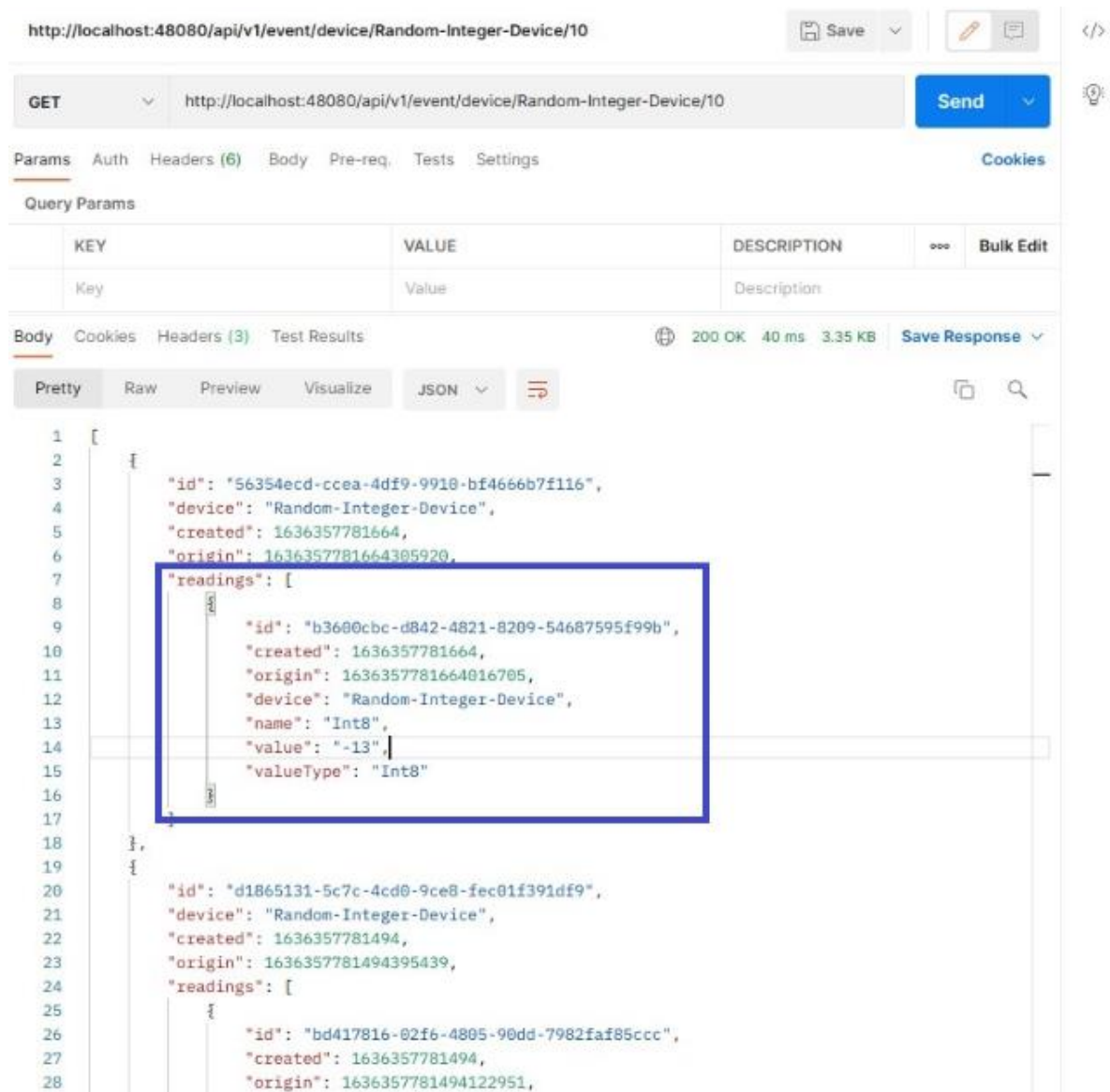
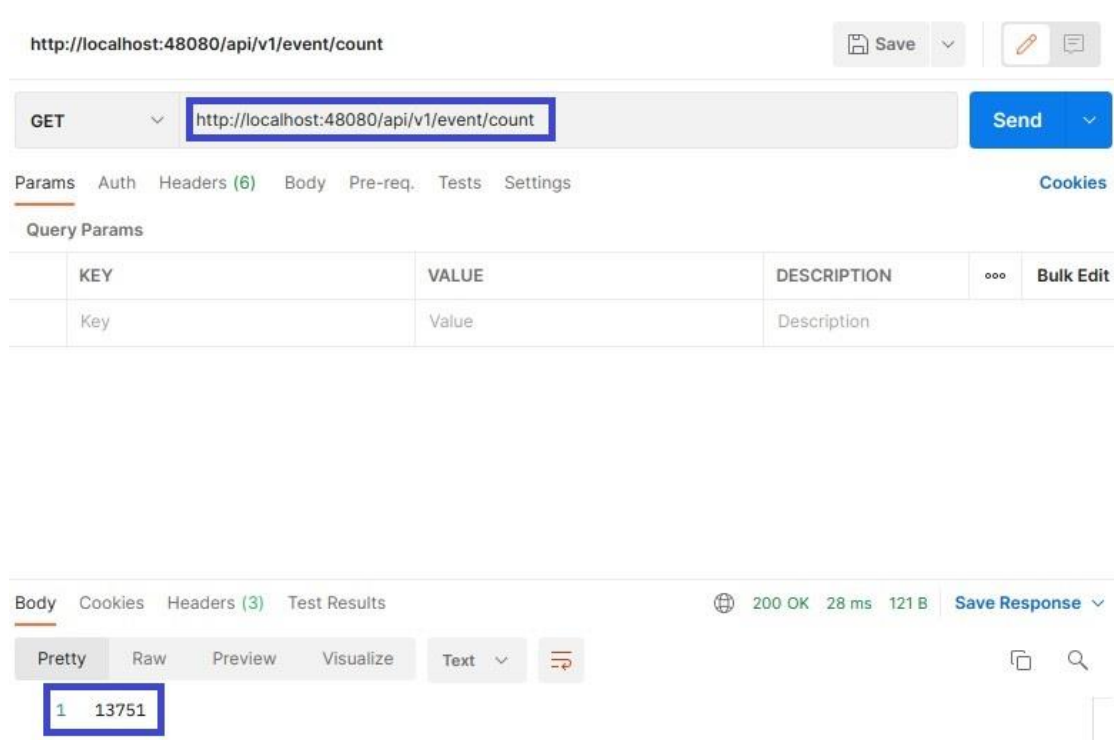


Figura 3.11 Lectura de datos a través del Postman.

Finalmente, una vez que se ejecute el comando asociado a la cantidad de datos generados, se verifica que se están recibiendo datos a través de la figura 3.12.



The screenshot shows a REST client interface with the following details:

- URL: `http://localhost:48080/api/v1/event/count`
- Method: `GET`
- Status: `200 OK`, `28 ms`, `121 B`
- Response Body (Pretty): `1 13751`

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Figura 3.12 Cantidad de datos generados.

3.4 Conclusiones del capítulo

Utilizando un sensor virtual, que generaba lecturas de forma aleatoria, se ha podido verificar, a través de llamadas a las API de los microservicios que forman la plataforma, cómo el sensor se registraba correctamente en nuestra plataforma, cómo las lecturas recolectadas por el sensor eran enviadas al servicio correspondiente para su almacenamiento temporal y la cantidad de datos que se reciben. Con este caso de estudio se ha podido verificar la funcionalidad del conjunto de microservicios que proporciona EdgeX y que ofrece la capacidad mínima de una plataforma *edge*.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Se considera que los objetivos planteados inicialmente en este trabajo se han logrado ya que:

- 1 El extenso estudio realizado sobre los conceptos claves tratados en este trabajo permite conocer y entender elementos importantes como IoT y computación de borde, asimismo se logra ver los problemas actuales a los que se enfrentan los sistemas de IoT y cómo tecnologías emergentes como EC intentan dar solución a dichos problemas.
- 2 Se ha podido comprobar que con los avances en Internet de las cosas (IoT), inteligencia artificial y tecnologías de la comunicación, la computación de borde ofrece una nueva solución al problema de procesar todos o parte de los datos localmente en el borde de un sistema.
- 3 Se ha demostrado que la Inteligencia Artificial (IA) se está convirtiendo en una plataforma para el cálculo y la interacción entre máquinas y objetos inteligentes en aplicaciones en tiempo real y muchos aspectos críticos de la vida diaria, por lo que el estudio de las plataformas vinculadas a IoT y EC se hace necesario para el desarrollo de estas tecnologías.
- 4 Específicamente con EdgeX Foundry se realiza un caso de estudio que permite verificar que dicha plataforma contribuye a la estandarización de IoT, facilita la adopción de estos sistemas y cumple con la capacidad mínima de una plataforma *edge*.

Recomendaciones

- 1 Continuar indagando en la búsqueda de nuevas plataformas vinculadas a la informática de borde y a IoT, y verificar su funcionamiento.
- 2 Profundizar más en el proyecto EdgeX Foudry, por ejemplo, conectar dispositivos reales a la plataforma para validar su comportamiento y ver cómo los diferentes microservicios interactúan entre ellos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. M. Merchan Miller, "Análisis de seguridad en plataformas de computación distribuida con arquitectura de edge computing," Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ..., 2020.
- [2] H. Cai, B. Xu, L. Jiang, and A. V. J. I. I. o. T. J. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," vol. 4, no. 1, pp. 75-87, 2016.
- [3] M. S. Sánchez, "Estudio de una Plataforma para Edge Computing e IOT," 2020.
- [4] J. L. Mela, G. D. Cedeño, and E. C. J. V. A. Herrera, "Edge Computing: Aplicaciones y desafíos actuales," vol. 5, no. 1, pp. 75-91, 2021.
- [5] J. Galiana, "Conocimiento y Tecnología clave en Internet de las Cosas (IoT)," ed, 2016.
- [6] I. X. Sitton Candanedo, "Resumen de tesis. Arquitectura inteligente Edge Computing para entornos IoT," 2020.
- [7] B. A. Hernández and D. P. Ortiz Galeano, "Análisis general del enfoque IOT en redes," Editorial Universitaria San Mateo, 2019.
- [8] A. J. González García, "IoT: Dispositivos, tecnologías de transporte y aplicaciones," 2017.
- [9] O. Vermesan *et al.*, "Internet of things strategic research roadmap," vol. 1, no. 2011, pp. 9-52, 2011.
- [10] I. Peña-López, "ITU Internet report 2005: the internet of things," 2005.
- [11] A. Noronha, R. Moriarty, K. O'Connell, and N. Villa, "El valor de IoT: cómo pasar de conectar cosas a obtener información," ed: Cisco, 2015.
- [12] K. Wang, H. Yin, W. Quan, and G. J. I. N. Min, "Enabling collaborative edge computing for software defined vehicular networks," vol. 32, no. 5, pp. 112-117, 2018.
- [13] F. J. Merino Ricaurte and D. I. Valencia Leal, "Análisis y diseño de un sistema de seguridad distribuido para la plataforma de terapias médicas temonet fase II," Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ..., 2020.

-
- [14] Y. Zhao, W. Wang, Y. Li, C. C. Meixner, M. Tornatore, and J. J. I. A. Zhang, "Edge computing and networking: A survey on infrastructures and applications," vol. 7, pp. 101213-101230, 2019.
- [15] S. J. A. i. N. Wang, "Edge Computing: Applications, State-of-the-Art and Challenges," vol. 7, no. 1, pp. 8-15, 2019.
- [16] G. Ananthanarayanan *et al.*, "Real-time video analytics: The killer app for edge computing," vol. 50, no. 10, pp. 58-67, 2017.
- [17] Y. Wu, Y. Liu, S. H. Ahmed, J. Peng, and A. A. J. I. I. o. T. J. Abd El-Latif, "Dominant data set selection algorithms for electricity consumption time-series data analysis based on affine transformation," vol. 7, no. 5, pp. 4347-4360, 2019.
- [18] T. X. Tran and D. J. I. T. o. V. T. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," vol. 68, no. 1, pp. 856-868, 2018.
- [19] F. Wang, J. Xu, X. Wang, and S. J. I. T. o. W. C. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," vol. 17, no. 3, pp. 1784-1797, 2017.
- [20] Y. Mao, J. Zhang, S. Song, and K. B. J. I. T. o. W. C. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," vol. 16, no. 9, pp. 5994-6009, 2017.
- [21] E. Kristiani, C.-T. Yang, Y. T. Wang, and C.-Y. Huang, "Implementation of an edge computing architecture using openstack and kubernetes," in *International Conference on Information Science and Applications*, 2018, pp. 675-685: Springer.
- [22] R.-H. Hsu, J. Lee, T. Q. Quek, and J.-C. J. I. N. Chen, "Reconfigurable security: Edge-computing-based framework for IoT," vol. 32, no. 5, pp. 92-99, 2018.
- [23] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*, 2015, pp. 73-78: IEEE.
- [24] G. I. J. G. S. Klas, "Fog computing and mobile edge cloud gain momentum open fog consortium, etsi mec and cloudlets," vol. 1, no. 1, pp. 1-13, 2015.
- [25] S. Guerrero, J. Carrero, O. Andrés, and C. J. B. Gómez, Colombia: Universidad Industrial de Santander Bucaramanga, "Análisis del Paradigma Fog y Edge Computing."
- [26] L. Aljure, "Computación en el borde y en la niebla," *Revista SISTEMAS*, pp. 22-34, 09/16 2020.
- [27] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge computing perspectives: architectures, technologies, and open security issues," in *2019 IEEE International Conference on Edge Computing (EDGE)*, 2019, pp. 116-123: IEEE.
- [28] A.-G. Gheorghe, C.-C. Creana, C. Negru, F. Pop, and C. Dobre, "Decentralized storage system for edge computing," in *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*, 2019, pp. 41-49: IEEE.

- [29] K.-C. Chang, Y.-C. Lin, K.-C. J. T. F. o. S. Chu, Science, and Technology, "Application of edge computing technology in the security industry," vol. 1, no. 10, 2019.
- [30] F. J. R. U. I. Rozo-García, "Revisión de las tecnologías presentes en la industria 4.0," vol. 19, no. 2, pp. 177-192, 2020.
- [31] L. J. M. A. E. Rouhiainen, "Inteligencia artificial," 2018.
- [32] G. Nguyen *et al.*, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey," vol. 52, no. 1, pp. 77-124, 2019.
- [33] A. J. S. Strong, "Applications of artificial intelligence & associated technologies," vol. 5, no. 6, 2016.
- [34] K. Ashton, "That 'internet of things' thing, in the real world things matter more than ideas. RFID J.(2009)," ed, 2013.
- [35] A. M. Alberti *et al.*, "Platforms for smart environments and future Internet design: A survey," vol. 7, pp. 165748-165778, 2019.
- [36] R. Ke, Y. Zhuang, Z. Pu, and Y. J. I. T. o. I. T. S. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," 2020.
- [37] X. Wang *et al.*, "Convergence of edge computing and deep learning: A comprehensive survey," vol. 22, no. 2, pp. 869-904, 2020.
- [38] J. Qian, S. P. Gochhayat, and L. K. Hansen, "Distributed active learning strategies on edge computing," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2019, pp. 221-226: IEEE.
- [39] F. Al-Turjman, *Artificial intelligence in IoT*. Springer, 2019.
- [40] T. Mucha and T. Seppala, "Artificial Intelligence Platforms—A New Research Agenda for Digital Platform Economy," 2020.
- [41] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. J. A. C. S. Milojicic, "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture," vol. 50, no. 6, pp. 1-37, 2017.
- [42] B. Vogel, Y. Dong, B. Emruli, P. Davidsson, and R. J. F. I. Spalazzese, "What Is an Open IoT Platform? Insights from a Systematic Mapping Study," vol. 12, no. 4, p. 73, 2020.
- [43] A. Bröring *et al.*, "Enabling IoT ecosystems through platform interoperability," vol. 34, no. 1, pp. 54-61, 2017.
- [44] J. Mineraud, O. Mazhelis, X. Su, and S. J. C. C. Tarkoma, "A gap analysis of Internet-of-Things platforms," vol. 89, pp. 5-16, 2016.
- [45] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A survey of commercial frameworks for the internet of things," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015, pp. 1-8: IEEE.

- [46] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. J. I. n. Khoukhi, "IoT technologies for smart cities," vol. 7, no. 1, pp. 1-13, 2017.
- [47] S. Kuila, N. Dhanda, S. Joardar, and S. Neogy, "Analytical survey on standards of Internet of Things framework and platforms," in *Emerging Technologies in Data Mining and Information Security*: Springer, 2019, pp. 33-44.
- [48] B. Weinberg, "The internet of things and open source," in *Interoperability and open-source solutions for the internet of things*: Springer, 2015, pp. 1-5.
- [49] X. Liu and P. S. Nielsen, "Air quality monitoring system and benchmarking," in *International Conference on Big Data Analytics and Knowledge Discovery*, 2017, pp. 459-470: Springer.
- [50] C. Jinbo, Z. Yu, and A. J. W. P. C. Lam, "Research on monitoring platform of agricultural product circulation efficiency supported by cloud computing," vol. 102, no. 4, pp. 3573-3587, 2018.
- [51] J. H. Jeon, K.-H. Kim, and J.-H. Kim, "Block chain based data security enhanced IoT server platform," in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 941-944: IEEE.
- [52] D.-H. Park, H.-C. Bang, C. S. Pyo, and S.-J. Kang, "Semantic open IoT service platform technology," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 85-88: IEEE.
- [53] S. Kianoush, M. Raja, S. Savazzi, and S. J. I. I. o. T. J. Sigg, "A cloud-IoT platform for passive radio sensing: Challenges and application case studies," vol. 5, no. 5, pp. 3624-3636, 2018.
- [54] I. Andreev, "Advanced open IoT platform for prevention and early detection of forest fires," in *World Conference on Information Systems and Technologies*, 2018, pp. 319-329: Springer.
- [55] F. Ramparany, F. G. Marquez, J. Soriano, and T. Elsaleh, "Handling smart environment devices, data and services at the semantic level with the FI-WARE core platform," in *2014 IEEE International Conference on Big Data (Big Data)*, 2014, pp. 14-20: IEEE.
- [56] P. Fernández *et al.*, "SmartPort: A platform for sensor data monitoring in a seaport based on FIWARE," vol. 16, no. 3, p. 417, 2016.
- [57] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [58] J. An *et al.*, "Toward global IoT-enabled smart cities interworking using adaptive semantic adapter," vol. 6, no. 3, pp. 5753-5765, 2019.
- [59] A. Alonso *et al.*, "An identity framework for providing access to FIWARE OAuth 2.0-based services according to the eIDAS European regulation," vol. 7, pp. 88435-88449, 2019.

- [60] A. Medvedev, A. Zaslavsky, S. Khoruzhnikov, and V. Grudin, "Reporting road problems in smart cities using OpenIoT framework," in *Interoperability and open-source solutions for the internet of things*: Springer, 2015, pp. 169-182.
- [61] J. Soldatos *et al.*, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and open-source solutions for the internet of things*: Springer, 2015, pp. 13-25.
- [62] *NetApp y Lenovo aceleran la inteligencia artificial en el borde | Noticias y Actualidad | Almacenamiento IT.* Available: <https://almacenamientoit.ituser.es/noticias-y-actualidad/2021/04/netapp-y-lenovo-aceleran-la-inteligencia-artificial-en-el-borde>
- [63] J. Peregrina. *Plataformas de IA para el borde de red industrial - Permitiendo la automatización industrial impulsada por IA.* Available: <https://www.lanner-america.com/es/ultimas-noticias/plataformas-de-ia-para-el-borde-de-red-industrial-habilitando-la-automatizacion-industrial-impulsada-por-ia/>
- [64] T. L. Foundation. *Welcome.* Available: <https://www.edgexfoundry.org>
- [65] A. Das, S. Patterson, and M. Wittie, "Edgebench: Benchmarking edge computing platforms," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 175-180: IEEE.
- [66] *Red Hat impulsa la siguiente fase del edge computing con la nueva versión de la plataforma empresarial de Linux líder a nivel mundial - DCD.* Available: <https://www.datacenterdynamics.com/es/noticias/red-hat-impulsa-la-siguiente-fase-del-edge-computing-con-la-nueva-versi%C3%B3n-de-la-plataforma-empresarial-de-linux-1%C3%ADder-a-nivel-mundial/>
- [67] T. Kondo, K. Isawaki, and K. Maeda, "Development and evaluation of the MEC platform supporting the edge instance mobility," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, vol. 2, pp. 193-198: IEEE.
- [68] *Contributor Suggestions (how can you be a part of EdgeX Foundry?) - EdgeX Wiki - EdgeX Confluence.* Available: <https://wiki.edgexfoundry.org/pages/viewpage.action?pageId=4063305>
- [69] *EdgeX Foundry Project Wiki - EdgeX Wiki - EdgeX Confluence.* Available: <https://wiki.edgexfoundry.org/>
- [70] J. J. M. a. d. o. t. n. a. t. Fowler, "Marthin; Lewis, "Microservices: a definition of this new architectural term,"" 2014.
- [71] *EdgeX Foundry Project.* Available: <https://github.com/edgexfoundry>
- [72] *Docker Documentation | Docker Documentation.* Available: <https://docs.docker.com/>
- [73] *Introduction - EdgeX Foundry Documentation.* Available: <https://docs.edgexfoundry.org/2.0/>
- [74] *Get started with EdgeX as snaps.* Available: <https://ubuntu.com/tutorials/get-started-with-edgex-as-snaps>

-
- [75] *Installing snapd / Snapcraft documentation.* Available:
<https://snapcraft.io/docs/installing-snapd>
- [76] D. C. Baun, "Edge-Computing Framework (EdgeX)," (in en), p. 48.
- [77] *Virtual Device - EdgeX Foundry Documentation.* Available:
<https://docs.edgexfoundry.org/1.2/microservices/device/virtual/Ch-VirtualDevice/#core-command-and-the-virtual-device-service>

ANEXOS

Anexo I Servicios de EdgeX Foundry iniciados de manera automática tras la instalación [74].

SERVICIO	DESCRIPCIÓN
Cónsul	proporciona servicios de configuración y registro a EdgeX
Kong	proporciona control de acceso para un sistema EdgeX
Postgres	proporciona soporte de configuración para Kong
Redis	proporciona servicios de bus de mensajes y bases de datos a EdgeX
Datos básicos (EdgeX)	gestiona la persistencia a corto plazo de las lecturas del dispositivo / sensor
Comando básico (EdgeX)	proporciona servicios de comando a clientes REST externos
Metadatos básicos (EdgeX)	gestiona los metadatos del servicio del dispositivo
Seguridad - * - configuración	trabajos únicos utilizados para arrancar EdgeX

Anexo II Servicios deshabilitados de forma predeterminada tras la instalación de EdgeX Foundry [74].

SERVICIO	DESCRIPCIÓN
Servicio de aplicaciones configurable	proporciona filtrado de datos para eKuiper
Dispositivo virtual	un servicio de dispositivo virtual, útil para conocer EdgeX
eKuiper	un motor de reglas basado en SQL utilizado para análisis locales
Notificaciones de soporte (EdgeX)	proporciona un servicio de notificación genérico
Programador de soporte (EdgeX)	proporcionar un servicio de programación genérico
Agente de administración del sistema (EdgeX)	proporciona servicios básicos de gestión del sistema