Universidad de Oviedo

Programa de Doctorado en Energía y control de procesos

INVESTIGACIÓN EN LA APLICACIÓN DE TECNOLOGÍAS IIOT PARA LA SUPERVISIÓN DE PROCESOS INDUSTRIALES Y DE GESTIÓN ENERGÉTICA

Pedro de Arquer Fernández

Gijón, Noviembre 2024

Universidad de Oviedo

Dpto. de Ingeniería Eléctrica, Electrónica, de Comunicaciones y de Sistemas

Programa de Doctorado en Energía y control de procesos

INVESTIGACIÓN EN LA APLICACIÓN DE TECNOLOGÍAS IIOT PARA LA SUPERVISIÓN DE PROCESOS INDUSTRIALES Y DE GESTIÓN ENERGÉTICA

Tesis presentada para la obtención del título de Doctor en Energía y Control de Procesos por la Universidad de Oviedo con Mención Industrial

Pedro de Arquer Fernández

Directores:

Pablo Arboleya Arboleya

Juan Luis Carús Candás

Gijón, Noviembre 2024

Universidad de Oviedo

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Investigación en la aplicación de tecnologías IIoT para la supervisión de procesos industriales y de gestión energética	Inglés: Research on the Applicability of IIoT Technologies for Industrial Processes Monitoring and Energy Management

2.- Autor	
Nombre: Pedro de Arquer Fernández	
Programa de Doctorado: Energía y Control de Procesos	
Órgano responsable: Centro Internacional de Postgrado	

RESUMEN en español (máximo 4000 caracteres)

En la actualidad, la energía eléctrica es un recurso estratégico fundamental de cualquier tejido industrial. Dentro de los sistemas de gestión energética, las plantas de generación renovable representan una parte crítica que debe afrontar cada vez más desafíos y adaptarse a nuevas exigencias y condiciones de mercado. En este contexto de constante cambio y creciente demanda, la automatización y el control eficientes se vuelven cruciales para garantizar la competitividad y sostenibilidad de estas instalaciones.

En este sentido, los SCADA juegan un papel esencial, sirviendo como herramienta central para la supervisión y gestión de los procesos industriales. No obstante, estos sistemas han mostrado limitaciones al ser entornos excesivamente estáticos y anquilosados frente a la rápida evolución tecnológica, lo que impide una asimilación óptima de los avances y limita los beneficios de la innovación.

Dentro del marco de la cuarta revolución industrial, las tecnologías IIoT son un elemento decisivo para desarrollar SCADA más flexibles, capaces de incorporar las nuevas mejoras de manera más eficiente. Aunque las tecnologías IIoT han alcanzado la madurez suficiente para su implementación en industria, sigue presente el reto de construir una infraestructura que las aproveche con la versatilidad, robustez y fiabilidad necesarias para un entorno productivo.

Para afrontar este desafío, se propone la creación de una arquitectura de monitorización y control basada en IIoT que contemple las necesidades de la industria, analizando paso a paso su diseño e implementación para lograr resultados más generales que puedan servir de referencia a nuevas soluciones.

Para evaluar esta propuesta, se seleccionan casos de uso variados en los que se implementa la arquitectura planteada. Las aplicaciones elegidas responden a necesidades reales de la industria, identificadas a través de la colaboración con la empresa TSK, que participa en el desarrollo e implantación de las soluciones. Gracias a ellas, es posible analizar tanto las capacidades como los límites de la arquitectura y detectar posibles puntos críticos.

RESUMEN en Inglés

Nowadays, electrical energy is a strategic resource in any industrial setting. Within energy management systems, renewable energy plants have become a key component, facing an increasing number of challenges and the need to adapt to evolving market demands and conditions. In this environment of constant change and rising demand, efficient automation and control are essential to ensure the competitiveness and long-term sustainability of these facilities.

Therefore, SCADA systems play a crucial role, acting as a central tool for the monitoring and management of industrial processes. However, these systems have limitations, as they tend to be overly rigid and outdated in comparison with rapid technological advancements, disrupting the optimal adoption of innovations and restricting their potential benefits.

In the context of the Fourth Industrial Revolution, IIoT technologies are a decisive factor in developing more flexible SCADA systems capable of a seamless integration with new systems. IIoT technologies have already reached a level of maturity suitable for industrial implementation, but there are still pending challenges. The most fundamental issue is the construction of an infrastructure that can leverage these technologies with the required versatility, robustness, and reliability for productive environments.

To address this challenge, the development of an IIoT-based monitoring and control architecture is proposed. This architecture considers the specific needs of the industry, with its design and implementation carefully analysed to produce more generalised outcomes that can serve as a reference for future solutions.

For the assessment of this proposal, the proposed architecture has been implemented in multiple use cases. The chosen applications address real industry needs, identified through collaboration with the company TSK, which is actively involved in the development and deployment of the solutions. These cases allow for a thorough analysis of the capabilities and limitations of the proposal, as well as the identification of potential critical issues.

SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN _____

Agradecimientos

Esta tesis es fruto de un trabajo y aprendizaje que abarca muchos años. A lo largo de este tiempo son muchas las personas que han pasado por mi vida y contribuido directa o indirectamente al enseñarme, ayudarme o inspirarme. No cabe duda de que debo un agradecimiento a cada una de ellas. Sin embargo, es justo decir que este proyecto sería impensable si no fuera por algunas personas a las que quiero dedicar unas palabras especiales.

En primer lugar, a Juan Luis Carús, compañero y mentor, cuya dedicación incansable, consejos y entrega a esta tesis han sido imprescindibles para su desarrollo y del que tanto he aprendido en el ámbito profesional y personal a través de su sencillez, cercanía y compromiso.

A Pablo Arboleya, cuya guía a través del mundo académico y optimismo me han ayudado y motivado en los momentos de mayor incertidumbre. Su conocimiento y experiencia han sido claves para allanar el camino de esta tesis y realizar una investigación de calidad.

A mi esposa Ana, por su amor inquebrantable, por su aliento, por su paciencia, por sufrir la tesis a mi lado, por creer en este proyecto más que yo mismo, por enseñarme a vivir más intensamente cada momento, pero sobre todo porque ser su cómplice y compañero en la vida me hace feliz cada día.

A mis familiares, en especial a mis padres y hermanos, que siempre me han dado su apoyo incondicional, por ser una fuente de inspiración y disfrute por igual. A mis amigos, en particular a Alejandro e incluso Eloy, por estar ahí en momentos clave con palabras de ánimo, optimismo e incluso nuevas ideas. También a aquellas personas que han recorrido este mismo camino a la vez que yo, cuya perseverancia ha servido de norte y esperanza.

Y, por último, a mis compañeros de trabajo, que han contribuido en algunos de los aspectos más técnicos de esta tesis con su tiempo, conocimiento y experiencia, haciendo gala de un gran compañerismo, esmero y dedicación.

A todos vosotros, mi más sincero agradecimiento.

Resumen

En la actualidad, la energía eléctrica es un recurso estratégico fundamental de cualquier tejido industrial. Dentro de los sistemas de gestión energética, las plantas de generación renovable representan una parte crítica que debe afrontar cada vez más desafíos y adaptarse a nuevas exigencias y condiciones de mercado. En este contexto de constante cambio y creciente demanda, la automatización y el control eficientes se vuelven cruciales para garantizar la competitividad y sostenibilidad de estas instalaciones.

En este sentido, los SCADA juegan un papel esencial, sirviendo como herramienta central para la supervisión y gestión de los procesos industriales. No obstante, estos sistemas han mostrado limitaciones al ser entornos excesivamente estáticos y anquilosados frente a la rápida evolución tecnológica, lo que impide una asimilación óptima de los avances y limita los beneficios de la innovación.

Dentro del marco de la cuarta revolución industrial, las tecnologías IIoT son un elemento decisivo para desarrollar SCADA más flexibles, capaces de incorporar las nuevas mejoras de manera más eficiente. Aunque las tecnologías IIoT han alcanzado la madurez suficiente para su implementación en industria, sigue presente el reto de construir una infraestructura que las aproveche con la versatilidad, robustez y fiabilidad necesarias para un entorno productivo.

Para afrontar este desafío, se propone la creación de una arquitectura de monitorización y control basada en IIoT que contemple las necesidades de la industria, analizando paso a paso su diseño e implementación para lograr resultados más generales que puedan servir de referencia a nuevas soluciones.

Para evaluar esta propuesta, se seleccionan casos de uso variados en los que se implementa la arquitectura planteada. Las aplicaciones elegidas responden a necesidades reales de la industria, identificadas a través de la colaboración con la empresa TSK, que participa en el desarrollo e implantación de las soluciones. Gracias a ellas, es posible analizar tanto las capacidades como los límites de la arquitectura y detectar posibles puntos críticos.

Abstract

Nowadays, electrical energy is a strategic resource in any industrial setting. Within energy management systems, renewable energy plants have become a key component, facing an increasing number of challenges and the need to adapt to evolving market demands and conditions. In this environment of constant change and rising demand, efficient automation and control are essential to ensure the competitiveness and long-term sustainability of these facilities.

Therefore, SCADA systems play a crucial role, acting as a central tool for the monitoring and management of industrial processes. However, these systems have limitations, as they tend to be overly rigid and outdated in comparison with rapid technological advancements, disrupting the optimal adoption of innovations and restricting their potential benefits.

In the context of the Fourth Industrial Revolution, IIoT technologies are a decisive factor in developing more flexible SCADA systems capable of a seamless integration with new systems. IIoT technologies have already reached a level of maturity suitable for industrial implementation, but there are still pending challenges. The most fundamental issue is the construction of an infrastructure that can leverage these technologies with the required versatility, robustness, and reliability for productive environments.

To address this challenge, the development of an IIoT-based monitoring and control architecture is proposed. This architecture considers the specific needs of the industry, with its design and implementation carefully analysed to produce more generalised outcomes that can serve as a reference for future solutions.

For the assessment of this proposal, the proposed architecture has been implemented in multiple use cases. The chosen applications address real industry needs, identified through collaboration with the company TSK, which is actively involved in the development and deployment of the solutions. These cases allow for a thorough analysis of the capabilities and limitations of the proposal, as well as the identification of potential critical issues.

Lista de publicaciones

1. de Arquer Fernández, P., Fernández Fernández, M. Á., Carús Candás, J. L. y Arboleya Arboleya, P. An IoT open source platform for photovoltaic plants supervision. *International Journal of Electrical Power and Energy Systems* **125,** 106540. ISSN: 01420615 (feb. de 2021).

2. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. Determining Operational Constraints for IoT-Based Advanced Metering Infrastructure. *2021 IEEE Madrid PowerTech,* 1-6 (jun. de 2021).

3. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. en *Encyclopedia of Electrical and Electronic Power Engineering* (Elsevier Ltd, 2022).

4. Fernández Villán, A., Fernández Fernández, M. Á., Carús Candás, J. L., de Arquer Fernández, P., Arias Linacero, N. y Usamentiaga Fernández, R. *Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography* en *2018 2nd International Research Conference on Sustainable Energy, Engineering, Materials and Environment (IRCSEEME2018)* (jul. de 2018), 206-207.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Históricamente, los sistemas de supervisión y adquisición de datos (SCADA por su siglas en inglés "Supervisory Control And Data Acquisition") han sido el núcleo de la automatización industrial. Los SCADA son responsables de la monitorización, el control y la gestión de la información y de coordinar los procedimientos de trabajo entre todos los dispositivos en la industria.

Como se puede ver en la Figura 1.1, la estructura general de un SCADA contiene múltiples sensores y actuadores, Unidades Terminales Remotas (Remote Terminal Unit, RTU) para manejar la comunicación y establecer procedimientos estáticos para los sensores y actuadores, e Interfaces Hombre-Máquina (Human-Machine Interface, HMI) para introducir las acciones humanas en el sistema de control. También integración con aplicaciones de mayor abstracción como planificadores de producción.

Con el paso de los años, esta estructura se ha ido haciendo más compleja. Por una lado la lógica de negocio requiere más control para mantenerse competitiva y por otro la sucesión de avances tecnológicos e informáticos ponen a prueba la capacidad de ampliación de los SCADA y los dispositivos de menor nivel. Con el tiempo, muchos SCADA han evolucionado a conectarse directamente con verticales, para facilitar su integración, pero dificultando cada vez más su

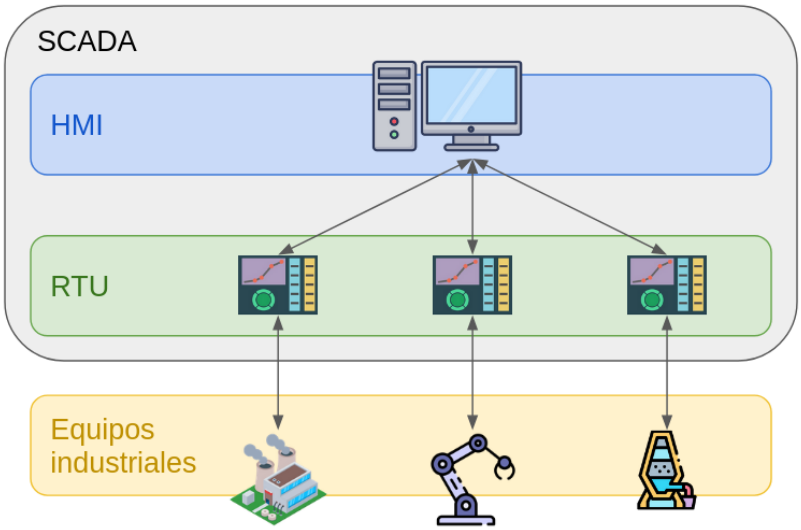


Figura 1.1. Estructura de un SCADA. El SCADA es base de cualquier industria automatizada moderna.

ampliación.

La Industria 4.0 es un paradigma emergente que se basa en tecnologías habilitadoras (Key Enabling Technologies, KET) como la robótica avanzada y drones, la impresión en 3D o el análisis de datos masivo (Big Data). Proporciona nuevas oportunidades para redefinir los SCADA y adaptarlos a un modelo industrial más moderno y sostenible en el futuro pues proporcionan características de alto rendimiento. Tales mejoras resultan críticas para mejorar los modelos SCADA tradicionales.

Dentro de estas tecnologías, una de las más prometedoras y ya lista para llevar a entornos productivos es el Internet de las cosas (Internet of Things, IoT). IoT es un patrón o estrategia de diseño de arquitectura de comunicaciones. Inicialmente motivado por las tecnologías de comunicación inalámbricas y de bajo consumo, ha evolucionado desde los entornos domésticos y educativos hasta la industria. Este tipo de arquitecturas están preparadas para ser más fáciles de mantener, mejorar la integración con otros servicios y utilizar canales de comunicación más

eficientes. Además se pueden implementar con un buen rendimiento económico gracias a su escalabilidad.

1.1. Motivación

Sin embargo, al igual que está pasando con otras KET de la Industria 4.0, la introducción de IoT en el proceso productivo supone un reto. Es necesario adaptarlo para ofrecer sus características al nivel de exigencia y competitividad que se espera en estos entornos. Específicamente, la adopción de IoT en industria ha resultado un problema tan específico que ha dado lugar al término propio de Industrial Internet of Things (IIoT).

Para el desarrollo de estas adaptaciones, se hace necesario llevar a casos de uso reales en industria el conjunto de tecnologías IIoT disponibles para contrastar su usabilidad y aportación al proceso productivo. Esto es necesario tanto para validar su posible uso como para encontrar los puntos de mejora que aún le pueden faltar para ser plenamente operativo, pues en origen IoT no se planteó para la industria, sino para entornos donde la falta de eficiencia o fiabilidad no son tan críticos. Es por esto que la separación entre IoT e IIoT es cada día mayor.

Por otro lado, la industria energética se enfrenta en la actualidad a retos que la están transformando por completo y que exceden las capacidades de sistemas tradicionales de monitorización y control. Algunos ejemplos de estos retos son el cambio de roles de generadores y consumidores, la evolución a energías renovables o la electrificación de servicios.

En este contexto, la empresa TSK Electrónica y Electricidad S.A. (en adelante solo TSK) es un claro ejemplo de estas necesidades en la industria energética. A pesar de su compromiso con la excelencia y la innovación la empresa llegó a encontrarse con la falta de respuesta del mercado para proveer aplicaciones tecnológicas de alto rendimiento en los proyectos que lideraban.

Como consecuencia de esto, se adoptó la decisión estratégica de generar conocimiento interno a través de la investigación en las tecnologías de Industria 4.0 que le permitiera incorporar sus principales fortalezas en las soluciones ofertadas y ejecutadas por la empresa.

Esta tesis surge como consecuencia del marco estratégico presentado para dar respuesta a esta necesidad, buscando mejorar las soluciones presentes y preparar a la empresa y la industria en general para los retos futuros con la aplicación de las tecnologías IIoT. Al mismo tiempo, al estar sometida a entornos productivos reales, permitirá hacer una validación más allá de lo académico de las propuestas, dando lugar a soluciones competitivas en el mercado, duraderas en el tiempo y robustecidas por la exposición a incidencias reales.

1.2. Objetivos y metodología

Para poder abordar estas necesidades, se propone un trabajo en cuatro partes bien diferenciadas: análisis de la industria, estado del arte de las tecnologías IIoT, definición de una solución y validación de la solución en entornos productivos.

Las dos primeras partes están destinadas a comprender en profundidad el estado actual tanto de la industria como de las tecnologías IIoT. En el primer caso se busca identificar las deficiencias de los entornos industriales productivos y esbozar los retos principales. También se realizará una valoración inicial del impacto que las tecnologías de la Industria 4.0 están proporcionando a la industria energética en la actualidad. En el segundo caso, el objetivo es detectar las fortalezas y debilidades de las tecnologías IIoT para entender con mayor profundidad cómo deben aplicarse para obtener un rendimiento óptimo. Adicionalmente, se exploran las múltiples tecnologías existentes (protocolos, software, hardware o estándares entre otros) con el objetivo de obtener una visión de conjunto de las posibilidades que ofrece IIoT.

Tras la revisión de ambas facetas, se parte de los retos identificados en el primer punto y, considerando el panorama tecnológico presentado en el segundo, se va definiendo cada vez con más detalle la solución idónea para lograr un desempeño productivo óptimo. Esta solución se definirá desde la arquitectura básica que debe guiar toda la implementación hasta la selección de tecnologías específicas, de forma que se pueda construir a partir de ella aplicaciones concretas capaces de afrontar todas las deficiencias de la industria actual.

Por último, se valida la solución propuesta, desarrollando un conjunto de aplicaciones reales trasladadas a casos de uso en producción. En este punto, TSK proporciona los entornos necesarios para todas las implantaciones y representa un actor clave en la validación de la solución. Al depender de la buena ejecución de las soluciones para la oferta y servicios proporcionados a los clientes, se presenta como un elemento imparcial en la evaluación del rendimiento de las aplicaciones desarrolladas frente a soluciones de terceros o frente a la competencia. Esta aplicación en un entorno a la vez productivo y competitivo, pone a prueba la solución IIoT y las tecnologías implicadas en igualdad de condiciones frente al resto de los sistemas industriales actuales, dando una mayor dimensión a la validación.

Como consecuencia de esto, el objetivo de la tesis es validar que la industria puede beneficiarse de las arquitecturas IIoT en las siguientes cuestiones:

- Modernización de las redes industriales: aprovechar las arquitecturas IIoT para prepararlas para el futuro a la vez que se garantiza la fiabilidad y seguridad.

- Optimización de los sistemas de comunicación industriales: mejorar las comunicaciones gracias al amplio catálogo de tecnologías inalámbricas presentes entre las redes IIoT para hacer más eficientes los procesos industriales.

- Aumento de la capacidad de integración: facilitar la actualización posterior ante tecnologías y sistemas que se desarrollen tiempo después de la puesta en marcha original.

- Optimización del coste de las soluciones finales: reducir los costes de automatización y control aprovechando que las aplicaciones IIoT se apoyan en soluciones electrónicas de alto rendimiento a un coste muy ajustado.

Al mismo tiempo, desde el punto de vista de las tecnologías IIoT se busca alcanzar los siguientes objetivos adicionales:

- Validación de las arquitecturas IIoT en entornos productivos: aunque a día de hoy la tecnología IIoT es lo bastante madura para llevarse a la industria, es necesario investigación adicional para que esta aplicación sea realmente óptima.

- Perfeccionamiento de las arquitecturas IIoT: para abordar los retos de la industria, se diseña una arquitectura IIoT más eficiente para su integración industrial. Además se describe el proceso de diseño, dando pie a definir nuevas características para ampliar el presente estudio.

- Diseño de una arquitectura IIoT versátil: el planteamiento de la solución debe de ser adaptable a los diversos casos de uso analizados de manera que la solución validada pueda servir para muchos más escenarios.

Por tanto, se trata de realizar aportaciones tanto a la industria como al estudio de las tecnologías IIoT mediante la integración de ambas y validar que esta integración es beneficiosa para el entorno productivo.

1.3. Hipótesis

El enfoque de este trabajo se puede resumir en dos hipótesis principales:

1. Las arquitecturas IIoT permiten solucionar los retos actuales de la industria
2. Las arquitecturas IIoT están listas para integrarse en un entorno productivo a cualquier nivel

En mayor detalle, en la primera hipótesis se plantea que, tras un análisis de los retos a los que se enfrenta la industria actual, es posible definir, diseñar e implementar una arquitectura IIoT robusta que aborde cada uno de los retos encontrados y los solucione. De esta manera, se identifican las arquitecturas IIoT como una pieza clave de la automatización y control de la industria para los próximos años.

La segunda hipótesis propone que las tecnologías IIoT no solo son aptas para la industria, sino que su nivel de madurez tecnológico ya permite implantarlas en entornos productivos y que son lo bastante flexibles y robustas para aplicarse en una solución pequeña (como monitorizar algunas variables concretas) o en una implantación íntegra (asumiendo la labor completa de un SCADA).

1.4. Organización del documento

Para exponer los contenidos de este trabajo, se estructura el documento de forma paralela a la metodología planteada. En este primer capítulo se describe la motivación de la que nace esta investigación, los objetivos del trabajo y las hipótesis planteadas que guían toda la investigación. A continuación en el Capítulo 2 se analiza la situación actual de la industria, para obtener una visión de conjunto de los retos a los que se enfrenta. También se revisa el estado del arte de las tecnologías IIoT en el Capítulo 3 para conocer en profundidad sus posibilidades y cómo se abordan las soluciones de dicho ámbito.

Con estos dos puntos de partida, en el Capítulo 4 se define una solución IIoT que aprovecha las fortalezas de estas tecnologías para cubrir los principales retos de la industria. Dicha solución se propone primero de forma abstracta y

luego aterrizada a una topología y tecnologías concretas. Luego, para validar esta solución, se realizan implementaciones de la misma en múltiples casos de uso en el Capítulo 5. Al emplearse en entornos productivos reales, se puede validar su versatilidad, fiabilidad y eficiencia. Finalmente, en el Capítulo 6 se resumen las principales conclusiones del trabajo realizado.

1.5. Publicaciones asociadas a la tesis doctoral

En el curso de este proyecto se han publicado un artículo de revista Q1, dos artículos de conferencia internacional y un capítulo de un libro:

1. de Arquer Fernández, P., Fernández Fernández, M. Á., Carús Candás, J. L. y Arboleya Arboleya, P. An IoT open source platform for photovoltaic plants supervision. *International Journal of Electrical Power and Energy Systems* **125,** 106540. ISSN: 01420615 (feb. de 2021).

2. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. Determining Operational Constraints for IoT-Based Advanced Metering Infrastructure. *2021 IEEE Madrid PowerTech,* 1-6 (jun. de 2021).

3. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. en *Encyclopedia of Electrical and Electronic Power Engineering* (Elsevier Ltd, 2022).

4. Fernández Villán, A., Fernández Fernández, M. Á., Carús Candás, J. L., de Arquer Fernández, P., Arias Linacero, N. y Usamentiaga Fernández, R. *Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography* en *2018 2nd International Research Conference on Sustainable Energy, Engineering, Materials and Environment (IRCSEEME2018)* (jul. de 2018), 206-207.

Se describe a continuación en mayor detalle el trabajo asociado a estos artículos en relación a la tesis.

1.5.1. An IoT open source platform for photovoltaic plants supervision

Este artículo expone en mayor detalle uno de los casos de uso que se presentan en el Capítulo 5, de monitorización de plantas fotovoltaicas con un SCADA basado íntegramente en tecnologías IIoT de código abierto. Se publicó en la revista del primer cuartil *International Journal of Electrical Power and Energy Systems* (IJEPES) y puede usarse como referencia del resto de los casos de uso y aplicaciones descritos a lo largo de esta tesis. Destacar que el sistema desplegado con motivo de este artículo sigue en producción en la actualidad.

1.5.2. Determining Operational Constraints for IoT-Based Advanced Metering Infrastructure

Este artículo se realizó como un estudio secundario dentro de esta tesis al ser necesario detallar los requisitos hardware para el software utilizado en la solución IIoT planteada. Para este estudio se plantean equipos reales de uso industrial, junto con alguno de prestaciones domésticas/educativas. La consecuencia directa de este estudio son los requisitos que se exponen en la selección de tecnologías y hardware del Capítulo 4

1.5.3. Internet of Things (IoT) for Power Systems Applications

A consecuencia de la publicación en la revista IJEPES, se propuso la preparación de un capítulo en la *Ecyclopedia of Electrical and Electronic Power Engineering* donde se condensara el estado del arte y aplicaciones de IoT para sistemas de potencia. La documentación recopilada para este artículo es una

parte de lo expuesto en los capítulos 2 y 3, donde se describe el escenario actual tanto en la industria como las tecnologías IIoT.

1.5.4. Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography

En este artículo se presenta una primera pincelada de un algoritmo de detección de anomalías en subestaciones mediante la captura de imágenes termográficas. Este algoritmo se incorporó horizontalmente a la infraestructura IIoT construida en el contexto de esta tesis para darle un uso más generalizado, tal como se expone en el Capítulo 5.

Capítulo 2

Estado del arte en industria

La industria es un sector siempre en evolución. Durante muchos años el culmen de esta evolución fue la presencia de cadenas de montaje automatizadas o los sistemas automatizados, pero existen tecnologías capaces de hacer a la industria dar un nuevo salto evolutivo hacia un proceso más eficiente y sostenible. Estas tecnologías se engloban en los conceptos de la Industria 4.0 y suponen en algunos aspectos una ruptura con las arquitecturas tradicionales [5, 6].

2.1. Hacia la Industria 4.0

Desde mediados del siglo XX, la automatización y los sistemas eléctricos y electrónicos han intervenido cada vez más en la industria y los procesos productivos . Con el tiempo, los autómatas programables (o PLC por sus siglas en inglés, Programmable Logic Controllers) y los servidores industriales se han ido convirtiendo en la infraestructura básica de cualquier proceso industrial. Para coordinar toda la complejidad de estos sistemas, el control en industria se ha estructurado en cinco capas que se ordenan conforme la pirámide de automatización representada en la Figura 2.1 [7].

Este diagrama representa con precisión la relación y magnitud de las capas.

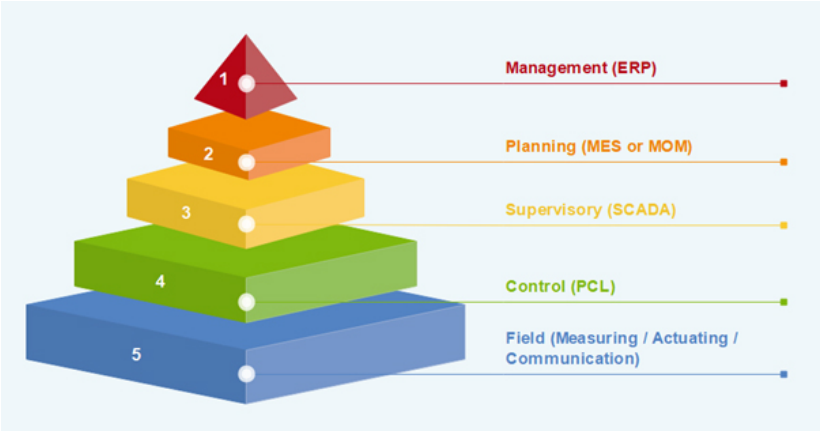


Figura 2.1. Pirámide de automatización. Tradicionalmente los niveles solo se comunican con el inferior o superior directo.

Así, las capas de menor nivel requieren más componentes, porque así consiguen un control preciso del proceso en todos sus puntos. Estos equipos suelen ser muy específicos para la tarea a realizar y se centran en que la operativa de la planta industrial se mantenga óptima. Por otra parte, las capas superiores se componen de menos herramientas y se centran en aspectos más estratégicos y abstractos del proceso productivo, como el abastecimiento de materias primas o la planificación de la producción en su conjunto.

La relación dentro de la pirámide es siempre a los niveles inmediatamente inferior o superior. En especial, a mitad de la pirámide se produce un salto muy significativo porque se pasa de un conjunto de tecnologías conocido como Tecnologías de Operación (OT por sus siglas en inglés, Operational Technology), propias de la mitad inferior de la pirámide, a Tecnologías de la Información (IT por sus siglas en inglés, Information Technology), propias da la mitad superior. En la Figura 2.2 se puede apreciar que el principal ámbito de conflicto entre ambos conceptos se da a la altura del SCADA.

En este contexto, la aparición de Internet, que ha supuesto una revolución completa en la sociedad, economía y producción, ha permitido a todos los sistemas

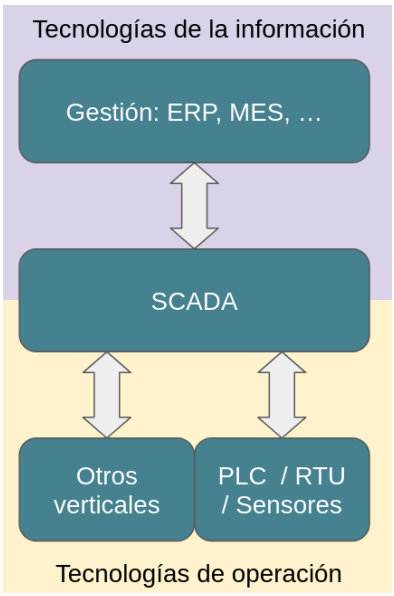


Figura 2.2. Separación de las capas IT y OT en una arquitectura industrial habitual. Las OT abarcan todo lo relativo a las capas inferiores de la pirámide de automatización y las IT la parte superior. El SCADA hace de puente entre ambos entornos.

de esas capas superiores, los correspondientes a las IT, mejorar su operativa y adquirir nuevas funcionalidades. Uno de los ejemplos más ilustrativos es cómo Internet ha permitido a grandes empresas orquestar producciones deslocalizadas para que funcionen como un solo sistema productivo.

Desde este escenario de partida, la industria se enfrenta ahora a una nueva revolución de la automatización. Esta revolución se apoya sobre un conjunto de tecnologías que han surgido de múltiples ámbitos como educativo, militar o gobernanza civil y han generado grandes expectativas en su puesta en marcha en la industria desde las primeras pruebas de concepto [8]. La incorporación de estas tecnologías son lo que se conoce como la Industria 4.0 [9].

2.1.1. Industria 4.0

La Industria 4.0 es el término con el que se designa la cuarta revolución industrial en la que se busca hacer de la industria un entorno más eficiente, inteligente, autónomo y sostenible. El camino para lograrlo es integrar en la esfera digital la maquinaria y los procesos del proceso productivo y los seres humanos relacionados, tanto operarios o coordinadores como clientes y proveedores.

La consecución de estos objetivos se basa en la aplicación de nuevas tecnologías de digitalización y actuación sobre las cuales se está desarrollando esta integración. Las tecnologías habilitadoras que destacan en la Industria 4.0 son [10]: *Big Data*, robótica autónoma, simulación y gemelos digitales, integración vertical y horizontal, interacción con la nube, Internet de las cosas, fabricación aditiva y realidad aumentada.

Big Data se refiere a un conjunto de tecnologías para el procesamiento masivo de datos. Estas herramientas permiten trabajar con un volumen mucho mayor que el tradicional, lo que posibilita análisis que hasta ahora no eran alcanzables y puede extraer correlaciones con variables que no era posible incluir en el análisis.

Robótica autónoma son un conjunto de elementos robóticos capaces de desplazarse de forma autónoma y con inteligencia y herramientas suficientes para resolver algunas tareas. De esta manera, por ejemplo, se puede desplegar un dron que realice una supervisión aérea completa de una zona o que acceda a una parte remota y potencialmente peligrosa para un ser humano con el propósito de llevar a cabo una tarea de mantenimiento. A día de hoy la actuación está siempre supervisada ya que estos equipos aun no han adquirido la fiabilidad necesaria que garantice la seguridad en una planta industrial, pero la monitorización sí resulta más viable.

Simulación y gemelos digitales comprende la capacidad de construir una

réplica de la planta industrial en la que sea posible ver cambios en tiempo real, plantear nuevos escenarios y validar puestas en marcha. En la medida en que el gemelo digital sea más preciso y ajustado a la situación real de la planta, esta herramienta podrá contribuir a obtener resultados predecibles en el entorno industrial [11].

Integración vertical y horizontal permite incorporar a una planta industrial nuevos sistemas y dispositivos. De no hacerse una integración adecuada, es fácil que una planta acabe teniendo muchos sistemas aislados y cada vez más difíciles de mantener. Esta es una característica claramente diferenciadora de las IT frente a las OT que se trata de llevar a dicho ámbito. Destacar que, para esta integración, una de las bases principales es el IoT.

Interacción con la nube es otra de las aportaciones directas de integrar las IT en el entorno de operación. La nube es el conjunto de equipos y conexiones presentes en Internet. Mediante esta integración se han reducido los costes asociados a transferir información a ubicaciones remotas y ha desencadenado la creación de clústeres especializados. Estos clústeres especializados proporcionan a un precio económico y de manera escalable una cartera de servicios que de otra manera un particular no podría obtener sin una fuerte inversión. Así, es cada vez más común encontrar servicios para el procesamiento remoto de datos sin necesidad de disponer de un Centro de Procesamiento de Datos (CPD) o para almacenar información de forma redundante sin sufrir problemas de escalado ante el aumento del volumen de datos.

Internet de las cosas es una pieza fundamental para la incorporación de la Industria 4.0, ya que sirve de nexo de unión entre otras muchas tecnologías. Su foco principal es proporcionar una alta interoperabilidad entre sistemas de forma eficiente y segura. Como la adaptación de IoT al entorno industrial

comprende dificultades específicas, se emplea con más frecuencia el término IIoT en este ámbito. Se presentará en más detalle en el Capítulo 3.

Fabricación aditiva conocida comúnmente como impresión 3D, es la creación de elementos o piezas a medida mediante maquinaria que va incorporando material sobre una cama de impresión. Esta tecnología se puede aplicar sobre plásticos o metales y ofrece una capacidad de prototipado rápido o de obtener reemplazos a medida en poco tiempo. En el primer caso ayudaría a la innovación e incorporación de nuevas herramientas y en el segundo permitiría minimizar las paradas de producción ante la espera de un repuesto. Con esta tecnología también es posible producir piezas que otras tecnologías como la inyección o forja no permiten construir.

Realidad aumentada y virtual ofrece una visualización inmersiva de la planta industrial. Este concepto va muy ligado a la creación del gemelo digital que proporciona los datos a los modelos virtuales o aumentados de estas soluciones. La realidad aumentada es particularmente práctica para obtener información adicional o difícilmente accesible en planta mientras se está presencialmente en ella, así como notificación de alertas en tiempo real o asistencia remota. La realidad virtual es un acceso remoto y seguro a una planta industrial que se puede usar para llevar a cabo la supervisión remota de la misma o para formar a nuevo personal en escenarios hipotéticos realistas.

La introducción de estas tecnologías en la industria ya se ha evaluado extensamente y son capaces de proporcionar grandes mejoras en el proceso productivo. De hecho, existen muchas soluciones que ya están aprovechándose de ellas tales como [12-15].

No obstante, estas tecnologías también han puesto sobre la mesa una necesidad que la industria debe resolver para adoptar estas tecnologías:

la conexión con Internet. Gran parte de las ventajas que vienen con estas tecnologías van de la mano del uso de Internet. Esta integración, que resulta más sencilla con las IT y ya se ha llevado a cabo en muchos casos, supone un gran reto para las OT. No solo por cuestiones tecnológicas como pueden ser los distintos protocolos usados, sino también más prácticas como fiabilidad o mantenimiento, o críticas como puede ser la ciberseguridad. Este problema es lo bastante grave para hacer que casos de uso exitosos de la Industria 4.0 no pasen de prototipos por la incapacidad de la industria para asimilarlos.

Por ejemplo, el uso de un algoritmo con aprendizaje en tiempo real puede usarse en cualquier sistema cuyos equipos vayan deteriorándose con el paso del tiempo y deban ir ajustándose a nuevas consignas. Los parámetros para alimentar el modelo analítico se encuentran recogidos en los PLC locales, pero aguas arriba del proceso productivo lo normal sería considerar solo unas pocas consignas (temperatura o caudal en un punto, revoluciones por minuto en un motor, etc.). Construir un prototipo que se integre localmente con estos PLC para descargar los datos y hacer una analítica puntual puede ser fácilmente realizable en una prueba de concepto. En cambio, implementar esto de forma continua supone un problema grande para la planta.

La aproximación más directa parece ser integrar estos datos con el resto de la planta, pero no es la más sencilla ya que implica actuar en muchas capas hasta obtener estos datos en la parte superior de la pirámide, además de saturar sistemas intermedios con mucha información muy poco relevante. Conectar directamente esta red a Internet para extraer los datos remotamente supondría una brecha de ciberseguridad muy evidente. También se podría incorporar un equipo que extrajera los datos y los enviara a ese servicio remoto, que tiene también implicaciones en ciberseguridad y confidencialidad, aunque no tan graves. Una aproximación cada vez más viable es simplemente aplicar el procesamiento en cercanía con un servidor local. El bajo coste y la minimización

de la electrónica facilitan este tipo de estrategias. Se trata de un ejemplo de integración de un sistema de IT en un entorno de OT.

Este tipo de soluciones es cada vez más común pero también tiene sus desventajas. Conforme se suceden soluciones que pueden aplicarse por este proceso, la convergencia entre los entornos IT y OT acorta la distancia entre ambos, generando un reto que si no se trabaja adecuadamente puede dar lugar a problemas graves de escalabilidad, fiabilidad y sobre todo seguridad.

2.1.2. Convergencia IT/OT

Según se ha visto, la estructura presentada en las capas de la Figura 2.1 da un enfoque global, pero existe una categorización más abstracta que separa los dispositivos de la planta por el tipo de equipos y redes de comunicación involucrados, lo que se correspondería con la brecha entre IT y OT (Figura 2.2).

Las IT tienen su fuerza principal en la conexión a Internet. La utilización de servicios de en línea y herramientas remotas, acelera el desarrollo, aumenta la capacidad de resolución de problemas complejos y mantiene la información actualizada para una toma de decisiones adecuada. No obstante, esto expone a los dispositivos a problemas de ciberseguridad, latencia y menos fiabilidad en la comunicación en tiempo real.

Por otra parte, las OT están centradas en proporcionar un servicio con la máxima fiabilidad posible. Debido a esto, se busca que las comunicaciones sean locales, en algunos casos incluso deterministas. En este caso la capacidad de procesamiento es mucho más simple pero el comportamiento es más predecible.

A causa de estos enfoques, las dos tecnologías muestran diferencias que dificultan su integración [16, 17]. Estas características serían las siguientes, que se resumen además en la Tabla 2.1:

- Comunicación: tal como se ha comentado, las IT se apoyan en la comunicación a Internet, mientras que las OT lo hacen sobre redes locales.

Característica	IT	OT
Comunicación	Internet	Redes locales
Aplicación	Negocio	Industria
Latencia	Segundos	<Milisegundos
Versatilidad	Alta	Baja
Procesamiento	Avanzado	Simple
Variedad	Poca	Alta
Equipos	Pocos	Numerosos
Ambiente	Controlado	Severo
Mantenimiento	Frecuente	Escaso
Actualización	Sencilla	Compleja
Ciberseguridad	Alta	Baja

Tabla 2.1: Resumen de las diferencias entre IT y OT. En el primer caso destacan la conexión a Internet y la gran capacidad de datos, mientras en el otro priman el aislamiento y la fiabilidad de los sistemas.

- Ámbito de aplicación: en la planta industrial se emplean principalmente las OT, mientras que las IT se usan para la parte de negocio o planificación. Usando como referencia la Figura 2.1, las OT se situarían en los niveles inferiores y las IT en la parte de negocio y planificación en la parte superior. Los SCADA estarían a caballo entre ambas.

- Latencia: muchas veces las OT trabajan en tiempo real, por debajo de los milisegundos, para garantizar el correcto funcionamiento y sincronización entre sistemas. En cambio, en las IT se valora la velocidad pero la conexión remota (a Internet) marca un tope por el que se debe asumir siempre un cierto retraso. Esto no significa que las IT no puedan realizar tareas de muy alta velocidad, pero hay que tener en cuenta que la transmisión sí es falible o se puede retrasar.

- Versatilidad: el tipo de dispositivos y funcionalidades utilizados en las IT

suele permitir la asignación de muchas tareas variadas al mismo equipo. Así, un PC puede ser usado para realizar un cálculo de simulación de fluidos o monitorizar un SCADA. Sin embargo, en las OT, los equipos suelen tener funcionalidad específica, lo que les permite ser más eficientes y garantizar la fiabilidad.

- Capacidad de procesamiento: las IT disponen de una capacidad de procesamiento avanzada por el tipo de equipos empleados y por el acceso a recursos externos de terceros a través de Internet. Las OT, en cambio, realizan procesamientos mucho más sencillos para resolver eventos y acciones concretos sobre el sistema en el que se emplean.

- Variedad: debido a la versatilidad de los dispositivos usados en IT, habitualmente se trabaja con muy poca variedad o se abstrae dicha variedad con un mismo Sistema Operativo (SO). En las OT, por el contrario, es muy común encontrarse con muchos verticales y dispositivos de todo tipo conviviendo en una misma planta industrial, para proporcionar la máxima eficiencia en todas las fases del proceso productivo.

- Número de equipos: gracias a la gran capacidad y versatilidad de las IT, es poco frecuente encontrarse muchos equipos en estas redes. Lo más habitual es que existan unos pocos: algunos para supervisión por parte de los operarios y uno o varios CPD con servidores industriales. En cualquier caso se trata de un número relativamente reducido pero equipos de altas prestaciones. Por otro lado, en las OT cada máquina o a veces cada sensor cuenta con un equipo específico para la tarea y de poca capacidad, lo que da lugar a una cantidad mucho mayor de dispositivos.

- Condiciones ambientales: en las OT es necesario que los componentes de una solución sean capaces de soportar las condiciones de una planta industrial, por lo que son más robustos y están mucho mejor protegidos. En las IT, por

contra, no solo no se espera que se sometan a condiciones extremas sino que en concreto en los CPDs es necesario disponer de una atmósfera controlada.

- Mantenimiento y actualización: al margen de la fragilidad debida a las condiciones ambientales, el hardware de las IT sufre mucho la obsolescencia y tiene una vida útil relativamente corta (menor de diez años). El software también debe mantenerse con mucha mayor frecuencia al día, aunque las IT presentan muchas facilidades para este tipo de actuaciones. Sin embargo, en las OT rara vez es necesario intervenir porque la vida útil suele ser mucho mayor (de veinte a treinta años). Las actualizaciones ni siquiera son aconsejables, pues pueden comprometer el proceso productivo o la integración con otros sistemas circundantes.

- Ciberseguridad: como consecuencia de la falta de mantenimiento y del trabajo en redes locales, las OT tienen capacidades muy limitadas frente a ciberataques. Su exposición en caso de acceso a Internet sería muy alto. Las IT, en cambio, están diseñadas para proporcionar una conexión segura a Internet y su constante actualización las mantiene protegidas de las vulnerabilidades que van apareciendo.

Se trata de dos formas de trabajo tan distintas que resulta muy compleja su convivencia en el mismo entorno. A pesar de eso, en la actualidad, ambas tienden hacia su convergencia en el marco de la industria. Una de las razones de esta evolución se ha valorado ya en el apartado anterior, donde se describía un ejemplo en el que instalar un equipo IT en un entorno OT puede ser la mejor forma de desplegar una solución. Pero también las OT obtienen muchos beneficios de las IT, como su versatilidad y facilidad de gestión. Además, algunas tecnologías de la Industria 4.0 como el gemelo digital requieren una integración plena de las OT con dicho gemelo, que no es más que un ejemplo directo de convergencia IT/OT [18].

Dado que esta convergencia genera muchos problemas [19] (vulnerabilidades de seguridad, pérdida de fiabilidad o incremento del mantenimiento, entre otros) se hace necesaria una arquitectura capaz de integrar elementos de IT y OT indistintamente. El objetivo es que elementos de los dos ámbitos puedan interactuar de forma acorde a sus propios entornos. Una de las mejores posibilidades tecnológicas actuales son las redes IoT industriales [20].

En líneas generales, IoT está pensado para incorporar la conectividad de Internet a objetos de bajo nivel. Si bien la integración entre IT y OT es más compleja, las herramientas y tecnologías necesarias para afrontar la convergencia son comunes a las empleadas en soluciones IoT, en especial de IIoT, para llevar la conexión de Internet de forma segura y fiable. Así, las arquitecturas IIoT pueden contribuir a acortar la brecha entre IT y OT y mejorar el rendimiento de la convergencia de ambos.

2.2. Retos

La Industria 4.0 supone a la vez una gran oportunidad y también un desafío. Esto se une a otros ya presentes en la industria desde hace tiempo, como las duras condiciones de trabajo, y a otros que se suman en los últimos años consecuencia sobre todo del avance tecnológico o la globalización.

Aunque el alcance de estos retos puede ser muy diverso y depende en gran medida del sector en el que se evalúen, hay una serie de problemas que se dan de forma transversal en varios entornos industriales. A partir de [21-24] se pueden recopilar principalmente: ciberseguridad, evolucionabilidad, convergencia IT/OT (presentada anteriormente en este capítulo), análisis de datos, fiabilidad y condiciones ambientales. No son problemas aislados ya que por ejemplo la convergencia impacta en la evolucionabilidad, la fiabilidad en el análisis, etc. De todos modos, se detallará a continuación cada reto para entender bien las causas

de los mismos y cómo se están afrontando.

2.2.1. Ciberseguridad

El reto más destacado de la era digital para la industria es con diferencia la ciberseguridad ([25-27]). En los últimos años la cantidad de ciberataques sufridos por todo tipo de compañías, asociaciones y hasta naciones ha ido incrementando hasta poner este reto en primera línea. Estos ataques pueden suponer desde una simple incomodidad hasta pérdidas millonarias o un peligro para la vida de personas en la planta industrial.

En concreto en la industria, los ciberataques más comunes serían los siguientes:

- Sabotaje: al infiltrarse en una red industrial es posible actuar sobre el proceso productivo. Esto permite por ejemplo cambiar la producción a parámetros que deterioren la calidad, o estropear la maquinaría al llevarla a un ritmo de trabajo de estrés. Otros peligros son la inutilización de un sistema por corrupción de su software o el daño a los usuarios de las máquinas. A pesar de ser el ataque más frontal contra el proceso productivo, este tipo de ciberataque busca principalmente una actuación discreta que le permita estar activo durante mucho tiempo sin que la empresa pueda localizar el problema de raíz, sino que se limite a cubrir los problemas generados de mantenimiento o producción.

- Denegación de servicio: sin siquiera entrar en la red industrial, un ataque puede forzar la caída de un servicio de una compañía frente a sus clientes. Entrando en la red puede hacer lo mismo también con servicios internos para la producción. El objetivo de hacer caer estos servicios es disminuir la fiabilidad de la producción y la confianza de los clientes, dañando la imagen de la empresa.

- Confidencialidad: habitualmente mediante la infiltración en la red interna de una compañía un atacante puede hacerse con información sensible. De esta manera pueden obtenerse datos de clientes, disposición y configuración de la maquinaria, secretos industriales o previsiones de producción. Con esta información se espera realizar un chantaje, aprovecharla para obtener alguna ventaja industrial o poner en evidencia a la empresa y dañar su imagen con la publicación de la información. Esto último también llevaría asociada la pérdida de ventaja competitiva si se difunden detalles de los procesos de producción.

Para combatir este tipo de ataques, la principal estrategia es el aislamiento de los elementos de la red. Lo que se busca en general es que, incluso si algún equipo presenta vulnerabilidades y sufre un ataque, este ataque no se extienda a otros. A bajo nivel, el responsable de este trabajo es el firewall. Un firewall permite diseñar una red industrial de forma que dos dispositivos no se puedan comunicar entre sí sin pasar por el firewall. El firewall por su parte se ocupa de garantizar que ese tráfico es el esperado y está autorizado. No comprueba que la información sea válida ni es una garantía de seguridad sobre la planta, pero sí ayuda a mitigar la mayoría de los ataques que suele sufrir una red.

Existen otras herramientas adicionales como el análisis de la red, que puede realizarse con un software adicional en el firewall o desde un equipo independiente. El análisis de la red consiste en controlar el tráfico existente para detectar anomalías. Un ejemplo de ellas podría ser un tráfico de red especialmente alto o en un protocolo no esperado. El objetivo es evitar aquellos problemas que el firewall no puede paliar y disponer de un informe más detallado del ataque recibido en caso de que se produzca. En algunos casos incluso puede ayudar a prevenir ataques.

Hay que tener en cuenta que la correcta configuración de estos equipos es compleja y requiere conocimiento experto. Muchas empresas no disponen de él y

no pueden o no están dispuestas a asumir el coste de contratar externamente este servicio. Un firewall mal configurado puede no suponer ningún tipo de defensa, producir falsa sensación de seguridad e incluso abrir vulnerabilidades adicionales. Aun cuando estos equipos funcionan óptimamente, no existe nunca una garantía de seguridad, aunque sí proporcionan herramientas para mitigar o analizar los ataques sufridos. Es por esto que la principal línea de defensa para una red debe venir siempre desde una concienciación de los usuarios en una cultura cibersegura y por la actualización frecuente de su software.

Sin embargo, estos dos puntos en particular han sido tradicionalmente muy complicados. Por un lado, la falta de actualización de software en los sistemas industriales es un problema muy extendido debido a la prioridad de la fiabilidad del proceso productivo y la amortización de equipos. Así, los dispositivos que permiten actualización de software no se suelen modificar para evitar poner en peligro la ejecución de los programas de control y monitorización. Además, los que no admiten actualizaciones se mantienen sin reemplazar aunque puedan ser vulnerables a la espera de amortizar la adquisición de los mismos. Por ambos motivos, es fácil encontrar en las redes industriales software con vulnerabilidades conocidas en ejecución. En algún caso incluso con vulnerabilidades identificadas como críticas.

Por otro, la concienciación de los usuarios resulta difícil en entornos industriales porque de nuevo prima la fiabilidad y la seguridad para los trabajadores. Un ciberataque es concebido como un riesgo intangible y se subestima enormemente. Es por eso que, en general, sea poco común que un usuario de una red industrial sea plenamente consciente de hasta que punto un ataque puede suponer un problema de producción o seguridad en su entorno ya sea por desconocimiento del problema global o por confianza de que no se van a sufrir ese tipo de ataques.

Además, muchos usuarios no han recibido formación en cuanto a qué

actuaciones pueden comprometer la ciberseguridad como, por ejemplo, abrir un correo desconocido, conectar un USB a su ordenador o mantener información sensible a la vista. De hecho, los usuarios son una fuente muy habitual desde la que se inician los ciberataques, ya que la ingeniería social es más sencilla que tratar de atacar un firewall bien configurado.

Dentro de la ciberseguridad, la industria energética y en particular la de distribución eléctrica debe atender con especial detalle a las cuestiones de la privacidad y la fiabilidad.

- Privacidad: los datos personales de una persona o empresa son cada vez más valiosos en la medida que existen más empresas capaces de sacar provecho de su uso para fines comerciales o de otro tipo. Por eso, también se hace cada vez más hincapié el proteger esa información y regularla para evitar los abusos en torno a ella. Así, toda empresa debe tener mucho respeto con el tratamiento de estos datos. En particular, la industria de la energía tiene una relación muy estrecha con los ciudadanos como cliente final y por tanto más responsabilidad en esta materia. A futuro, las estrategias de gestión de energía *behind-the-meter* o la relación con un cliente prosumidor intervienen en la vida doméstica y por tanto tienen una entrada en la privacidad de las personas mucho mayor de lo normal en otras industrias.

- Fiabilidad: al ser un recurso crítico, la industria energética tiene unos requisitos de fiabilidad mucho mayores que otras industrias. Por poner un ejemplo, un daño a la red que implique un sobrecoste de mantenimiento no sería tan grave como uno que produzca un apagón, que impactaría en la gobernanza, industria y vida de las personas. Pero también es importante en casos menos extremos. Por ejemplo adelantar una parada de mantenimiento de una central térmica porque se ha sufrido sabotaje puede desajustar la producción de un país y forzar una importación energética

indeseada, causando pérdidas económicas muy grandes.

2.2.2. Evolucionabilidad

Mantenerse al día es cada vez más una necesidad imperativa en el mundo industrial. La velocidad a la que evolucionan las tecnologías, las tendencias del mercado y las demandas de los consumidores exige que los profesionales y las empresas estén constantemente actualizados y adapten sus prácticas y conocimientos para seguir siendo competitivos. Debido a esto, la capacidad de evolucionar y adoptar innovaciones ha pasado de ser una ventaja a una necesidad para que una producción sea rentable conforme pasa el tiempo.

Las causas principales de esta aceleración son sobre todo la globalización y la digitalización, haciendo que la obsolescencia sea una amenaza real tanto para grandes empresas como pymes (Pequeña y Mediana Empresa) [28, 29]. Las que no se mantienen al tanto de los últimos avances corren el riesgo de quedarse retrasadas, perder oportunidades e incluso, si el avance es lo bastante relevante, quedar fuera del mercado.

Es por eso que, para mantenerse competitiva, una empresa debe hacer un constante esfuerzo de I+D+i a la vez que disponer de un entorno industrial capaz de integrar los nuevos avances que se incorporen. Este segundo aspecto implica no solo conservar la fiabilidad, seguridad y productividad de una planta tras la incorporación de un nuevo sistema, sino también ser capaz de agregar las nuevas funcionalidades y posibilitar la sinergia con las viejas funcionalidades, no imponer una gran complejidad en la planta y conservar un mantenimiento sostenible.

Estos dos puntos, complejidad y mantenimiento, serían los focos claros a abordar en este reto, ya que son los principales enemigos de esta necesidad de mantenerse al día y al mismo tiempo son consecuencia de ella. A continuación se detalla esta relación.

Complejidad

Conforme pasa el tiempo, cualquier instalación industrial necesita la incorporación de nuevos sistemas. Incluso sin impulso de innovación, esto se produce por el mero reemplazo de equipo obsoleto, defectuoso o del que se ha perdido servicio de asistencia por parte del fabricante. Cuando esto ocurre, los nuevos sistemas incorporados a veces se pueden integrar de forma horizontal con la planta, como puede ser un sensor de temperatura de un horno, que se incorpora a una lectura de un PLC y no genera mayor complejidad. No obstante, en la mayoría de los casos es necesario hacerlo de forma vertical, siendo un sistema aislado adicional con el que se tiene solo algunos puntos de conexión al resto de la red industrial.

Dicha integración vertical puede ser muy beneficiosa porque suele integrar soluciones eficientes, pero no siempre es fácil conservar la coordinación con el resto de la planta. De ser necesario, se desarrolla una integración *ad hoc* para trasladar una funcionalidad a otro sistema de la planta. La consecución de soluciones similares origina al cabo de un tiempo un montón de servicios añadidos que deben ser gestionados y actualizados.

Por ejemplo, se puede dar el caso de que se incorpore a un motor un sistema para detectar, por vibraciones, una anomalía. Este sistema puede comprender varios sensores de los que se recopilan datos y sobre los que se hace un análisis. Lo normal sería que un fabricante proporcionase toda esta funcionalidad en una solución compacta. Sin embargo, a nivel de planta industrial no solo sería interesante obtener estas lecturas sino quizá también emitir una alarma luminosa, o incluso detener el motor. Al ser funcionalidades que pueden depender del caso de uso, no es probable que la solución de monitorización la contemple de forma nativa y la propia empresa deberá generar un servicio que implemente esta funcionalidad.

Similar a este ejemplo, se producen constantemente nuevas necesidades y más con la instalación de un nuevo equipo. Dado que el SCADA de una

industria no se suele cambiar con frecuencia, pueden pasar más de 15 años sin una reestructuración importante. Al cabo de ese tiempo, la complejidad es tan grande que la planta empieza a perder fiabilidad y ya no es tan robusta. Cuando se llega a este punto, habitualmente la incorporación de nuevas soluciones es más difícil y mucho más arriesgada, ya que no está claro a qué otros puntos del proceso pueden llegar a afectar.

Partiendo del ejemplo del motor, si un tiempo más tarde se intenta incorporar un automatismo para arrancarlo si se cumplen unas condiciones (como que se detecte un movimiento cercano o similar), podría existir algún fallo si al mismo tiempo el detector de anomalías intenta detenerlo. En un caso ideal, se valorarían estas situaciones y se actualizarían todos los servicios que se puedan ver afectados. Sin embargo, muchas veces esto no se puede llevar a cabo. Si es un factor crítico, podría no llegar a instalarse por una dificultad práctica a pesar de que tecnológicamente sea perfectamente viable.

Hay muchas causas que dificultan, ante una nueva solución, que se pueda valorar o asumir el riesgo [30]. Las más habituales serían:

- Falta de documentación que impida reconocer correctamente todo lo que está instalado y en qué manera se relaciona.
- Ausencia de las personas involucradas en la puesta en marcha de otras soluciones que podrían proporcionar experiencia para suplir deficiencias de documentación.
- Incapacidad para contactar con los instaladores, escenario muy común cuando interviene una empresa externa subcontratada, ya que en muchos casos la rotación de personal de estas es alta.
- Falta de personal de desarrollo que pueda hacer las modificaciones necesarias para hacer compatible la nueva solución con otras anteriores.

- Cierre de empresas, que puede referirse al fabricante de un equipo tanto como a la empresa ocupada de su instalación o desarrollo.
- Otros procesos de la misma planta cuya relación pueda suponer también un problema y por tanto generen más capas de complejidad.
- Incluso el mero paso del tiempo puede suponer una barrera significativa al provocar el deterioro de documentación, falta de asistencia técnica o finalización de contratos comerciales.

La mejor manera de abordar esta dificultad es estableciendo una infraestructura en la que se pueda intervenir en una parte sin afectar demasiado a otras, es decir, que sea modular de diseño. El SCADA típico en industria sigue el concepto clásico de SCADA monolítico como sinónimo de eficiente y robusto. Con las tecnologías actuales, en especial las provenientes del ámbito IT, es posible hacer un entorno de trabajo eficiente, robusto y duradero que sea a la vez modular y distribuido. Esto no significa necesariamente descentralizado. La mayoría de las industrias trabajan por necesidad con una topología centralizada, pero eso no tiene por qué ser incompatible con una estructura modular.

Esto no solo contribuiría a una menor complejidad con la inclusión de nuevas soluciones, sino también a la mejora del mantenimiento de la planta industrial, ya que facilita la intervención en equipos sin alterar significativamente a otros. Esto no significa que no haya que parar la planta, sino que se reduce el riesgo de que se rompan compatibilidades o se generen complicaciones imprevistas por falta de conocimiento de las relaciones entre dos sistemas.

Mantenimiento

El mantenimiento es una preocupación común para cualquier industria. Sin embargo, la frecuencia y magnitud de las tareas de mantenimiento varían mucho. Algunas veces son tareas prácticamente diarias, como la limpieza de rodillos de

impresión, y otras son tareas que se realizan cada muchos meses o incluso años, como puede ser la limpieza de un horno.

Sin embargo, el mantenimiento es una parte poco deseada por la industria ya que implica detener la producción para poder realizar el mantenimiento y conlleva un gasto extraordinario de tiempo y dinero que repercute en los costes finales [31]. Lo deseable es tener que realizar estas tareas lo menos posible y poder preverlas para que no supongan una interrupción importante del servicio o la producción. Este esfuerzo de controlar las pausas ha evolucionado desde el mantenimiento correctivo o reactivo (que es desencadenado por un fallo de algún sistema) al preventivo (normalmente periódico en base a los tiempos estimados de duración) y luego al condicional (en base a unas medidas de referencia de calidad o rendimiento). De esta manera se pueden planificar mejor estas tareas para que no generen tanto impacto en la producción.

Si bien el mantenimiento condicional ya es bastante eficiente respecto al punto de partida, hoy día es posible utilizar algoritmos más complejos de análisis que permitan alargar más la vida de un equipo o sistema y dar pautas de mantenimiento más productivas en base al análisis de la producción por patrones de comportamiento. Este mantenimiento predictivo ha sido una de las primeras ventajas derivadas de la integración de las IT en el ámbito productivo, pero, una vez que se ha empezado a dar este salto la capacidad de análisis incorporada a las OT, ha dado como fruto también el comienzo del llamado mantenimiento basado en el riesgo.

Incluso con toda la información derivada de la predicción, hay fallos de equipos, servicios o producción cuyo impacto puede ser asumido en caso de que se produzcan, sabiendo que es una posibilidad, frente a la certeza de dedicar el tiempo a ejecutar la tarea de mantenimiento preventivo. No hay que olvidar que el mantenimiento predictivo, aunque sea muy preciso, no deja de ser también preventivo. Ponderar este riesgo requiere un conocimiento preciso y holístico del

proceso productivo y una buena calibración. Un intento mal ejecutado puede llevar a pérdidas mayores. En todo caso, a día de hoy es posible y se denomina mantenimiento basado en riesgo.

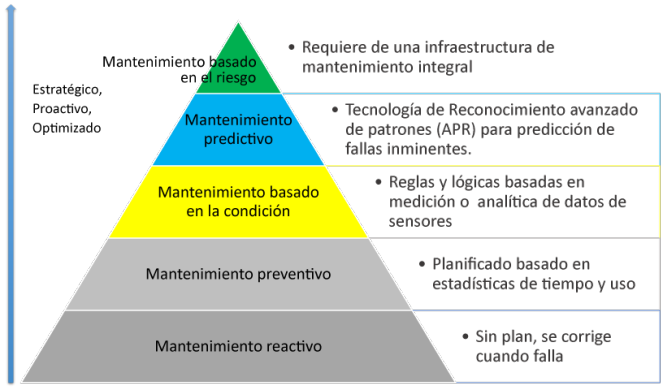


Figura 2.3. Pirámide y evolución del mantenimiento. El mantenimiento basado en riesgo exige que toda la base de la pirámide esté bien desarrollada para poder desplegar su potencial.

La dificultad radica en que, para poder utilizarlo de manera eficiente, el análisis debe disponer de una simulación lo más completa posible de toda la planta, de una infraestructura de mantenimiento ya instaurada y hasta de la previsión de producción, gestión de stock, etc. El grado de digitalización que hay que alcanzar es un reto ingente para muchas industrias que pueden permitirse mantenerse en niveles más bajos de productividad. Es por esto que se encuentra en la parte más alta de la pirámide del mantenimiento (Figura 2.3) como una pieza que se apoya en todas las demás. En aquellas en las que es necesario, la concepción del gemelo digital de la Industria 4.0 es idónea para afrontar este reto.

Además de estos mantenimientos, que se podrían considerar de naturaleza mecánica, con la aparición de la electrónica y la informática en industria se añaden los propios del software, principalmente actualizaciones. A pesar de eso, durante

muchos años, la informática en industria ha seguido la premisa de actualizarse solo si no funcionaba bien, lo que ha generado mucha obsolescencia informática. Es muy común el software que ya no tiene soporte operativo e incluso está desaconsejado utilizar, como puede ser el uso de versiones de Windows antiguas por miedo a que se produzca alguna incompatibilidad con una aplicación vieja.

El hincapié en seguridad llevado a cabo por las grandes tecnológicas y los integradores industriales ha reducido mucho en los últimos años esta brecha de obsolescencia pero sigue siendo una dificultad presente en la industria. Además, la atención a las actualizaciones hace necesario vigilar que la fiabilidad, eficiencia y seguridad (en este caso en el ámbito de riesgos laborales) no se vean comprometidas.

De la mano del apartado anterior, la complejidad juega aquí un papel relevante ya que, conforme se complica la funcionalidad de un entorno productivo, soportar un mantenimiento eficiente se hace cada vez más complejo. Cuantas más aplicaciones o servicios se ejecutan en una planta, más difícil es asegurar que todos están al día de los últimos parches de seguridad y que un vertical determinado no está comprometido.

2.2.3. Interoperabilidad

La interoperabilidad es la capacidad de los sistemas de entenderse entre sí. Esto presenta muchas ventajas cuando se puede ejecutar bien, pero supone un reto alcanzar una buena comunicación [32, 33]. Este reto de la automatización industrial es una consecuencia directa del marco de convergencia IT y OT ya expuesto. Se podría dividir entre conectividad e integración.

La integración, tanto vertical como horizontal, se refiere a la capacidad de un sistema para poder conectar su funcionalidad con otros sistemas adyacentes en el proceso de manera que puedan operar con la máxima eficiencia. Si es horizontal, se trata de una conexión directa y entre equipos de mismo nivel, mientras que la

vertical es entre dos niveles de operación distintos. Si no hay integración directa y es necesario usar un intermediario lo más común es considerarla también vertical, ya que ese intermediario actuaría verticalmente sobre lo que se quiere integrar.

La integración en OT se hace habitualmente desde un equipo hacia un PLC verticalmente y entre los propios PLC de forma horizontal. Entre los servidores industriales es más dispar ya que pueden estructurarse en jerarquía o como cluster. En cualquier caso, lo normal es que haya un servidor central a nivel de planta. El tipo de protocolos usado suele ser maestro-esclavo, lo que favorece escenarios verticales. De hecho, la incorporación de nuevos equipos suele hacerse siempre de forma vertical con lo existente, sin importar lo que se haya incorporado, de manera que los equipos nuevos se integren como maestros y así no sea necesario alterar la funcionalidad de algo ya existente.

La integración en IT puede ser indistintamente vertical u horizontal. Sin embargo, como suele estar orientada a servicios, lo normal es que sea predominantemente horizontal. En este caso los protocolos empleados suelen ser de tipo cliente-servidor o publicación-suscripción. La orientación a servicios hace que las interrelaciones se den de forma bidireccional. Por ejemplo, un servicio desplegado para el control de la producción usa y es usado por un servicio que gestiona las finanzas de la empresa. Esto se propaga también a las modificaciones posteriores con más facilidad, de forma que la tendencia de integración horizontal suele conservarse más fácilmente con el paso del tiempo.

Esta diferencia se traslada a la ciberseguridad, el mantenimiento o la versatilidad de forma directa, dando pie a los puntos de mejora ya conocidos de las OT. Pero no hay que olvidar que también actúa en favor de la fiabilidad. El objetivo de la convergencia entre tecnologías es que las IT sean capaces de favorecer esta integración horizontal, en especial a largo plazo, para reforzar las debilidades sin comprometer las fortalezas. Como es evidente, esto es bastante complejo y lo más normal es que se dé una convergencia imperfecta en la que las

IT se conviertan en un nuevo vertical en la planta o en las que las IT dificulten el mantenimiento o supongan una amenaza de ciberseguridad.

En el caso de la conectividad, el núcleo del reto es la comunicación que se usa para la integración. Como se ha dicho, en las OT lo normal es utilizar protocolos de tipo maestro-esclavo. También es común que cada protocolo tenga su propia definición para los datos proporcionados y que tenga poca seguridad. La definición de los datos mejora la fiabilidad de la comunicación, ya que no hay que interpretar los datos una vez recibidos. Por poner un caso simple, si se solicita una señal digital se obtiene una señal digital, no una cadena de texto que deba ser interpretada. De esta manera, los protocolos empleados pueden llegar a ser muy costosos de implementar, pero una vez hecho no requiere mucho procesamiento usarlo.

La falta de seguridad, en cambio, es una consecuencia histórica de los protocolos de automatización originales que han perdurado por ser muy compatibles o estar muy extendidos. El ejemplo más claro es Modbus, que está muy extendido, es sencillo, tiene compatibilidad universal, pero carece de las seguridades más básicas como la autenticación y la encriptación. Otros, por el contrario, están bien preparados para los escenarios de seguridad actuales, como OPC UA, que se cuenta entre los protocolos propios de las OT.

En general en las OT existe una amplia variedad de protocolos. Muchos fabricantes llegan a diseñar su propio protocolo con el fin de proteger su solución con la máxima eficiencia. Para facilitar luego su aplicación ofrecen documentación sobre dicho protocolo, herramientas de integración o una integración vertical a través de otro protocolo conocido.

En las IT ocurre todo lo contrario: los protocolos suelen ser una carcasa que proporciona seguridad, y, sin embargo, la información a transmitir no suele estar tan determinada. Esto ha favorecido, por ejemplo, la aparición de la web moderna, al ser posible crear muchas aplicaciones sobre un conjunto de

protocolos muy pequeño y, por tanto, prácticamente universal. En las IT, los protocolos rara vez son particulares. Se aspira a que los servicios tengan un rango más universal, en consonancia con el alcance de Internet. Por este motivo, se prefiere desarrollar soluciones estándares. Estos protocolos a su vez exigen más capacidad de procesamiento en tiempo real, pero este requisito es perfectamente asumible para los equipos usados en IT. A medida que la electrónica se vuelve más económica y compacta, también los equipos de una planta industrial están más preparados para estos retos.

Al mezclar ambas tecnologías, se pueden aunar ventajas pero se conjugan dificultades también, como la variedad de protocolos de las OT o la necesidad de seguridad de las IT. Un inconveniente que proviene de ambos a la vez es la heterogeneidad de los datos y su formato [34]. En el caso de las OT al menos está limitado a las señales tradicionales (digitales, analógicas, enteros, flotantes, etc.) pero las IT pueden llegar a aportar otro tipo de datos muy dispar (imágenes, coordenadas, etc.). Además, en las IT los datos no vienen tan claramente definidos, lo que dificulta la conectividad.

La unificación o armonización de estos datos es imprescindible para que los equipos se puedan entender y, sobre todo, si se quiere agregar la información para hacer análisis de mayor nivel entre varias partes de una planta. Cualquier algoritmo o procesamiento dejaría de ser válido si los datos con los que se le alimenta están mal armonizados y no son comparables.

2.2.4. Análisis de datos

La toma de decisiones es la principal aportación humana a cualquier proceso productivo, ya sea determinando los niveles de producción, cuánto es posible forzar las capacidades de un sistema, en qué situación el proceso alcanza un mayor rendimiento, etc. Sin embargo, la cantidad de tomas de decisión que pueden ser necesarias en una planta industrial es ingente. Muchos parámetros de

producción son ignorados y simplemente configurados por defecto. En ocasiones han sido optimizados durante la puesta en marcha pero no se han modificado para adaptarlos a nuevas situaciones. A un nivel más elevado, la gestión de la cadena de suministro y de la distribución se vuelven muy complejas, de modo que, con frecuencia, se recurre a simplificaciones. En general, una correcta toma de decisiones puede tener un impacto muy alto en el rendimiento de cualquier industria.

Hoy en día la inteligencia artificial (considerando aquí una definición amplia de toda la variedad de algoritmos y tecnología que engloba este término) ya puede asumir algunas de estas tomas de decisión a bajo nivel con plena solvencia y puede ayudar en gran medida en las de alto nivel [35]. Por supuesto, esto no significa eliminar el factor humano que debe seguir supervisando o validando estas decisiones, así como configurar correctamente esta tecnología. Pero de la misma manera que una hoja de cálculo puede reducir enormemente el análisis de datos en bruto, el uso de inteligencia artificial puede aligerar el trabajo de supervisión en una planta industrial e incluso proponer optimizaciones que, por falta de tiempo o consideración, un ser humano no llegaría a valorar.

La principal dificultad en este análisis, al igual que sucede en otros ámbitos en los que se usa, tiene que ver con la generación de los algoritmos adecuados, desde la detección de la necesidad hasta la implementación práctica del procesamiento. Pero específicamente en industria, una de las mayores barreras se encuentra ligada a la ubicación de estos procesamientos. Lo normal sería alimentarlos con la máxima cantidad de datos relacionados con el problema a tratar, pero hay mucha información de un proceso que no se recoge a alto nivel porque no tiene utilidad y sobrecarga la red. Otra estrategia similar es la compactación de datos (obtener la media cada minuto o diez minutos de una señal en vez de sus lectura cada segundo). Debido a esto, la ubicación ideal para desplegar un procesamiento (ámbito IT) carece de los datos necesarios para hacerlo, mientra que dónde si

hay datos abundantes y mejor respuesta en tiempo real (ámbito OT) no está plenamente preparado.

La mejor aproximación a este problema hasta ahora ha sido el desarrollo de tecnología *edge*. Esto se refiere principalmente a equipos capaces de funcionar a nivel de planta pero con potencia de procesamiento propias de un servidor pequeño. Así, es posible instalar estos equipos cerca de la información sobre la que debe trabajar. El uso de estos equipos supone un ejemplo de convergencia IT/OT que ya se ha tratado, pero facilitan acceder a los datos en el lugar adecuado sin necesidad de trasladar un volumen alto de lecturas por toda la red de planta.

Un pilar importante para esta toma de decisiones está reflejada en los retos anteriormente descritos, y consistiría en la recopilación y armonización de la información de la planta. Sin la aportación de datos sobre los que trabajar, toda toma de decisiones será susceptible de desviarse con facilidad y si los datos obtenidos no pueden ser procesados de manera comparable, carecerá de sentido su adquisición. También es fundamental que la información sea válida y fiable, esto es, que refleje la realidad y sea lo más completa posible.

2.2.5. Fiabilidad

La fiabilidad es un concepto asociado a la industria desde hace mucho tiempo. La producción industrial siempre ha necesitado mucha fiabilidad en sus equipos y procesos por razones tanto de productividad como de seguridad laboral [36]. Como consecuencia de esto, algunos sectores, entre los que se puede destacar las redes de distribución eléctrica, se han apoyado tradicionalmente en la replicación de sistemas críticos. En otros sectores, esta fiabilidad va también de la mano del control de calidad, que cada vez es más importante para una producción competitiva.

A estas consideraciones se unen otras en los últimos años como la ciberseguridad (ya tratada en esta sección) y la integridad de los datos. La

integridad es fundamental para la fiabilidad, dado que es la base para cualquier análisis que se haga con la información. Para que un conjunto de datos se pueda considerar íntegro, debe ser completo, preciso y coherente, [37] y debe mantenerse así durante todo su ciclo de vida. Es evidente que conseguir esto tiene un impacto en la fiabilidad de cualquier proceso.

Decir que un dato obtenido es completo, significa que no necesite nada más para representar la información a la que se refiere y que no tenga un histórico fragmentado. Por ejemplo, para obtener el rendimiento de una planta fotovoltaica al final del día es necesario disponer de toda la curva de producción. Podría usarse una curva construida con datos agregados cada varios minutos, pero lo que no sería admisible para un análisis es tener una medida cada segundo de una hora del día y unos pocos datos más del resto. Un conjunto de datos así haría que fuera muy difícil lograr un análisis relevante de la producción.

Al hablar de precisión no solo se entiende tener muchos decimales en el caso de variables analógicas, sino que también hay que tener en cuenta otros formatos y, por tanto, puede referirse a la resolución de una coordenada de geoposicionamiento o imagen, e incluso considerar la pérdida por compresión de un dato. Este contexto debe acompañar al dato siempre que sea necesario para interpretarlo correctamente y usarlo para su análisis. Una variable dada (del tipo y formato que sea) debe ir acompañada de toda la información para usarla. De esta manera será completa y precisa a la vez.

Para garantizar la coherencia de un dato, es primordial tener una referencia con la que hacerlo. Esto no siempre es posible o tiene sentido aunque sí es habitual que en las mediciones recogidas haya una cierta relación entre los datos. También es común que los datos mantengan un patrón. Por ejemplo, si se tienen varias sondas de temperatura en un horno, no sería razonable que una de ellas tuviera un valor mucho menor que el de todas las demás. Eso podría indicar algún defecto en la medida. Tampoco tendría sentido que su temperatura cambiara 200 grados en el

lapso de unos milisegundos, ya que la temperatura debería variar paulatinamente. Este tipo de medidas puede indicar que la variable está mal obtenida. Pero, incluso así, también podría indicar otras situaciones, como que la sonda está estropeada, que el horno está deteriorándose por algún punto o, si el valor es mayor de la cuenta, que el horno podría estar sufriendo una fuga. Como puede verse, garantizar la coherencia de un dato podría ser una tarea compleja y requiere muchas veces análisis cerca del origen del dato. Esto puede abordarse cada vez mejor gracias al procesamiento *edge* ya mencionado en el apartado anterior. De esta manera es posible evitar que lleguen datos discordantes a un algoritmo de procesamiento.

Finalmente, incluso consiguiendo todas estas características para cada dato, faltaría una pieza fundamental que es asegurarse de que se mantiene completo, preciso y coherente durante todo su ciclo de vida. Esto significa no solo en la adquisición, sino en el destino de almacenamiento final, los procesamientos que se operen sobre él, etc. La dificultad de hacerlo radica en los intermediarios que transmiten el dato. Por muy precisa que sea una medida, si se pasa por un protocolo o servicio que no es capaz de gestionarlo, es fácil perderlo. Un ejemplo cotidiano de ello es la transferencia de imágenes. Muchos servicios de mensajería realizan compresiones con pérdida de información cuando se utilizan para enviar imágenes. También un sensor de geoposicionamiento obtiene mucha información sobre la calidad de su medida, pero no todos los servicios están preparados para gestionarlo. Otro ejemplo podría ser una pérdida de datos históricos por problemas de conexión con el equipo que los adquiere o por la corrupción de un mensaje. Son todo ejemplos de casos donde un dato era completo, preciso y coherente en origen, pero pierde parte de su contexto al pasar por una arquitectura de supervisión.

2.2.6. Condiciones ambientales

En cualquier planta industrial, a los retos anteriores siempre hay que añadir las condiciones ambientales. Polvo, temperatura, humedad, exposición a la intemperie, golpes o vibraciones son algunas de las condiciones más habituales que pueden sufrir los equipos. Es común disponer de armarios eléctricos herméticos en las instalaciones industriales para evitar en que los dispositivos usados se vean afectados por el ambiente, pero no siempre se logra y hay que contemplar la posibilidad de algunas fugas, aperturas del armario, etc.

De esta manera, los equipos de OT suelen estar preparados para estas eventualidades, con el denominado grado industrial que entre otras cosas determina un rango mayor de lo normal de temperatura para su operación o garantiza el funcionamiento del equipo sin necesidad de ventilación adicional (lo que lo haría muy vulnerable a la acumulación de polvo).

Habitualmente, esto no es así para los servidores industriales. Estos tienen que estar cerca de la operación a la vez que instalados en un entorno más controlado, que se suele ubicar en salas aisladas. No es común en estas tener una atmósfera perfectamente controlada, como podría ser la de un CPD, pero sí que esté separada de las condiciones extremas de algunos procesos industriales.

La convergencia de las OT y las IT en este caso supone un reto para las IT, al ser necesario considerar este requisito adicional con equipos que no están tan preparados para entornos agresivos y son más vulnerables a condiciones extremas. De todos modos, el mercado ya dispone de bastantes equipos (con una oferta creciente día a día) con buena capacidad de procesamiento y mucha resistencia. El mejor enfoque en este caso es mantener la solución independiente del equipo para que se pueda utilizar el que mejor convenga en cada caso.

Por ejemplo, para una misma solución de monitorización fotovoltaica podría ser útil un equipo hermético capaz de permanecer a la intemperie en el caso de pequeñas plantas donde no suele haber salas de operación. Sin embargo, sería un

requisito excesivo para instalaciones mayores, donde es común que se agrupen *arrays* de inversores en salas técnicas.

Como se puede ver, este reto no es tanto una limitación técnica severa como un requisito que hay que considerar siempre en cualquier solución industrial.

2.3. Sistemas de gestión energética

En paralelo al desarrollo de la Industria 4.0 para mejorar la eficiencia, la industria se enfrenta también a una tendencia global de construir una producción más sostenible. La sostenibilidad es una estrategia a nivel mundial para construir un tejido industrial y productivo apto para un consumo de bienestar, pero con un impacto medioambiental bajo, que pueda conservarse indefinidamente en el tiempo con los recursos existentes.

Entre estos recursos, la energía eléctrica es uno de lo más universales, ya que en mayor o menor medida se utiliza en todas las industrias. Debido a esto, la transformación hacia la sostenibilidad pasa inevitablemente por una buena gestión de esta energía.

En términos generales, los sistemas de gestión energética son aquellos que se ocupan del control, supervisión y optimización de la generación y consumo de energía. Esto abarca desde una planta de producción, como una central nuclear, hasta un contador eléctrico de un hogar, pasando por redes de distribución, centros de transformación, plantas fotovoltaicas, hidráulicas, térmicas, ciclos combinados, generadores por motor a gas o diésel, vehículos eléctricos y estaciones de recarga, etc.

Esta industria, centrada sobre todo en la generación, traslado y consumo de la energía eléctrica, es uno de los grandes pilares para el resto de las industrias a día de hoy y se estima que cada vez será más importante. Además, en los países europeos, la industria energética juega también un papel crucial también en la

sostenibilidad del transporte a largo plazo.

Sin embargo, esto también conlleva sus retos. El más importante sería quizá la electrificación masiva que esto supone. Con la evolución de la industria, el consumo energético en Europa ha tenido un incremento muy significativo, pero se prevé que estrategias como la integración del vehículo eléctrico, la expansión de los sistemas electrónicos (hogares fuertemente domotizados, ciudades inteligentes, etc.) o la transición al uso de hidrógeno representen un esfuerzo adicional para estas redes de distribución [38].

Otro reto significativo es la descarbonización de la producción, en el sentido de que se busca depender cada vez menos de combustibles fósiles y recurrir más a las energías renovables. Este enfoque aborda frontalmente la reusabilidad de recursos, pero exige una mayor gestión energética, dado que controlar plenamente las condiciones de producción resulta más complejo que, por ejemplo, en una central térmica clásica.

También se agudizan condiciones que se han considerado siempre en las redes como es la importancia de un suministro garante de energía. Al aumentar la dimensión y la complejidad del sistema, su control y supervisión se torna más difícil. A esto hay que añadir que la dependencia de la electricidad es muy alta en la mayoría de los países y, por tanto, un problema que puede tener consecuencias profundas. Estrategias como la diversificación de fuentes de energía, la duplicidad de vías de conexión o la gestión de los grandes consumos para evitar demandas excesivas se vuelven fundamentales para construir una red de distribución fuerte.

También se están buscando constantemente nuevas estrategias que mejoren la eficiencia de la red eléctrica y su predecibilidad. La actuación más allá del contador (*behind-the-meter*), mismamente, consiste en acceder a las aplicaciones de mayor consumo para hacer que su uso sea más eficiente para la red eléctrica. Esto, que ya se lleva a cabo en los casos de grandes consumidores, puede aplicarse también a los pequeños gracias a tecnologías como IIoT.

Otros enfoques buscan reorganizar los consumos para evitar grandes picos de demanda. Además, se intenta abordar a gran escala una disminución del consumo tanto a nivel industrial como a través de iniciativas de ahorro doméstico. Una de las más interesantes a la vez que compleja es la estimulación de pequeñas producciones. La facilidad viene de una disminución directa en el consumo sin necesidad de inversión a gran escala. Sin embargo, esto también tiene dificultades porque no se conoce bien el consumo real de la red al estar parte de él absorbido por esquemas de autoconsumo.

Estos serían algunos ejemplos de retos y tendencias que se pueden encontrar en los sistemas de gestión energética. Como se puede ver, es un sector en el que el impulso tecnológico y la investigación pueden jugar un papel muy grande a medio plazo para cualquier organismo y las tecnologías de la Industria 4.0, entre ellas IIoT, están ayudando a abordar muchas cuestiones [39].

A este respecto, dentro de la abundante literatura en torno a este tema, cabe destacar las siguientes aportaciones:

- Predicción meteorológica para ajustar el consumo con antelación [12]. Esta propuesta podría incrementar notablemente el rendimiento de redes con un consumo significativo de energías renovables (sujetas a condiciones meteorológicas), reduciendo de este modo su dependencia de los combustibles fósiles. Es particularmente interesante en redes con alta dependencia de las energías renovables como es el caso de España [40].

- Mejora de la distribución del consumo mediante el análisis en puntos intermedios para evitar los picos [41]. Es un uso directo de la tecnología de procesamiento distribuida que posibilita IIoT. Este caso de uso se enfoca además en reducir el coste energético del consumidor.

- Análisis más allá del contador para identificar hogares que dispongan de producción fotovoltaica [42]. El planteamiento consiste en identificar

también los hogares que no tienen vertido a la red, para poder realizar estimaciones más precisas de consumo. El uso de paneles podría ocultar un potencial demandante en aquellos momentos en que la producción fotovoltaica no está disponible.

- Ejemplo de uso de tecnologías IoT para mejorar el rendimiento energético de un equipo doméstico [43]. Este caso de uso mejora las capacidades de una nevera para que siga satisfaciendo los intereses del usuario, pero con un consumo menor y más eficiente. Con este ejemplo se validan otros caminos alternativos a la sustitución del electrodoméstico para mejorar su eficiencia energética.

Como se puede ver, la aplicación de IoT en la industria energética puede dar lugar a muchas mejoras significativas en cualquier nivel, como el de los grandes productores, la distribución o el uso doméstico. A grandes rasgos, la integración de esta tecnología ayuda a recopilar y organizar la información y habilita la actuación local. Ambas cosas pueden ayudar a gestionar un entorno de grandes dimensiones de manera que acciones pequeñas (como aplazar unos minutos la recarga de un coche eléctrico), al realizarse de forma masiva, provoquen un impacto enorme en el conjunto del sistema.

Capítulo 3

Estado del arte de las tecnologías IoT

El término Internet de las Cosas o Internet of Things (IoT) ha ido evolucionando con el paso de los años desde su primera aparición en 1999. La idea original se centraba en la conexión de objetos físicos a Internet para añadirles un valor, ya sea recogiendo datos que dicho objeto produce o ampliando su funcionalidad. Esto también puede orientarse a una conexión local con objetos colindantes de forma que forme un ecosistema digital.

El núcleo de esta idea sigue siendo válido, pero los avances en las TIC le confiere una profundidad nueva al concepto. Avances como la llegada de los teléfonos inteligentes, las tecnologías inalámbricas, los vehículos no tripulados o la realidad virtual son solo una parte del escenario tecnológico disponible a día de hoy, del que dependen las posibilidades del IoT de cara a la conexión de dispositivos y potenciar el valor de esa unión digital.

Es por esto que, aunque a veces se utilice el término tecnología para definir IoT, en realidad se trata más bien una estrategia susceptible de ser aplicada en una arquitectura de comunicaciones.

3.1. Historia

En sus primeros años, incluso anteriores a la creación del término IoT, esta idea ya se estaba llevando a cabo con la tecnología existente: conexiones ethernet, protocolos de redes como HTTP o interfaces mecánicos. Sin embargo, conforme el concepto ganaba fuerza, traccionaba nuevos desarrollos tecnológicos, definiendo necesidades frente a nuevos protocolos y asentando incluso la hoja de ruta de estándares y tecnologías de las que podía sacar provecho.

Algunas de estas necesidades tractoras serían la portabilidad, el bajo coste, el tamaño reducido o la versatilidad [44]. A tenor de estos requisitos, la tecnología ha evolucionado para proporcionar conectividad inalámbrica (WiFi, Bluetooth, ZigBee, 6LowPAN, etc.), protocolos de comunicación ligeros (MQTT, CoAP, REST, etc.), capacidades de procesamiento de bajo coste o de muy variada aplicación (como en las grandes familias de microcontroladores) y otras funcionalidades que completan el escenario, como el autodescubrimiento (por ejemplo con el protocolo Web of Things) o la comunicación en malla (aplicable sobre múltiples protocolos).

A medida que iban apareciendo más dispositivos y aplicaciones específicamente diseñadas en el campo de IoT, surgía una necesidad adicional: abstracción. Al tratarse de una estrategia de comunicación, la cantidad de escenarios en los que se aplicó IoT fue incrementándose hasta alcanzar un grado de variedad considerable, al tiempo que las soluciones centradas en un solo contexto acababan por ser reutilizadas en aplicaciones completamente ajenas a su planteamiento original. Es por eso que también aparecieron herramientas para abstraer esas aplicaciones para la gestión de equipos, redes y software de las arquitecturas IoT.

Una de las herramientas más comunes en este ámbito es la de la plataforma IoT. Dicha plataforma no es otra cosa que un servicio centralizado que se

encuentra conectado a los distintos equipos IoT y permite trabajar con la arquitectura de forma unificada. También existe la aproximación, compatible con la anterior, de introducir la abstracción en el dispositivo IoT, otorgándole herramientas de integración estandarizadas o un software común. Esta última opción, además, sirve para facilitar el despliegue de aplicaciones de análisis en los dispositivos, lo que se conoce como Edge Computing y puede facilitar mucho la escalabilidad de una arquitectura [45].

Gracias a este desarrollo, IoT ha pasado de representar un conjunto de pruebas de concepto prometedoras y soluciones aisladas a vertebrar estrategias empresariales e impulsar nuevos modelos de negocio. En su inicio, su impacto se concentró en los sectores medioambiental [46] y educativo [47], si bien pronto dejaría su huella en la Medicina [48] o a la Domótica [49]. A día de hoy está penetrando cada vez más en los entornos industriales, hasta el punto de que se ha desarrollado un concepto independiente de *IoT para industria* (Industrial IoT o IIoT). Este enfoque específico se debe a que en industria, el catálogo de requisitos de la solución se desvía un poco de otras aplicaciones actuales de IoT, principalmente en lo relativo a seguridad y robustez [50, 51]. Es por esto que en los sucesivo se orientará todo en torno al concepto de IIoT en lugar de IoT.

3.2. Arquitectura

La principal aportación de IIoT es la capacidad de hacer frente a los retos de conectividad y masificación de un entorno. Es por esto que la topología concreta de la arquitectura que permite esto es un punto todavía difícil de esclarecer de forma genérica. Es decir, existen múltiples arquitecturas con un enfoque IIoT válido pero de muy distinta composición.

De hecho, en el momento en que se redacta esta memoria no existe aún un estándar que haya sobresalido definiendo una serie de capas o contexto a respetar

[52-56]. Por tanto, no es posible encontrar el equivalente a las capas ISO/OSI en las redes o la pirámide de automatización (Figura 2.1) ya presentada. Además, junto con la adaptación a nuevas soluciones, está el desarrollo aún incipiente de nuevas tecnologías, software o protocolos, que pueden modificar el balance de carga y funcionalidad en una arquitectura.

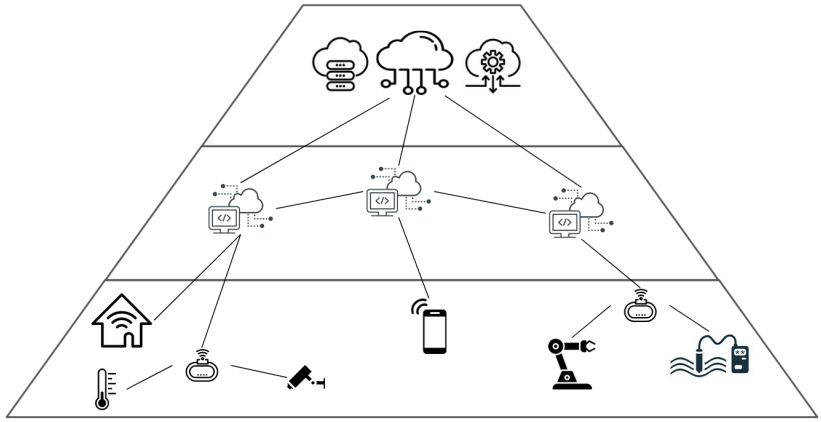


Figura 3.1. Capas de una arquitectura IIoT. La capa inferior (percepción) interactúa con una colección heterogénea de dispositivos físicos. La superior (plataforma) sirve de nexo común e interacción con terceros. La del medio (edge) proporciona servicios de diversa índole e integra las capas.

A pesar de eso, se ha tratado de hacer aproximaciones a una estructura unificada, como puede verse en [57-60], donde se intenta encontrar puntos en común entre múltiples soluciones. Según el enfoque aplicado, dicha aproximación puede llevarse a cabo, por ejemplo, a nivel físico (en cuanto a los dispositivos empleados), a nivel de funcionalidades, o a nivel de tecnologías implicadas. En todo caso siguen constituyendo definiciones muy difusas de cómo debe constituirse una infraestructura IIoT. En base a estos trabajos se propone utilizar una versión simplificada de capas a partir de las presentadas en estos análisis. La propuesta en este caso es interpretar tres capas de arquitectura tomando como referencia la funcionalidad. También se tendrán en cuenta las características habituales de los dispositivos empleados para adquirir dichas funcionalidades [61].

Por tanto, las capas que se van a utilizar, presentadas en la Figura 3.1, son:

- Percepción: primera capa de la arquitectura, que conecta con los dispositivos físicos. Sus principales retos son la variedad de equipos con los que interactúa, la comunicación y las restriccciones de espacio y batería.
- *Edge*: sin traducción consensuada en español (borde, extremo, límite, ...), es la capa de procesamiento de datos y servicios locales, cuya principal función es la de proporcionar robustez frente a las pérdidas de conexión. Sus principales retos son las tecnologías de comunicación de área ancha y la integración de algoritmos de procesamiento, entre los que cada vez tiene más protagonismo la inteligencia artificial (IA).
- Aplicación / Plataforma: capa de uso de los datos. Sus principales retos son el almacenamiento de la información, el procesamiento de alto nivel, la gestión de los equipos y sobre todo la integración con herramientas de terceros para la explotación de los datos.

A pesar de esta definición de las capas, la tecnología y los dispositivos implicados están en constante evolución. En consecuencia, una funcionalidad propia del *Edge* puede asociarse a la capa de percepción si se considera que de este modo tiene más sentido [62]. No es, por tanto, una estructura rígida o con límites cerrados. En todo caso, esta arquitectura sí permitirá exponer de forma clara el resto de las explicaciones de este documento e identificar retos, ventajas, tecnologías y los componentes de la solución.

3.2.1. Capa de percepción

La conexión de la arquitectura con los equipos físicos exige que esta capa tenga una gran capacidad de adaptación, no solo en métodos de conexión con el medio, sino también en tecnologías y protocolos de comunicación, capacidades de

procesamiento y otras características asociadas con la variedad de equipos con los que se deba interactuar [63].

Además, al ser la capa por la que entran los datos a la arquitectura, cumple una misión adicional de homogeneización de la información. De esta manera, en capas superiores puede usarse independientemente de dónde o cómo se haya obtenido. Para que esto sea posible, debe establecerse para los datos un formato común, aunque lo suficientemente versátil para servir de definición a cualquier información que se introduzca en la plataforma. Adicionalmente, el dispositivo deberá completar los datos obtenidos con una referencia de contexto (el origen del dato o la identidad del dispositivo que lo ha adquirido) de forma que pueda ser correctamente interpretado y de forma unívoca en toda la plataforma.

Por ejemplo, en una aplicación doméstica, una misma solución podría estar adquiriendo datos binarios del estado de las bombillas (encendidas/apagadas) y, al mismo tiempo, imágenes de una cámara. Por un lado, existe mucha diferencia entre un dato binario y otro de una imagen. Por otro lado, si estos datos de estado de las bombillas se trasladan a capas superiores de la arquitectura sin contextualizarlos (asignarles una habitación o zona de la casa), no sería posible distinguir los datos de una bombilla de los de otra, haciendo que la adquisición de los datos resulte inútil.

El otro reto principal al que debe atender esta capa es la comunicación. Dado que la fuente de datos puede ser muy variada, es necesario que disponga de los medios para integrarse en múltiples escenarios y dar capacidad de comunicación a algo que originalmente no lo tenía. Continuando con el símil de la aplicación doméstica, una bombilla tradicional no dispone de los medios para conectarse a una red de forma cableada porque no se contemplaba esa aplicación. Por tanto, una bombilla inteligente deberá ser capaz de sortear esa dificultad utilizando una conexión inalámbrica, a ser posible disponible en un hogar como WiFi, para disminuir el coste y la complejidad de la nueva aplicación.

Sin embargo, en otros entornos los requisitos pueden cambiar mucho y las tecnologías de comunicación inalámbricas han bebido en los últimos años de las condiciones industriales para las aplicaciones de IIoT. Algunas de estas tecnologías son ZigBee, LoRa, 6LowPAN o BLE. Estas tecnologías difieren mucho según su uso. Las hay de corto alcance y de largo alcance, unas priman el ahorro energético mientras que otras priman el tráfico de datos o la robustez de la red de comunicación, etc.

Debido a esto, la cuestión principal en este ámbito es mantener una amplia variedad de opciones tanto en hardware como en software para adecuarse bien a los casos de uso. También la disminución en el coste de la electrónica facilita la identificación de equipos que puedan ser baratos, de muy bajo consumo, portátiles (en muchas ocasiones), inalámbricos y ligeros (o al menos pequeños).

Tal como se ha planteado, el sector industrial fuerza a replantear algunas de estas definiciones más extendidas de IoT para adecuarse a sus necesidades. En este caso, lo más habitual es que en esta primera capa haya una mixtura de equipos IIoT y equipos de automatización industrial. Si bien los equipos IIoT serían los más adecuados para afrontar la abstracción de los datos, la comunicación remota y la variedad de equipos, la fiabilidad que se exige en industria y la alta inversión de algunos equipos obliga a los equipos IIoT a convivir con los equipos de nivel 1 de la pirámide de automatización: PLCs y dataloggers principalmente (ver Figura 2.1) [64, 65]. De esta manera, IIoT puede aportar sus características distintivas sin comprometer los sistemas anteriores, además de aprovechar el conocimiento experto de los fabricantes de esos otros equipos.

Un ejemplo de esta situación puede darse, por ejemplo, al integrar una carretilla elevadora con una arquitectura IIoT. Por un lado, el equipo IIoT puede usarse para monitorizar de forma directa algunas variables a las que la carretilla no esté sacando partido, tales como las condiciones ambientales en la cabina. Además puede enviar esos datos de manera inalámbrica si es necesario

por las condiciones de movilidad. Sin embargo, carece de sentido monitorizar la posición de las palas elevadoras de forma independiente, siendo preferible integrar dicha monitorización en la controladora ya existente en el equipo.

3.2.2. Capa edge

En esta capa, los datos adquiridos en la capa anterior son procesados, servidos localmente y enviados a la capa superior a través de Internet para una explotación y un almacenamiento más centralizado.

En este nivel ya no es necesario atender a una variedad tan amplia de interfaces de comunicación, centrándose en su lugar en aquellas ya elegidas para la solución tanto aguas abajo (para integrar los dispositivos de la capa de percepción) como aguas arriba (para conectarse con la plataforma o nube). Por poner un ejemplo, se plantea una planta fotovoltaica. En este caso podrían disponerse equipos de percepción para comunicarse con los contadores eléctricos, recibir información de sensores ambientales o monitorizar la producción de los paneles. Todos ellos irían provistos de una misma tecnología de comunicación, por ejemplo ZigBee. Un dispositivo edge solo necesitaría radio ZigBee para comunicarse con todos ellos, simplificando su escenario y requisitos.

Dado que el formato ya se ha homogeneizado en la capa inferior, esta capa ya ofrece escalabilidad en caso de ampliarse el catálogo de equipos de medida. Además, dicha armonización de datos permite realizar operaciones entre las variables adquiridas, al ser comparables y procesables desde un mismo software. De esta manera, un dispositivo edge puede disminuir el volumen de datos si la aplicación lo requiere. Por ejemplo, en el caso de la planta fotovoltaica, su ubicación aislada podría causar en la planta problemas de conexión escasa o deficiente en momentos puntuales. Una solución IIoT podría aprovechar la capa edge para mantener la planta gestionada localmente y enviar solo unos datos (como los referidos a la productividad de la planta o aquellos necesarios para

realizar algún informe) reducidos al nivel superior durante esa caída mediante una tecnología de comunicación alternativa. El resto de los datos los puede acumular temporalmente el dispositivo edge hasta tener una mejor conexión con la nube. Este tipo de actuaciones también serían deseables en caso de tener una batería limitada.

El dispositivo edge también puede servir los datos entre los equipos locales, sobre todo de variables generadas en el propio edge, para otros sistemas que los puedan usar. En la planta fotovoltaica podría realizarse un cálculo sencillo del rendimiento de un panel en relación a la radiación solar para saber si el panel está funcionando mal. Esta información podría utilizarse para enviar una alerta de inspección, activar un agitador del panel para limpiarlo o incluso forzar una inspección automática mediante un equipo móvil, como un dron. Si bien esta funcionalidad podría realizarse a alto nivel desde una plataforma remota, igualmente se llevaría a cabo a través del dispositivo edge, que estaría actuando de intermediario entre la red local ZigBee y la conexión a Internet con la plataforma. El control local podría ser deseable para mantener la funcionalidad en caso de desconexión, o bien para mejorar los tiempos de respuesta.

Con la mejora de la IA, esta capa puede ganar importancia para descargar procesamientos de un servidor centralizado, así como para ejecutar algoritmos crecientemente complejos, lo que es cada vez más común en trabajos que incluyen procesamientos avanzados vinculados a esta capa [66].

Como se puede ver, los retos a los que se enfrenta esta capa son, por un lado, las comunicaciones tanto locales como de larga distancia, principalmente inalámbricas; y, por otro, la infraestructura de procesamiento. Por tanto, el perfil típico de estos equipos responde al de un equipo con capacidad de conexión inalámbrica local y remota y buena capacidad de procesamiento. No se espera que este tipo de equipos sean móviles ni por batería ni por tamaño. Los equipos de la capa edge también se llaman *gateways IIoT* por su labor intermediadora

[67].

En el escenario industrial, esta capa y la de percepción tienden a difuminarse. Como se ha planteado en el apartado 3.2.1, muchas veces la sensorización de la planta ya se ha realizado con anterioridad a la adopción de la solución IIoT o requiere de equipos específicos que puedan soportar unas condiciones dadas o garantizar una calidad de medida o acción. Al integrar este escenario con la arquitectura IIoT, muchas veces se realiza una comunicación directa entre estos equipos y la capa edge. En estos casos, el protocolo y formato tendrían que ser adaptados en esta capa y no en la de percepción, pero no habría que considerar otros requisitos como la necesidad de ser compactos o de bajo precio porque se seguiría usando un solo gateway IIoT centralizado. Debido a esto, la capa edge es con frecuencia la pieza clave de una arquitectura IIoT.

3.2.3. Capa de plataforma

La última capa de la arquitectura sería la plataforma IIoT. Si bien en los comienzos las plataformas IIoT eran herramientas principalmente de recolección y visualización de datos, a día de hoy tienen un uso mucho más amplio y fundamental para cualquier arquitectura.

Esta capacidad para servir como centro de comunicaciones sigue estando vigente de cara a los dispositivos de las capas inferiores, en especial desde la capa edge. De esta manera, los datos adquiridos pueden ser almacenados o procesados, aunque no necesariamente dentro de la propia plataforma IIoT. En muchas ocasiones, este servicio es solo un intermediario temporal entre equipos IIoT y un servicio de almacenamiento o de visualización más específico.

Todas las plataformas suelen proporcionar al menos una versión básica de estas funciones: almacenamiento, visualización y procesamiento. Esto permite simplificar los despliegues de una solución en la medida en que ésta escala. Así, una aplicación que no deba procesar un gran volumen de datos

no requiere de una infraestructura específica para el almacenamiento de la información. También puede salvarse la necesidad de desplegar una herramienta de procesamiento avanzado o Big Data si solo se quieren realizar procesamientos sencillos, que puede alojar la propia plataforma IIoT. No es habitual confiar a estos servicios estas funciones a un nivel avanzado, pero la posibilidad de hacerlo a nivel básico puede resultar diferenciador. Paradójicamente, lo más común es confiar menos procesamiento a esta capa que a la de Edge, porque resulta más eficiente usar otras herramientas una vez se dispone de un servidor. Sin embargo, ya existen aproximaciones de IA generativa para las plataformas IIoT como la de Microsoft, que dispone de integración más íntima entre las herramientas, lo que puede suponer una ventaja competitiva.

Lo más habitual es que las plataformas IIoT dispongan de métodos para transmitir los datos recibidos desde los dispositivos IIoT con otros sistemas. En estos casos suelen proporcionarse conectores específicos para las herramientas más comunes. Por ejemplo, que disponga de un interfaz JDBC para enlazar con bases de datos, o conectores MQTT, REST o WebSocket para integrar otros servicios o verticales como Amazon Web Services (AWS) o Azure IIoT de Microsoft.

La otra característica crítica de una plataforma IIoT es la gestión de los equipos IIoT. Aunque este requisito se está afrontando en algunos casos con desarrollos específicas para cada caso de uso, estas soluciones tienden a convertirse en verticales poco reusables. El enfoque más común, en cambio, es plantear un protocolo de gestión agnóstico a una aplicación concreta, ya que las necesidades de los equipos IIoT suelen ser similares en todos los contextos. Las funciones más típicas son: la instalación o actualización de software, la configuración de parámetros de control, el control de los flujos de trabajo (arranque, parada y modificación), el aprovisionamiento (adquisición de credenciales y alta en la plataforma), la comunicación con equipos locales y la ejecución de comandos remotos (para permitir adaptaciones o extender la funcionalidad del interfaz de

gestión).

Lo habitual en esta capa es que no exista una amplia variedad de equipos y sistemas que compartan espacio en la misma arquitectura IIoT, apostándose en su lugar por una sola plataforma y se despliegue en un clúster de servidores desde el que pueda proporcionar todo el rendimiento exigido en condiciones de alta disponibilidad (sin caídas).

3.3. Tecnologías IIoT

Existe un catálogo inmenso de protocolos, software, estándares y dispositivos que se pueden usar en soluciones IIoT. De hecho, parte de esta pila de tecnologías no son específicas de IIoT sino heredadas de otros ámbitos, si bien adaptadas para IIoT. Este apartado recoge algunas de las más empleadas en el sector industrial en la actualidad

Entre estas tecnologías se encuentran estándares de comunicación inalámbrica, protocolos de comunicación ligeros y software específico para IIoT. Se destaca que, aunque los estándares de comunicación inalámbrica y los protocolos de comunicación a veces tienen solape, se distinguen en la medida en que los primeros pertenecen a la capa física de la comunicación, mientras que los segundos se relacionan con el formato del mensaje transmitido sobre esa misma capa.

Los estándares de comunicación cableados también tienen algunas peculiaridades para su adopción en IIoT, pero no han sufrido cambios tan grandes como los inalámbricos. En muchos casos se siguen usando tecnologías mucho más extendidas como comunicación serie, I2C, SPI, Ethernet, etc.

Aunque son relevantes para una solución IIoT, se descarta la realización de un listado de dispositivos. Una de las grandes ventajas de IIoT son la variedad de enfoques que pueden adoptarse y el hardware disponible para ello. Es por

esto que acotar o enumerar de forma exhaustiva la totalidad de fabricantes y productos se revela inviable. Baste decir que para cualquier tecnología o necesidad asociadas a las capas edge y de percepción existe algún equipo que puede proveerlo a según qué coste y calidad. Como punto de referencia se pueden citar tres ejemplos de equipos básicos que pueden emplearse, por ser los más comunes en el sector educativo: Raspberry Pi, Arduino o ESP32. La facilidad de uso, bajo coste y amplia comunidad hace estos equipos muy útiles para la construcción de prototipos y elaboración de pruebas de concepto. De todos modos,estos equipos sufren la competencia de otras alternativas con características similares pero dotadas de ventajas específicas (tales como bajo consumo o procesamiento de imágenes), y por tanto no pueden proponerse como referencias para una aplicación que ya tenga requisitos específicos.

3.3.1. Comunicaciones inalámbricas

El crecimiento del concepto de IIoT ha ido acompañado de las tecnologías de comunicación inalámbricas emergentes. En los últimos años, además, han ido ampliando su funcionalidad de acuerdo con las nuevas necesidades de IIoT.

En la Figura 3.2 se presentan las tecnologías inalámbricas más importantes distribuidas según su rango de alcance y su velocidad de transmisión. Como se muestra, en este momento es posible cubrir infinidad de casos de uso ya que hay tecnologías presentes en prácticamente todo el espectro. A grandes rasgos se pueden distinguir tres categorías: corto alcance, celulares y LPWAN (Low Power Wide Area Network). Las comunicaciones de corto alcance están concebidas para ser empleadas en recintos acotados tales como hogares o plantas industriales. En una arquitectura IIoT se utilizan sobre todo en la capa de percepción o en la capa edge.

Para distancias largas se acudiría a los otros dos grupos. Entre ellos, las celulares se distinguen por su buena velocidad de transmisión y por la prioridad

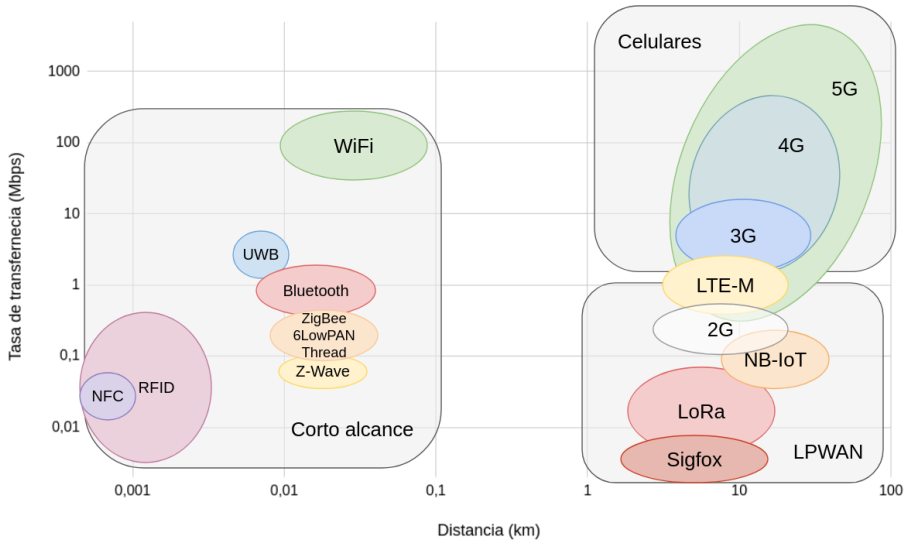


Figura 3.2. Comparativa de tecnologías inalámbricas. Se pueden clasificar según su uso y capacidad en: corto alcance, celulares y LPWAN.

que adquiere su valor para el usuario, serían las más habituales para conectar la capa edge con la plataforma. Las tecnologías LPWAN, como su propio nombre indica, están pensadas para ajustar el consumo y sirven habitualmente como medio de conexión directa de la capa de percepción con la de plataforma, ya que permite el acceso a Internet a equipos de bajas prestaciones.

En las Tablas 3.1, 3.2 y 3.3 se resumen las características técnicas de todas las tecnologías citadas en este epígrafe con la finalidad de aligerar las explicaciones sucesivas, omitiendo una enumeración metódica de rango, anchos de banda, dispositivos admitidos, etc. Éstas se obtuvieron a partir de [68-72]

Corto alcance

Las dos tecnologías más conocidas de comunicación de corto alcance son WiFi y Bluetooth (incluido Bluetooth Low Energy, BLE). La mayoría de las soluciones IIoT incluyen o están basada en alguna de estas dos, debido a que ya están muy extendidas en el mercado tanto doméstico, como industrial.

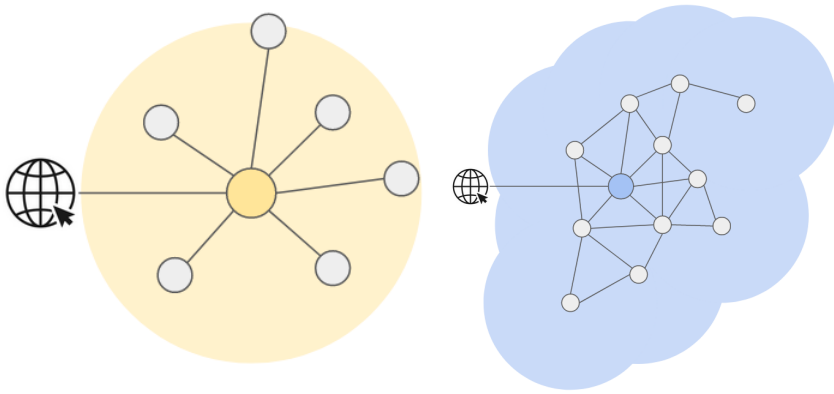
Sin embargo, algunas soluciones presentan retos que no pueden afrontarse debidamente con estas tecnologías. Algunos de estos retos pueden venir por la cantidad de dispositivos conectados, el alcance de la red o el consumo eléctrico.

Para abordar a estos problemas existe una solución a nivel de redes (no de una tecnología concreta) que consiste en estructurar la red de comunicaciones en malla, en contraposición con su forma tradicional en estrella. Una red en estrella se caracteriza por tener un punto donde se concentran las comunicaciones. Cualquier dispositivo que quiera comunicarse con otro debe conectar con este nodo central y desde ahí se dirigirá el mensaje al destino final. De esta manera, como se ve en la Figura 3.3a la red está siempre limitada en alcance por este dispositivo central y es vulnerable a un fallo interno. Por el contrario, en una red de malla, un dispositivo transmite su mensaje a todos los nodos cercanos, indicando el destino deseado, y estos nodos intermedios, a su vez, lo retransmiten a otros. Este tipo de topología posibilita que cada nodo extienda el alcance de la red tal como se presenta en la Figura 3.3b. También ofrece una mayor robustez ante la caída de un nodo, porque la ruta de comunicación puede establecerse a través de otros nodos.

A pesar de todo, en ambos casos es necesario disponer de un *gateway* para el acceso a otras redes (por ejemplo para llegar a Internet). No obstante, en el caso de la red de malla, se obtiene redundancia a través de otros nodos. Otra ventaja adicional de las redes de malla es que permiten una mayor cantidad de dispositivos conectados a la red simultáneamente porque ya no dependen de la capacidad de procesamiento de un nodo central. Por consiguiente, el único límite es el que establezca el protocolo al definir las direcciones de los equipos.

Las redes de malla también poseen desventajas. Una de ellas es que exige mayor capacidad en los nodos para procesar los saltos y garantizar robustez y eficiencia. Dos ejemplos comunes de estos procesamientos adicionales son el establecimiento de caminos óptimos (para reducir los reenvíos circulando en la

red) o la selección colectiva de un nuevo gateway si se pierde el actual.



(a) Topología de estrella y alcance. **(b)** Topología de malla y alcance.

Figura 3.3. Comparación entre topologías de estrella (3.3a) y malla (3.3b). El alcance de la red de malla llega tan lejos como su nodo más lejano, mientras que en la de estrella solo puede usarse la cobertura del nodo central.

Aunque tanto Bluetooth como WiFi están haciendo avances para ofrecer la característica de malla bajo su estándar normal, existen protocolos diseñados de un modo específicamente acorde con esta topología. Algunos de los más conocidos son ZigBee, Z-Wave o 6LowPAN.

ZigBee es una tecnología de comunicación inalámbrica que se apoya en el concepto de malla para ofrecer un alcance de la red amplio manteniendo al mismo tiempo un consumo de batería muy bajo y una velocidad de transferencia muy buena. Cada nodo solo tiene unos 10 - 20 metros de alcance en un escenario real pero, en conjunto, la red puede cubrir área muy extensas ya que el límite de la red es de 65535 dispositivos. Si bien ZigBee inicialmente solo definía la capa de comunicación más baja, dejando al usuario la selección de un protocolo sobre ella, en las últimas especificaciones (desde la versión 3.0), ya plantea un protocolo y mensajes de mayor nivel que deben respetar los equipos que implementen ZigBee, manteniendo en parte la libertad original pero favoreciendo la

interoperabilidad.

Z-Wave es similar a ZigBee en la mayoría de sus características. La principal diferencia es que se trata de una tecnología propietaria y para soportar la comunicación con ella es necesario certificar el equipo. Esto ofrece ventajas a nivel de compatibilidad respecto a ZigBee. Además, Z-Wave tiene un alcance mayor nodo a nodo que le da un poco de ventaja, aunque solo para casos de uso relativamente pequeños, ya que tiene un límite de direccionamiento de solo 232 dispositivos.

6LowPAN es otra tecnología que se apoya sobre el mismo espectro de comunicaciones, pero trata de aprovechar toda la tecnología de enrutado ya existente en la comunicación tradicional por TCP/IP. De esta manera, utiliza el estándar IPv6 para el direccionamiento de los equipos y para el envío de los mensajes. Gracias a esto, 6LowPAN es compatible con todos los protocolos típicos de las redes típicas: HTTP/S, MQTT/S, FTP/S o SSH, entre otros. La particularidad del protocolo se da solo al traducir el mensaje a la capa física, de manera similar a como un router WiFi hace de intermediario entre una comunicación cableada y una inalámbrica. El diseño tanto de la red como de los dispositivos es, por lo tanto, mucho más simple que en las otras dos tecnologías, cuyos protocolos de comunicación son más específicos. En comparación con la comunicación por WiFi, el alcance punto a punto de esta red es mucho menor, ya que se limita a 30 metros. Sin embargo, el consumo es muy inferior y es una red que permite el mallado de forma nativa.

Thread es un protocolo propietario de Google que se sustenta sobre una red 6LowPAN y aportar características que a día de hoy no soporta el estándar de 6LowPAN. Estas son principalmente seguridad, fiabilidad y facilidad de integración con otras redes. Como ejemplo, incorpora un procedimiento

para reelegir el nodo que se comunica con el exterior de la red en caso de fallo, cubriendo una posible vulnerabilidad de la red mallada. Además, para simplificar la integración, Google ha desarrollado una implementación de código abierto en múltiples lenguajes de programación llamada OpenThread, que ayuda a desarrollar soluciones desde cualquier tipo de dispositivo que posea la tecnología. Debido a esta sencillez y que las características que añade son muy prácticas en los casos de uso típicos, a día de hoy es más frecuente encontrar redes basadas en Thread en lugar de 6LowPAN puro.

Bluetooth Mesh es una extensión del estándar BLE con el cual es posible crear una red mallada de dispositivos Bluetooth. Está pensada para la construcción de redes de sensores y tiene una capacidad teórica de 65535 equipos conectados a la vez. Dado que su funcionalidad de malla está basada en un cambio de especificación de firmware, este protocolo es compatible con todos los dispositivos que tengan Bluetooth 4 o superior, lo que le confiere una sustancial ventaja a la hora de apoyarse en equipos comerciales.

Además de las presentadas, existen otras tecnologías de comunicación inalámbrica local que son comunes encontrar en pruebas de concepto de IIoT, tales como Ultrawide Band (UWB), Radio Frequency Identification (RFID) o Near Field Communication (NFC).

UWB surge como una tecnología de gran fuerza en casos de uso con muchos obstáculos, ya que su principio de funcionamiento (que consiste en ocupar una banda muy ancha de frecuencias) le permite ser robusta ante esas situaciones. Es por eso que una de las principales aplicaciones de esta tecnología se esté dando en la localización en interiores. Al igual que otras, UWB también mantiene en sus conexiones un consumo relativamente bajo aunque su alcance también es bastante corto, siendo poco fiable a partir de los 30 metros. En todo caso es una

Tecnología	Ancho de banda	Alcance	Nodos	Topología	Consumo
WiFi	450 Mbps	10 - 100 m	250	Estrella	Alto
WiFi Mesh	450 Mbps	10 - 100 m	-	Malla	Alto
Bluetooth	2 Mbps	15 - 20 m	7	Estrella	Bajo
BLEMesh	1 Mbps	15 - 20 m	32000	Malla	Bajo
ZigBee	250 kbps	10 - 100 m	65535	Malla	Bajo
Z-Wave	100 kbps	30 - 50 m	232	Malla	Bajo
6LowPAN	250 kbps	10 - 100 m	250	Ambas	Bajo
Thread	100 kbps	10 - 100 m	250	Ambas	Bajo

Tabla 3.1: Resumen de tecnologías inalámbricas de corto alcance. Los datos presentados sirven como referencia comparativa aproximada, puesto que la capacidad real depende mucho del hardware empleado.

tecnología que todavía tiene un uso muy marginal a pesar de llevar varios años en el mercado.

Las tecnologías RFID (ente las que se englobaría NFC también), son muy versátiles y llevan aplicándose en el mercado desde mucho antes de la aparición del concepto de IIoT. Si bien no permiten la construcción de una red de comunicaciones como tal, su conexión punto a punto tiene una gran ventaja sobre todas las tecnologías citadas hasta ahora que es la posibilidad de tener nodos *pasivos*. Estos nodos no requieren ningún tipo de batería y su única alimentación sería la inducida en su antena por la onda electromagnética del nodo que le consulta la información. De esta manera, las soluciones basadas en esta tecnología pueden llegar a resultar muy económicas. NFC es un subgrupo de RFID que permite transmisión a muy poca distancia (habitualmente un par de centímetros o incluso contacto) y que está muy extendida para el uso de tarjetas de uso personal. Tanto en RFID como en NFC, los protocolos de comunicación suelen ser propietarios y muy específicos. Sin embargo, su uso se da con frecuencia en soluciones IIoT porque ofrecen una forma de identificación

y seguimiento muy eficaz y barata.

Celulares

En lo relativo a las tecnologías inalámbricas de largo alcance, la más utilizada es con mucha diferencia el 4G o LTE (Long-Term Evolution), aunque podría ser rápidamente reemplazada por el 5G [73]. El 4G es una comunicación de telefonía móvil celular comúnmente conocida gracias al teléfono inteligente. Esta tecnología permite establecer llamadas telefónicas y acceso a Internet a un tiempo.

Tecnología	Ancho de banda	Alcance	Consumo	Cobertura
2G	40 kbps	-	Medio	Mundial
3G	15 Mbps	-	Alto	Mundial
4G	300 Mbps	15 km	Medio-alto	Mundial
5G	30 Gbps	0.5 km	Adaptable	Europa, Asia, EEUU

Tabla 3.2: Resumen de tecnologías celulares. Los datos presentados sirven como referencia comparativa aproximada, la capacidad real depende mucho del hardware empleado.

Como antecedente al 4G, existieron las tecnologías 2G y 3G. La más conocida de estas es sin duda la de 3G, que fue la que posibilitó el uso masivo de Internet desde cualquier parte. Se desplegó por primera vez en 1998 con la intención de reforzar las características del 2G pero su aparición resultó crítica sobre todo para el acceso a Internet en el que se daban mejores velocidades. Esto catapultó el número de soluciones basadas en comunicación a Internet y dio pie a la invención del teléfono inteligente y la aparición de redes sociales. La red predecesora (2G) no ofrecía una conexión a Internet lo bastante potente para que un usuario común la empleara (cuya principal navegación es web y resulta muy pesada), pero sí permitió la comunicación remota para telemedida en el sector industrial. De esta manera, la llegada del 3G no supuso la misma revolución para la automatización

y medida remota, pero sí amplió las posibilidades de intervención y la fiabilidad de la conexión.

Es la llegada del 4G la que ha generado nuevas opciones en la industria pues, además de ofrecer prestaciones mucho mejores que el 3G, permite una mejor selección de las características de la comunicación para desarrollar soluciones muy bien ajustadas. Es lo que se denominan las *categorías*. En 4G es posible establecer comunicación con la red móvil seleccionando una categoría a medida, lo que limita la velocidad de conexión pero permite disminuir el consumo. Además, las categorías más habituales permiten un volumen de datos bastante grande, lo que significa que el coste de enviar grandes bloques de información es muy bajo para los sectores que manejan gran cantidad de datos de forma remota.

El 5G, la siguiente generación de tecnología móvil, permite ampliar la capacidad de las redes móviles tanto en velocidad de transmisión como en volumen de dispositivos [74]. También dispone de categorías de comunicación para ajustar consumo y transferencia de datos, incluso con mayor rango que el 4G, por lo que sirve a equipos de muy bajo consumo y a los que requieran mucha velocidad al mismo tiempo. Si bien no dispone aún de cobertura plena, se espera obtener plena capacidad pronto ya que muchas compañías de telecomunicaciones están apostando a este avance. Hay que hacer notar además que el despliegue de 5G es compatible con el de 4G en hardware con solo una actualización de software. Si bien no tiene la misma eficiencia, esto facilita la puesta en marcha de la red en aras de una mejora a posteriori por reemplazo del hardware.

A día de hoy, la tecnología 3G ha sido completamente sustituida por el 4G. Aunque no va a permanecer mucho más tiempo, el 2G sí conserva mayor cuota de uso porque permite un consumo más ajustado que el 4G todavía. En cualquier caso, en muchos países se está ya realizando un *apagado* de las redes 2G y 3G (algunos con mayor permanencia del 2G por lo ya explicado). A la espera de un pleno despliegue del 5G, el 4G es a día de hoy la única tecnología con cobertura

prácticamente global.

Sin embargo, las redes celulares, tienen algunos inconvenientes para la creación de arquitecturas IIoT como son la asimetría de tráfico, el consumo o el coste por conexión.

Por asimetría de tráfico se entiende la diferencia entre el ancho de banda y cantidad total de tráfico de subida desde el dispositivo a la red y la del tráfico de bajada de la red al dispositivo. Debido a su uso doméstico, las redes celulares se han centrado principalmente en ofrecer unas buenas prestaciones de tráfico de bajada a costa del de subida. Esto significa que es mucho más fácil descargar datos que subirlos. Dado que la mayoría de los usuarios utilizan estas redes para consultar servicios web y solo ocasionalmente para subir su propia información, el dimensionamiento es ideal para su uso actual. Incluso el uso de redes sociales y la facilidad de publicar no logran igualar ambos tráficos. Sin embargo, en IIoT, el tráfico es habitualmente al revés: lo más común es que el dispositivo IIoT suba constantemente información sin que necesite apenas descargar datos o que se le envíe nada. Esto hace que el uso de una red celular resulte poco óptimo, aun cuando pueda ser igualmente válido.

En lo referente al consumo, a pesar de poder ajustarlo, todavía hay soluciones IIoT donde la restricción en uso de batería lleva a exigir unas características mejor. La categoría más sencilla de 4G es excesiva para algunos casos de uso de IIoT. Con 5G se espera solucionar esta cuestión al ampliar las opciones a menores consumos. No obstante, también impactan en el consumo otros elementos de la comunicación como el procedimiento de conexión/desconexión o las cabeceras de los mensajes.

En cuanto al coste por conexión, las redes celulares se establecen en bandas que están reguladas por los estados. De esta manera, es necesario adquirir una licencia de explotación para dar cobertura en la misma. A efectos prácticos, esto significa que para cada dispositivo que se quiere conectar a una de estas redes,

debe abonarse dinero a un operador de la red que haya adquirido la licencia, de manera que el coste por dispositivo sufre siempre un sobrecoste en mantenimiento. Como IIoT se centra en la conexión de múltiples objetos a Internet, muchas veces de forma independiente, este coste adicional por conexión puede resultar excesivo.

También hay que considerar que una red móvil como 4G depende del uso de unas frecuencias de acceso privado que los gobiernos distribuyen a subasta entre entre lo que se denomina Proveedores de Servicios de Internet (PSI). Esto por un lado proporciona la ventaja de que la compañía adjudicataria amplía y promueve la cobertura de las bandas de frecuencia que ha adquirido y tratará en la medida de lo posible de optimizar costes a nivel global. Por otro, representa un coste externo y sujeción a condiciones no siempre deseadas por parte de los usuarios finales.

Con la llegada del 5G están surgiendo también redes privadas. Si bien esto ya era posible con las redes 4G, el beneficio potencial es mucho mayor con el 5G, ya que una red privada puede aprovechar toda la velocidad de esta tecnología sin los retardos de navegar por Internet. Esta forma de despliegue puede mejorar la escalabilidad de una solución al no ser necesario depender del PSI para cada equipo, sino solo para la creación de la red (la banda de frecuencias sigue estando atribuida al PSI).

Sin embargo, entre las tecnologías de largo alcance IIoT existen alternativas como redes privadas, en las que solo puede lograrse la comunicación vinculándose a un vertical, o completamente abiertas, en las que un usuario puede elegir desplegar su propia red, de manera similar a una red WiFi de largo alcance. Estas tecnologías están mucho mejor adaptadas a casos de uso de bajo consumo.

LPWAN

Las redes LPWAN son aquellas que ofrecen una conexión de largo alcance con un consumo mucho menor que la conexión celular [75-77]. Estas tecnologías

deben su nacimiento a las propias aplicaciones de IoT e IIoT, ya que surgen como medio para solucionar la conexión universal de equipos y alcanzar aplicaciones de bajo consumo.

En el caso de las LPWAN, algunas de las tecnologías se apoyan en frecuencias de uso libre. Esto significa que no es necesario contratar su uso como en el caso de las empleadas para comunicación celular. A cambio, existen algunas restricciones relacionadas con el tiempo de uso o la potencia de emisión. Aprovechando estas características es posible establecer medios de comunicación más versátiles para algunas soluciones IIoT y a cambio mejorar la escalabilidad al eliminar el coste por conexión.

Tecnología	**Ancho de banda**	**Alcance**	**Consumo**	**Tipo banda**
SigFox	600 bps	10 - 40 km	Muy bajo	Libre (operador)
NB-IoT	200 kbps	1 - 10 km	Bajo	Atribuida
LTE-M	1 Mbps	1 - 10 km	Bajo	Asignada
LoRaWAN	50 kbps	5 - 20 km	Muy bajo	Libre

Tabla 3.3: Resumen de tecnologías LPWAN. Los datos presentados sirven como referencia comparativa aproximada, la capacidad real depende mucho del hardware empleado.

Las tecnologías LPWAN más relevantes a día de hoy son Sigfox, LTE-M, NB-IoT, LoRa y LoRaWAN. Se puede encontrar una comparativa aproximada en la Tabla 3.3.

Sigfox [78] fue una de las redes IIoT pioneras en proveer a equipos de comunicación de largo alcance con consumos mínimos posibilitando la creación de soluciones que con una batería pequeña pudieran durar años enviando datos remotamente. Se trata de una red propietaria que nació en Francia y se ha desplegado ya por la mayor parte de Europa occidental y central. Está también presente en muchos otros países, aunque solo

como despliegue parcial en algunas regiones. En esta red la transferencia de datos admisible es extremadamente baja y asimétrica, en este caso en beneficio de la subida de datos. En un día apenas se puede subir unos pocos mensajes de pocos bytes y la descarga es aún menor.

Aunque se apoya sobre una banda libre, se trata de una red propietaria por lo que es necesario contactar con Sigfox para usarla. Solo ellos (o una compañía colaboradora) pueden proporcionar la conexión. A cambio es posible obtener soporte en los despliegues y, según la zona, solicitar una ampliación de cobertura Sigfox para acomodarla a la solución que se esté desarrollando. Debido a esta centralización en su uso, Sigfox presenta bastantes inconvenientes en la escalabilidad de una solución si el entorno de trabajo no es el marcado por su cobertura.

LTE-M (Long Term Evolution for Machines) es una tecnología basada en la infraestructura 3G/4G que permite una velocidad y volumen de transmisión aceptable manteniendo un consumo relativamente bajo. Si bien no permite duraciones de batería tan largas como Sigfox, su aplicación es muy útil para soluciones de movilidad y de telefonía, pues admite tráfico de datos y voz, porque su cobertura es mucho mayor (aunque todavía no está presente en Asia y Europa del este) y porque el ancho de banda admite muchas más soluciones que el de Sigfox anteriormente presentado. Aunque no es un protocolo propietario, LTE-M depende de PSIs para el uso de las frecuencias.

NB-IoT (Narrow Band IoT) [79] es también una tecnología basada en la infraestructura celular tradicional de 3G/4G. Está pensada de manera similar a LTE-M para proporcionar comunicación de bajo consumo con una velocidad aceptable. NB-IoT proporciona una penetración algo mayor que LTE-M aunque a día de hoy no es posible la transmisión de voz por

esta tecnología. También se estima que NB-IoT tiene un coste por terminal algo menos aunque al depender de proveedores de Internet, está sujeto a condiciones de mercado. El despliegue actual está en Europa, Asia, América y Oceanía, pero se estima que acabará siendo mundial. Gracias a su mayor penetración, NB-IoT es muy interesante para aplicaciones en interiores o en regiones remotas a pesar de tener unas velocidades algo menores que el LTE-M.

LoRa (Long Range) [80] es un protocolo propietario de comunicación que se apoya en la banda de frecuencia libre (para propósitos industriales, educativos y sanitarios) y que permite la comunicación a unos pocos kilómetros de distancia. Al estar en una banda de frecuencia libre, no se accede a través de un PSI. Ni siquiera es necesario contratar una plataforma, como ocurre en Sigfox. Es decir, cualquier usuario puede crear su propia red utilizando una radio LoRa. La desventaja del uso de esa banda de frecuencia es que se limita su acceso para una misma aplicación a solo el 1 % de la red, por lo que no es posible realizar conexiones masivas en la misma.

Por eso precisamente las redes LoRa optimizan mucho el uso de datos, y están dimensionadas para la subida de datos desde los dispositivos a una plataforma. La bajada de datos se plantea de manera ocasional y siempre que el dispositivos lo admita y tras una conexión del mismo para la subida de datos. Por tanto, la bajada de datos hacia un dispositivo no es determinista, sino que está siempre supeditada a que el dispositivo sea quien inicie la conexión.

LoRaWAN es una especificación para que en una red LoRa se proporcione acceso a Internet. Esta conexión se hace a través de una red privada que proporciona los gateways para que los dispositivos LoRaWAN se conecten a

través de ellos a su plataforma online. El uso de esas redes sí es propietario aunque no necesariamente de pago y puede ser desplegado en una red privada. Una desventaja de LoRaWAN es que su cobertura depende de la instalación de gateways en muchos puntos y con buena cobertura. En este contexto hay empresas como *The Things Network* que han optado por un modelo abierto que incentiva el uso de la plataforma y otras como *Helium* que han optado por un modelo de recompensa por criptomoneda al dueño del gateway y proveedor de la conexión.

Destacar que, a pesar de estar listado entre las redes inalámbricas de largo alcance, LoRa es una tecnología que está teniendo gran empuje también en redes de más corto alcance para, por ejemplo, soluciones de ciudades inteligentes [81, 82] e incluso para comunicación en interiores [83]. De hecho, se han planteado ya casos de uso para localización en interiores basados en una red LoRa [84, 85].

Como se puede apreciar, el catálogo de tecnologías inalámbricas de conexión en general es muy variado y ofrece muchas posibilidades de adaptar las soluciones. Aunque en un inicio se planteaba que alguna de estas tecnologías acabara por mostrar un potencial de mercado mayor y destacar en su uso, a día de hoy el futuro más probable es que varias de ellas lleguen a convivir durante bastante tiempo ya que sus diferencias son bastante significativas.

3.3.2. Protocolos

De la misma manera que han aparecido tecnologías de comunicación específicas para las necesidades de IIoT, también se han desarrollado protocolos para la comunicación e integración en estas arquitecturas [86-88]. Por un lado, se ha trabajado en protocolos que permiten la transferencia de información de manera que resulte ligera en consumo y procesamiento para adaptarse a medios de comunicación restrictivos y a dispositivos de bajo rendimiento. Por otro lado, existen protocolos a mayor nivel que facilitan la gestión de los activos de una

red IIoT. Estos últimos pueden o no apoyarse sobre los anteriores.

Comunicación

Estos protocolos están pensados para resultar ligeros en transferencia de datos, de manera que se ahorre mucho la batería con su selección frente a otros protocolos más sobrecargados. Además, también están planteados para que la desconexión de dispositivos no suponga un problema, porque es muy común en IIoT encontrarse con equipos que se comunican de forma esporádica, que tienen muy mala conexión o que no mantienen la sesión abierta porque al momento de enviar vuelven a un modo de bajo consumo.

Dentro de los protocolos de comunicación IIoT, el más destacado es MQTT, aunque existen alternativas como AMQP o CoAP, e incluso en muchos casos se utiliza REST sobre HTTP.

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y eficiente diseñado para la comunicación entre dispositivos en redes con ancho de banda limitado. Está orientado a la integración mediante publicación/suscripción a través de un broker IIoT.

MQTT-SN (MQTT for Sensor Networks) es una variante del protocolo MQTT diseñada específicamente para redes de sensores y dispositivos con recursos limitados. Sigue el modelo de publicación/suscripción pero resulta más ligero que el MQTT, por lo que se adapta bien a escenarios donde la implementación completa de MQTT puede ser demasiado pesada, como en redes de sensores IIoT. Destacar que aunque el estándar es compatible con MQTT, no todas las implementaciones lo son.

AMQP (Advanced Message Queuing Protocol) es un protocolo que permite la comunicación eficiente y confiable entre aplicaciones y sistemas distribuidos

de manera similar al MQTT. La principal diferencia es que es más robusto a costa de ser menos ligero.

CoAP (Constrained Application Protocol) es un protocolo ligero orientado a la ejecución de procedimientos RESTful (GET, POST, PUT, DELETE) remotos. Esto lo hace idóneo para la gestión remota aunque admite también la transmisión de datos bajo estos procedimientos remotos.

REST (Representational State Transfer) al igual que CoAP, es un protocolo para la ejecución de procedimientos RESTful remotos. Las principales diferencias son que por un lado no es tan ligero como CoAP, pero sí mucho más extendido.

WebSocket es un protocolo de comunicación bidireccional, muy ligero y con implementaciones en muchas plataformas. Aunque su uso está más extendido en el ámbito de chats o juegos online, está cada vez más presente en IIoT.

También cabe destacar que algunos protocolos tradicionalmente industriales como el OPC se están reinventando ara facilitar la interoperabilidad con sistemas e incluso SCADA ya existentes a través de nuevos estándares (en este caso el OPC UA, OPC Unified Architecture).

Gestión

La gestión de un equipo no implica solamente poder ejecutar acciones remotas, también mantener trazabilidad sobre su estado, conocimiento sobre los recursos que tiene disponibles, diagnóstico ante incidencias o adaptar su configuración con el tiempo.

Aunque lo ideal de estos estándares o protocolos es que no estén vinculados a una tecnología o protocolo de comunicación concreto, muchos de ellos surgen

de casos de uso reales y su especificación está muy vinculada a este caso de uso inicial. De todos modos, muchos están evolucionando a día de hoy para superar esa limitación. Algunos ejemplos de protocolos de gestión que se emplean actualmente en IIoT son:

Web of Things es un estándar de definición de un equipo IIoT que permite integrar su recursos y funcionalidades con otros sistemas. Utiliza el formato JSON-LD como base para las definiciones, pero puede usarse para múltiples protocolos de comunicaciones como MQTT, CoAP o HTTP.

LWM2M (Lightweight M2M) es un protocolo de gestión de dispositivos desarrollador por el OMA (Open Mobile Alliance) para utilizarse con dispositivo de recursos muy limitados, con comunicación CoAP. Permite la configuración, monitoreo y actualización de firmware de equipos.

Sparkplug es un protocolo que se apoya estrictamente sobre MQTT y que proporciona información sobre el estado y ciclo de vida de un equipo. Al ser muy abstracto y tener un uso muy concreto, puede usarse en combinación con cualquiera de los demás protocolos.

UPnP (Universal Plug and Play) es un conjunto de protocolos y estándares que permiten a los dispositivos electrónicos descubrir y comunicarse entre sí en una red local de manera automática y sin configuración manual. UPnP es utilizado comúnmente en entornos domésticos para habilitar la conectividad y comunicación entre dispositivos como impresoras, cámaras, routers, televisores y otros electrodomésticos.

ZigBee 3.0 es un estándar definido dentro de la última especificación de ZigBee y que comprende no solo la comunicación bajo este protocolo sino también herramientas para gestión, abstracción y hasta catalogación de equipos.

En IIoT la mayoría de las comunicaciones están basadas en el protocolo IP (TCP o UDP) que es el usado en Internet en general. No obstante, como se puede ver, no todos los protocolos están asociados a esta base de comunicación. En una aplicación con muchos actores, resulta muy complejo hacer que un solo protocolo de gestión pueda abarcar todas las necesidades. En todo caso, los protoclos de gestión es una rama en constante crecimiento y aparecen aún nuevos protocolos o se redefinen los ya existentes.

Formato de datos

En el vasto ecosistema del IIoT, el formato de los datos desempeña un papel fundamental al definir la estructura y la representación de la información intercambiada entre dispositivos y plataformas IIoT.

Un formato de datos para IIoT debe ser principalmente ligero y sencillo de interpretar, pero también versátil para muchas circunstancias y fiable. Algunos de los más extendidos son:

JSON (JavaScript Object Notation) es con diferencia el formato más extendido en las redes a día de hoy y se basa simplemente en la definición de objetos y listas anidados. Al tener un formato ligero y poco específico, permite definir cualquier información y resulta sencillo de codificar y decodificar.

JSON-LD (JSON Linked Data) es una ampliación sobre JSON que permite aportar semántica al mismo, más flexibilidad en las definiciones y anotaciones sobre el JSON. Aunque no está tan extendido como el JSON sipmle, se trata de una expansión que podría hacerse necesaria con el tiempo.

Protocol Buffers (o Protobuf) es un formato desarrollado por Google para la codificación binaria de cualquier tipo de datos. El mensaje resultante resulta

muy compacto, eficiente y robusto, aunque requiere acordar previamente el formato a emplear para permitir la decodificación.

Avro es un formato para la codificación de datos que admite tanto codificación binaria como en texto, de manera que puede usarse para obtener salidas legibles o más eficientes para un ordenador, según el caso de uso. Se apoya sobre JSON para el codificado legible con la salvedad de que se debe acordar antes de la comunicación el formato de datos a intercambiar. Resulta especialmente eficiente en transferencias de datos muy extensas.

XML (eXtensible Markup Language) es un formato versátil para definir objetos mediante etiquetas y atributos. Aunque se usó de forma extensa durante muchos años, a día de hoy está cada vez más en desuso en favor del JSON, que proporciona la misma información, o el JSON-LD, que incluso cubre toda la funcionalidad de XML.

3.3.3. Software

Probablemente el aspecto de IIoT donde más evolución constante haya sea en las implementaciones con las que se despliega una solución. Aunque ya se ha comentado que recopilar el hardware disponible sería extenso y poco práctico por obsolescencia continua, en el caso del software sí hay mayor persistencia en el tiempo de las herramientas.

Destacar que el software disponible depende, como es natural, de la capa en la que se esté trabajando. En la capa de percepción se busca principalmente software para sistemas embebidos que permita un control muy preciso aunque no resulte tan versátil. En la capa edge, el tipo de middleware a emplear deberá proporcionar mucha versatilidad, capacidad de procesamiento y abstracción. En la capa de plataforma se buscan entre otras cosas usabilidad de cara al usuario o integración tanto aguas arriba como aguas abajo.

Antes de presentar las distintas alternativas de software, es necesario señalar que muchos de estos proyectos surgen como iniciativas *open source*. El código abierto u *open source* es un tipo de licencia de software que permite que el código fuente de un programa sea accesible y modificable por cualquier persona, por lo que los usuarios y desarrolladores tienen la libertad de estudiarlo, modificarlo y distribuirlo según sus necesidades (aunque con algunas matizaciones según la licencia exacta sobre la que se apoye el desarrollo de código abierto en particular).

Este tipo de proyectos se da cada vez con más frecuencia incluso en software comercial porque tiene algunas ventajas que lo hacen muy competitivo. Entre estas se puede destacar el aprovechamiento del conocimiento, sugerencias y trabajo de usuarios, al tiempo que lo hace más accesible al ser posible utilizarlo de forma gratuita. Además, la transparencia del código hace que sea constantemente auditado y, por consecuencia, suele ser más seguro que las alternativas. La desventaja principal para el usuario es la falta de garantía si se usa gratuitamente.

Un buen ejemplo de un proyecto de código abierto exitoso es Android de Google, que a pesar de tener ya mucha competencia cuando salió al mercado, se posicionó con firmeza como el sistema operativo más usado en dispositivos móviles, con un amplio margen.

Soluciones embebidas

Las soluciones embebidas son aquellas que se desarrollan para un propósito muy específico y con un software dedicado enteramente a la necesidad. Es el tipo de equipos más abundantes en la capa de percepción, ya que da la máxima eficiencia para una necesidad dada. En la industria, este papel lo desempeñan aún con frecuencia PLCs, pero cada vez es más abundante la presencia de otras soluciones basadas en microcontroladores que dan más flexibilidad a un coste menor.

Una ventaja adicional es que este tipo de dispositivos pueden tener un consumo muy bajo y ajustado, lo que posibilita usarlos con batería y los hace idóneos para aplicaciones inalámbricas o distribuidas.

El software de este tipo de equipos ha cambiado radicalmente en los últimos años gracias a múltiples iniciativas para permitir la portabilidad de aplicaciones entre los productos de múltiples fabricantes. Algunos ejemplos son [86, 89, 90]:

FreeRTOS sistema operativo de código abierto para sistemas embebidos. Es muy liviano y eficiente y cuenta con una serie de herramientas para el desarrollo de aplicaciones de tiempo real como los semáforos, colas y temporizadores. De esta manera muchas funcionalidades son fácilmente portables entre distintos microcontroladores.

Zephyr OS es un sistema operativo de código abierto gestionado por la Linux Foundation. Además de tener funcionalidad de tiempo real como FreeRTOS, dispone de una capa de abstracción para los recursos y entradas/salidas del microcontrolador. Esto le da aún más portabilidad entre distintos hardwares aunque significa que no está soportado por tantos microcontroladores como FreeRTOS.

Arduino es un sistema operativo de código abierto creado para facilitar el uso de microcontroladores a programadores no expertos. Se planteó sobre los equipos la marca Atmel aunque otros microcontroladores han proporcionado también compatibilidad con Arduino para facilitar la entrada de usuarios. Se centra sobre todo en la interacción con las entradas/salidas y comunicación del microcontrolador aunque dispone de muchas librerías y funcionalidades comunes para cualquier desarrollo en un microcontrolador.

MicroPython implementación de Python 3 para sistemas embebidos. Aunque Python es en origen un lenguaje de alto nivel pensado para ordenadores.

La ligereza de su implementación permite bajarla a tal nivel que se pueden traducir sus instrucciones a un microcontrolador sin mucha sobrecarga. Si bien esta implementación no soporta todas las librerías de Python, sí cubre un amplio espectro de ellas y permite desarrollar muchas aplicaciones con facilidad.

MBED OS es un sistema operativo de código abierto liderado por ARM que proporciona una capa de abstracción sobre todos los microcontroladores de la compañía ARM. Es más o menos similar a lo que ofrece Zephyr OS aunque específico para esta familia de productos. El soporte está garantizado por el propio fabricante aunque no está tan extendido.

Contiki OS sistema operativo de código abierto diseñado para equipos con muy bajas prestaciones. Aunque inicialmente se pensaba que acabaría por tener un uso muy intenso en IIoT, su funcionalidad quedó claramente sobrepasada por alguno de los anteriores.

ROS (Robot Operating System) es un proyecto de código abierto con librerías y herramientas para el desarrollo de robots sobre una abstracción software. Está perfectamente adaptado para sistemas distribuidos y es de instalación modular, lo que ayuda a mantener las soluciones compactas y sencillas de mantener.

Como se ve, la tendencia en el caso de los sistemas embebidos es claramente hacia el código abierto. La razón principal es la falta de interés de los fabricantes por aportar abstracción a sus SDK (Software Development Kits), centrándose más en promocionar las mejoras específicas de sus productos. Esto ha llevado a que sea la comunidad la que haya puesto en marcha las iniciativas para mejorar la portabilidad de aplicaciones. Incluso en el caso en el que los fabricantes se implican (MBED OS) les interesa que la comunidad de desarrolladores se involucre para

dar más fuerza a los desarrollos precisamente ante la presencia de estos otros proyectos más generalistas.

Si bien este tipo de desarrollos permite a una solución IIoT adaptarse a casos de uso muy específicos, las soluciones desplegadas sobre estos equipos están pensadas para ser específicas y sus capacidades de integración suelen estar bastante limitadas. Para cubrir esta necesidad existen los middlewares.

Middleware

Un middleware es un software que se sitúa entre dos capas de software para interconectarlas y provee al conjunto de la solución de escalabilidad y alta integración. Suele disponer de gran variedad de protocolos de comunicación, mucha versatilidad y posibilidad de configuración que le permite ser ampliado con el paso del tiempo. Esta capa de software se encuentra siempre en la capa edge de una arquitectura IIoT. Aunque todavía se encuentra situada en la planta y cerca de los equipos finales, lo común es que existan pocos o un solo equipo haciendo esta labor de middleware. Cuando se trata de uno solo habitualmente es un gateway IIoT.

Tal como se comentó en la Sección 3.2.2, un gateway IIoT proporciona el interfaz entre una comunicación local y remota, pero este tipo de equipos cada vez dispone de mejores prestaciones y pueden aprovecharse este intermediario para sacar mayor partido de su ubicación central en la solución. Así, con el tiempo, el middleware ha asumido funciones adicionales como la de procesamiento y abstracción de datos, capa de seguridad para la planta industrial e interfaz de gestión [91]. También puede usarse para acceder directamente a dispositivos de bajo nivel directamente y para integrar otros verticales. Según el caso de uso, este elemento central podría ser la única pieza en una implantación de IIoT al proporcionar la mayoría de las características, siendo las soluciones embebidas solo una forma de ampliar aspectos dónde el gateway IIoT no alcanza.

De forma similar al caso anterior, en el middleware también destaca por ser predominantemente de código abierto. En este caso la causa viene principalmente por la incapacidad de la mayoría de productos comerciales de seguir el paso de todos los casos de uso y mantener una buena explotación de las implementaciones, mientras que las iniciativas de código abierto se han beneficiado precisamente de esa variedad de casos de uso y aportaciones. Algunas ejemplos de estos proyecto son [86, 92, 93]:

Node RED es un software de código abierto para la programación de flujos de datos y procesamiento de forma sencilla. Es visual, fácil de usar, ampliable mediante componentes y muy ligero. Además dispone de una comunidad enorme de contribuyentes al proyecto, lo que facilita la disponibilidad de todo tipo de componentes y una funcionalidad en constante crecimiento. Su uso en aplicaciones IIoT está muy extendido aunque principalmente para domótica y educación, aunque también hay algunos fabricantes del ámbito de la automatización de lo han empezado a incorporar entre sus productos. Probablemente es el software más emblemático y conocido de esta categoría.

Eclipse Kura es un middleware industrial de código abierto desarrollado por la Eclipse Foundation con gestión completa sobre MQTT y está también orientado a la programación de tareas por flujos de trabajo. Al igual que Node RED resulta sencillo de configurar y dispone de un interfaz web, pero es más pesado en requisitos al ejecutarse sobre Java. La principal ventaja es la robustez que ofrece, el enfoque industrial y la capacidad de controlarse por completo remotamente a través de una API MQTT. Está basado en una infraestructura de servicios que pueden correr de forma independiente y ofrecer funcionalidades al conjunto lo que lo hace fácilmente escalable y muy modular.

EdgeX Foundry es un proyecto de código abierto liderado por la Linux

Foundation basado en microservicios que se orienta a enlazar servicios y datos entre las capas de percepción y plataforma. Tiene un enfoque tanto industrial como domótico y es fácilmente configurable a través de API REST. Da herramientas para procesamiento de datos, pero sobre todo modularidad para ampliar su funcionalidad.

Apache NiFi es un proyecto de código abierto de la Apache Software Foundation diseñado para automatizar el flujo de datos entre sistemas heterogéneos en tiempo real. Proporciona una plataforma visual y fácil de usar para construir, controlar y administrar flujos de datos complejos en una arquitectura distribuida y tolerante a fallos. Es más abstracto que los anteriores, lo que lo hace más adecuado para integraciones verticales y contra plataforma, pero no es tan potente en el sector industrial.

Siemens Industrial Edge entorno desarrollado por la empresa Siemens, para el despliegue de aplicaciones encapsuladas en contenedores. El papel de la empresa, además de proporcionar el entorno y algunos servicios integrados con dispositivos Siemens, es actuar como repositorio de aplicaciones para facilitar la distribución desde la parte desarrolladora y la puesta en marcha en la parte de cliente.

AWS IoT Greengrass es un servicio de Amazon Web Services (AWS) que permite extender a un dispositivo local, las capacidades del servicio de plataforma IIoT de Amazon (AWS IoT Core). En este sentido permite desplegar toda la potencia de la nube en un entorno local de manera muy escalable, según la aplicación. Es un enfoque totalmente distinto a los anteriores ya que la integración y sincronización con la capa de plataforma puede ser desplazada a un segundo plano, cosa que no ocurre con la mayoría de los middleware antes citados. De todos modos tiene una integración directa con AWS IoT Core.

Como se puede ver, aunque las grandes tecnológicas han desarrollado soluciones para IIoT, los proyectos más exitosos son aquellos ejecutados por las principales organizaciones de código abierto. Algunos middleware como Google IoT Core han llegado a ser retirados y otros como Windows for IoT no acaban de arraigar por la falta de adaptación en las capas inferiores de la arquitectura IIoT.

En cualquier caso, en todas los middleware presentados la tendencia clara es a la modularidad para que la solución sea versátil y admita adaptación a casos de uso variados o futuros escenarios que se puedan necesitar. También se ha impuesto tanto la configuración como la gestión de datos mediante flujos de trabajo, que facilita la mantenibilidad. Estas dos características reflejan bien el concepto del middleware como intermediario que distribuye la información entre otras herramientas.

Aunque en un equipo de esta capa podría llegar a acumularse muchas capacidades de procesamiento y gestión, confiarle demasiada funcionalidad comprometería la escalabilidad de cualquier solución al limitarse solo a casos de uso *on-premise* y aplicarse sobre equipos de capacidades limitadas. Asimismo, la extracción de la gestión y algunos procesamientos a un servicio remoto, aligera esta capa y concentra en la capa de plataforma todas las complejidades que no sean realmente necesarias afrontar en la capa edge.

Plataformas

Si la capa de percepción se centra en la conexión con los equipos físicos y la capa edge en la gestión de los datos obtenidos (abstracción, procesamiento y enrutado), el objetivo de la capa de plataforma es dotar al conjunto de escalabilidad, integración con verticales de negocio y una gestión sencilla de la arquitectura.

Las primeras plataformas IIoT tenían herramientas muy heterogéneas, eran

muy específicas de un caso de uso o sufrían problemas de escalabilidad. Con el paso de los años, se han establecido bastantes características y protocolos de referencia. Aunque es un campo que aún sigue en evolución, la mayoría de las plataformas en uso a día de hoy proporcionan al menos:

- Gestión de dispositivos: capacidad de gestionar remotamente un dispositivo, enviando mensajes o ejecutando métodos remotos. De esta manera se puede mantener un equipo actualizado, modificar los parámetros de funcionamiento y controlar su estado en tiempo real.
- Integración: conectar la arquitectura IIoT con verticales de negocio e industria (MES, SRM o ERP), de forma que se pueda sacar partido más allá del control de la propia planta.
- Escalabilidad: crecimiento tanto en número de dispositivos como en funcionalidad para que toda la arquitectura pueda evolucionar con el paso del tiempo y adaptarse a nuevas necesidades.
- Seguridad: en esta capa la seguridad se traduce en el control de los equipos conectados, gestión de usuarios y roles, acceso a la información, etc.
- Pérdidas de conexión: debido a la conexión inalámbrica, la plataforma debe estar preparada para que el equipo esté desconectado en un momento dado y aún así no perder los eventos, de manera que se puedan notificar en cuanto se vuelva a conectar.

Hay otras características como la estructuración de datos (para dotarlos de cierta semántica), la incorporación de procesamientos avanzados, la visualización o la replicación on-premise que también se pueden dar en las plataformas IIoT pero las citadas arriba serían las que se encuentran siempre presentes.

A pesar de compartir estas capacidades, cada plataforma tiene un enfoque distinto y suelen disponer de algún elemento diferenciador. Además, al contrario

que en los otros casos, el software para la capa de plataforma sí se encuentra liderado por las grandes empresas tecnológicas (Amazon, Google, Microsoft). La principal razón es que la garantía de plena disponibilidad y escalabilidad que se busca en esta capa es muy difícil de ofrecer en modelo gratuito y desplegar estas soluciones es un reto complejo para muchas empresas. Esto hace perder a las iniciativas Open Source bastante fuerza.

Algunas de las particularidades en las plataformas más utilizadas serían [86, 94, 95]:

- Google Cloud IoT Core: la fortaleza principal de la integración con Google es la potencia de sus servicios en la nube. Además de proporcionar muchos servicios comunes, es posible desplegar aplicaciones propias para el procesamiento de los datos recogidos o la integración con otros entornos.
- Azure IoT Suite: desarrollada por Microsoft, de nuevo se centra en utilizar la plataforma IIoT sobre todo como puerta de entrada a un entorno de procesamiento en la nube y dar a los clientes un amplio servicio que facilite el éxito de un caso de uso.
- AWS IoT Core: Amazon Web Services es a día de hoy la plataforma IIoT comercial líder gracias a que tiene una vía de entrada sencilla, es muy escalable y dispone también de servicios en la nube para facilitar la explotación de los datos. La razón de su éxito es principalmente que resulta sencillo para una empresa pequeña empezar a trabajar con los servicios de Amazon y es posible hacer una plataforma con plena flexibilidad. También tiene una integración con las capas inferiores muy abstracta que puede ser adaptada a cualquier caso de uso.
- Wonderware IoT: evolución de un SCADA de Schneider Electric adaptado para la integración con dispositivos y aplicaciones IIoT. Al ser una evolución desde un SCADA tradicional, dispone de todas las herramientas propias de

este tipo de sistemas como gestión de eventos, creación de pantallas de control, integración con herramientas de mantenimiento o de planificación y una gran robustez.

- Eclipse Kapua: proyecto de código abierto liderado por Eclipse Foundation que se integra perfectamente con Eclipse Kura y que tiene una integración muy avanzada para controlar un middleware IIoT. Se mantiene en línea con la solución comercial Everyware Cloud de la empresa Eurotech, que es la principal contribuidora al proyecto.

- Thingsboard: plataforma de código abierto agnóstica cuya principal característica es ser muy adaptable a cualquier aplicación gracias a la programación por flujo de trabajos y una vía de comunicación muy abstracta que se puede adaptar a cualquier dispositivo. La misma aplicación se comercializa con algunas características adicionales y servicio en la nube por la empresa The ThingsBoard dentro de un conjunto de herramientas y servicios para arquitecturas IIoT.

- Thingspeak: herramienta de código abierto desarrollada por MathWorks que permite conectar fácilmente dispositivos IIoT a las demás herramientas de procesamiento desarrolladas por la misma empresa. Una de las fortalezas de Thingspeak es la gran versatilidad y potencia para dar estructura a los datos adquiridos, lo que facilita su explotación a futuro.

- Thingworx: plataforma comercial desarrollada por PTC con un enfoque en grandes despliegues personalizados y agilizar la creación de soluciones ad hoc gracias a un amplio esfuerzo en soportar otros verticales, proporcionar herramientas de desarrollo y dar peso a los problemas de seguridad.

Además de estas, hay muchas otras plataformas IIoT, algunas que no han podido mantenerse en el mercado y otras que están en expansión. Se podrían

citar también Kaa Project, Balena, Kepware, Evrythng, Carriots, etc.

Como se ha visto, existen también varios proyectos Open Source que se comercializan en paralelo como servicio. Este modelo de negocio puede parecer débil, ya que se ofrece simultáneamente la plataforma gratuita y el servicio con pago. Sin embargo, la complejidad de desplegar y gestionar la plataforma justifica contratar el servicio en vez de alojarlo. De esta manera un proyecto se nutre de las aportaciones de la comunidad a través de su faceta Open Source y refuerza la estabilidad y sustentación del desarrollo mediante una empresa adjunta rentable. La mayoría de los casos de éxito Open Source se han constituido de esta manera.

Capítulo 4

Diseño y propuesta de la solución IIoT

Habiendo analizado las necesidades existentes en la industria actual en el Capítulo 2, se pone de manifiesto en qué medida pueden las tecnologías IIoT contribuir a una industria más eficiente y sostenible, en especial para los sistemas de gestión energética.

A partir de la arquitectura y tecnologías expuestas en el Capítulo 3, se propone a continuación una solución basada en IIoT que afronta y resuelve los retos de la industria como se describe en la primera hipótesis planteada (Sección 1.3).

4.1. Caracterización de las necesidades

Al tratar de construir una arquitectura IIoT para un entorno industrial, es necesario tener en cuenta los retos identificados en el Capítulo 2. Estos retos definen una serie de necesidades y requisitos que la solución debe paliar para una implantación exitosa.

El primer reto a analizar es la *evolucionabilidad*. Para dotar a una solución de ella es necesario que las herramientas puedan modificarse, reemplazarse y

adaptarse. Esto no se trabaja de la misma forma en todas las capas de la arquitectura. A más bajo nivel, es necesario disponer de una solución modular que pueda incorporar hardware y software acorde a las necesidades, sin crecer de más cuando no es necesario. Así, se facilita optimizar el coste de la solución. También debe ser flexible en su concepción, de forma que no esté fuertemente ligado a un dominio concreto. Por ejemplo, no sirve que esté perfectamente adaptada a una planta fotovoltaica si luego no es posible readaptarla a su uso en ciclos combinados.

A más alto nivel, esta adaptabilidad se traduce más bien en una facilidad para el mantenimiento de toda la arquitectura y también la posibilidad de realizar gestión remota de los equipos, soportando la modularidad de los equipos. En muchos casos, ser capaz de enviar mensajes de vuelta al equipo puede ser suficiente para desencadenar una instrucción remotamente, que puede ir desde un cambio de configuración menor a una actualización completa del software instalado. Disponer de esta funcionalidad a alto nivel habilita obtener el máximo rendimiento de una solución versátil a bajo nivel.

En segundo lugar, lograr una buena *interoperabilidad* se traduce sobre todo en conseguir que la arquitectura a diseñar no se convierta en un vertical más de la planta industrial. Por el contrario, debe ser muy capaz de integrarse en cualquier nivel y a cualquier otro servicio o aplicación, lo que implica también favorecer tanto integraciones verticales como horizontales. No solo se traduce en disponer de métodos de interacción con la solución en los equipos y servicios desplegados, sino también dar las herramientas para facilitar esas interacciones mediante el uso de protocolos extendidos y conocidos, un formato de los datos que pueda exportarse a otros servicios con sencillez o simplemente la capacidad de comunicarse por más medios que TCP/IP sobre ethernet, como RS-232, bus CAN o, en soluciones más adaptadas a la Industria 4.0, LoRa o BLE.

En este punto resulta relevante considerar la *ciberseguridad* como base para

estas comunicaciones ya que una solución es tan segura como la menos segura de sus partes. Debido a esto, es necesario abordar el reto de forma holística y en todas las capas de la arquitectura planteada, como mantener trazabilidad de qué dispositivos están en uso y qué hace cada uno. También controlar que no sea posible acceder a la información sin la autorización adecuada. Esto se puede resumir en los requisitos de autenticación, encriptación y trazabilidad. El primero permite identificar en todo momento cualquier persona, equipo o servicio que esté interactuando con la solución y a partir de ello determinar a qué tiene y a qué no tiene acceso. Además esto va ligado también a la trazabilidad ya que una vez identificado un servicio, persona o equipo, se puede conocer su actividad durante toda su vida útil. También implica tener un control de las autorizaciones entregadas para poder revocarlas en caso de verse comprometidas. La encriptación permite codificar la información en todo su ciclo de vida para que todo aquel que no esté identificado sea incapaz de obtenerla.

Respecto al *análisis de datos*, lo fundamental en este caso es que la solución tenga capacidad para desplegar algoritmos de tratamiento de datos lo más avanzados posibles en función de la capa en la que se trabaje. Esto significa soportar procesamiento edge en los equipos de más bajo nivel, y análisis más complicados a mayor nivel. Un enfoque básico sería trabajar con la información tan pronto como se disponga de los datos suficientes. Por ejemplo si un equipo de una capa intermedia ya puede procesar los datos de varios sensores para obtener una información con garantías, se haga el cálculo en ese nivel. Se destaca la necesidad de hacerlo con garantías porque una mera pasarela de datos podría no garantizar que la información está correctamente sincronizada o no tratar adecuadamente ventanas de tiempo largas.

También se considera parte de este requisito que la capa superior, la plataforma, pueda integrarse plenamente con herramientas de procesamiento avanzadas, principalmente aquellas comprendidas en el paraguas de tecnologías

de Big Data o de inteligencia artificial. A pesar de que la integración puede venir más marcada por requisitos anteriores, en este caso el enfoque es más próximo a poder integrarse forma bidireccional para la explotación de estos datos a través de la infraestructura IIoT construida.

La *fiabilidad* está muy ligada con la parte de análisis por esa garantía de hacer un cálculo preciso. Esto se traduce en un requisito principal que es mantener la integridad de los datos para que sean completos, precisos y coherentes como ya se comentó en la Sección 2.2.5.

Por último, las condiciones ambientales en industria tienen un impacto irregular en la implementación de una solución. En la capa de plataforma apenas influye ya que se despliega sobre un CPD o en la nube. Sin embargo, los equipos implicados en las otras capas sí tienen que adecuarse a entornos hostiles y variados. Es por eso que es necesario que la solución tenga por un lado facilidad para desplegarse en equipos con rango industrial y por otro que pueda usarse en equipos variados. Un punto clave para lograrlo es que los requisitos necesarios para lanzar la solución sean relativamente bajos y por tanto se pueda acceder a un amplio catálogo. Dado que el hardware es difícil de cambiar pero está continuamente mejorando, esto también beneficia a la evolucionabilidad de la solución al facilitar el cambio a equipos siempre actuales.

También vinculado a estas condiciones ambientales, se puede considerar el coste de la solución en su conjunto. Una solución que pueda funcionar en muchos equipos permite disminuir los costes. Si se considera en general, también evita la construcción de soluciones muy ligadas a un fabricante o una tecnología en particular. Todo esto ayuda a utilizar en cada situación los equipos más adecuados y de esta manera optimizar no solo la solución en términos de software, sino también en el hardware implicado. Adicionalmente, considerar el coste lleva también a controlar el mantenimiento y el consumo. Ambas cuestiones pueden llegar a suponer un coste muy significativo por ejemplo en

licencias o reemplazos en el primer caso. O un impacto directo en el beneficio para plantas de generación de energía en el caso del consumo.

En conclusión, los retos presentados en el Capítulo 2 delimitan los requisitos no funcionales de la solución final. En la Tabla 4.1 se presentan asociados a los distintos retos desde los que se han definido. Sin embargo, la relación entre requisitos es permeable ya que muchos estás relacionados. Por ejemplo la integración es la base de análisis de datos o la versatilidad se apoya también en los bajos requisitos de software.

Reto	**Requisito**
Ciberseguridad	Identidad
	Trazabilidad
	Encriptación
Evolucionabilidad	Versatilidad
	Modularidad
	Gestión
	Mantenimiento
Interoperabilidad	Integración
	Conectividad
Análisis de datos	Despliegue de aplicaciones
	Procesamiento avanzado
Fiabilidad	Integridad de datos
Condiciones ambientales	Rango industrial
	Bajos requisitos
	Coste

Tabla 4.1: Catálogo de requisitos de la solución IIoT a partir de los retos identificados en la industria. Estos requisitos determinarán las características de la solución final.

A partir de estos requisitos, se definen a continuación las características de la arquitectura IIoT a construir.

4.2. Diseño de la arquitectura

Para definir una arquitectura IIoT flexible y robusta, se parte de una solución en tres capas como se describe en el Capítulo 3. Esta arquitectura se acopla a los sistemas de una planta industrial en todos los niveles, según las capas. Las capas inferiores se integran de forma más horizontal, llegando a asimilar las redes y sistemas industriales. Conforme se trabaja con mayor abstracción, en las capas superiores, esta integración se realiza de forma más vertical, quedando los sistemas fuera de la arquitectura IIoT. En la Figura 4.1 se representa este escalonamiento de integración.

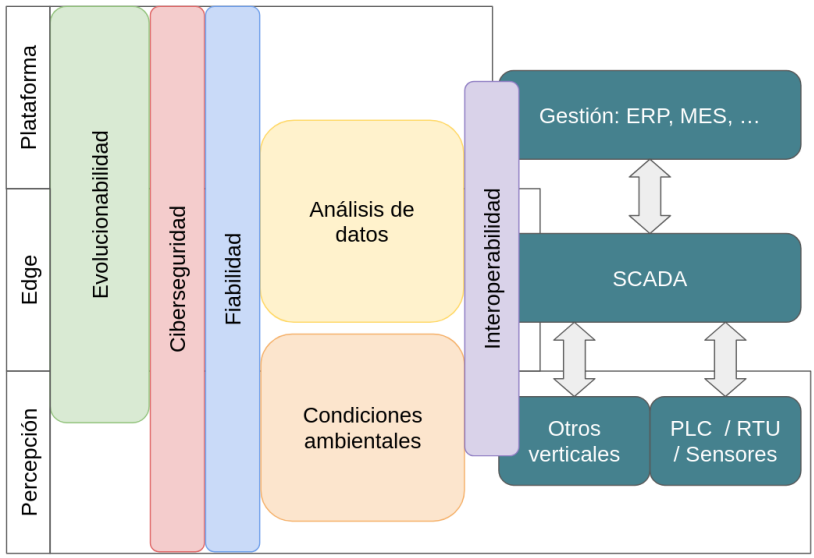


Figura 4.1. Impacto de los retos industriales en cada capa de una arquitectura IIoT. Algunos como la seguridad afectan de forma transversal a todos los niveles.

Debido a esto, y a las características de cada capa de la red IIoT, el impacto de los requisitos planteados no es homogéneo en todas las capas de la arquitectura. En la propia Figura 4.1 se puede ver un esbozo inicial de cómo impacta cada reto en cada capa según lo planteado en la sección anterior. Se analiza a continuación en más detalle qué consideraciones hay que tener capa a capa para cada requisito.

Este análisis se resumen en la Tabla 4.2.

Reto	Requisito	Percepción	Edge	Plataforma
Ciberseguridad	Identidad	✓	✓	✓
	Trazabilidad	✓	✓	✓
	Encriptación	✓	✓	✓
Evolucionabilidad	Versatilidad		✓	
	Modularidad		✓	
	Gestión		✓	✓
	Mantenimiento	✓	✓	✓
Interoperabilidad	Integración	✓	✓	✓
	Conectividad		✓	
Análisis de datos	Despliegue de aplicaciones		✓	✓
	Procesamiento avanzado			✓
Fiabilidad	Integridad de datos	✓	✓	✓
Condiciones ambientales	Rango industrial	✓	✓	
	Bajos requisitos	✓	✓	
	Coste	✓	✓	

Tabla 4.2: Catálogo de requisitos de la solución IIoT por capa. Algunos son estructurales de toda la arquitectura como la ciberserugidad y otros afectan solo a capas concretas.

4.2.1. Capa de percepción

Esta capa, que corresponde a la digitalización de la información, se aborda en la industria por dos vías. La más directa es el uso de los PLC u otros equipos ya instalados para la automatización industrial. La otra es la instalación de nuevos equipos que la automatización industrial no es capaz de controlar pero que aportan valor al sistema productivo.

Esta ambivalencia determina en gran medida cómo se afrontan los distintos requisitos ya que por ejemplo un equipo ligado únicamente a la arquitectura IIoT necesitará de mantenimiento dentro de la propia arquitectura, mientras que uno ya integrado en la automatización de la planta probablemente no lo admita o suponga un riesgo para la producción y seguridad integrar su gestión.

De hecho, en el caso de los dispositivos industriales ya instalados, lo

fundamental, y a lo que se supedita el resto de su integración, es no comprometer ni su seguridad ni su funcionamiento. Para dar esta garantía de funcionamiento, lo habitual es interactuar con estos sistemas en modo lectura o con comandos que ya se contemplen desde un HMI (Human-Machine Interface). En cambio, el primer punto (la seguridad) es más complejo y potencialmente mucho más peligroso. Para mantener segura una red de automatización, es importante que los equipos que se incorporen no abran un acceso inseguro a Internet o potenciales atacantes. Cualquier otra estrategia de ciberseguridad más avanzada (como el uso de credenciales, certificados o encriptación) es una posibilidad si alguno de estos equipos lo permite mediante algún protocolo, pero hay que asumir que en muchos casos no es posible.

Más allá de los equipos industriales preexistentes, hay otros sistemas que son de utilidad para un entorno productivo, pero que una red de automatización industrial tradicional no puede gestionar. Un ejemplo claro de esto son las redes de sensores inalámbricas (WSN por sus siglas en inglés, Wireless Sensor Network). Para estos equipos, la principal limitación para su uso en industria son la seguridad y el coste.

En el caso de la seguridad se espera como mínimo que dispongan de medios de identificarse (mediante credenciales o certificados, según el tipo de conexión) y de encriptar la información si se trata de conexiones inalámbricas. Protocolos como LoRaWAN o ZigBee tienen definiciones específicas para ambas cosas de manera que si se utilizan estos protocolos para la comunicación, se obtienen estas características. Destacar que otros como el WiFi no disponen de ello necesariamente (en las últimas especificaciones sí se contempla). En cuanto a la trazabilidad, en muchos casos solo se puede conocer su estado de conexión (conectado/desconectado de la red), pero suele ser suficiente para seguir su ciclo de vida.

El coste y el rango industrial es el otro gran reto para los equipos en esta capa.

Hay una amplia gama de equipos comerciales que se pueden utilizar directamente cumpliendo con el rango industrial. Encontrar entre ellos los que resultan más eficientes en coste y garantizar que cumplan otros requisitos es más complejo. Aunque dependerá del caso de uso, se intentan fijar algunos patrones para la selección de estos dispositivos en base al resto de los requisitos.

Otras características relevantes son la integridad de los datos y la conexión equipos de terceros. La integridad a este nivel se entiende sobre todo en términos de precisión. No es posible abordar la coherencia de los datos cuando se miden individualmente. A lo sumo puede descartarse una temperatura si está fuera del rango de medida del sensor, pero tratar de interpretar si es razonable exige normalmente más información local y un análisis que no suele ser eficaz llevar a esta capa.

La integración con otros equipos se enfoca principalmente como una recolección vertical de la información, aunque en algunas redes de sensores podría tener sentido integrarse horizontalmente con otros dispositivos de la red. En todo caso, no es necesariamente una consideración a este nivel ya que estos equipos rara vez incorporan mecanismos para integrarse con una variedad de sistemas. Lo más común es que se adapten a uno concreto y sobre todo que estén preparados para conectar con la capa edge.

Otras características como la versatilidad o el análisis no aplican de forma general. Si bien hay casos de uso que podrían justificar un procesamiento Machine Learning a bajo nivel, la problemática de mantener y sobre todo depurar dicho algoritmo hace que en la práctica las aplicaciones industriales ejecuten estos cálculos más adelante (capa edge o plataforma). En todo caso, una de las ventajas de las arquitectura IIoT frente a las redes de automatización tradicionales es que sí se contempla la capacidad para soportarlo.

La cuestión del mantenimiento presentada (sobre todo actualización remota y configuración), también es en sí misma una cuestión relevante. No obstante, en

la capa de percepción es difícil fijar un método estándar. Esto se valora con mejor perspectiva en la capa edge y sobre todo la plataforma.

4.2.2. Capa edge

En la capa edge se aloja el gateway IIoT propiamente dicho. Si bien la capa anterior adquiere la información convirtiéndola en un dato digital, en esta capa esa información se organiza para optimizar su explotación en la solución. A este nivel están disponibles características más complejas gracias a que tanto los equipos como el software implicado ofrece más capacidad.

Así, avanzando requisito a requisito, el gateway IIoT cubre los siguientes ámbitos:

- Identidad: ser capaz de utilizar credenciales o, preferentemente, certificados para las conexiones con el resto de equipos en cualquier otra capa de la arquitectura.
- Trazabilidad: informar a la plataforma de su actividad básica (conexión, desconexión y modificaciones al menos) para poder conocer su estado en todo momento y actuar en caso de quedar comprometido.
- Encriptación: el uso de protocolos basados en SSL o similares es fundamental para estos equipos, sobre todo hacia las capas superiores. Hacia capas inferiores está encriptación no siempre es posible. En ese caso se busca mantener los equipos lo más aislados posibles para no favorecer sus ataques, sobre todo evitar que se pueda acceder a un equipo de estos desde Internet.
- Versatilidad: esto exige mantener una configuración y diseño agnóstico. Es decir, no tener funcionalidad muy ligada a un caso de uso específico, sino admitir una gran variedad de usos. También se traduce en tener una configuración modificable y precisa.

- Modularidad: en la medida de lo posible, cualquier funcionalidad que se desarrolle para un caso de uso tiene que poder reusarse para otros. Esto amplía la necesidad de agnosticismo del punto anterior a una capacidad de asumir funciones de forma granular.

- Gestión: si bien es un requisito más ligado a la capa de plataforma por la magnitud en la misma, los gateway IIoT deben ser capaces de gestionar equipos por debajo de ellos. Esto suele suponer más un esfuerzo de integración que de gestión si cada equipo necesita un protocolo. Por supuesto, también se extiende a habilitar esa gestión y la del propio gateway remotamente desde la plataforma.

- Mantenimiento: de nuevo más ligado a las capas superiores, pero también relevante a este nivel ya que el equipo debe disponer de los medios para ser actualizado y revisado remotamente. En un equipo de la capa de percepción no es tan común realizar actualizaciones por robustez, pero para un gateway IIoT, expuesto a Internet, es crítico.

- Integración: al encontrarse funcionalmente en el centro de la arquitectura, un gateway IIoT debe contemplar más medios de integración que las otras capas porque puede ser necesario conectarlo con PLC de bajo nivel, aplicaciones de planificación avanzadas (ambas integraciones verticales) o servidores industriales (integración horizontal). La variedad de medios necesarios para ello involucra también al requisitos de modularidad. Por ejemplo, si fuera necesario incorporar un broker MQTT, o sobre este añadir el protocolo de un fabricante específico. Las posibles necesidades serían imposibles de enumerar por completo o de incluir en la solución de primeras. De la misma manera que el software, esto puede afectar también al hardware, siendo necesaria cierta *modularidad* en las opciones de comunicación del gateway.

- Conectividad: consecuencia de lo anterior, es necesario que el equipo edge tenga capacidad para asimilar toda la información integrada. Esto significa tener medios para armonizar los datos para usarse de forma indistinta en todas las aplicaciones que dependan de esa información. Es importante que dentro de la arquitectura IIoT planteada se definan uniformemente los datos para facilitar la conectividad con otros servicios.

- Despliegue de aplicaciones: muy vinculado también a la modularidad, en un equipo edge debe ser posible poner en marcha algoritmos de procesamiento que puedan aportar valor a los datos, o contribuir a mejorar su integridad.

- Integridad: en la capa edge, no solo debe preservarse la precisión de los datos heredada desde la de percepción, sino que ya es posible realizar agregaciones y comprobaciones cruzadas de los mismos para revisar su coherencia y garantizar que son completos. Esto puede hacerse mediante análisis estadísticos u operaciones de ámbito general entre la información obtenida, aunque la lógica específica y la confirmación/descarte de datos sí son específicos de las aplicaciones.

- Rango industrial: disponer de rango industrial para cualquier aplicación a la vez que se mantienen las opciones de integración ya comentadas traduce este requisito que la solución software (middleware) sea independiente del hardware. En la capa de percepción, el desarrollo hardware y software van ligados en todo momento. En esta capa, en cambio, el middleware tiene que ser capaz de desplegarse en cualquier equipo. Siempre son necesarias unas características mínimas (como un sistema operativo o similar), pero deben ser lo menos limitantes posible. La selección de un dispositivos industrial una vez consideradas estas características depende principalmente de la disponibilidad en mercado que, como se verá más adelante, es muy extensa y variables.

- Bajos requisitos: en línea con lo anterior, el middleware no solo debe partir de poca necesidad de abstracción, sino también solicitar pocos recursos al sistema. Por tanto, debe ser un middleware que no consuma mucha CPU o RAM y que no requiera mucho espacio. No limitar esto, podría hacer la característica anterior quedara inutilizada por la ausencia de equipos con rango industrial y suficiente rendimiento para correr la aplicación.

- Coste: aunque el coste del hardware en esta capa es menor que en la de percepción (por cantidad) o la de plataforma (por magnitud), otros costes como mantenimiento, consumo o licencias sí pueden llegar a ser relevantes y hay que tenerlos en consideración al dimensionar la solución.

Como puede verse, la capa edge es con diferencia la más solicitada en términos de requisitos. Esto se debe a que tiene que afrontar los retos de bajo nivel y alto nivel al mismo tiempo. En realidad es el punto de conflicto principal en la convergencia IT/OT ya presentados en la Sección 2.1.2. Estas dificultades se corresponden también con una menor cantidad de soluciones tecnológicas exitosas y sobre todo una clara falta de extensión de las mismas, aunque esto se detalla más adelante.

4.2.3. Capa de plataforma

La última capa de la arquitectura es la plataforma. En la plataforma IIoT convergen todos los nodos de las capas anteriores y se concentran aquellas funcionalidades de gestión, interacción con el usuario o de procesamiento avanzado.

Como en todos los niveles inferiores, la plataforma también debe atender a las cuestiones de ciberseguridad como base de su diseño. En este caso la principal dificultad viene de la identidad, ya que no solo gestiona la autorización de los dispositivos IIoT, sino también la de los usuarios y otras aplicaciones, además

es necesario hacerlo de forma aislada según el cliente o la organización. Así, en la plataforma IIoT conceptos de confidencialidad y protección de datos no solo afectan a la encriptación de la información, sino también a su acceso mediante la gestión de la autorización.

A caballo entre la gestión de equipos y la trazabilidad está la gestión del ciclo de vida de los dispositivos, en particular el proceso para darlos de alta o de baja. Esta característica, que es fundamental para la solución, es quizá uno de los puntos más conflictivos para una solución IIoT a nivel de plataforma. Entraña suficiente complejidad como para que sea muy difícil establecerla de forma general, pero al mismo tiempo debe ser accesible para gran variedad de equipos de la propia arquitectura y potenciales integraciones externas. Al mismo tiempo, y este problema es común a toda la gestión desde la plataforma, se coordina con equipos remotos cuya conexión puede ser inestable o sufre la transformación a protocolos distintos o más lentos (por ejemplo en el caso de redes malladas).

Ahondando más en la gestión, la plataforma también se comunica con los dispositivos para ejecutar procedimientos remotos y supone una complicación considerar que pueden encontrarse muchos tipos de dispositivos. Por ejemplo un comando que en un equipo es crucial, en otro podría no tener sentido. Esta diversidad puede asemejarse a la que afronta la capa edge en los datos que procesa, homogeneizando información muy variada bajo un mismo modelo de datos. De forma similar, la plataforma interactúa con muchos tipos de dispositivos y homogeneiza la interacción con ellos. Por lo tanto, esta gestión está también relacionada con la capacidad de integración y los protocolos empleados para la comunicación aguas abajo.

Hacia otras aplicaciones la integración habitualmente es más sencilla en este nivel ya que no existen tantos protocolos de comunicación en el ámbito IT como lo había en el OT. Además los datos ya se han armonizado en la capa edge. La principal complejidad se establece en la creación del enlace seguro que se puede

orientar como un proceso de alta contra otro servicio o como una integración independiente.

Uno de las principales integraciones a considerar es contra herramientas de análisis de los datos. Si en la capa de percepción se contempla el análisis como una excepción y en la capa edge se orienta a simplificar los datos o distribuir cálculos sencillos, en la capa de plataforma se despliega el resto del análisis necesario para explotar los datos. Esto abarca desde cálculos sencillos como estadísticos a análisis avanzados mediante *machine learning* o inteligencia artificial.

Para implementar estos procesamientos, se pueden incorporar dentro de la propia arquitectura como una parte de la plataforma o establecer como una herramienta externa. En el primer caso, es necesario que la plataforma tenga medios para desplegar módulos funcionales que accedan a los servicios de la plataforma internamente. Es un enfoque bastante común para los cálculos más sencillos o despliegue de modelos pre-entrenados. En el segundo caso se traduce más en una necesidad de integración de nuevo. En ambos casos, se contempla la opción de trabajar con procesamientos de inteligencia artificial cuyo ámbito de aplicación está en constante crecimiento y es parte del escenario a futuro para cualquier herramienta de software. Si bien en capas inferiores se pueden aplicar algunos procesamientos de esta naturaleza, en la plataforma es posible desplegarlos con mayor eficiencia al disponer de un conjunto de datos más consolidado y más tiempo y capacidades.

Es por esto que la plataforma es la capa de la arquitectura donde más hincapié se hace en la fiabilidad e integridad de los datos. A este nivel la precisión no es posible aumentarla, ya que viene determinada por las capas inferiores, pero la coherencia y la completitud sí son responsabilidad de la plataforma. De la misma manera que se puede analizar la coherencia de los datos en la capa edge, la plataforma valida que los datos tienen sentido no solo por entrar en un rango aceptable, sino también por asociación con otros parámetros. Para conservar los

datos completos se garantiza que la persistencia es robusta frente a caídas y cortes. También es importante asimilar los datos considerando las mismas políticas que en los equipos edge. Por ejemplo, si un equipo está varios días desconectado por alguna incidencia y al restablecerse envía datos antiguos a la plataforma, dichos datos no pueden desecharse porque contribuyen a una colección de datos completa. Sin embargo, tampoco se pueden tratar como información en tiempo real, en especial si son relativos al estado del propio equipo.

Por último, en lo relativo al hardware, tal como se ha comentado en la Sección 3.2.3, el escenario ideal es el despliegue en un CPD. En un CPD habitualmente se sirven los recursos del equipo (RAM, CPU o almacenamiento) a través de una capa de virtualización. Esto facilita mucho la adecuación de un servidor cualquier con un servicio o plataforma cualquiera. Por esa razón, es necesario que la plataforma cumpla con las características propias de un despliegue en CPD. Principalmente se traduce en dos cuestiones. Por un lado que admita replicación y distribución de sus servicios y por otro que su instalación, actualización y mantenimiento se realice de forma virtual y sencilla. Esto último se puede valorar fácilmente según si soporta alguna de las herramientas de despliegue y orquestación más extendidas como puede ser Docker o Kubernetes.

4.2.4. Arquitectura de la solución

Tras el análisis capa a capa de cómo afrontar los distintos requisitos y retos identificados, se desarrolla a continuación una visión de conjunto de toda la arquitectura. En la Figura 4.2 se resumen todas las cuestiones consideradas hasta este punto en la arquitectura IIoT y su relación con la red industrial.

De esta visión de conjunto, se pueden extraer las características a nivel global que tiene la arquitectura IIoT que se propone construir:

- Centralizada: a pesar de admitir redes de malla, facilitar integración en todos los niveles o distribuir funcionalidad sobre todo en la capa edge,

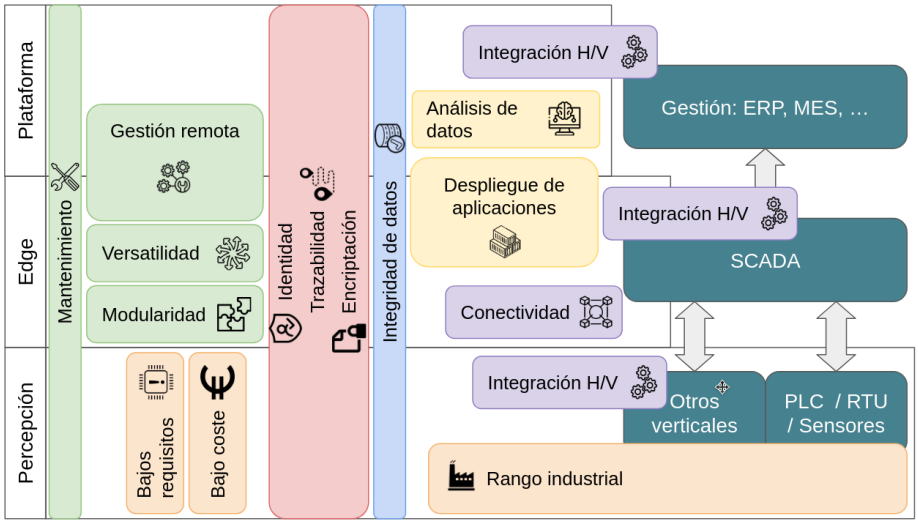


Figura 4.2. Requisitos de la aplicación en las distintas capas de una arquitectura IIoT. La integración con los sistemas industriales está presente en cada capa para facilitar la conexión con nuevos procesos y verticales.

existe un nodo central en la plataforma IIoT que es fundamental para la gestión de la arquitectura internamente.

- Permeable: al contrario de lo visto en la pirámide de automatización de la Figura 2.1, la interacción entre las distintas capas y con terceras soluciones es parte de la solución. Aunque se separe la capa de percepción de la de plataforma a nivel funcional, sí puede haber conexión directa entre ellas.

- Escalable: a pesar de ser centralizada, la arquitectura IIoT propuesta descarga mucha funcionalidad en los gateways intermedios. En la medida en que estos gateways aportan recursos significativos a la arquitectura, el crecimiento de la misma no es tan solicitante como en otras topologías cuyo procesamiento también está centralizado. Además la permeabilidad combate un aumento de complejidad con la incorporación de nuevos equipos.

- Integración: la característica motora de esta solución es la mejora en la

integración, por lo que es posible hacerlo en cualquier nivel y con una amplia cantidad de métodos.

- Expandible: ya sea por integración con terceros, modularidad o por despliegue de procesamientos, la solución en conjunto se puede mejorar o actualizar conforme las necesidades del proceso evolucionen.

- Segura: a todos los niveles, la solución se plantea sobre la base de la ciberseguridad exigiendo métodos de autenticación y encriptación y facilitando la verificación del estado de la misma para una gobernabilidad sencilla a los administradores.

- Industrial: la solución está pensada para poder entenderse con un entorno industrial productivo. Algunos ejemplos son el uso de hardware, o la integración con sistemas de planificación. También es capaz de interactuar con el proceso productivo sin necesidad de desactivar equipos antiguos.

- Bidireccional: la comunicación dentro de la arquitectura es bidireccional en tanto que equipos y plataforma se comunican entre sí para enviarse datos adquiridos, telemetría de su estado o ejecución de procedimientos remotos.

Estas características definen las líneas generales para crear una arquitectura IIoT adaptada a los retos de la industria y que considera todos los requisitos identificados. Sin embargo, para obtener una aplicación final eficaz también deben considerarse en cada toma de decisión a la hora de implantar cada solución o caso de uso concreto.

4.3. Selección de tecnologías

Partiendo de la arquitectura descrita se pueden construir multitud de soluciones IIoT distintas con la tecnología existente. El objetivo de esta sección

es determinar entre el software, hardware y protocolos disponibles, cuáles son los que mejor se adaptan a la arquitectura buscada.

Para ello primero se selecciona el software a emplear en cada capa. Esto determina a su vez los requisitos finales para el hardware, que además de cumplir con lo ya planteado tendrá que ser capaz de ejecutar el software seleccionado. Finalmente, se traza un esbozo de las tecnologías de comunicación preferentes para la solución, sin ir en detrimento de otras en caso de ser necesarias para un caso de uso.

Destacar que de las tecnologías identificadas en el Capítulo 3, algunas no se contemplan como opciones para la arquitectura presentada por tener un uso muy distinto. Por ejemplo, el middleware Apache NiFi que se describe en la Sección 3.3.3 está pensado para un enfoque más simple de la capa edge como enrutado de datos a otras plataformas. Las características que se contemplan en este caso son mucho más complejas de las que puede llegar a gestionar este software.

4.3.1. Software

Dentro del software, se distinguen las tres capas de la arquitectura. Cada uno parte de un entorno de trabajo y unas capacidades distintas. Para la capa de percepción, donde lo principal serían los sistemas embebidos, se busca un software capaz de ejecutarse en un microcontrolador. Para el edge, donde se trabaja principalmente con un PC industrial, se busca un middleware que disponga a la vez de flexibilidad y robustez. En la capa de plataforma el entorno de trabajo es un CPD por lo que se pueden conjugar varias herramientas coordinadas en un mismo entorno para cubrir toda las características, aunque lo ideal sería una plataforma IIoT que contemple todo lo descrito en un mismo servicio.

Sistemas embebidos

Los sistemas embebidos se caracterizan por ser muy específicos para una aplicación concreta. Esto los hace mucho más fiables y sencillos. Debido a esto, es muy difícil con los requisitos planteados cerrar un software concreto, sino que es necesario contemplar las alternativas de cara a los casos de uso concretos. Además hay que tener en cuenta que en muchos entornos industriales esta parte ni siquiera es necesaria al disponerse ya de una capa de percepción compuesta por PLC o sensores industriales.

Lo que sí se plantea, es el uso de un sistema operativo de tiempo real para dar una base mínima de homogeneización a los desarrollos. Para esto, la principal restricción que se contempla es que sirva para la mayor parte de los microcontroladores, y limite lo menos posible el hardware y los desarrollos construidos sobre el mismo.

Framework	**Portabilidad**	**Equipos soportados**	**Eficiencia**
FreeRTOS	Media	Casi todos	Muy alta
ZephyrOS	Media	Muchos	Alta
MBED OS	Alta	Bastantes	Media
Contiki OS	Baja	Bastantes	Alta
RIOT OS	Media	Pocos	Alta

Tabla 4.3: Comparativa de OS de tiempo real. La implantación de FreeRTOS en los microcontroladores comerciales es casi universal.

Valorando algunas de las más extendidas (FreeRTOS, ZephyrOS, MBED OS, Contiki OS y RIOT OS), la que tiene mejor portabilidad y es soportada en la mayoría del hardware es FreeRTOS, como se muestra en la Tabla 4.3. En comparación, ZephyrOS es más reusable entre proyectos, pero no tiene un catálogo tan amplio de dispositivos donde realmente se puedan usar sus prestaciones, lo que contrarresta en parte esa reusabilidad. En el caos de MBED

OS esto se acentúa más ya que tiene una reusabilidad mayor y disminuye bastante el tiempo de desarrollo pero está mucho más acotado el catálogo de equipos disponibles. Contiki por su parte está relativamente poco extendido en comparación con las otras porque se centra en el nicho de equipos con mayores restricciones. Por último RIOT OS está todavía en expansión y mejorando su portabilidad. Si bien podría acabar siendo una de las principales alternativas en la actualidad aún es superada por FreeRTOS.

Por tanto, FreeRTOS es la mejor opción en lo que se refiere a abarcar una gran parte de los dispositivos disponibles. Algunas de sus principales características son:

- Determinista: garantiza tiempos de respuesta predecibles, lo cual es crucial para aplicaciones en tiempo real.
- Portabilidad: como se ha comentado, está soportado en una gran gama de dispositivos, ARM Cortex, AVR, PIC, etc.
- Bajos requisitos: tiene unos requisitos de potencia y memoria mínimos, lo que permite usarlo incluso en aplicaciones de consumo extremadamente bajo o con batería.
- Código abierto: se distribuye bajo licencia de código abierto y no es necesario pagar un cargo por implantación o proyecto. Esto impacta mucho en el coste final de una solución, sobre todo en la capa de percepción.
- Herramientas básicas: proporciona una colección de herramientas para gestionar tareas y aplicaciones de tiempo real y de forma unificada para cualquiera de los procesadores soportados.
    - Gestión de hilos: API para crear, destruir y gestionar tareas de forma dinámica
    - Colas: para facilitar el traspaso de mensajes e información entre tareas

- Sincronización: semáforos y mutex para proteger recursos compartidos y sincronizar el acceso a los mismos
- Eventos: basado en el sistema de colas, permite desencadenar tareas y acumularlas para no perder ninguno
- Temporización: conteo, retrasos y eventos temporales

Destacar que muchas de las herramientas disponibles por FreeRTOS, también están disponibles a bajo nivel en muchos microcontroladores, la ventaja de usar FreeRTOS es que no es necesario ahondar en las características a bajo nivel, porque el propio sistema operativo las proporciona a través de un API que es común a todos los microcontroladores en los que se implemente, sin perder eficiencia.

Middleware

Lo fundamental a la hora de seleccionar un middleware es que sea versátil y modular, de forma que se pueda adaptar a muchos casos de uso. También es necesario que disponga de muchas herramientas de integración. En general, todos los que se valoran contemplan estas dos características de base. Sus principales diferencias entran en los otros requisitos.

En la Tabla 4.4 se presenta una comparación de estas otras características que sí resultan diferenciadoras. Para su elaboración, se ha considerado que un middleware dispone de un buen enfoque industrial si es robusto y su integración está pensada con protocolos industriales. El procesamiento no solo contempla la capacidad de desplegar y correr aplicaciones de desarrollo avanzado, sino también lo que ya tenga incluido para facilitar el tratamiento de datos. La gestión remota va de la mano del mantenimiento, ya que siempre se contempla hacerlo remotamente. Se añade la característica de código abierto como un reflejo del requisito de coste.

Solución	Industrial	Procesamiento	Requisitos	Ciberseguridad	Gestión	Código abierto
NodeRED	No	Básico	Bajo	Básico	Nada	Sí
AWS Greengrass	No	Avanzado	Medio	Avanzado	Básica	No
EdgeX Foundry	Sí	Avanzado	Medio	Medio	Básica	Sí
vNode	Sí	Básico	Medio	Avanzado	Avanzada	No
ESF	Sí	Avanzado	Medio	Avanzado	Avanzada	No
Eclipse Kura	Sí	Avanzado	Medio	Avanzado	Avanzada	Sí

Tabla 4.4: Comparativa cualitativa de middleware IIoT. El enfoque industrial comprende robustez e integración con protocolos industriales.

Se detallan a continuación fortalezas y debilidades de estos middlewares para las solución que se propone:

NodeRED aunque se trata de un software sencillo, ligero y quizá de los más extendidos de los que se analizan, NodeRED presenta muchas deficiencias para un uso en industria. Su enfoque genérico lo hace idóneo para pruebas de concepto pero no presenta de base las capacidades de ciberseguridad, escalabilidad y robustez que se buscan en una solución industrial. Su principal ventaja es que posee una comunidad extensa y es posible ampliar su funcionalidad para cubrir las desventajas.

AWS Greengrass sus capacidades son de uso general y no están adaptadas específicamente para industria. Lo que sí presenta es mucha fortaleza en ciberseguridad y escalabilidad. A pesar de disponer de un SDK (Software Development Kit) para el desarrollo de ampliaciones, el framework de base no es de código abierto y la gestión remota que permite es muy básica.

EdgeX Foundry aunque es agnóstico en su enfoque, está claramente preparado para un entorno industrial. Su principal debilidad a día de hoy es la gestión remota de este software, pero el número de desarrolladores contribuyendo a este proyecto es cada vez mayor y podría posicionarse como una de las mejores opciones en no mucho tiempo.

vNode está perfectamente adaptado para industria y dispone de herramientas

de ciberseguridad avanzadas. Sin embargo, su coste es bastante alto por dispositivo desplegado. También tiene serias deficiencias en la escalabilidad de los procesamientos y en la ampliación de funcionalidad.

ESF Everyware Software Framework tiene un enfoque exclusivo para industria, herramientas de ciberseguridad avanzadas y una gestión muy granular. Este middleware se adapta perfectamente a los requisitos planteados para la capa edge.

Eclipse Kura de forma similar a ESF está muy bien adaptado a los requisitos para el middleware. El principal inconveniente es la falta de soporte y el retraso de una versión respecto a ESF (excepto en parches de seguridad). La ventaja es que es de código abierto, lo que reduce los costes por equipo y permite suplir las carencias destacadas.

Se considera que Eclipse Kura es el middleware idóneo para una solución IIoT industrial. Además de cumplir con la mayoría de los requisitos y ser de código abierto, tiene una sólida comunidad de desarrolladores, no solo en la plataforma sino también en módulos adicionales. Esto proporciona componentes listos para usar y allana la curva de aprendizaje en la creación de componentes propios. Las principales desventajas son la necesidad de incorporar las carencias con respecto a ESF y la exigencia de unos requisitos algo elevados que se trasladarán al hardware.

Entrando a mayor detalle, Eclipse Kura es un software basado en OSGi (Open Services Gateway initiative) con licencia de código abierto. Incorpora nativamente muchas características necesarias en cualquier proyecto de IIoT y se puede ampliar mediante plugins desarrollados por particulares. Además, tiene una gran versatilidad para configurar distintos flujos de trabajo según los requisitos de cada solución.

Algunas de las principales características de Kura son:

- Comunidad Open Source: existe una amplia comunidad detrás del

desarrollo de Kura que, además de proporcionar parches de seguridad y nuevas funcionalidades cada cierto tiempo, ofrece un soporte a los usuarios. Además de esta comunidad oficial hay otros proyectos Open Source de módulos para Kura que se pueden instalar a través de una plataforma oficial (Eclipse Marketplace). Estos desarrollos también contribuyen a allanar la curva de aprendizaje para crear componentes propios.

- Infraestructura de servicios: dispone de una infraestructura para tareas de IIoT. Incluye herramientas básicas para la conexión a dispositivos IIoT, equipos industriales y plataformas remotas. También para la gestión del propio equipo (localización, red, configuración, gestión de certificados, etc.) y un panel de configuración del flujo de trabajo.

- Arquitectura de plugins: permite la instalación de plugins o módulos que amplían la funcionalidad desde la estructura básica presentada en el punto anterior.

- Flujo de trabajo: posee un interfaz donde se puede diseñar de forma gráfica la interacción de distintos servicios y funcionalidades para configurar el trabajo a ejecutar. Este entorno está basado en bloques funcionales, como puede ser un temporizador, una lectura de un dispositivo, el almacenamiento, etc. Gracias a esta configuración, se pueden programar o cambiar fácilmente las tareas del equipo.

- Ciberseguridad: la configuración local se almacena siempre de forma cifrada para reducir la vulnerabilidad de datos sensibles. También dispone de un servicio de gestión de certificados que facilita la conexión autenticada y cifrada. Finalmente, el interfaz web y REST poseen autenticación por token y despliegue nativo por SSL.

- Servicios integrados: aunque la principal fuerza es la ampliación mediante

servicios personalizados, ya incluye algunos muy interesantes sea con servicios preinstalados o con plugins creados por la comunidad. Algunos de estos servicios son: localización, servidor MQTT, cliente MQTT con protocolo de gestión remota, almacenamiento en base de datos, servidor REST, gestión de interfaces de red, gestión de firewall, algunos drivers de comunicación industrial comunes (Modbus, S7 y OPC UA), etc.

Se destaca que Eclipse Kura tiene facilidades para asumir algunas características de la capa de percepción, por ejemplo con la lectura/escritura de señales digitales. Este enfoque es parte de la permeabilidad entre las capas de la arquitectura IIoT y da mucha flexibilidad a la hora de implementar soluciones. Dado que la electrónica industrial (principalmente PLC) cubre mucho terreno de la capa de percepción, es común prescindir de ella por completo en una aplicación. Si el middleware puede asumir algunas capacidades de la capa de percepción, se mejora el rendimiento para estos casos y se ahorra el uso de equipos adicionales. Aunque este solapamiento de funcionalidades podría suponer un problema para la automatización tradicional, la abstracción de la capa edge anula los efectos negativos en una arquitectura IIoT.

Plataforma

La plataforma es en términos básicos la herramienta que permite gestionar la flota de dispositivos y centralizar la información. De esta plataforma se busca que tenga la mejor integración posible con el middleware pero no debe ser exclusivo. Hay que considerar que ninguna de las encontradas cubre los requisitos a la perfección y que es necesario que se pueda ampliar con nueva funcionalidades.

Al analizar las alternativas, se detecta que algunas plataformas están pensadas para uno o unos pocos casos de uso específicos. Principalmente debido a que se han desarrollado desde otro software o porque ha bebido de un conjunto de usuarios reducido. Esto las hace más eficientes en esas aplicaciones pero mucho menos

en otras, a veces incluso inviables. Otras, sin embargo, no han puesto el foco en ningún caso de uso en particular y tienen una perspectiva más abstracta de la gestión de datos y dispositivos. Esto, en el ámbito tecnológico, se conoce como una plataforma agnóstica. Habitualmente pueden alcanzar la misma eficiencia que las otras pero son más complejas de configurar. La gran ventaja es la flexibilidad que permite. Esto se valorará también entre las características a pesar de no provenir de los requisitos iniciales.

Plataforma	Gestión	Integración	Análisis	Visualización	Personalizable	On-premise	Agnóstica	Código abierto
Azure	Avanzado	Medio	Avanzado	Avanzado	Básico	No	No	No
AWS	Medio	Avanzado	Avanzado	Medio	Avanzado	Sí	Sí	No
Everyware	Avanzado	Medio	Básico	Básico	Básico	Sí	Sí	No
Kapua	Avanzado	Básico	Básico	Básico	Medio	Sí	Sí	Sí
Thingworx	Básico	Avanzado	Medio	Avanzado	Medio	Sí	Sí	No
ThingSpeak	Básico	Medio	Avanzado	Avanzado	Medio	No	No	No
Kaa	Básico	Avanzado	Avanzado	Básico	Avanzado	Sí	Sí	Sí
Thingsboard	Medio	Medio	Medio	Avanzado	Avanzado	Sí	Sí	Sí

Tabla 4.5: Comparación cualitativa de plataformas IIoT [95-98]. El agnosticismo de la plataforma se refleja sobre todo en la gestión de los dispositivos y puede marcar la diferencia a la hora de adaptarla a una solución.

En la Tabla 4.5 se resumen algunas de las opciones más relevantes del mercado, de las cuales se podría comentar:

Azure IoT Suite es de las mejores dotadas para el procesamiento. Aunque no se pueda considerar agnóstica están haciendo muchos esfuerzos por paliarlo.

AWS IoT Core la gran versatilidad que ofrece en la plataforma la hace muy adecuada para casi cualquier solución. Sus herramientas de gestión son básicas por el bajo nivel de especificación aunque cada vez van teniendo más importancia.

Everyware Cloud tiene puesto el foco en la gestión remota y agnóstica de dispositivos, aunque dispone de herramientas de visualización y análisis muy rudimentarias.

Eclipse Kapua al igual que pasaba entre ESF y Kura, Kapua está por detrás de

Everyware Cloud en algunas características y soporte. La principal ventaja respecto a la anterior es la posibilidad extender su funcionalidad por su diseño modular.

ThingWorx dispone de un modelado de datos muy avanzado y personalizable aunque tiene una gestión remota e integración muy inferiores a otras plataformas..

ThingSpeak está muy adaptada al análisis e inspección de datos, pero sus características de gestión no permiten mucha operativa remota.

Kaa tiene una gran fuerza en la capacidad de análisis y visualización de datos. Al ser principalmente un hub de información la gestión de equipos es muy básica. Una de sus principales ventajas es que es muy personalizable.

Thingsboard tiene un enfoque de gestión de dispositivos muy ligero pero también muy personalizable en múltiples protocolos de comunicación, lo que la hace muy versátil en este campo. Dispone de una arquitectura modular y tiene un diseño robusto de interacción con los equipos.

Al compara las alternativas, hay que tener en cuenta que una aportación propia de la plataforma IIoT y no que no puede obtenerse de ninguna otra herramienta es la gestión de los dispositivos conectados. Por esto, se le dará prioridad a aquellas que den más facilidades en ese aspecto. Otras características como el análisis y la visualización pueden aportarse a posteriori siempre que la plataforma pueda integrarse con ellas, pero la gestión es muy compleja de delegar a un servicio separado.

También se debe dar importancia a la opción de despliegue On-Premise (localmente). Aunque muchas empresas no tienen problema en integrar sus sistemas con un entorno *cloud*, para otras resulta crítico que la información

nunca salga de sus plantas o empresa. Esto es particularmente importante en ámbitos militares o de recursos estratégicos, como es la energía eléctrica.

Dentro de las alternativas de código abierto, en estos dos aspectos destacan principalmente Eclipse Kapua y Thingsboard. Por sus características ambas opciones serían viables para una arquitectura IIoT. Para acotar la selección, se valora la proyección a futuro de ambos proyectos Open Source. El proyecto de Thingsboard solo está soportado por ThingsBoard, Inc., mientras que Kapua, aunque proviene del producto de la empresa Eurotech, está gestionado por la fundación Eclipse. En una situación de empate, la implicación de una fundación de código abierto tan estable como Eclipse garantiza la continuidad del proyecto. También, facilita las integraciones dentro de la infraestructura Eclipse, que es toda de código abierto y contiene muchos proyectos relacionados con IIoT (Eclipse Ditto, Eclipse Hono, Eclipse Arrowhead, ...). En particular, Kapua tiene la ventaja de tener una integración plena con el middleware seleccionado, Eclipse Kura, lo que garantiza una gestión robusta.

Entrando en detalle, Eclipse Kapua es una plataforma IIoT de código abierto y estructura modular desarrollada por el equipo de Eclipse IoT que posee integración específica para Kura y proviene de la solución comercial de Everyware Cloud. Entre sus características se puede nombrar:

- Broker MQTT: el núcleo de la plataforma es su broker MQTT, a través del cual recibe y envía los datos. Sobre este mismo también integra el protocolo de gestión remota de la plataforma, facilitando la integración para todo tipo de dispositivo IIoT.

- Gestión remota: capacidad para instalar/actualizar software, enviar configuraciones y gestionar todo tipo de servicios a través de su protocolo de configuración basado en MQTT. Dispone además de integración específica con Kura sobre este protocolo.

- Consulta de datos: permite la consulta de la información recibida del dispositivo IIoT para un análisis básico de los datos proporcionados. Sin embargo, este entorno está pensado específicamente para una gestión de mensajes y canales MQTT, por lo que más que disponer de una visualización de series temporales y gráficas, es un entorno de análisis de *topics*, mensajes y dispositivos que los han publicado.

- Programación de tareas: dispone de un interfaz para la creación y programación de tareas sobre los dispositivos para que se realicen de manera asíncrona y no depender de que estén conectados en un momento dado.

- Gestión de usuarios: la plataforma tiene un método avanzado de gestión de usuarios, roles, permisos y dominios de trabajo, de forma que puede seccionarse el acceso a la información y definir con detalle lo que un equipo o persona puede o no hacer.

- Integración: la plataforma puede comunicarse con otros servicios vía MQTT pero también servir su funcionalidad a través de un API REST. También dispone de un interfaz web para el uso directo de un usuario.

- Ciberseguridad: además de la gestión de usuarios avanzada, posee la capacidad de conectarse bajo protocolo SSL en las integraciones por MQTT, web y REST. También gestiona los certificados de los equipos conectados y su validez.

- Despliegue local: permite el despliegue de la plataforma y los servicios de forma local en la instalación (on-prem) bajo Docker u OpenShift con sincronización con una nube central.

A pesar de tener una integración específica con el middleware de Eclipse Kura, el protocolo empleado en Kapua para la gestión de equipos IIoT es público y puede

implementarse sobre cualquier cliente MQTT con el grado de complejidad que se desee. Esto otorga mucho control sobre cualquier equipo que se desee conectar a la plataforma.

4.3.2. Hardware

Para desplegar el software presentado en el apartado anterior, es necesario seleccionar un hardware sobre el que pueda ejecutarse aprovechando todas las capacidades. El tipo de dispositivo depende, evidentemente de la capa en la que se trabaje.

Capa de percepción, aquí se busca trabajar en tiempo real con funcionalidades muy específicas y robustas. Es por esto que se seleccia un software adaptado a la medida de los microcontroladores, que son ideales para este tipo de usos. Esto permite diseñar según la aplicación un hardware que se ajuste a la perfección a la situación concreta, ya sea porque se busque tener un consumo especialmente bajo, integrar alguna señal eléctrica o sensor fuera de lo común, o desplegar una lógica un poco más compleja. La colección de microcontroladores disponibles para estas situaciones es inmensa. Para ejemplificarlo, se muestra en la Tabla 4.6 una serie de módulos seleccionados para una aplicación media basada en Bluetooth. Evidentemente se dejan de lado muchas otras opciones. Como se puede ver, solo entre los equipos presentados ya existen fuertes diferencias según la capacidad de procesamiento, el consumo o el precio. También en los protocolos adicionales que soportan. Según otros condicionantes del caso de uso, será necesario elegir uno u otro, lo que impactará en el resto del desarrollo de hardware.

Esto no significa necesariamente incurrir en un desarrollo electrónico completo para cada solución, ya que a día de hoy existen muchas soluciones comerciales que cubren mucho territorio para cualquier solución. Así, se pueden encontrar en el mercado módulos de uso general como puede ser el ESP32 o Arduino sobre

Característica	ESP32	nRF52840	BGM220PC22HNA	STM32WB5MMG
CPU	Xtensa Dual-Core LX6	ARM Cortex-M4	ARM Cortex-M33	ARM Cortex-M4
SPI Flash	16 MB	1 MB	512 kB	1 MB
SRAM	512 kB	256 kB	32 kB	256 kB
Consumo medio	80 mA	7 mA	5 mA	9 mA
Coste	2 - 4 €	2,5 - 6 €	6 - 7 €	2 - 5 €
Bluetooth	BLE 4.2	BLE 5.0	BLE 5.2	BLE 5.0
WiFi	Sí	No	No	No
ZigBee	No	Sí	No	Sí
Otros	CAN	NFC, Thread	-	Thread
DAC	Sí (2x)	No	No	No
UART	3	2	3	2
I2C	2	2	2	2

Tabla 4.6: Comparativa de microcontroladores para Bluetooth. Se presentan algunas alternativas del mercado con las características más comúnmente usadas de estos dispositivos.

una placa que permite un uso directo. También integraciones de placas similares con ampliaciones como puede ser un módem NB-IoT o LoRa. Según la escala de la implementación puede hacerse necesario avanzar al desarrollo de una PCB, pero otros casos no necesitan llegar tan lejos. Una de las fortalezas precisamente de una arquitectura IIoT es que se puede gestionar la heterogeneidad derivada de usar muchos tipos de equipos sin que la solución se vuelva más compleja de mantener.

Tal como se ha comentado, en el ámbito industrial la capa de percepción también considera los sistemas de automatización ya presentes o que se vayan incorporando. Estos dispositivos, aunque se integran en arquitectura, no se consideran dentro del alcance de las especificaciones de hardware, ya que el fabricante o integrador de esos equipos son los que marcan los requisitos. Por poner un ejemplo, en una planta fotovoltaica los inversores suelen incorporar ya la lógica necesaria para gestionar los paneles y comunicarse, todo provisto por una solución cerrada del fabricante en la que poco se puede incorporar. Otro ejemplo en el mismo caso de uso es el de los contadores, gestionados por la compañía eléctrica que de nuevo incorporan ya la capacidad de medir todo lo

relevante a ese nivel y la comunicación, por lo que no hay margen para selección de hardware alguno ni despliegue de aplicaciones. En estos casos, la arquitectura IIoT trata de establecer una integración horizontal con estos equipos en la medida de lo posible. Así, además de recoger los datos que generen, se podrá interactuar con ellos como si realmente formaran parte de la arquitectura.

La capa edge debe ser capaz en primer lugar de ejecutar el software seleccionado, que tiene unos requisitos mínimos. Más allá de estos requisitos, podría necesitar más prestaciones según el caso de uso para llevar a cabo un procesamiento más avanzado. También tiene que integrar los equipos de la capa de percepción, que puede implicar incorporar algún módulo de comunicaciones, de nuevo según el caso de uso.

En el caso de Eclipse Kura, los requisitos mínimos no están claramente descritos en la documentación. Esto hace necesario estudio específico con el que esclarecerlos en detalle. Los detalles de este estudio pueden encontrarse en [2], donde se comparan equipos industriales con características diversas para poner a prueba el software. Como resultado, se obtienen una serie de requisitos mínimos y recomendados que se presentan en la Tabla 4.7. Como se puede ver, son muy inferiores a cualquier plataforma de control industrial y, sin embargo, puede proporcionar medios de integración muy superiores a la mayoría de ellos.

Característica	**Mínimo**	**Recomendado**
Procesador	1 GHz armv7	1,8 GHz x86_64
RAM	500 MB SDRAM	2 GB DDR3
Almacenamiento	500 MB Flash (eMMC)	2 GB HDD
Sistema Operativo	Linux (Yocto)	Linux (distro estándar)

Tabla 4.7: Requisitos mínimos y recomendados para Eclipse Kura. Establecidos a partir de pruebas en equipos reales en el estudio [2].

Estos requisitos son lo bastante bajos como para que la gran mayoría de los productos industriales enfocados a tareas de Gateway IIoT sean capaces de

ejecutar Eclipse Kura. También los requisitos recomendados son sobrepasados por muchas soluciones comerciales. Esto permite centrarse en otros aspectos como puede ser el consumo, el factor de forma, la resistencia o la comunicación local. Este último es habitualmente la principal restricción a la hora de seleccionar el equipo que se usará como gateway IIoT. Una necesidad muy habitual es la comunicación serie, muy extendida en la automatización industrial. En el ámbito de IIoT, conexiones por WiFi, BLE o LoRaWAN también son muy comunes. Cada vez hay más productos que incorporan estas capacidades de forma nativa, ampliando aún más las posibilidades y mejorando la adaptabilidad en esta capa.

Al necesitar pocos recursos, Eclipse Kura puede acceder a hardware que otras soluciones pueden incorporar. Por ejemplo la necesidad de ejecutar Microsoft Windows o de disponer de mucho espacio de almacenamiento, muy común en muchas aplicaciones industriales, fuerza a usar equipos comparables a servidores industriales donde las opciones de integración inalámbricas rara vez se incorporan.

Nótese que no se ha entrado al detalle de la integración con la plataforma, la capa edge y otros sistemas al mismo nivel, que también es necesaria. Esto se debe a que no supone una restricción relevante: estas conexiones suelen ser a través de TCP o UDP y por tanto son bastante homogéneas a nivel de hardware y entra dentro de las capacidades de cualquier gateway IIoT que se seleccione. Incluso si es necesario incorporar comunicación celular, se trata de un caso de uso muy común y existen muchos productos en el mercado que cubren los requisitos con creces e incorporan comunicación inalámbrica.

La capa de plataforma se desplegará en un CPD. En este sentido Eclipse Kapua ya está preparado para correr de forma distribuida y desplegarse mediante Docker (necesaria versión 1.2 o superior) u OpenShift (versión 1.4.1 o superior). Gracias a esta facilidad, solo es necesario considerar los recursos de hardware a asignar. Dado que Kapua es una plataforma preparada para desplegarse On-Premise, estos requisitos se pueden escalar para correr con unos recursos muy

básicos y luego aumentar conforme crezca su uso. Esto le permite correr desde equipos de muy bajas prestaciones para un servidor (desde 4 GB RAM, CPU de 1,8 GHz y 5 GB de almacenamiento según pruebas empíricas).

En líneas generales, las necesidades de hardware en toda la arquitectura son muy bajas. Esto dará facilidades al implantar la solución, ya que será posible elegir de una gama más amplia de equipos y por tanto mejora la flexibilidad de la arquitectura.

4.3.3. Comunicaciones

Una vez seleccionadas las tecnologías software y hardware a emplear, se aborda la cuestión de las comunicaciones. Aunque es necesario estar abierto a las necesidades de los casos de uso, se valorará la familia de tecnologías más adecuada para comunicación local y remota y los protocolos que mejor se adaptan a las funcionalidades descritas para la interacción entre las capas de la arquitectura IIoT.

En lo relativo a la comunicación local, se valoran por igual tanto las comunicaciones cableadas como inalámbricas y dentro de estas últimas sería aceptable cualquiera que cumpla los niveles de seguridad requeridos. De las primeras se destaca claramente la comunicación Ethernet por ser preponderante en la industria. Otras como la comunicación serie (RS-485 y RS-232 principalmente) o bus CAN están también muy presentes y hay que tenerlas en cuenta al seleccionar el hardware. Entre las inalámbricas, la comunicación WiFi es la más extendida y facilita la comunicación con todo tiop de equipos. También son destacables la presencia de Bluetooth (en particular BLE), por la capacidad de comunicación en bajo consumo y la tecnología LoRa, que proporciona capacidades también de bajo consumo con mayor penetración que BLE.

La comunicación remota, se propone principalmente por red móvil, opción

dominante en la industria a falta de conexión por fibra óptica o Ethernet. Otras tecnologías como Sigfox o NB-IoT pueden proporcionar una comunicación con mejor compromiso de consumo pero se contemplan solo como métodos de refuerzo o caídas de la comunicación principal. La razón de esta preferencia es el volumen de datos a transmitir. Aunque se puede optimizar la comunicación para minimizar los datos conectados con la nube, las redes móviles permiten también ajustar este volumen y consumo en función de la categoría seleccionada en la conexión y en caso de ser necesaria una batería en un entorno industrial tiene más sentido recurrir a tecnologías locales de largo alcance como LoRa antes que NB-IoT.

En relación a los protocolos de comunicación, la selección de software marca ya una pauta clara en pro de MQTT. Este protocolo es con diferencia el principal en todas las arquitecturas IIoT y se ha establecido como un estándar de facto (posteriormente formalizado a través de la ISO/IEC-20922 que fijo la versión 3.1.1 como referencia para encolado de mensajes [99]). El uso de MQTT garantiza la conectividad de la solución con futuras soluciones IIoT y también representa el primer punto de integración para soluciones de terceros. Su estructura de publicación suscripción lo hace ideal para futuras integraciones mientras que la ligereza del protocolo ofrece buenas prestaciones para soluciones de bajo consumo.

Dentro de la comunicación por MQTT, se plantea la utilización de un protocolo para la gestión de los dispositivos. Este protocolo ya se ha presentado brevemente con la selección de Kapua y Kura pero no es solo un protocolo desarrollado para estas herramientas, sino que está basado en el estándar de gestión de dispositivos Sparkplug [100]. Sparkplug define una familia de mensajes, *topics* y funcionalidades que tanto dispositivo como plataforma deben respetar para realizar una gestión y telemetría completa. Es un protocolo impulsado por la fundación Eclipse que ya ha sido ampliamente implantado por empresas grandes de la comunicación MQTT como HiveMQ. Se destaca la adhesión a este protocolo como referencia porque, si bien el protocolo nace

precisamente de la aproximación que usa Kapua, la propia plataforma no se ciñe a la última versión del mismo. En caso de haber alguna diferencia relevante se opta antes por mantener el protocolo estándar aunque sea necesario realizar alguna modificación en Kapua para así disponer de una mejor integración y mayor versatilidad de cara al futuro. Además, tanto Kura como Kapua contemplan plena integración por Sparkplug en su hoja de ruta aunque aún no lo tengan plenamente incorporado.

4.4. Solución propuesta

Con las tecnologías seleccionadas, es posible aterrizar la solución IIoT que se propone para validar la hipótesis y afrontar los retos actuales en la industria. Esta solución está compuesta de tres capas: percepción, edge y plataforma. Estas capas se integrarán con la red industrial según su nivel de abstracción, enlazando las capas superiores de la pirámide de automatización (MES y ERP) con la plataforma, el SCADA con el edge y los PLCs y sensores con la percepción.

A nivel tecnológico, en la capa superior se despliega Eclipse Kapua en una única instancia centralizada, con opción a desplegarse on-premise en el interior de la planta industrial. Se acompaña también de otros servicios en caso de ser necesario.

En la capa edge se disponen gateways IIoT industriales con Eclipse Kura, que actuan como nodo principal de la arquitectura IIoT, redirigiendo datos, facilitando la integración y optimizando la gestión de la red industrial. En principio cada planta industrial cuenta con al menos un gateway IIoT, lo cual es suficiente para muchos de los casos de uso.

En la capa de percepción se consideran los dispositivos industriales ya presentes y otros que se requiera añadir como parte de la arquitectura. También se admite la opción de desarrollo de soluciones hardware a medida sobre

FreeRTOS para aquellos casos donde sea necesario para mejorar la integración o conectividad de un sistema.

A nivel de comunicaciones se prioriza el uso de red celular para larga distancia y ethernet para conexión con equipos industriales. Para las soluciones a medida basadas en FreeRTOS se conserva una aproximación abierta que pueda adaptarse a las necesidades del caso de uso si es necesario, ya sea por conexión inalámbrica o con acceso directo a la plataforma o a través de la capa edge. También se dará prioridad a la conexión MQTT con la capa de plataforma, siendo un protocolo ligero y robusto sobre el que también se establecerá la vía de gestión de los equipos remotos.

En la Figura 4.3 se puede ver un resumen de las tecnologías implicadas construido sobre el análisis de los requisitos de la Sección 4.2.

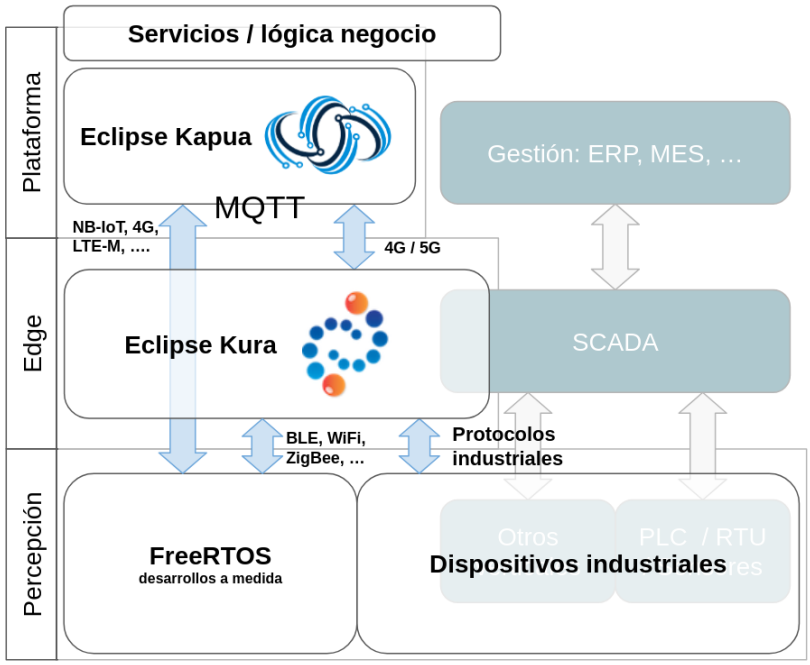


Figura 4.3. Stack de tecnologías de la solución propuesta. La permeabilidad entre las capas permite conectar directamente un equipo del nivel de percepción con la plataforma IIoT para los casos de uso que lo necesiten.

En el próximo capítulo se pone a prueba esta solución en casos de uso diversos. De esta forma se valida que la solución realmente atiende a los requisitos identificados en este capítulo.

Capítulo 5

Validación de soluciones

A partir de las características y tecnologías descritas en el Capítulo 4, se proponen varios casos de uso en los que poner a prueba las posibilidades de una solución IIoT en un entorno industrial real, y validar a la vez las dos hipótesis planteadas en Sección 1.3. Estos casos de uso reproducen varios retos dentro de los sistemas de gestión energética desde la supervisión completa de una planta de generación hasta la integración en un escenario domótico y son ejemplos de aplicaciones concretas que pueden ser mejoradas por el uso de tecnologías IIoT.

Para esta validación hay que destacar la posición de la empresa TSK Electrónica y Electricidad S.A. en el mercado, que ha permitido implantar en entornos productivos reales la solución propuesta.

TSK es una sociedad de prestación de servicios de ingeniería a las empresas industriales de distintos sectores de actividad. Está orientada a la realización de proyectos completos abarcando los campos de Equipamiento, Automatización e Instrumentación, Instalaciones y Montajes Industriales, Fabricaciones Electromecánicas, Mantenimiento Especializado e Innovación Digital. La empresa cuenta con una contrastada experiencia en el desarrollo de proyectos en los sectores de energía, telecomunicaciones, siderurgia, metalurgia, alimentación, papel, petroquímicas, cemento, medio ambiente, fertilizantes, puertos y plantas

industriales en general.

En la actualidad, TSK es la primera ingeniería de Asturias y uno de los diez grandes grupos españoles de ingeniería, con experiencia internacional en más de 50 países repartidos por todo el mundo. El grupo TSK ha tenido un crecimiento exponencial en los últimos años en cuanto a su volumen de ventas y también, respecto a las ventas por empleado, especialmente teniendo en cuenta que el número de empleados no ha dejado de crecer. TSK ha ejecutado proyectos en más de 50 países, adaptándose a las particularidades técnicas y culturales de cada uno, herramienta fundamental para culminar con éxito todos los proyectos internacionales.

Las soluciones que se plantean en esos escenarios están basadas en sistemas de alta calidad y fiabilidad en línea con el valor de excelencia técnica de la empresa. Sin embargo, se ha detectado que estos sistemas de supervisión y control pueden quedar obsoletos en los próximos años, debido a la tecnología emergente. Es por eso que, en la línea estratégica de TSK, la apuesta por el I+D+i es fundamental para mantener los proyectos siempre a la vanguardia. Esta perspectiva de innovación es la base que ha permitido la aplicación de una arquitectura IIoT en los casos de uso que se presentan en este capítulo. Dentro de esta apuesta por la innovación, se desataca el desarrollo de la plataforma *SIS*.

La plataforma SIS es la pieza central de todas las aplicaciones presentadas en este capítulo. Antes de describir los casos de uso se detalla el funcionamiento de esta plataforma. También se detalla como se ha desarrollado una arquitectura IIoT siguiendo los principios identificados en el Capítulo 4 sobre la misma.

Luego, para cada caso de uso, se presenta un análisis general del problema a abordar, seguido de un detalle del caso de uso concreto y problemas a resolver. A partir de ambas, se elabora la solución atendiendo a las particularidades encontradas y se exponen los puntos críticos que se han encontrado. Finalmente, se valora el resultado y beneficios obtenidos con el uso de la arquitectura IIoT.

5.1. Plataforma SIS

SIS es una plataforma desarrollada por TSK como vía para aunar los esfuerzos de desarrollo de Industria 4.0 de la empresa. Al mismo tiempo es el producto final que se incorpora transversalmente a servicios entregados desde la empresa, como la monitorización de las plantas construidas. Aunque la base de la plataforma es la visualización y control a través de un entorno web, a día de hoy SIS dispone de módulos y herramientas avanzados entre los que se encuentra la capacidad de análisis Big Data, aplicaciones de realidad virtual o integración con drones u otros sistemas robóticos.

Las tres patas fundamentales de la plataforma serían la infraestructura de procesamiento de datos, la de lógica de negocio y la de visualización web:

El procesamiento de datos se realiza en una arquitectura Big Data construida principalmente sobre Apache Flink capaz de ejecutar algoritmos tanto en tiempo real como en *batch*. La captación de datos para esta arquitectura se realiza a través de Apache Kafka, que sirve a su vez como medio para conectar internamente los procesamientos.

La lógica de negocio está implementada sobre una capa de microservicios que interactúan con las herramientas desplegadas en SIS y entre ellos para formar el backend de toda la plataforma. Los microservicios de comunican entre ellos mediante peticiones REST.

La visualización web sirve junto con otros medios para la monitorización y control de las soluciones conectadas y es el nexo principal para la gestión de todo SIS. Está construido sobre una arquitectura de microfrontales que facilita la creación de vistas personalizadas y soluciones adaptadas.

Sobre esta infraestructura, se desarrolla toda la arquitectura IIoT conforme a los principios presentados en el Capítulo 4. La base de la misma está igualmente

basada en las tres capas propuestas atendiendo a las particularidades de SIS y de las soluciones desplegadas. En la capa de plataforma se sitúa la integración entre SIS y los equipos de las capas edge y percepción, gobernada principalmente por Eclipse Kapua ampliada con lógica de negocio de SIS y un portal web. En la capa edge se encuentra el software de Eclipse Kura, cuyo flujo de trabajo se presenta en más detalle en cada aplicación aunque siempre partiendo de algunas configuraciones básicas. Por último está la capa de percepción, sobre la cual no se puede detallar una versión generalizada ya que depende mucho más del tipo de aplicación y muchas veces se cubre con los equipos preexistentes en la planta. Se puede ver en la Figura 5.1 un esquema general de la infraestructura constituida entre SIS y la arquitectura IIoT.

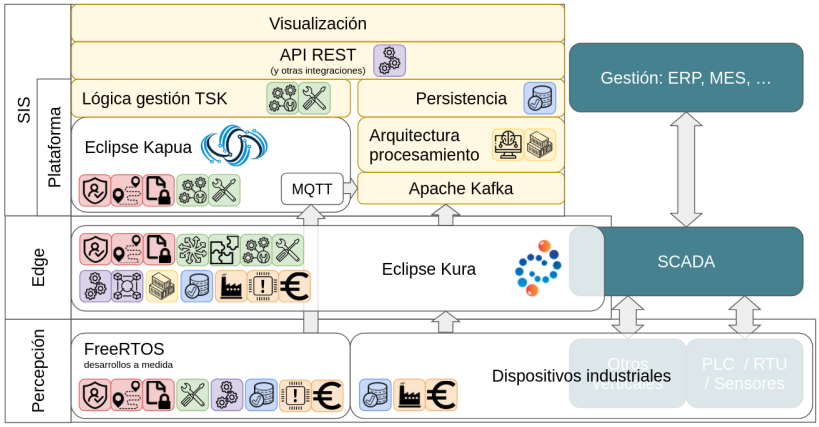


Figura 5.1. Esquema general de la plataforma SIS con la arquitectura IIoT construida. Algunos elementos de la arquitectura IIoT se alojan en SIS para cubrir requisitos de la capa de plataforma.

5.1.1. Plataforma IIoT en SIS

La capa de plataforma IIoT desarrollada en SIS parte de la tecnología elegida en el capítulo anterior, Eclipse Kapua. Es la pieza central ya que permite gestionar el ciclo de vida completo de cualquier equipo, dar de alta, actualizar firmware y configuración y además servir como medio de entrada de datos. Sin embargo,

Kapua no resulta eficiente en la cobertura de todos los requisitos. Para mejorar su utilidad, se aprovecha la infraestructura de SIS para construir desarrollos adjuntos a Kapua que complementan sus debilidades y cubran parcial o totalmente los requisitos, tal como se puede ver en la Figura 5.1:

- Ingesta y procesamiento de datos: para mejorar la capacidad de procesamiento y garantizar la persistencia de los datos recogidos.
- Lógica de negocio: para automatizar y robustecer las tareas conforme a las necesidades de SIS y TSK.
- Visualización: para facilitar la gestión de los dispositivos, en especial a los usuarios finales.

El conjunto de estos servicios interrelacionados constituye la capa de plataforma IIoT desplegada. A continuación se desarrolla la interacción entre Kapua y estos tres elementos con los que se integra para expandir su funcionalidad.

La ingestión de datos directa a Kafka se hace utilizando certificados firmados por TSK para validar la identidad del equipo conectado y se realiza la conexión a un topic específico para cada organización y tipo de dato a integrar. Directamente a esta entrada de datos, se conecta el análisis Big Data de SIS que incluye desde cálculos aritméticos sencillos hasta índices y correlaciones complejas. La conexión a Kafka o mediante certificados es asequible para un equipo medio y una comunicación por red móvil, pero resulta muy pesada para algunas tecnologías de comunicación como NB-IoT o compleja para equipos de menores capacidades como los microcontroladores. Es por esto que se mantiene también el camino alternativo que supone Kapua (basado en un broker MQTT), aunque se conecta internamente a Kafka y la arquitectura de procesamiento.

Como se comentó en el Capítulo 4, Kapua puede ser ampliada de forma modular para modificar o ampliar su funcionamiento. En principio esto permite

incorporar la lógica de negocio necesaria. Sin embargo, por la estructura ya existente en SIS, se ha optado por recurrir a su capa de microservicios REST permitiendo el despliegue de nueva funcionalidad de forma integrada dentro de SIS. Para lograr todos estos beneficios, se han implementado tres servicios interrelacionados:

- Recursos: que controla toda información adicional necesaria para la gestión de los equipos. Es el caso por ejemplo del catálogo de plugins disponibles desde TSK para instalar en la capa edge o las plantillas para configuraciones según la aplicación.

- Plataforma: actor intermediario entre el modelo de datos de Kapua y los de SIS. También sirve de traductor de métodos para instrucciones de cierta complejidad. Este microservicio permite aislar las características específicas de Kapua para una migración más sencilla a otra plataforma en caso de ser necesario.

- Inventario: destinado a las tareas más generales de la gestión o el mantenimiento de la arquitectura IIoT como actualizar los equipos periódicamente o coordinar el proceso de alta.

Sobre estos servicios se despliega también un microfrontal web. Esta visualización está pensada para facilitar las tareas de supervisión de la propia arquitectura IIoT y la gestión remota. De esta manera, se dispone por un lado de un entorno de trabajo pensado para acciones globales que no distinguen de un equipo a otro (como la actualización de certificados) y por otro se presenta el detalle de la lógica y estado de cada dispositivo (útil para ajustes de la configuración).

La combinación de Kapua y SIS constituye la capa de plataforma de la arquitectura, dotándola de funcionalidades avanzadas y muy bien adaptadas a

las necesidades de TSK, al tiempo que se amplían las opciones de integración. Algunas de estas características fruto de la unión de ambas plataformas son el alta de dispositivos, el tipado de los equipos, las configuraciones simplificadas o unificación de permisos. Estas y otras adaptaciones, que son naturales en cualquier integración, se pueden superar gracias a la conexión por API REST con la plataforma. Como toda la funcionalidad de Kapua está disponible por este protocolo (hasta el punto de que algunas opciones no están disponibles en la web pero sí por esta vía), los microservicios automatizan todos los mecanismos y lógica necesaria para que SIS y Kapua actúen como una unidad frente a los dispositivos por un lado y frente al usuario por otro. Se ahonda a continuación en las adaptaciones más relevantes que se llevaron a cabo.

Alta de dispositivos La plataforma Eclipse Kapua permite el alta de un dispositivo automática por su propia conexión con ella. Sin embargo, esto no es deseable en una arquitectura centralizada y con un fuerte compromiso con la ciberseguridad como la de SIS. Además, para integrarse con el resto de la plataforma es necesario asociarle una organización y que algún usuario pueda realmente acceder a ese equipo.

Para hacer todo esto, el proceso de alta realmente se desencadena desde la actuación de un usuario que elige registrar un dispositivo. La información mínima a proporcionar para eso es un identificador que vaya a usar el dispositivo, unas credenciales y unos permisos de acceso. Con esto, SIS registra en Kapua un usuario específico para el dispositivo. Como el propio usuario ya está ligado a una organización concreta, se puede saber automáticamente a qué organización asociarlo y se incorporan los permisos introducidos por el usuario.

Una vez realizado este alta, el equipo ya es capaz de conectarse a la plataforma y por tanto acceder a SIS. Destacar que en esta primera conexión se realizan algunos métodos remotos sobre el equipo, principalmente modificar la contraseña

por una generada aleatoriamente y dar al equipo los metadatos necesarios para que los datos enviados sean válidos y relevantes. Dado que ya se ha establecido un canal de comunicación encriptado punto a punto mediante SSL, el intercambio de información ya es seguro.

Tipo de dispositivos Aunque la mayor parte de la gestión de los equipos es común y sobre todo el protocolo sobre el que se hace está unificado, en SIS se contemplan varios tipos de dispositivos bien por la lógica que implementan, bien por el hardware sobre el que se ejecuta. Por ejemplo, ya se ha comentado la diferencia de envío desde un microcontrolador que debe enviar los datos a Kapua en contraste a un equipo que está conectado a Kafka.

Para tratar correctamente estas diferencias, se realizaron sobre SIS adaptaciones que permiten gestionar los dispositivos según el tipo detectado al conectarse y por tanto dar una vista de gestión distinta según el caso. En las Figuras 5.2 y 5.3 se presentan dos ejemplos de vistas de gestión distintas según el tipo de dispositivo. Si bien el protocolo y la gestión a bajo nivel se hace de la misma manera, resulta más intuitivo para un usuario apreciar las diferencias en la visualización para comprender mejor el proceso implicado y su configuración.

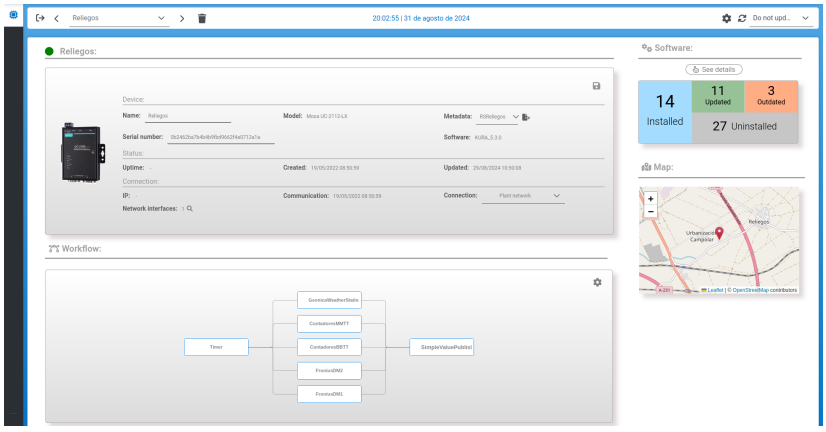


Figura 5.2. Visualización en SIS para la gestión de un equipo edge. En este equipo se gestionan flujos de trabajo avanzados.

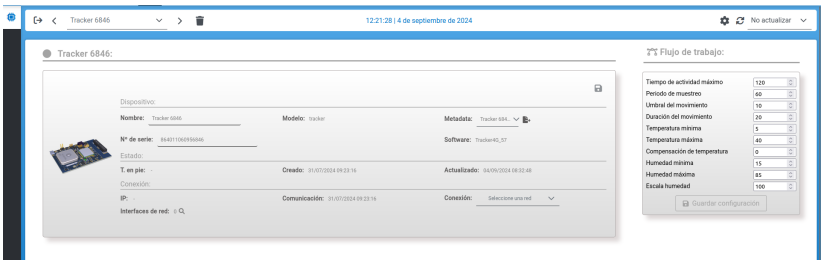


Figura 5.3. Visualización en SIS para la gestión de un equipo de percepción. Este tipo de equipos solo permite configurar parámetros puntuales.

Configuraciones simplificadas A pesar de la versatilidad de configuración de la solución IIoT establecida, hay mucha configuración común o similar entre las distintas aplicaciones. En parte porque la mayoría tienen muchas necesidades en común, como puede ser con la configuración de la hora de los equipos o la propia conexión a Kapua; y en parte porque se busca cierta uniformidad dentro de las soluciones desplegadas en SIS para facilitar el mantenimiento, como puede ser el uso de unidades de medida comunes o un convenio de nombres para las variables leídas.

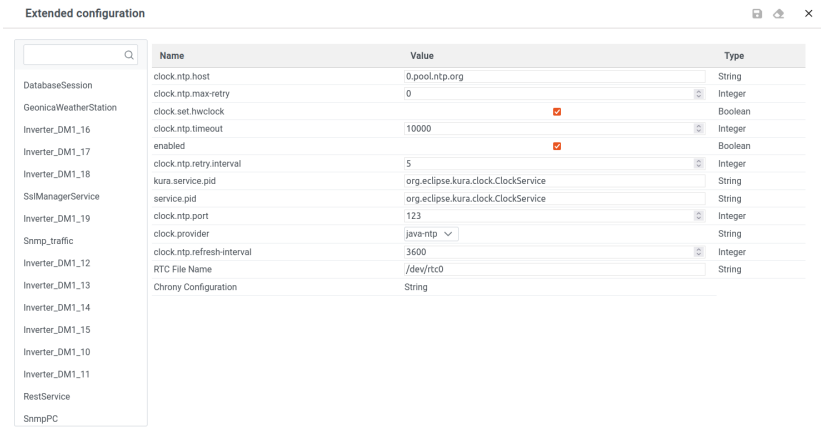


Figura 5.4. Configuración completa de un equipo edge. Se presentan todos los servicios que están realmente usándose en el equipo aunque muchos no tienen relevancia específica para el usuario.

Partiendo de estas restricciones, es posible crear configuraciones solicitando al

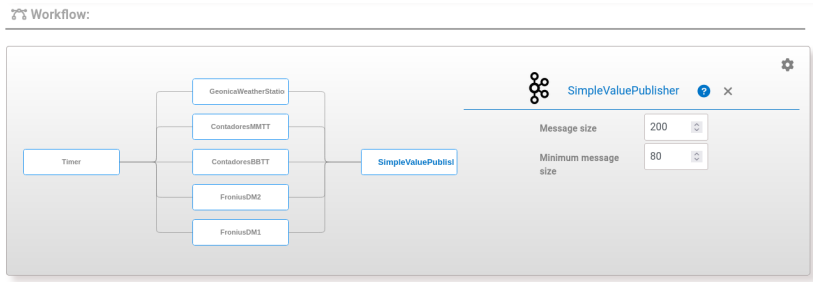


Figura 5.5. Configuración simplificada de un equipo edge. Se presentan solo los servicios más habituales que va a usar un usuario y los parámetros más relevantes.

usuario solo información relevante para su caso de uso o a la inversa presentar al usuario solo aquello que realmente le influye. En las Figuras 5.4 y 5.5 se pueden ver ejemplos de la configuración completa de un equipo y de la versión simplificada que realmente tiene que tocar un usuario en su día a día. Así se facilita el trabajo y también se reducen las posibilidades de introducir un fallo, mejorando la robustez de la aplicación.

Unificación de permisos Kapua dispone de una gestión de permisos muy granular para configurar con mucho detalle lo que un usuario puede y no puede hacer. Sin embargo, este nivel de detalle no es directamente comparable con el de SIS. Así, por ejemplo, Kapua distingue en permisos distintos la conexión de un equipo y el envío de datos desde ese equipo. En SIS esta distinción no tiene sentido. Sin embargo, Kapua no permite distinguir la creación de un usuario personal y la creación de un usuario de dispositivo. En SIS es necesario que el alta vaya asociado de un nuevo usuario para el equipo pero no hace falta ser administrador de la plataforma para dar de alta un equipo, basta con ser un operario mantenedor.

5.1.2. Capa edge en SIS

Eclipse Kura es un software concebido para ejecutar servicios conectados entre sí y admitir funcionalidad de forma modular. Esta característica permite utilizar un conjunto limitado de piezas o módulos y construir con ellas flujos de trabajo muy variados. Para poder comprender mejor las implementaciones descritas en el Capítulo 5 es necesario conocer los módulos más empleados y el flujo de trabajo básico.

Los módulos más habituales y disponibles en su mayoría de forma nativa en el software permiten no solo gestionar el propio software, sino también el hardware, otras aplicaciones del equipo y también agilizar el desarrollo de futuros módulos. En esta tesis se han identificado e implementado los módulos básicos con el objetivo de disponer de un biblioteca de plugins que permita afrontar cualquier caso de uso. Debido a esto, fue necesario construir algunos de cero o partiendo de otros similares existentes. Algunos de estos servicios básicos, transversales a cualquier funcionalidad o aplicación que se quiera utilizar se presentan a continuación (entre paréntesis se detalla si son propios de Kura o desarrollados desde TSK):

- Sincronización de hora (Kura): habilitado por servidor NTP y/o por hardware para sincronizar la hora del sistema.

- Variables de proceso (TSK): para proporcionar un espacio de persistencia a valores importantes para el proceso, como un cálculo intermedio o un identificador. De esta manera los servicios no tienen que acceder a la base de datos para cada parámetro y se pueden cachear a través de este servicio.

- Geoposicionamiento (Kura): a través de un módulo GNSS disponible por hardware, puede integrarse para recopilar la información y entregarla internamente para cualquier otra aplicación.

- Red (Kura): control de los interfaces de red, módems o conexiones WiFi disponibles en el dispositivo.
- Despliegue de contenedores Docker (Kura): como medio para ampliar la funcionalidad más allá del propio alcance de Eclipse Kura, este servicio ayuda a integrar cualquier otra aplicación.
- Gestión de certificados (Kura): tanto crear, importar, actualizar o borrarlos, así como servirlos internamente para que cualquier otro módulo pueda acceder a esta información sin incorporar la complejidad de interactuar con los almacenes de certificados y su encriptación.
- API REST (Kura): que sirve toda la funcionalidad disponible a través de la web para una automatización eficaz. Es además ampliable para mantenerse al día de nueva funcionalidad.
- Credenciales (Kura): tanto para el portal web como la API REST es posible gestionar usuarios y sus permisos.

Destacar que no es una lista sistemática ya que hay bastantes más (para gestión de logs, identificación del equipo, configuración, gestión de paquetes, etc.) que en su conjunto permiten controlar el equipo plenamente desde el software. De esta manera, Eclipse Kura convierte al propio equipo en un gemelo digital de sí mismo para poder manipularlo desde la plataforma. A partir de aquí, la integración del equipo queda habilitada para cualquier aplicación interna o externa.

Internamente, toda la información y servicios descritos está disponible para cualquier módulo que se quiera desarrollar de forma directa sin que cada servicio deba, por ejemplo acceder a un módulo GNSS para obtener la posición, sino simplemente solicitarla al servicio de posicionamiento. Externamente la información está disponible bajo la autenticación adecuada para cualquier otro

servicio a través de la API REST o cualquier otro protocolo que un usuario desee implementar. En particular, Eclipse Kura ofrece esta capacidad de gestión absoluta también a través de un broker MQTT contra el que esté conectado.

Una vez estos servicios proporcionan la capa básica de funcionalidad, el flujo de trabajo puede enfocarse en lo estrictamente necesario para el proceso. En este caso se presenta un flujo de trabajo básico típico. Este se compondría de:

1. Lectura programada de dispositivos de campo o adquisición de información por suscripción a eventos

2. Transformación de la información. Según el caso puede ser una transformación simple como una agregación o una derivada respecto al tiempo, más compleja como puede ser el procesamiento de imágenes, o avanzada como análisis mediante un modelo de redes neuronales.

3. Almacenamiento de la información, aun cuando se logren enviar las variables exitosamente a la plataforma, es útil para el proceso conservar siempre una imagen de la información leída en cada momento para facilitar la integración y para facilitar un análisis local

4. Envío a la plataforma para una persistencia más garante y facilitar integraciones y análisis más avanzados de los que es capaz la capa edge.

Además, es necesario conocer en todo momento el estado del propio dispositivo, para prevenir posibles incidencias y ataques. Además de controlar los parámetros básicos del equipo, significa monitorizar la adquisición de datos para confirmar que todos los equipos a los que el equipo se conecta siguen operativos y enviando datos en un momento dado.

Para complementar este flujo de trabajo, se dispone de una amplia familia de módulos sobre Eclipse Kura que ayudan a adaptarse a cada aplicación afrontada. Algunos de estos módulos están proporcionados con la instalación básica, otros se

pueden ampliar usando la plataforma de aprovisionamiento de plugins de Eclipse (Eclipse Marketplace), y otros han sido desarrollados para los distintos casos de uso de TSK de manera que enriquezcan las funcionalidades disponibles. Estos componentes son:

- Temporizador (Kura): incluido con la versión básica, este componente permite programar el inicio de una tarea en el flujo de trabajo.
- Drivers (Kura): para leer de los dispositivos industriales conforme al protocolo concreto. Se concretará más adelante.
- Operador básico (TSK): para realizar operaciones básicas entre dos variables tales como suma, resta, multiplicación, etc. [1]
- Operador booleano (TSK): para realizar operaciones booleanas sobre una o dos variables.
- Operador evolutivo (TSK): para realizar operaciones en las que los valores anteriores influyen en el resultado actual, tales como una derivada o una integral.
- Operador de ventana (TSK): para realizar operaciones que requieren de un conjunto de valores para cada variable, tales como la media o una desviación típica.
- Filtro (Kura): para eliminar valores que resulten claramente aberrantes y que podrían distorsionar procesamientos posteriores.
- Unión (Kura): para unir dos grupos de variables en uno solo. Útil para sincronizar dos lecturas independientes de cara a un cálculo posterior.

[1] Destacar que este y todos los módulos de transformación son parametrizables para aplicarse no solo sobre una variable determinada, sino contra un conjunto de variables, de forma que, por ejemplo, una etapa de cálculo de integral pueda aplicarse para obtener una energía a partir de una potencia y un volumen a partir de un caudal.

- Mapeo (TSK): conversor de información entre dos dominios. Muy común para pasar del dominio de la planta al de la plataforma modificando el formato del valor o el identificador asociado.

- Agregación temporal (TSK): para agrupar los valores de acuerdo a un patrón de tiempo tales como cada minuto o cada hora.

- Gestión de eventos (TSK): para descartar valores que no han variado con respecto a una medida anterior y por tanto pueden descartarse si se busca optimizar el almacenamiento.

- Base de datos (Kura): para guardar la información obtenida en una base de datos local. Muy eficiente para valores numéricos o booleanos.

- Sistema de archivos (TSK): para guardar la información en el sistema de archivos local. Más eficiente que la base de datos para cierto tipo de datos como imágenes, vídeos o configuraciones grandes.

- Buffer de datos (TSK): sistema de persistencia temporal para que los distintos métodos de envío puedan garantizar que no se pierden datos a pesar de que el servidor remoto no responda.

- Conexión MQTT bruto (TSK): también práctico para integración local. Componente para enviar información a un broker MQTT sin un formato predefinido de manera que pueda aceptar las transformaciones anteriores.

- Conexión Kafka (TSK): publicación de datos al servidor Apache Kafka, muy común en integraciones tanto de IT como IIoT y utilizado para gestionar el flujo de datos principal de TSK. En adelante se asume que el envío de datos se hace siempre a Kafka aunque pueda utilizarse también la plataforma IIoT en paralelo para gestionar la arquitectura o para integrar algún dato en concreto.

- Conexión MQTT Eclipse Kapua (Kura): específico para la integración con Eclipse Kapua, permitiendo una robustez de la conexión mayor y habilitar la gestión remota con la plataforma.

- Notificación de eventos (TSK): recogida y notificación de eventos en el propio gateway a través de los métodos de envío que se elijan.

Como se puede ver, la gama de procesamientos es sin duda la más amplia ya que es donde más se puede aprovechar la modularidad del framework para dar lugar a configuraciones de proceso muy variadas. Al igual que sucede con los servicios básicos, se pueden añadir a estos módulos otros desde el Marketplace (como la analítica con modelos preentrenados o cálculos de contabilidad) o de desarrollos propios.

Dentro de la colección de módulos disponibles, la mayoría son relativos a la integración industrial. La colección de drivers disponibles para Kura es muy extensa. Solo en el Marketplace oficial están disponibles por ejemplo Modbus, OPC UA, S7, Ethernet/IP, BACnet, Fanuc, iBeacon, Eddystone, MBUS o DNP3 entre otros. También se pueden encontrar otros entre proyectos Open Source de terceros. Aunque la parte propia de los protocolos (canal físico de comunicación, procedimiento pregunta/respuesta, credenciales, etc.) es específico de cada driver de comunicación, Eclipse Kura proporciona una capa de funcionalidad común en toda lectura de datos.

Esta funcionalidad incluye lógica tanto para lecturas desencadenadas desde el propio equipo como para aquellas en las que el equipo activa una escucha y atiende a los eventos producidos por el dispositivo a monitorizar. También admite la configuración variable a variable del procedimiento de lectura, tipo de dato, escalado y *offset*, además de las variables necesarias para cada protocolo. Así, un desarrollador no necesita ahondar en las partes comunes para cualquier proceso de lectura, sino aprovechar esta funcionalidad para incorporar solo lo relativo al

protocolo a implementar.

5.1.3. Capa de percepción en SIS

La capa de percepción planteada para SIS se compone principalmente de dispositivos industriales. Las necesidades de fiabilidad y condiciones ambientales apuntan a estos dispositivos como idóneos para la digitalización de los datos frente a otros desarrollos electrónicos o equipos de bajo coste.

No obstante, se valora el uso de desarrollos ad hoc basados en FreeRTOS como medio para ampliar la funcionalidad de la arquitectura bajo demanda. Además de por flexibilidad, estos productos propios pueden ser importantes para situaciones que requieran bajo coste o un despliegue muy extenso de equipos y por tanto sea más efectivo una adaptación a medida. También pueden emplearse para garantizar una mayor seguridad en el origen de los datos o mejorar la integración de un sistema.

5.2. Supervisión de plantas termosolares

5.2.1. Descripción

Las plantas termosolares, en términos simples, utilizan el sol para generar vapor a presión que ponga en marcha un generador eléctrico mediante una turbina. Este tipo de plantas son bastante complejas y requieren una inversión inicial alta, pero proporciona una producción más estable y eficiente que otras alternativas renovables basadas en la radiación solar como pueden ser las fotovoltaicas. Se pueden implantar distintas topologías según las tecnologías implicadas y la estabilidad y eficiencia que se busque alcanzar. A continuación se detallan algunos elementos habituales que componen una planta termosolar:

- Campo solar: área en la que se captura la radiación solar para calentar un fluido térmico. Habitualmente se compone de una serie de tuberías por las

que circula el fluido expuesto a la radiación solar que se concentra con unas superficies reflectantes cilindro-parabólicas. En este punto es donde más se distinguirían otros tipos de plantas termosolares, ya que capturarían el calor sobre el fluido térmico mediante otra distribución.

- Tanque de sales: para almacenar la energía calorífica y poder estabilizar la producción durante todo el día. El tipo de sales y tamaño de los tanques influye en la estabilidad productiva de la planta.
- Circuito de fluido térmico: utilizado para calentar el circuito de vapor. Se calienta en el campo solar y estabiliza su temperatura en el tanque de sales antes de llegar al intercambiador de calor.
- Circuito de vapor: circuito de fluido que acciona la turbina. Se calienta en un intercambiador de calor con el circuito de fluido térmico, se descarga en la turbina y se enfría con el circuito de refrigeración antes de comprimirse para empezar el ciclo de nuevo.
- Turbina y generador: donde se genera la energía eléctrica a partir de la inyección de vapor.
- Circuito de refrigeración: enfría el circuito de vapor. Recupera su temperatura normal en una torre de refrigeración.

Algunas instalaciones tienen, además de estos elementos, una segunda etapa de calentamiento y accionamiento de turbina. Otras optimizan el circuito de refrigeración con recalentamiento, según las temperaturas de operación. Las variaciones sobre esta base son abundantes. Dado que la solución IIoT a aplicar no es específica para la gestión de este tipo de plantas, la composición en detalle no resulta tan significativa para el diseño de la misma.

Lo que sí hay que considerar es que las plantas termosolares son extensas en terreno, debido al campo solar, lo que influye en la construcción de la red

de comunicaciones. También tienen condiciones críticas en algunos puntos del proceso que exigen un control en tiempo real muy severo, sobre todo para evitar riesgos de fugas y en general riesgos laborales para los trabajadores. Una primera consecuencia es que el ritmo de monitorización y volumen de datos es muy alto.

La ventaja es que no es necesario mantener un diseño escalable pues no existen plantas termosolares *pequeñas* como sí puede ser el caso de una instalación fotovoltaica.

Las plantas termosolares están operadas por personal local aunque la mayor parte del proceso se podría operar de forma remota. Por un lado se hace necesario para atender emergencias y por otro es asumible al haber muchas tareas de mantenimiento que se deben ir haciendo a lo largo de toda la planta. Sin embargo, es común la adquisición remota para el análisis de la planta, centralizar información y producción con otras plantas, realizar operaciones a alto nivel, evaluar el rendimiento de la planta o planificar el mantenimiento de sistemas. En este caso se vuelve a destacar el gran volumen de datos a manejar que se remontan a capas superiores de la arquitectura.

De cara a procesar esta cantidad de datos es necesario por un lado realizar una criba en local para descartar, agregar o simplificar parte de la información que no sea necesario llevar a las capas superiores de la arquitectura. Por poner un ejemplo, la temperatura de cada parte del circuito podría no ser tan relevante como simplemente saber que no hay ninguna parte que rebase un umbral. Otro planteamiento posible es disponer solo de la media cada dos minutos de la temperatura en el tanque de sales, aun cuando el operario local sí pueda acceder a los datos con muestreo menor por razones de seguridad. Este procesamiento en el edge tendría un impacto clave en el rendimiento de la solución al simplificar los requisitos de comunicación y del procesamiento en la nube.

Por último, la interacción de varias categorías de personal (mantenimiento, producción, operación, incidencias, análisis, etc.) hace necesario una gestión

adecuada del acceso a la información para evitar filtración de información sensible de la planta que pueda comprometer tanto a la empresa como a la propia seguridad de la planta.

En resumen, se necesita que la solución final proporcione:

- Autorización: debido a las dimensiones de una planta termosolar y los riesgos de la maquinaria involucrada, no solo es importante por privacidad, sino también por seguridad laboral para los trabajadores. También es necesario distinguir el perfil de cada trabajador, a que información tiene acceso y qué parte del proceso puede controlar.

- Integración horizontal: a nivel de planta, es necesario que la arquitectura se pueda conectar con todos los activos industriales para recopilar la información.

- Integración vertical: a nivel de plataforma, es necesario poder extraer los datos para tareas de planificación y análisis.

- Integridad de los datos: de cara a los análisis debe garantizarse que los datos sean fiables.

- Procesamiento en el edge: deben desplegarse algoritmos sencillos para simplificar toda la información que se utiliza en planta para el control en tiempo real.

- Bajo consumo: dado que es una planta de generación de energía, el consumo eléctrico impacta de forma directa en la productividad de la planta.

- Rango industrial: los equipos desplegados en planta deben ser capaces de resistir las condiciones, sobre todo si se disponen equipos en el campo solar, que está muy expuesto a altas temperaturas, polvo y condiciones meteorológicas.

5.2.2. Caso de uso

La validación de esta aplicación se realizó mediante el despliegue en la planta de Abdali. La planta termosolar de Abdali está ubicada en Kuwait (Figura 5.6) y tiene una potencia instalada total de 50 MW, ocupando una extensión de terreno de más de 200 hectáreas. Es la primera instalación de este tipo en el país y es parte fundamental de un plan estratégico de inversiones en producción de energía renovable. En la Figura 5.7 se presenta una vista de pájaro de la planta.

Figura 5.6. Ubicación de la planta termosolar de Abdali (fuente [101]).

Para la gestión de la instalación, se dispuso un SCADA robusto de la empresa Siemens cuya implementación se basa en OPC UA. Este SCADA permite controlar toda la planta y asistir en tareas de mantenimiento. No obstante, el contrato con TSK no se limita a la construcción de la planta, sino que incluye también la operación y mantenimiento durante los seis primeros años. Este régimen no es extraño en proyectos llave en mano de instalaciones complejas donde se utilizan esos primeros años para hacer el traspaso de competencias

Figura 5.7. Vista de pájaro de la planta termosolar de Abdali (fuente [102]). Ocupa más de 200 hectáreas de terreno y tiene una capacidad de 50 MW.

y formación a la parte compradora. Para esta tarea, TSK se ha apoyado en personal desplazado con amplia experiencia junto con otros contratados localmente. Esto facilita de nuevo la eventual transferencia de empleados. Además de esta operativa local, se plantea el despliegue de una supervisión remota.

Con esto se busca en primer lugar verificar por partida doble que el funcionamiento ordinario de la planta fuera correcto, pero también validar algunas innovaciones desplegadas en la propia planta de Abdali en su concepción inicial, de forma que se puedan exportar a otras soluciones de TSK con mayor conocimiento de su aplicabilidad y limitaciones. También se propone como forma de probar otros modos de funcionamiento y poner a prueba escenarios novedosos. En definitiva la supervisión remota proporciona un apoyo al control normal de la planta al tiempo que aporta valor a la labor de investigación e innovación constante que hace TSK para mantenerse competitiva.

Como es natural, este uso de la supervisión remota requiere una gran

flexibilidad en la adquisición de datos y en la integración con terceras partes. La adquisición de datos para la operación normal es relativamente fácil de estipular, pero la selección singular de variables a adquirir para cada una de las pruebas que se han ido y se siguen desarrollando, requiere una estrategia de adquisición más versátil, capaz incluso de bajar al nivel de percepción en algún caso. En cuanto a la integración, dado que cada prueba puede requerir análisis completamente distintos, es necesario tener dispuestos medios de integración diversos o flexibilidad para desarrollar nuevos. Este modo de trabajo invalida la creación de una integración específica ya que sería necesario hacer muchas integraciones individuales y no reaprovechables. El uso de una arquitectura IIoT, sin embargo, permite funcionar como elemento centralizador (sobre todo en la capa edge) y al mismo tiempo facilita la integración (en la capa de plataforma). De esta manera, todos los desarrollos de integración pueden ser reusados.

Además de estas consideraciones a priori, tras implantar la solución IIoT de SIS, también se habilita la integración con un sistema de realidad virtual (RV) de la supervisión. Esta integración contribuye a mejorar la calidad de la formación y también sirve como medio de mostrar las capacidades de TSK a potenciales inversores y clientes. La realidad virtual, por su capacidad inmersiva, ayuda a comprender mejor el funcionamiento real y la disposición de una planta termosolar. Además da una idea más visual de la magnitud de un proyecto de manera complementaria a un informe o presentación tradicional.

5.2.3. Diseño e implantación

Dado que la planta termosolar ya dispone de un SCADA completo basado en un servidor OPC, la integración con la misma se puede realizar sin recurrir a la capa de percepción como medio de adquisición de datos. Toda la información relativa al control y rendimiento de la instalación ya se encuentra disponible en este servidor por el propio funcionamiento en local de la planta. Por tanto, el

diagrama de conexiones a nivel de planta es mucho más sencillo que en otras soluciones, como se muestra en la Figura 5.8.

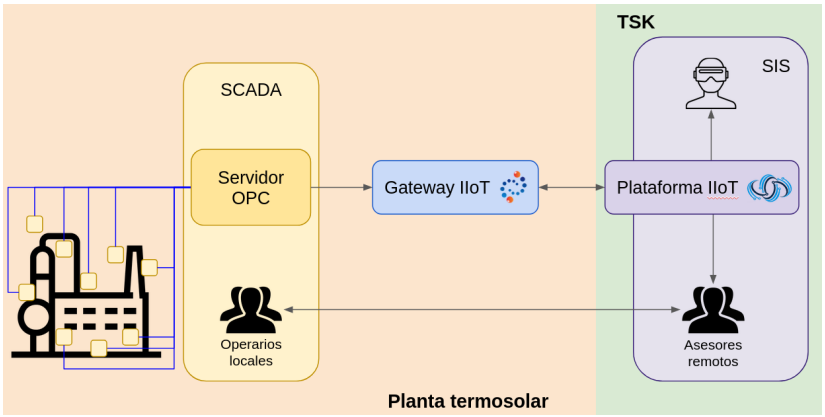


Figura 5.8. Diagrama funcional de equipos en la planta de Abdali. Además de proporcionar feedback al funcionamiento de la planta gracias a asesores, la arquitectura IIoT permite integrar nuevas herramientas a la solución.

El flujo de trabajo empleado en la capa edge se corresponde con el que se presentó en la Sección 5.1.2 en líneas generales con dos salvedades debidas al alto volumen de datos:

- Recuperación de históricos: se mantiene un flujo de trabajo independiente de envío de información en *batch* para los casos en que se acumulan envíos por falta de conexión. De no usarlo, se acumularía gran cantidad de datos en el equipo que tardarían mucho en ser extraídos. En caso de no haber información acumulada, este flujo de trabajo no impacta en el rendimiento.
- Agregación: aunque se leen datos cada segundo, en los análisis realizados hasta ahora no resulta de interés una frecuencia tan alta. Se implementa por tanto una agregación cada minuto de estos datos, que sí se envía.

Destacar también que aunque solo existe una conexión con el servidor OPC, se crean múltiples dispositivos virtuales a leer para proporcionar a la información una estructura más comprensible y facilitar el mantenimiento de la solución.

Con estas particularidades, el flujo de trabajo resultante sería el presentado en la Figura 5.9 que se compondría de:

1. Timer: lanza cada cinco segundos todo el flujo de trabajo.
2. Equipos: lectura de todos los equipos configurados. En la Figura 5.9 se omiten muchos de estos equipos para facilitar la visualización del flujo de trabajo en este documento.
3. Monitorización: sirve en primer lugar para unificar el flujo de trabajo de cara a futuras etapas y se aprovecha para controlar que todos los equipos están presentes.
4. Agregación: agrega los datos en ventanas de un minuto calculando su media. El resto del flujo de trabajo trabaja ya al ritmo de un minuto en vez de cada cinco segundos como hasta aquí.
5. Mapeo: conversión de los identificadores en Kura a unos universales.
6. Kafka: envío a la plataforma de procesamiento y visualización SIS por Apache Kafka.
7. Timer recuperación: conectado directamente al módulo de Kafka. Fuerza cada 10 segundos a enviar parte de las variables que se hayan almacenado por un fallo de conexión.
8. Kafka recuperación: envío a la plataforma de procesamiento y visualización SIS por Apache Kafka con un tamaño de datos superior al normal.

Aunque en la capa edge este procesamiento no tiene requisitos particularmente altos, sí es necesario atender atender a la política para cualquier equipo conectado a la red industrial de la planta. Estas son principalmente la integración en un *rack*, el uso del sistema operativo Windows y la instalación de un firewall intermedio.

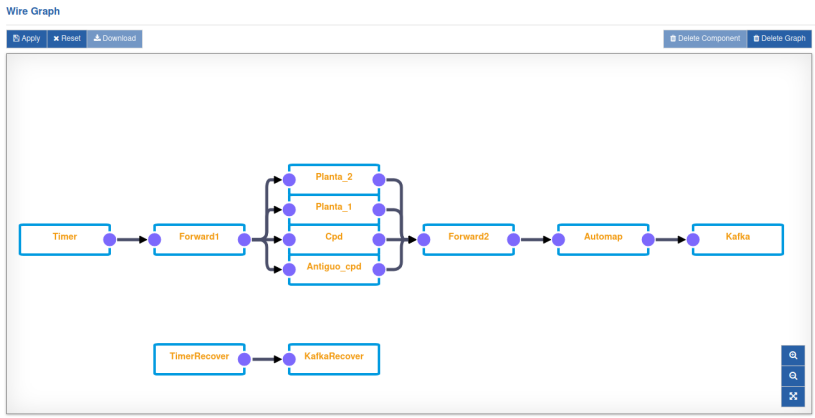


Figura 5.9. Flujo de trabajo configurado en Kura para Abdali. Se ha simplificado el número de equipos conectados y monitorizados para facilitar la visualización.

Este último requisito se logró evitar tras una reestructuración de la red de planta entre la que ya se pudo considerar este equipo adicional. Para cumplir con los otros dos, se opta por un servidor industrial enracable de coste muy ajustado. En concreto, el equipo empleado es el modelo ACP-2010/2330 de Advantech, que se presenta en la Figura 5.10. Sobre este se instaló Windows 10 primero y Kura sobre Windows.

Puntos críticos

Se destaca que en la propia puesta en marcha de la solución, surgen algunas dificultades. Las más relevantes se presentan a continuación:

Instalación OS Windows La instalación sobre Windows de Eclipse Kura no es nativa, ya que está desarrollado para Linux. Sin embargo, Kura está disponible también para instalar a través de Docker. Docker es una aplicación de empaquetamiento de software similar a una máquina virtual. En sistemas operativos Linux la eficiencia de Docker es muy superior a la de una máquina virtual ordinaria, ya que accede a un nivel de aislamiento a nivel nativo. Otros como Windows o MacOS soportan también Docker mediante una capa de

Figura 5.10. Equipo empleado como Gateway IIoT en Abdali. Se instaló en el interior de un armario de servidores o rack y debía respetar el factor de forma propio de estos equipos.

virtualización que resulta más eficiente que una virtualización común aunque no tanto como en Linux.

Al poder usar esta distribución de Kura, se sorteó la restricción de la planta de usar solo Windows. Igualmente fue necesario diseñar un despliegue específico considerando esta característica, pero una vez hecho es aplicable para cualquier Windows 10 y, al igual que en otros equipos, el control de Kura sobre el resto del equipo sigue siendo muy alto. En particular la única cuestión que se queda atrás con este despliegue es el control de la red que es muy específica del sistema operativo.

Cambio de variables exportadas Una dificultad que ya se ha presentado es la necesidad de cambiar constantemente el conjunto de variables a exportar a SIS y los procesamientos asociados a ellas. Esto se solventa con facilidad en la capa edge gracias a la posibilidad de configurar Kura remotamente. Además de poder

modificarse rápidamente, los cambios no comprometen la mantenibilidad de la solución.

En la parte de la plataforma, gracias a la centralización de la información y la estructura proporcionada a las variables, se facilita la creación de integraciones adaptadas a cada procesamiento, pero manteniendo la reusabilidad de dichas integraciones.

Caída de servidor OPC UA Debido a un problema de rendimiento del propio SCADA de la planta, el servidor OPC UA dejó de estar disponible para la monitorización paralela de la arquitectura IIoT al término de la instalación de la planta, cuando se conectaron todos los sistemas. Esto se debió a que la disponibilidad del servicio pasó a estar mucho más ajustada y por tanto la adquisición de datos dejó de ser fiable para leer en tiempo real y también costosa la adquisición de históricos.

El servidor de respaldo dispuesto para estas cuestiones permite acceder a la misma base de datos y conexión. Sin embargo, en este servidor no se aplican las mismas licencias que en el principal, lo que se traduce básicamente en que los protocolos habilitados en el mismo son el OPC DA (para tiempo real) y OPC HDA (para histórico). Estos dos protocolos son más antiguos que el OPC UA y plantean muchas complicaciones de integración multiplataforma. De hecho, así como en OPC UA sí existen librerías oficiales para la integración, en los otros dos no hay publicados recursos similares.

Ante esta circunstancia, a lo largo de los años se han lanzado varias iniciativas Open Source para desarrollar integraciones en múltiples sistemas operativos y lenguajes de programación. En particular, para la lectura desde Linux y con Java, hay varios proyectos tanto de particulares como de fundaciones de mayor envergadura. Al tener Kura una gran facilidad para integrar soluciones de terceros, se desarrolló un driver para estos nuevos protocolos.

Si bien en esta ocasión no existe un driver adaptado directamente a Kura, su capacidad para incorporar otras soluciones permite superar la situación con poco desarrollo adicional. Además al tener una capa de definición de drivers común, es posible reaprovechar la mayor parte de la configuración creada para la lectura por OPC UA.

Integración RV Como caso particular de integración, ya se ha comentado que surgió la necesidad de construir un entorno virtual con la adquisición de datos (Figura 5.11). Para que una recreación de realidad virtual resulte práctica debe ser muy fidedigna y tener una respuesta muy natural para la experiencia de usuario. A nivel de datos, se traduce en una transmisión que se acerque al tiempo real tanto como sea posible.

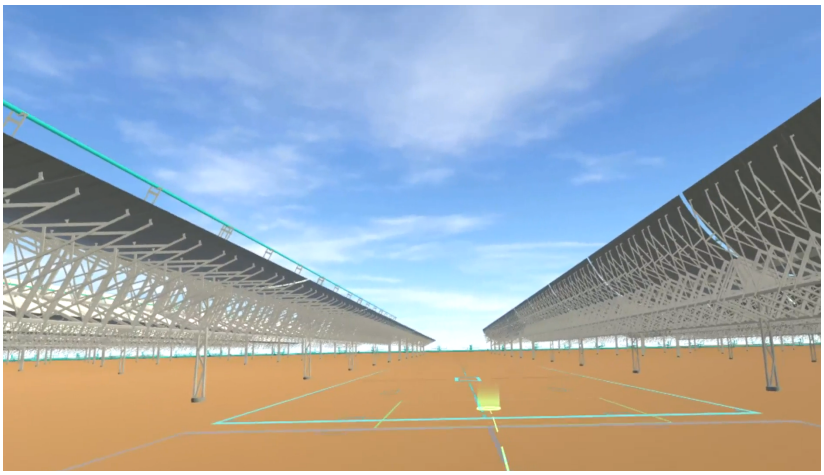


Figura 5.11. Vista de Abdali a través de un interfaz de realidad virtual. El uso de una arquitectura IIoT facilita la integración y mantiene un ritmo de refresco natural para el usuario.

También, para facilitar la información es necesario adaptar el conjunto de señales adquiridas ya sea para sintetizar un gran número de datos como para localizarlas correctamente. La capacidad de procesamiento y los metadatos asociados que se dan gracias a la incorporación en SIS por la arquitectura IIoT,

hacen posible trasladar estas facilidades al usuario.

5.2.4. Rendimiento de la solución

La solución desplegada permite una monitorización en tiempo real y remota de la planta termosolar. Esto se emplea para mejorar el rendimiento de la misma con asistencia y análisis remoto gracias a paneles como el mostrado en la Figura 5.12. También permite el despliegue de pruebas sobre la instalación, que contribuyen a la mejora de la propia puesta en marcha y el desarrollo de tecnología para plantas similares.

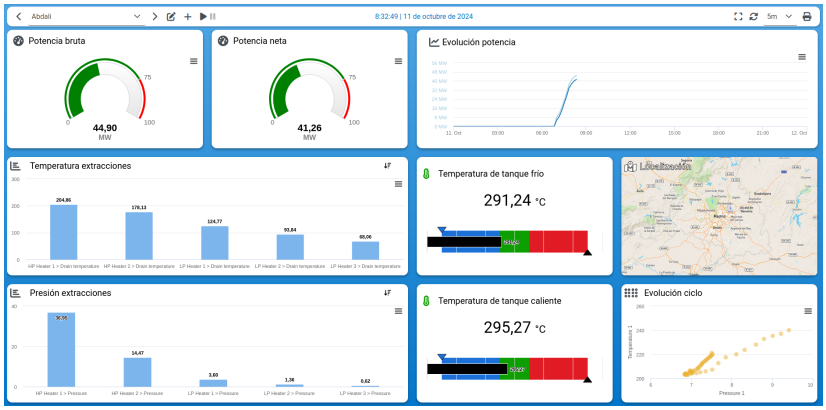


Figura 5.12. Panel de monitorización de la planta termosolar de Abdali. Resume algunos de los elementos críticos y más característicos del funcionamiento de la planta para identificar rápidamente ineficiencias en la producción.

De esta manera, aporta ventajas sobre la aplicación en aspectos como:

- Supervisión remota: complementaria a la supervisión local, la solución permite monitorizar remotamente el estado de la planta, lo que facilita la detección de incidencias y la adecuación de turnos de trabajo.

- Integración: gracias a la centralización de la información y la plataforma IIoT, es posible integrar los datos adquiridos en la planta con gran cantidad de sistemas, herramientas o software, siendo además integraciones reusables

para otras soluciones.

- Análisis avanzado: la obtención de datos remota y su integración abre la posibilidad de análisis avanzados tanto para mejorar la supervisión como para investigación.

- Preprocesamiento de los datos: la agrupación de los datos adquiridos para reducir la cantidad a señales por minuto permite un mejor procesamiento sin comprometer la calidad de la señal. También minimiza el impacto de almacenamiento y el ancho de banda de la conexión remota.

- Flexibilidad: la adaptabilidad de la capa edge hace posible modificar la exportación de las variables en función de las necesidades del momento, de forma que se pueda ampliar a lo largo de la vida útil de la instalación.

- Fiabilidad: la robustez de la adquisición y tratamiento de datos permite realizar una inspección y análisis fiable y por tanto obtener resultados de mucho valor.

- Bajo coste: al limitarse a un único PC de bajas características, el impacto de coste es relativamente pequeño. También se mantiene pequeño a pesar de la información procesada gracias a la agregación y reducción de información.

- Bajo consumo: dado que el equipo es de bajas prestaciones y aún así trabaja por debajo de sus posibilidades, el consumo es ínfimo en el conjunto, no ya de la planta, sino del propio armario de servidores industriales.

Adicionalmente a lo ya planteado, el éxito de este despliegue se manifiesta a partir de puestas en marcha posteriores en las que la solución, el flujo de trabajo y el procesamiento es prácticamente igual. Por ejemplo, esta misma solución se encuentra desplegada en cuatro plantas de Chile en las que se dio la necesidad de monitorizar bajo unas condiciones muy específicas para superar una auditoría

de calidad y generar garantías de las plantas. Estas condiciones, suponen solo modificaciones menores respecto a la solución gracias a su versatilidad ya que son referentes a frecuencias de lectura y variables a leer desde un servidor OPC. La implantación de estas cuatro soluciones se pudo realizar por tanto en cuestión de horas, frente a alternativas que hubieran exigido días de trabajo. De esta manera, la aplicación que se presenta como base para dar soporte adicional a una planta termosolar pasa a incorporarse también como servicio interno para TSKen la generación de garantías e informes de calidad en futuras puestas en marcha.

5.3. Gestión de infraestructura de red industrial

La inmensa mayoría de plantas industriales tienen una red basada en TCP/IP para la conexión de los activos del proceso industrial. Esta red contiene muchos equipos asociados con el proceso productivo, pero también contiene otros muchos solo para ofrecer esta infraestructura de comunicación. Entre estos últimos los principales son switches, firewalls y módems. También se puede considerar algunas cámaras o servidores industriales, según su uso.

Esta infraestructura habitualmente no se controla con el mismo rigor con el que gestiona el resto de la planta, porque se infravalora su impacto en la producción y muchas veces simplemente se diseña con redundancia para reemplazar equipos sin problemas y se le da poca más importancia. Sin embargo, las incidencias en estos equipos podrían llegar a ser tan críticas o más que en otros, pues puede llegar a aislar partes de la planta que no pueden funcionar de forma autónoma.

Esta necesidad se ve agravada por el escenario de ciberseguridad actual. En este momento se producen con frecuencia ciberataques contra industrias, más en el ámbito de la energía, y los equipos de la infraestructura de red son la primera línea de defensa para impedirlos pues contienen las reglas de acceso (módems y

firewalls) y enrutado (firewalls y switches) para comunicarse con toda la planta. Para garantizar que estos equipos están operativos y que su funcionamiento es correcto es necesario recopilar su información de estado en tiempo real y realizar sobre él análisis. De esta manera es posible encontrar anomalías y equipos comprometidos.

La presencia de un equipo edge IIoT puede usarse para realizar esta monitorización de infraestructura de red. En este caso la capa de percepción sería mucho más simple que en otros porque se trata de equipos que ya tienen contemplada su monitorización, la mayoría de ellos por el protocolo SNMP (Simple Network Management Protocol). El procesamiento de estos datos sí resultaría más relevante primero para la obtención de variables características y luego para la creación de alertas.

Aunque ya existen algunas herramientas para este propósito, la integración en una misma arquitectura IIoT ayudaría a reducir el número de verticales implicados, eliminando complejidad innecesaria a la planta y facilitando la integración con terceros de cara a alguna analítica de mayor orden. Por ejemplo, una arquitectura IIoT podría integrarse también con un analizador de red de un firewall o con el resto de la monitorización de la planta industrial.

Dado que las instalaciones industriales son de muy diverso tamaño, para adaptarse bien a esta aplicación, la solución debe ser altamente escalable y por tanto de coste muy ajustado a la situación. También será necesario mantener un consumo controlado porque, como parte de los equipos de infraestructura de red, estará probablemente conectado a un sistema de alimentación ininterrumpida (SAI). Un consumo alto impactaría mucho en la duración de dicho SAI ante un corte de alimentación. Una ventaja respecto a otras aplicaciones, son las buenas condiciones del ambiente, pues el equipo puede colocarse junto a otros servidores, que se ubican habitualmente en entornos con atmósferas controladas o por lo menos poco hostiles. Toda vez que se conecte a algún elemento de la red de

planta, podrá acceder a todos los conectados en la misma, no necesita cercanía específica a uno. En parte por esto, la ciberseguridad juega un papel importante en esta solución, donde el equipo va a tener posibilidad de acceder a una parte de la planta muy extensa.

Por último, la información obtenida debe tratarse con mucho cuidado. Primeramente por evitar que se filtre a terceros, ya que son datos que pueden poner en peligro la base de funcionamiento de la planta industrial. También por posibilitar un análisis a posteriori de un incidente. Como se ha comentado en la Sección 2.2.1, en ciberseguridad se da mucha importancia a las actuaciones a posteriori de un ataque, ya que permiten comprender por qué se ha producido y recuperar el control de una instalación.

En conclusión, esta solución debe poner el foco en:

- Ciberseguridad: la solución debe ser muy segura tanto frente a ataques como filtraciones de datos y dar ayudas para la ciberseguridad de la planta. Además el acceso a toda la infraestructura de la planta es potencialmente peligroso y debe cuidarse de no introducir vulnerabilidades.

- Integración vertical: para facilitar el análisis de los datos adquiridos.

- Escalabilidad: para adaptarse a distintos tamaños de plantas industriales y ser igualmente eficiente en todos.

- Integridad de los datos: debe disponerse de la información en tiempo real, pero es casi más importante que se garantice que no se pierda una vez adquirida para poder actuar también a posteriori de cualquier incidente.

- Procesamiento edge: aunque no sea crítico para la solución, puede aportar mucha información a la operativa dando herramientas para la ciberseguridad o una primera aproximación al análisis de datos necesario.

- Bajo consumo: para poder funcionar también bajo los sistemas de alimentación ininterrumpida sin suponer una carga excesiva.
- Bajo coste: en la medida en que la solución debe ser escalable, el coste se ve directamente repercutido.

5.3.1. Caso de uso

Para validar esta aplicación, se realizará el despliegue en Baní. Baní es una planta de producción fotovoltaica situada en la región de Arandia de República Dominicana (Figura 5.13) que tiene una capacidad de 60 MW y una extensión total de 95 hectáreas. En la Figura 5.14 se muestra una vista de pájaro de la misma.

Figura 5.13. Ubicación de la planta fotovoltaica Baní (fuente [101]).

En la ingeniería básica ofertada a la empresa Solaris Nacional (SN), promotora de la construcción, se propuso un sistema de monitorización de la infraestructura de la planta como medida adicional de ciberseguridad. Se

Figura 5.14. Vista de pájaro de la planta fotovoltaica Baní (fuente [102]). Ocupa un total de 80 hectáreas de terreno y tiene una capacidad de 50 MW.

buscaba aplicar esta supervisión a elementos básicos para el funcionamiento de la planta, esto es la electrónica de red, los sistemas de videovigilancia y seguridad perimetral, el control de accesos y el sistema de alimentación de emergencia. Esto significa conectarse a switches, firewalls, router, servidores críticos, cámaras y sistemas SAI, de donde se puede sacar al menos el estado básico (activo/inactivo, temperatura, etc.) y en el caso de los elementos de red también tráfico de datos en la planta.

La arquitectura IIoT actuará como enlace entre estas mediciones y el resto de la gestión de la planta, proporcionando los cálculos básicos para que sobre la información se puedan aplicar análisis de mayor calado y ajustados al tipo de control que se quiera hacer.

Los equipos a monitorizar son:

- Switch industrial Cisco Catalyst 9200L con 24 tomas ethernet y 4 de fibra óptica en disposición redundante uno frente al otro para función de núcleo de la comunicación. Dos unidades

- Firewall idnsutrial Cisco Firepower 2110 con 12 tomas ethernet y 4 de fibra

óptica. Dos unidades.

- Switch industrial Fortinet Fortswitch FSR-112D-POE con 8 tomas ethernet y 4 de fibra óptica en campo para servicio en la planta fotovoltaica. Ocho unidades.

- Switch industrial Fortinet Fortswitch FSR-112D-POE con 8 tomas ethernet y 4 de fibra óptica en campo para servicio en la red de control de Control Cerrado de Televisión (CCTV) de la planta. Seis unidades.

- Cámara de videovigilancia Axis Q6215-LE/Q6010-E/Q6074-E/I8016-LVE/P3715-PLVE para vigilancia perimetral de la planta y control de acceso. Siete unidades.

- Detector de presencia por RADAR Magos Systems SR-1000/500/250, para vigilancia perimetral de la planta. Nueve unidades.

- Servidores industriales Dell Precision 3630 y Milestone Husky con funciones diversas en el control de la infraestructura de red: control de acceso, grabación de CCTV, analítica del sistema de radares y estación de trabajo de operador. Cuatro unidades

- SAI Victron Cerbo GX con sistema de carga fotovoltaica compuesto por un panel, un inversor, una unidad de control, un cargador y una batería. Ocho de cada uno de los elementos.

Como se puede ver, incluso siendo una planta con un sistema de comunicaciones relativamente pequeño (hay pocos equipos implicados en la producción como tal), la electrónica de red está compuesta por un total de 71 equipos de hasta 12 modelos distintos. Un análisis de sus características permite limitar los protocolos empleados solo a dos: Modbus (todos los equipos de SAI) y SNMP (el resto). El protocolo SNMP simplifica mucho la configuración ya

que, aunque cada fabricante puede incluir sus propias métricas, hay muchas que suelen compartir todos los equipos, sobre todo las más generales como el estado, o la carga de CPU y RAM, también las tomas de red tienen todas los mismos parámetros a pesar de estar en distintos equipos. En todo caso, el trabajo de configuración seguirá siendo extenso, como se verá más adelante.

5.3.2. Diseño e implantación

Tal como se ha comentado, para monitorizar la infraestructura de red, no es necesario utilizar ningún equipo ni servicio de la capa de percepción, ya que todos los equipos están ya integrados en una misma red de comunicaciones, como se puede ver en la Figura 5.15.

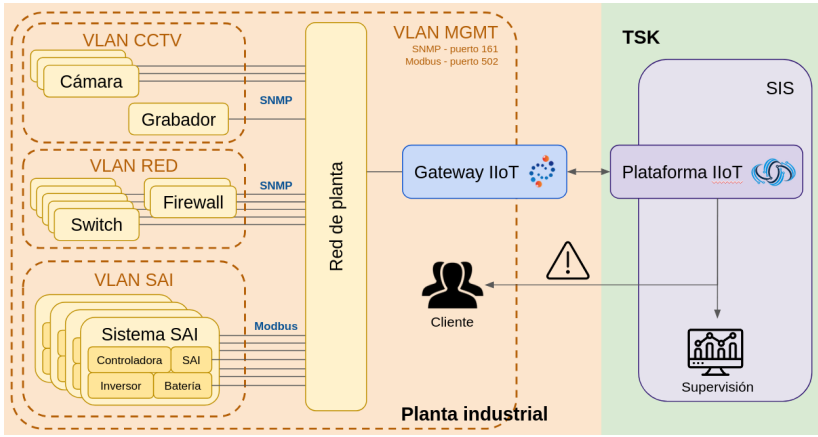


Figura 5.15. Diagrama funcional de equipos en la planta Baní. El Gateway IIoT accede a toda la información por la red de planta, con restricción a los protocolos SNMP y Modbus TCP.

Desde el punto de vista de la recopilación de información el flujo de trabajo responde al ya presentado en la Sección 5.1.2, sobre el que hay que particularizar algunas cuestiones. Por un lado para la lectura ya se ha visto que para recopilar todas las métricas basta con disponer de drivers de lectura para SNMP y Modbus. La lectura Modbus se puede habilitar con un módulo público de Kura. Sin embargo, la lectura SNMP no está implementada entre los drivers públicos, por

lo que se hace necesario desarrollarlo.

Por otro lado, el procesamiento de los datos está principalmente centrado en complementar las métricas básicas con algunas más específicas. Por ejemplo aunque algunos equipos proporcionan el tráfico de red en Mbps, las plantillas estándar de SNMP para un interfaz de red dan esta información como cuenta de paquetes enviados o recibidos. Esto significa que es necesario escalar y derivar las lecturas para poder obtener el dato deseado. El catálogo completo de operaciones desplegadas serían:

- Derivada: ya presentada para obtener tráfico de red a partir de paquetes enviados/recibidos.
- Porcentaje o porcentaje inverso: para calcular valores total, usado, libre y porcentaje de uso de disco, RAM o CPU cuando alguna de estas variables no está presente.
- Agregación: para obtener valores diarios o mensuales del consumo de GB de tráfico de red.

Este conjunto de operaciones contribuye no solo a completar la información obtenida, sino sobre todo a homogeneizarla. Gracias a estas operaciones, aunque algunos equipos no dispongan de las variables importantes para el proceso, tras estos cálculos se logran obtener, de manera que aguas arriba, en otros procesamientos, es posible analizar los datos de forma más abstracta. Una vez ejecutados estos procesamientos, la información debe enviarse a la plataforma para su persistencia. Desde la topología Big Data desplegada en SIS puede analizar patrones de tráfico, detectar anomalías e incluso detectar algunos ataques de red o acciones sospechosas.

El resumen de todo este flujo de trabajo se presenta de manera visual en la configuración de Eclipse Kura que se muestra en la Figura 5.16. El flujo completo sería:

1. Timer: lanza cada minuto todo el flujo de trabajo.

2. Equipos: lectura de todos los equipos configurados. En la Figura 5.16 se omiten muchos de estos equipos para facilitar la visualización del flujo de trabajo en este documento.

3. Monitorización: sirve en primer lugar para unificar el flujo de trabajo de cara a futuras etapas y se aprovecha para controlar que todos los equipos están presentes.

4. Cálculos: en este caso se configuran en paralelo ya que no dependen unos de otros. Podrían hacerse igualmente en serie.

5. Mapeo: conversión de los identificadores en Kura a unos universales.

6. Kafka: envío a la plataforma de procesamiento y visualización SIS por Apache Kafka.

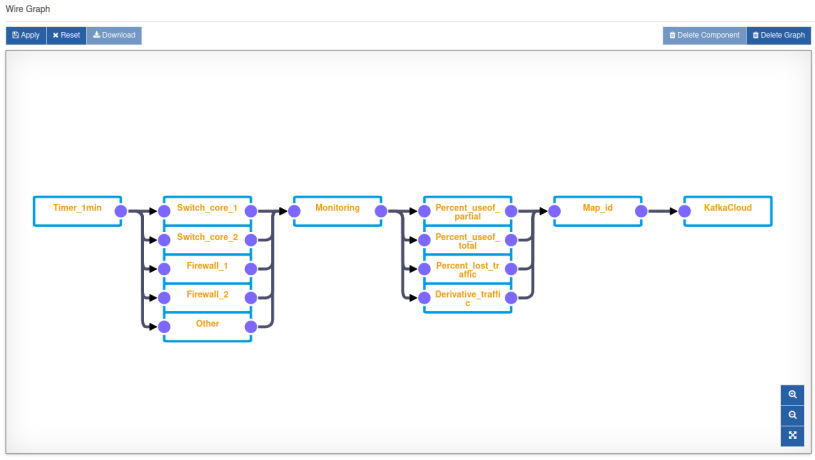


Figura 5.16. Flujo de trabajo configurado en Kura para Baní. Se ha simplificado el número de equipos leídos para facilitar la visualización.

Para alojar este procesamiento en Eclispe Kura, no es necesario un equipo de muchas prestaciones. La única restricción relevante considerada fue la necesidad

de incorporar el equipo en un rack de servidores industriales, por lo que se debía respetar cierto factor de forma. Dado que los requisitos en hardware no eran altos, se localizaron gran cantidad de equipos válidos con bajo consumo, precio ajustado y un volumen de 1 U (unos 45 mm de altura), que es el tamaño mínimo que puede ocupar un equipo en un rack. En concreto, el equipo empleado fue un servidor industrial enracable de Advantech modelo ACP-2010/2330, que se presenta instalado en la Figura 5.17.

Figura 5.17. Equipo empleado como Gateway IIoT en Baní. Se instaló en el interior de un armario de servidores o rack y debía respetar el factor de forma propio de estos equipos.

Como última consideración, el gateway IIoT deberá tener acceso a la infraestructura, pero no a los equipos conectados a ella. Por tanto, es necesario aislarlo mediante una red virtual (VLAN) que solo tenga acceos a estos equipos. También, aprovechando las capacidades para ciberseguridad que proporcionan los firewall industriales, esta VLAN se limitará a permitir comunicación por los protocolos y puertos de SNMP y ModbusTCP exclusivamente. Esto se refleja

también en la Figura 5.15. Con estas dos medidas, la puesta en marcha de la solución no introduce vulnerabilidades y puede servir a su propósito final de mejorar la seguridad de la planta.

Puntos críticos

Aunque se presenta la solución final, en el curso de la puesta en marcha se encontraron bastantes dificultades que se fueron subsanando para lograr el objetivo final. Entre las más relevantes se podrían indicar:

Diversidad de implementaciones SNMP Dentro de los equipos que soportaban SNMP, no todos tenían disponibles el mismo conjunto de variables. En algún caso, incluso las variables más comunes como RAM o CPU no se incluían según el estándar sino incompletas o en registros de fabricante.

A raíz de esto fue necesario hacer prácticamente una configuración distinta por modelo/fabricante y también hizo necesario crear cálculos para obtener una colección de variables comunes. Por ejemplo, algunos equipos proporcionaban la RAM disponible y la ocupada, mientras que otros proporcionaban la disponible y el total.

Para homogeneizarlo, se prepararon cálculos capaces de obtener toda la información desde cualquiera de los tipos de equipo de forma que en cualquiera de los casos se obtuviera al menos el uso porcentual, que es realmente el valor que se buscaba obtener para la monitorización.

Cambio de protocolo Entre los criterios de selección para los equipos Cerbo GX usados en el SAI se consideró que soportaba teóricamente la comunicación por SNMP. Sin embargo, una vez se empezó a implementar, se descubrió que las variables disponibles por ese protocolo únicamente permitían saber que el equipo seguía en funcionamiento. Consultando con el fabricante se confirmó que efectivamente por SNMP no era posible obtener apenas información.

Al consultar por los datos requeridos, el fabricante recomendó utilizar la comunicación ModbusTCP habilitada en el equipo, donde variables como estado de carga o tiempo de uso del SAI sí estaban disponibles. Gracias a la flexibilidad de Eclipse Kura para integrar múltiples protocolos industriales, se pudo incorporar la lectura de estos equipos sin que se diferenciasen aguas arriba del resto de las lecturas implementadas.

Unificación de modelos Aunque inicialmente estaba previsto utilizar un solo modelo de cámaras para todo el recinto, fue necesario utilizar hasta cinco modelos distintos debido a necesidades específicas en algunos puntos y falta de stock en dos de los casos. Aunque pertenecían todas al mismo fabricante, el acceso a los datos no era realmente igual ya que algunas soportaban un firmware más completo y actualizado que otras. Tras la integración en Eclipse Kura, los datos de todas estaban igualmente disponibles como si fueran un único modelo.

En los switch, siendo de Cisco y Fortinet, también se enfrentó esta cuestión, aunque se previó desde un principio.

Gestión a través de red cerrada El Gateway IIoT utilizado se configuró y probó antes de la instalación en planta. Sin embargo, en las puestas en marcha siempre es común disponer de un medio para conectarse remotamente al equipo mediante una VPN. Aunque esta era la propuesta inicial, una vez se intentó acordar la conexión concreta con la empresa SN, ésta determinó que no se podría acceder por VPN al equipo por motivos de seguridad.

Por fortuna, la conexión por MQTT de Eclipse Kura con la plataforma permite una gestión plena de todas las funcionalidades del equipo. Gracias a esto, a pesar de esta falta de acceso que podría haber supuesto un desplazamiento a la planta para un ajuste final, se logró corregir las dificultades de última hora remotamente. Entre otros cambios, se destaca que las direcciones IP de algunos equipos monitorizados tuvieron que cambiarse debido a una reestructuración de

las VLAN de la planta.

5.3.3. Rendimiento de la solución

La solución IIoT planteada aprovecha principalmente la capa edge para integrar toda la electrónica de red y demás infraestructura de una planta industrial. Así, sin entrar al detalle de las características del proceso, puede garantizarse un mínimo de operativa en la planta. Desde esta electrónica de red, la información se envía a una plataforma de procesamiento por Apache Kafka aunque se mantiene la capa de plataforma para la propia gestión de la arquitectura.

Algunas de las ventajas que proporciona esta solución es:

- Homogeneización: gracias al procesamiento edge, se homogeneizan los dispositivos monitorizados a un conjunto de variables común con todos los equipos y aptos para un procesamiento abstracto a pesar de las diferencias de modelos y protocolos empleados.
- Integración: a través de las herramientas de integración del gateway IIoT, se dispone de un modelo homogéneo de datos fácilmente accesible desde cualquier otro servicio, tanto local como remoto.
- Visualización resumen: en la integración básica con TSK ya están disponibles los datos para una visualización simplificada como se ve en la Figura 5.18. Otro sistema de representación similar como Grafana podría emplearse también a partir de la conexión MQTT del propio equipo.
- Reducción de complejidad: gracias a la homogeneización, se reduce la complejidad en el resto de los servicios que dependan de estos datos
- Coste bajo y escalable: dado que apenas se necesitan recursos para este procesamiento y depende sobre todo de la cantidad de equipos a

monitorizar, la solución es muy escalable a cualquier magnitud de planta industrial.

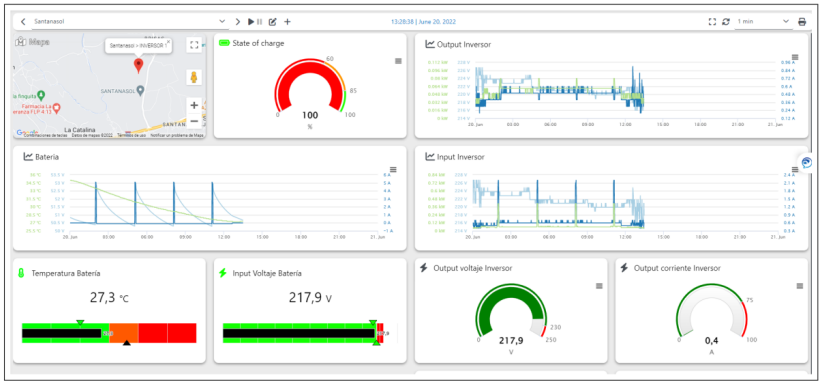


Figura 5.18. Ejemplo de visualización de datos de los equipos de Baní. Estos paneles permiten controlar de un rápido vistazo que toda la infraestructura de red, alimentación y control de accesos está disponible.

Gracias al uso de la arquitectura IIoT, toda la infraestructura de red, de alimentación, control de accesos y videovigilancia se puede tener monitorizada. Esto permite detectar más rápidamente los problemas que surjan en la propia planta y actuar con mayor diligencia en caso de fallo de un equipo. También contribuye a mejorar la ciberseguridad de la planta ya que se pueden desplegar análisis sobre la información obtenida para identificar anomalías de tráfico o estado.

5.4. Digitalización de instalaciones analógicas

Tanto en la industria en general como en la red eléctrica en particular es común encontrar equipos, sistemas o plantas enteras cuya funcionalidad es crítica pero no es posible integrarlo en un nuevo sistema o SCADA. Cuando se enfrentan estas situaciones, suele afrontarse de tres maneras: modificarlo, sustituirlo o ignorarlo. En el caso de la sustitución, podría ser muy costoso, sobre todo si no se ha amortizado aún la instalación original. También puede que no

esté clara la funcionalidad original en la medida suficiente para reemplazarlo con seguridad. Siguiendo esta misma cuestión, la modificación podría ser inviable o muy compleja por falta de documentación o personal con las competencias adecuadas. Sin recurrir a casos extremos, hay sistemas que pueden haber estado en funcionamiento 15 o 20 años y se plantea mantenerlos todavía un tiempo largo. De hecho, las empresas que lo instalaron pueden ya no existir siquiera y quizá no se contempló una aplicación como la que se plantea a día de hoy.

En esta situación, antes de ignorar la integración, se puede plantear una medición no invasiva, puede que incluso sin contacto, donde se monitorice los puntos relevantes del sistema a integrar. El planteamiento en este caso es común hacerlo partiendo de la interacción con los equipos adyacentes al sistema y los medios de actuación y monitorización para humanos. Estos métodos de integración indirectos se traducen en la instalación de una serie de sensores que permitan recopilar esta información y un procesamiento que logre aunarlos y darles un sentido coherente con el funcionamiento del activo a monitorizar.

Una vez efectuada esa digitalización, será necesario proporcionar medios de integración, sobre todo horizontal, para que el resto del SCADA pueda acceder a la información obtenida. Esto es más sencillo ejecutarlo desde una arquitectura IIoT, que cuenta con la fortaleza de estar diseñada para considerar el procesamiento en el edge y una buena diversidad de sensores.

Los principales requisitos a considerar en esta solución son:

- Versatilidad: la solución debe ser capaz de adaptarse a escenarios de digitalización muy diversos y sobre todo poder incorporar más con el tiempo, en caso de que se requiera ampliar la monitorización del activo.

- Mantenimiento: el conjunto de equipos a integrar no debe complicar adicionalmente el mantenimiento del sistema ni de la planta industrial por lo que es fundamental que sea sencillo y en lo posible con un bajo coste en

el largo plazo.

- Integración horizontal: es un requisito crítico para poder conectar la arquitectura IIoT al SCADA de la planta localmente, como un elemento más dentro de la planta.
- Conectividad: para poder integrar gran variedad de sensores y cualquier otro equipo necesario para digitalizar la información.
- Procesamiento edge: fundamental para tratar la información obtenida localmente y que la integración con el SCADA sea posible. Además de digitalizar la información será necesario tratar la señal obtenida para verificar doblemente que su valor tiene sentido.
- Integridad de los datos: para una buena digitalización la información debe ser completa, fiable y estar siempre disponible. De otra manera, toda la solución perdería su propósito.
- Rango industrial: los equipos utilizados para esta solución deben de estar preparados para afrontar las mismas condiciones que el equipo a monitorizar, lo que implica rango industrial y según el caso grados IP o IK específicamente altos.
- Coste contenido: vinculado a la necesidad de un mantenimiento bajo.

5.4.1. Caso de uso

Para la validación se aborda la situación de Energías Renovables S.A. (ERSA). Esta empresa dispone de varias minicentrales hidráulicas antiguas que se encuentran ubicadas en puntos de difícil acceso entre las provincias de León y Ourense (Figura 5.19). La denominación de las centrales, proveniente de pueblos o embalses cercanos es Cerroalto, Dehesa, Encinares y Fuentehermosa. En estas centrales, existe un sistema de monitorización y control basado en armarios

eléctricos. Aunque su funcionalidad es aún adecuada, ERSA no dispone de documentación sobre cómo funcionan o están montados dichos armarios ya que no se proporcionó esta información junto con la compra de las instalaciones.

Figura 5.19. Ubicación de las minicentrales hidráulicas propiedad de ERSA (fuente [101]).

Al intentar integrarlas en la plataforma de monitorización remota que usa ERSA para el resto de sus plantas, se encontró que era necesario introducir cambios que podrían estropear el funcionamiento de las mismas. Este camino quedó descartado ante el riesgo de introducir un problema inesperado. La sustitución no era tampoco posible en parte por la misma razón pero fundamentalmente porque al tener una producción baja, el retorno de la inversión (ROI) era muy lejano en el tiempo.

Sin embargo, valorando el coste de realizar inspecciones visuales cada poco tiempo para conocer la producción de las plantas, se optó por digitalizarlas de una manera alternativa. Dado que están pensadas para la monitorización humana por inspección visual, toda la información necesaria para conocer el estado de

las instalaciones estaba disponible en paneles eléctricos en forma de matrices de luces para reflejar alarmas, u otras señales digitales, y medidores de aguja para visualizar valores analógicos.

Más allá de esto, no existe la posibilidad de conectarse a los armarios para obtener estas señales de otra manera. La documentación de los mismos es escasa o nula y no disponen de ningún protocolo de comunicaciones. Estudiar el armario para realizar ingeniería inversa es costoso y también arriesgado en caso de que se produzca cualquier desperfecto. Por tanto, la solución menos invasiva es simular la inspección visual mediante una cámara y algoritmos de visión artificial.

Una vez digitalizada la señal, la solución propuesta deberá integrarla en el sistema de monitorización de ERSA. Para ello, se usará la red existente en la planta para otras necesidades (control de acceso, videovigilancia y monitorización de otros sistemas sí digitalizados) y se expondrá en esa red la información en un protocolo y formato soportado por el SCADA de la planta. Entre las propuestas ofrecidas se propuso Modbus o el protocolo IEC 60870-5-101/104.

5.4.2. Diseño e implantación

Según lo descrito, la digitalización de las instalaciones objetivo pasa por obtener una imagen visual de los paneles de inspección, procesarla para obtener las medidas necesarias e incorporar un servicio o cliente en alguno de los protocolos previstos para que el SCADA pueda obtener la información como si se tratase de un datalogger corriente.

Para hacerlo, se desplegaron cámaras delante de los paneles eléctricos, intentando abarcar lo más posible con cada cámara. Estas cámaras no necesitan tener grandes capacidades pero sí será necesario que tengan suficiente resolución y puedan soportar los rigores de las condiciones ambientales, que en este caso son principalmente polvo y humedad. En la Figura 5.20 se muestra la vista que se planteó obtener para cada panel.

Figura 5.20. Panel de monitorización analógica de Cerroalto visto desde la cámara de inspección instalada. Se pueden apreciar medidores de aguja y paneles luminosos para mostrar alarmas y otras señales digitales.

Conectado a estas cámaras se instaló un gateway IIoT, que ejecutaba sobre la imágenes obtenidas un procesamiento de visión artificial. Este procesamiento se desarrolló con la librería OpenCV para visión artificial y se encapsuló como un plugin de Kura. El principio del algoritmo consistía en separar primeramente las regiones a estudiar configuradas por el usuario y según su tipo buscar una aguja o una casilla de marcadores encendidos o apagados. La creación de este componente como un plugin de Kura viene motivado primero por una facilidad de integración y también por una facilidad para la configuración y actualización.

Destacar que en este punto del proceso, no se está dotando todavía a los datos de un sentido físico real. Es decir, este primer procesamiento ignora el significado de cada alarma y simplemente asocia a una región un valor. En el caso de los medidores también se ignora el rango del indicador de aguja. Todo esto permite una configuración ajustada del procesamiento y sobre todo que se pueda reusar.

Una vez se han obtenido los valores en bruto de los medidores y señales digitales, se derivan en dos caminos independientes. Por un lado se envían al

Figura 5.21. Ejemplo de panel analógico digitalizado de Fuentehermosa. El valor de los medidores se obtiene en dos etapas, primero se obtiene su ángulo y luego se aplica el rango del medidor.

SCADA a través de un servicio Modbus y por otro se envían a la plataforma SIS. Para el primer camino no se proporciona semántica a los datos obtenidos, solamente se escala el valor de los medidores analógicos para adecuarlo al rango real (Figura 5.21). El significado de cada uno viene dado por el mapa Modbus acordado con el operador del SCADA, ERSA. Para el camino a la plataforma SIS, sin embargo, se mapea esta información a identificadores únicos antes de enviarlos. Para esta asociación, simplemente se coordina en la propia plataforma la asignación y se envía remotamente al equipo. Se puede ver en la Figura 5.22 un ejemplo de asociación tal como se configura en la plataforma.

Por tanto, el flujo de trabajo necesario para integrar estos equipos sería el siguiente, que se resume de forma visual con la configuración en Eclipse Kura mostrada en la Figura 5.23:

1. Timer: lanza cada cinco minutos todo el flujo de trabajo.

2. Procesamiento: desencadena una consulta de las imágenes y procesamiento de la misma para obtener los sets de datos necesarios (un set para todas las

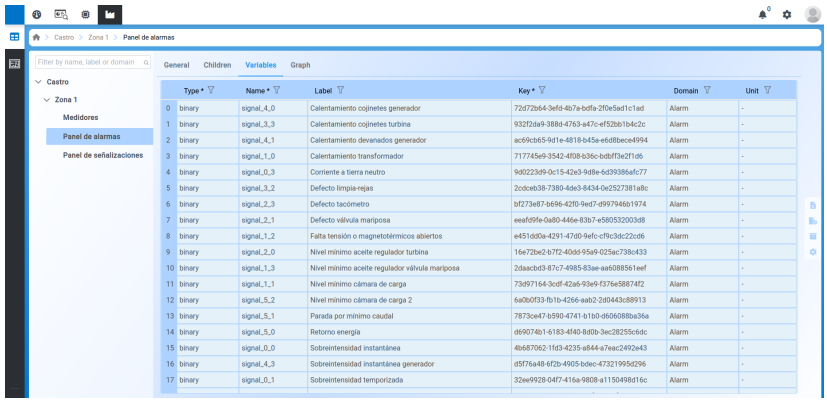


Figura 5.22. Descripción semántica de los datos de Cerroalto. La asignación cruza un mínimo de tres parámetros: la identificación única (key) en toda la plataforma, un nombre amigable (label) para los usuarios y la designación local de la variable (name) para el equipo edge.

analógicas y otro adicional por cada panel de digitales).

3. Cálculos: solo presentes para las variables analógicas. En concreto, además de la adecuación al rango ya descrita, se realiza un filtro de mediana adicional, dado que mejoraba el resultado del procesamiento ampliamente.

4. Integración SCADA: envío al SCADA a través del servicio Modbus.

5. Mapeo: conversión, para el flujo a SIS, de las variables a identificadores universales.

6. Kafka: envío a la plataforma de procesamiento y visualización de TSK. Destacar que a TSK se enviaban no solo los datos enviados al SCADA sino también aquellos sin tratar para validar el procesamiento y las imágenes.

En paralelo a este flujo de series temporales ordinarias (analógicas y digitales), se realizó también una recopilación directa de las imágenes proporcionadas por las cámaras. Si bien esto no es estrictamente necesario para el proceso (y de hecho el SCADA de ERSA carece de la capacidad para asimilar imágenes), se realizó para facilitar la usabilidad de la plataforma en SIS, ayudar en la resolución de

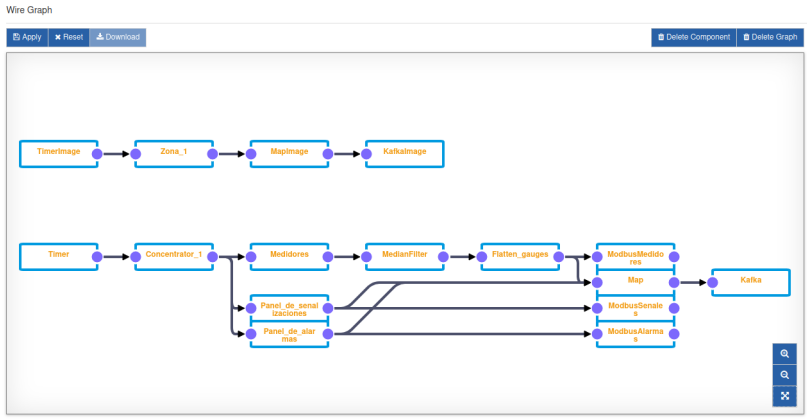


Figura 5.23. Flujo de trabajo programado en Eclipse Kura para la planta de Cerroalto. Se introduce un segundo hilo de procesamiento con un periodo mayor para el envío de imágenes.

incidencias y mejorar el algoritmo de visión artificial empleado. En este caso el flujo de trabajo es más o menos similar a otros en tanto que consiste en la adquisición, mapeo y envío de los datos. No se ejecutan procesamientos sobre estas imágenes.

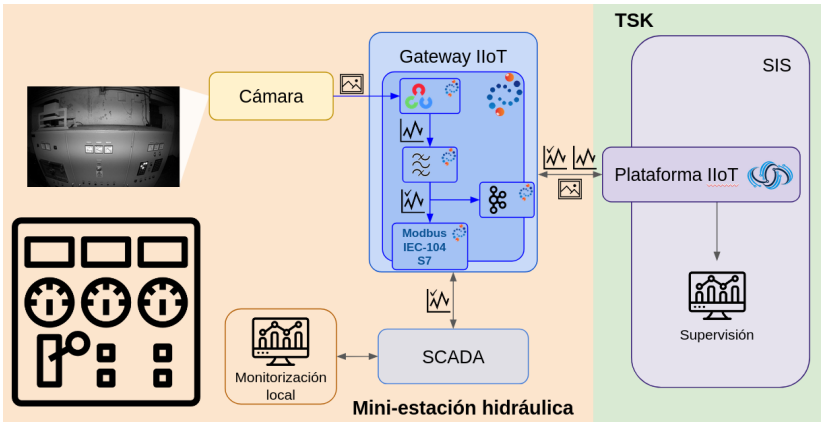


Figura 5.24. Diagrama funcional de Cerroalto. El procesamiento de la imagen y la integración con el SCADA se realiza a nivel local en gateway de la capa edge.

Como se puede ver en la Figura 5.24, la solución mantiene la arquitectura planteada con una capa de percepción compuesta por las cámaras y la posibilidad

de otros sensores, una capa edge donde está el gateway IIoT y la integración con el SCADA y la plataforma para controlar el equipo remotamente y recopilar también los datos en paralelo al SCADA.

Finalmente, destacar que todos los equipos implicados, no solo las cámaras, deberán estar preparados para asumir las condiciones industriales, en particular polvo y humedad, lo que fuerza a utilizar equipos sin ventiladores y que tengan grado industrial para soportar mejor la exposición continuada a estas condiciones. En este caso, el gateway IIoT se desplegó en un LEC 7230-M, que es un PC industrial de coste bastante acotado pero potencia suficiente para correr el procesamiento de visión artificial desarrollado. En este equipo, el framework de Kura puede ejecutarse sin ningún problema [2].

Puntos críticos

En el curso de la puesta en marcha de las cuatro plantas se encontraron algunas incidencias que la arquitectura IoT pudo asimilar sin mucha dificultad. Asimismo, durante la operación posterior se sucedieron otros problemas que también se pudieron paliar gracias a la versatilidad de la infraestructura desplegada.

Diferencias entre las instalaciones Si bien todas las instalaciones constaban de los elementos comunes ya descritos, todas ellas tenían diferencias.

Como primer punto, el número de zonas a monitorizar no era el mismo en todas la plantas. Aunque se ha presentado el caso de Cerroalto para ejemplificarlo, donde solo se debía monitorizar un panel, otros tenían más paneles y señales. Para abarcar estos nuevo elementos no es necesario replicar la capa edge, con incrementar la base de la percepción es más que suficiente para cubrirlos.

Asimismo, aunque se esperaba una colección de variables siempre igual en todas las plantas según la documentación inicial, al implantarlo se encontró que

el conjunto de variables a integrar era ligeramente distinto en cada panel y en cada planta. Dado que el procesamiento es abstracto a nivel de SCADA, esto no supuso un conflicto con las integraciones alternativas. Gracias a la fácil gestión de los metadatos asociados a las variables de la arquitectura construida, tampoco es un problema para la integración con la plataforma de visualización de TSK.

Cálculo de mediana El procesamiento para obtener la posición de una aguja en un medidor fue desarrollado primeramente con elementos de laboratorio, luego imágenes reales sacadas por operadores en distintos momentos y finalmente con una breve calibración durante la puesta en marcha.

Sin embargo, tiempo después de la puesta en marcha, la acumulación de polvo en el equipo a monitorizar cambió las condiciones de la calibración y se empezó a detectar que el algoritmo implementado arrojaba ocasionalmente valores nulos. Esto se entendía como una falta de detección de la propia aguja, que se asociaba con su posición en el borde del medidor.

Si bien se realizó una recalibración remotamente para solventar el problema, la eventualidad de que se volviera a producir llevó a valorar con el cliente posibles opciones. Una calibración automática periódica, por ejemplo, solucionaría el problema pero exigiría un tiempo alto de desarrollo. Otra opción puesta sobre la marcha fue la aplicación de un filtro sobre el valor detectado. De esta manera se buscaría omitir aquellos valores que sean claros errores en base a los datos conocidos recientes. En este caso se estimó que el estadístico más adecuado era el filtro de mediana, consistente en devolver la mediana de los últimos n valores.

Según lo estimado en los valores históricos, el filtro de mediana con una ventana de 5 valores eliminaba por completo los errores producidos y apenas producía desviación ante los cambios de estado de los valores. Este procesamiento se implementó en Kura de manera sencilla y rápida gracias a los plugins ya disponibles y una breve configuración del flujo de trabajo sin necesidad de

incorporar un nuevo procesamiento.

También era deseable hacerlo así en vez de incorporarlo al procesamiento de visión artificial para mantener el diseño modular. Una de las claves del éxito de una infraestructura modular es que cada parte haga una parte pequeña y no incorpore demasiada funcionalidad. De esta manera, si alguna no se usa en un momento dado, no basta con no desplegarla. Además, se evita sobrecargarlo de lógica, cuando Kura lo podía asumir de forma más sencilla.

Integración de algoritmo de visión artificial A pesar de que Eclipse Kura no es un hardware diseñado específicamente para incorporar algoritmos de visión por computador, su capacidad de incorporar funcionalidad permitió añadir este procesamiento integrado dentro del propio middleware como un módulo más. Esta incorporación mejora mucho la reusabilidad del desarrollo ya que permite desplegarlo en futuras soluciones también, como se hizo en el resto de la plantas contempladas en esta misma aplicación.

Una ventaja adicional de sistema de plugins utilizado en Kura es que se pudo separar la librería de OpenCV, que resulta bastante pesada, del componente implementado para esta aplicación. Esta separación permite mantener una única instancia de la librería aunque se incorporen varios algoritmos en paralelo. También permite actualizar rápidamente el componente desarrollado sin necesidad de volver a descargar de nuevo toda la librería en el equipo. Este tipo de estrategias ayuda a reducir el consumo de datos y resulta mucho más eficaz en entornos con una mala cobertura como es el caso.

Protocolos variados de integración La conexión con el SCADA de definió originalmente como un interfaz Modbus, donde el SCADA exponía el esclavo (o servidor) al que se enviarían las medidas. Sin embargo, en las distintas plantas se modificó esta conexión de acuerdo a la situación de cada una, de forma que resultase más sencillo de incorporar al resto de la lógica de control.

A pesar de esta heterogeneidad, se logró mantener una misma configuración abstracta en la integración de las variables gracias al tratamiento que tiene Eclipse Kura de los drivers industriales. En particular, se realizaron las siguientes implementaciones:

- Cerroalto y Dehesa: maestro Modbus enviando datos a un esclavo del SCADA
- Encinares: esclavo Modbus sirviendo datos para un maestro SCADA que consulta periódicamente
- Fuentehermosa: maestro S7 enviando datos a un PLC local del SCADA

Destacar de esta integración que en la parte de ERSA no fue necesario desplegar ningún equipo o lógica adicional relacionada con la integración ya que toda la adaptación se pudo hacer en el lado de la arquitectura IIoT. Solo fue necesario añadir los nuevos registros para las variables y los servicios/visualizaciones de explotación de los mismos que se habían previsto originalmente.

5.4.3. Rendimiento de la solución

Como se ha visto, la solución IIoT implementada logra digitalizar las señales visuales de un panel de monitorización para integrarlo en el SCADA como un activo digital más. Además se ha logrado con una solución que no pierde generalidad, ya que no necesita evaluar variables específicas, sino simplemente traslada la información a un entorno donde sí tenga más sentido dotar a la información de su sentido físico. Esto proporciona mucha escalabilidad y reusabilidad a la solución.

Algunas de las ventajas que proporciona esta solución es:

Figura 5.25. Ejemplo de visualización resumen de los datos de la planta de Cerroalto en SIS. Se presenta en un mismo panel la imagen obtenida con el procesamiento solapado, los datos leídos y la ubicación de la planta.

- Escalabilidad: debido a la abstracción de las señales a tratar, esta solución es muy escalable tanto horizontalmente (más cámaras) como verticalmente (instalaciones distintas). La información es trasladada de forma transparente al SCADA presente.

- Integración horizontal: a través de los protocolos industriales existentes en Eclipse Kura, se puede interactuar con el propio SCADA para proporcionar la información digitalizada.

- Bajo tiempo de desarrollo: el núcleo de la aplicación reside en el análisis de la imagen, el resto de las cuestiones (integración de SCADA, abstracción de datos o incluso filtros) pueden complicar la implementación sin aportar realmente valor. Gracias a la solución desplegada, esta parte ha sido mucho más sencilla y directa de preparar. También es posible abrir el abanico de integraciones SCADA disponibles sin añadir complejidad ni desarrollo.

- Desarrollo modular: es parte del éxito de esta solución pero también es una consecuencia ya que el desarrollo utilizado para digitalizar medidores y paneles es posible extrapolarlo ahora a otros servicios por su

implementación abstracta.

- Reducción de costes: a pesar del coste de los equipos (en particular las cámaras por el grado industrial), el coste global es bajo y permite a cambio reducir desplazamientos y riesgos de incidencias. En definitiva se traduce en un ahorro neto con un ROI muy rápido.

- Integración de imágenes: la capacidad de la arquitectura IIoT de asimilar todo tipo de datos, le permite integrar las imágenes sin generar complejidad, en la Figura 5.25 puede verse una visualización asociada. Esta adquisición se ha utilizado ya para solucionar incidencias relacionadas tanto con el propio mantenimiento del sistema como con incidencias de las instalaciones.

- Evolucionabilidad: gracias a la supervisión remota de los procesamientos y resultados, es posible controlar desde SIS el flujo de trabajo y mejorarlo continuamente.

- Análisis posterior: al empezar a generarse una serie histórica extensa de los datos de la planta, al contrario que con la inspección visual, es posible implementar algoritmos o análisis de mayor nivel, ya sea con los propios datos obtenidos, con su correlación con otra información de la instalación o con datos de otras instalaciones o fuentes.

En resumen, esta aplicación es un ejemplo directo de como las arquitecturas IIoT contribuyen a la creación de un gemelo digital de la planta. Este tipo de aplicaciones, además de generar un beneficio directo a la propia monitorización, permite explorar la aplicación de algoritmos o inteligencia artificial para habilitar un mantenimiento predictivo o conocer el rendimiento de la planta. El principal beneficio por tanto es la mera obtención de los datos como una fuente más de conocimiento y optimización.

5.5. Mantenimiento de subestaciones eléctricas

Las subestaciones eléctricas son puntos críticos de la red de distribución eléctrica. Se trata de estaciones en las que la energía eléctrica es preparada desde la generación para su transporte en la red o adaptada desde el transporte para su consumo. En general son nodos que concentran conexiones con varios puntos y tienen que mantener un funcionamiento continuo. De hecho, son estaciones donde es muy común mantener redundancia de todos los sistemas. Para garantizar este rendimiento, es muy importante que la planta reciba un mantenimiento permanente y garantizar que se dispone siempre de la redundancia ante fallos.

El mantenimiento habitual en estos casos es preventivo y muchas veces manual. Se realizan inspecciones de los puntos críticos de la instalación y se cambian aquellos que pueden dar algún problema antes del siguiente mantenimiento. De esta manera se anticipan muchas posibles paradas pero también se realizan inspecciones que no tienen como resultado ninguna intervención, se reemplazan equipos que podrían aguantar más tiempo y se corre el riesgo de una avería entre una inspección y otra. Se propone el uso de un sistema de monitorización digitalizado en continuo que pueda permitir un mantenimiento predictivo a partir del análisis del estado de la instalación, tratando de construir un periodo de vida útil para los equipos y ahorrando las inspecciones presenciales.

Las arquitecturas IIoT proporcionan mucha versatilidad en la capa de percepción y edge por lo que es posible encontrar soluciones fácilmente integrables en estas arquitecturas que permitan una monitorización no invasiva e integrarse en el SCADA como un sistema más de la forma que se disponga en la instalación. Así, por ejemplo, se plantea la monitorización en continuo de las barras colectoras (o embarrados) de una subestación.

Estos embarrados deben reemplazarse cada cierto tiempo y es muy directo

saber si están defectuosos o en el final de su vida útil si se mide su temperatura ya que se calientan más de lo debido. No obstante, medir esta temperatura supone un reto ya que no es posible instalar sensores de contacto en ellos y las medidas en remoto por sensores de infrarrojos resultan poco rentables para tantos puntos. En muchos casos lo que se hace es monitorizarlos manualmente durante otras operaciones de mantenimiento mediante un sensor de temperatura portátil por infrarrojos. Al no realizarse en continuo, no es posible analizar realmente el ciclo de vida de los embarrados y solo pueden realizarse mantenimientos preventivos. De nuevo se persiste el riesgo de que un embarrado se deteriorase entre una revisión y la siguiente.

Se propone un sistema de mantenimiento predictivo que obtenga localmente la información en bruto suficiente para una monitorización local efectiva y exporte en remoto un histórico íntegro para entrenar un algoritmo de mantenimiento predictivo. Para que este sistema funcione, será fundamental que en la capa edge de la arquitectura exista suficiente capacidad de procesamiento para proporcionar información útil y precisa en el SCADA local.

Un resumen de los requisitos a considerar sería:

- Ciberseguridad: al tratarse de una infraestructura crítica, no debe introducir vulnerabilidades en la red.
- Integración horizontal: para conectarse con el SCADA como un equipo más.
- Procesamiento edge: fundamental para obtener la información localmente a partir de la imagen termográfica.
- Integridad de los datos: la información obtenida debe llegar de forma completa y fiable para poder basar en ella el mantenimiento. Se hace aún más importante si se quiere iniciar un sistema predictivo asociado.
- Rango industrial: el hardware debe tener una alta fiabilidad y soportar

condiciones adversas. En especial si los embarrados están abiertos al exterior

- Coste: aunque no es crítico, se intentará que el coste permita un ROI bajo en comparación con la inspección manual periódica.

5.5.1. Caso de uso

La subestación eléctrica de Granadilla, situada en la isla de Gran Canaria (Figura 5.26), sirve de conexión a la red para un complejo de generación de energía renovable que integra energía eólica y fotovoltaica. Esta subestación cuenta con embarrados en el interior del edificio principal que se puede ver en la Figura 5.27, sobre los que se levantará un sistema de monitorización continua basada en la arquitectura IIoT descrita.

Figura 5.26. Ubicación de la subestación de Granadilla (fuente [101]). Ubicada en la isla de Gran Canaria.

La inspección de estos embarrados se realiza periódicamente empleando una pistola de infrarrojos o una cámara de mano termográfica. El operario apunta hacia los embarrados desde el suelo y comprueba que ninguno exceda

Figura 5.27. Vista de pájaro de la subestación de Granadilla (fuente [103]). La planta sirve de conexión para un complejo de generación renovable en la isla.

una temperatura dada. Aunque estas operaciones se realizan con frecuencia relativamente alta, pueden llegar a pasar varias semanas entre una y otra. Además no se realiza registro de la misma si no se detecta ningún fallo por lo que es difícil aprender de los errores en caso de darse.

Para mejorar el mantenimiento de esta planta se plantea un sistema de monitorización continua que permita obtener no solo la temperatura en cada momento, sino también desencadenar alertas automáticamente frente a un determinado umbral de temperatura y disponer de un histórico para analizarlo. De cara a obtener estos datos, se valoró el empleo de sensores de contacto o infrarrojos. Los primeros se desecharon por la dificultad y riesgo de instalarlos, mientras que los segundos requerían múltiples sensores apuntando con mucha precisión. Además sería necesario estimar un punto de referencia para un embarrado, a pesar de que se puede dar su rotura por varias partes.

La solución empleada finalmente fue una cámara termográfica que también se puede ubicar a distancia como los sensores de infrarrojos pero es capaz de abarcar un área mucho más extensa en una sola adquisición, tal como se puede ver en la Figura 5.28. Aunque la precisión de una cámara termográfica es suficiente para la

detección de puntos calientes como se quiere hacer, para mejorar la precisión de la medida se instalará también un sensor de temperatura y humedad ambiental. Con esta medida adicional, se pueden ajustar los datos obtenidos en la cámara.

Figura 5.28. Vista de la cámara instalada en Granadilla. En una sola imagen es posible monitorizar todos los embarrados de la subestación.

Sobre la imagen recogida se definirán una serie de áreas a monitorizar, que será aquellas en las que haya embarrados, y sobre las que se revisará que la temperatura no supera un umbral determinado. Además de un umbral de temperatura para reemplazo, se aprovechará la monitorización para introducir un umbral de alerta previa unos cuantos grados por debajo, que ayude a reaccionar mejor ante el potencial reemplazo posterior. De esta manera, se podrá mejorar la planificación de la reparación para un mantenimiento preventivo más eficaz que el previo a la instalación de esta solución.

5.5.2. Diseño e implantación

En esta solución hay tres puntos críticos a considerar: el procesamiento de los datos térmicos, la conexión con el SCADA y la generación de alertas.

Los dos últimos puntos, de forma similar a otros casos de uso, recaerán en la arquitectura IIoT. Sin embargo, para el procesamiento de los datos térmicos se buscará importar un desarrollo de terceros, reduciendo sensiblemente los tiempos de desarrollo de la implantación. Por ello, el esquema que se propone sería el que se ve en la Figura 5.29.

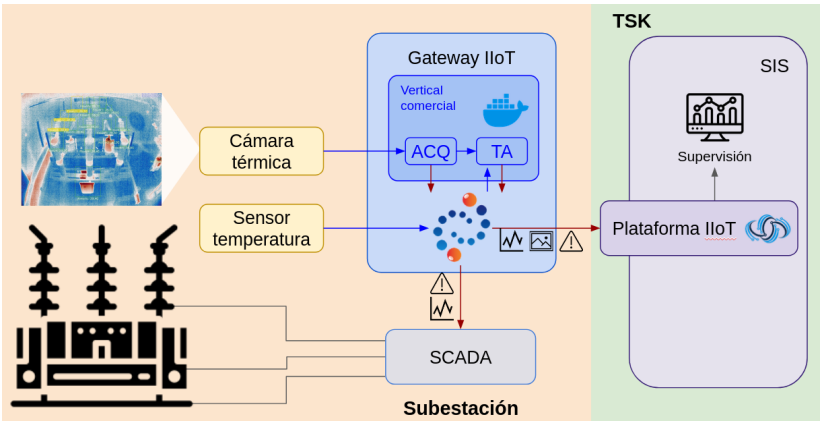


Figura 5.29. Diagrama funcional de la solución aplicada en Granadilla. El algoritmo de procesamiento de datos térmicos se incorpora como parte de la arquitectura IIoT a pesar de ser un desarrollo de terceros.

Aunque simplifica la solución, aplicar una solución de terceros como la que se plantea también puede suponer un reto. El procesamiento incorporado, denominado *Thermographic Analysis* por el proveedor (en adelante solo *TA*), tenía una funcionalidad muy rígida en cuanto a la integración con terceras soluciones. Gracias a la versatilidad de la capa edge proporcionada por Kura, fue posible paliar estas dificultades de manera que *TA* pasó a formar parte íntegra de la arquitectura IIoT. Las cuestiones afrontadas fueron:

- Despliegue: *TA* está preparado como un contenedor de Docker para su instalación, que facilita las puestas en marcha. Dado que Kura ya dispone de un sistema de gestión integrado con Docker, fue posible asimilar el despliegue con una sencilla configuración en la web de Kura.

- Adquisición de datos: como la obtención de datos térmicos depende bastante de cada cámara y su SDK, *TA* no incorpora esta obtención, sino solamente el procesamiento desde una imagen recogida por el sistema de archivos del OS. Para poder obtener estos datos, se desplegó desde Kura otro contenedor proporcionado por el desarrollador también para realizar esta etapa, denominado *Camera Acquisition* (*ACQ*).

- Salida por base de datos: los datos generados por *TA* se devuelven escribiéndolos en una base de datos local de tipo SQLite. Aunque esta base de datos no está soportada nativamente por Kura, se desarrolló un plugin específico que integraba las particularidades de SQLite con el resto de la lógica de gestión de base de datos de Kura, que es independientemente de la base de datos empleada. De la misma manera, se dispuso una conexión desde Kura con el sistema de archivos para obtener también las imágenes térmicas.

- Aportación de temperatura de referencia: para corregir los datos termográficos, *TA* permite la introducción de la temperatura ambiente en tiempo real desde una petición REST que realiza el algoritmo. Aunque es posible definir una dirección, el formato no es personalizable. Dado que este valor ya se integra desde Kura por Modbus, se abrió en Kura una llamada para que *TA* pudiera acceder a los datos en el formato necesario.

- Configuración: la configuración de *TA* se compone principalmente de dos archivos en formato JSON, uno para la definición de las regiones a analizar (definidas por un rectángulo) y otro para el flujo de trabajo general (frecuencia de trabajo, rutas en el OS, etc.). El envío y gestión de estos ficheros se puede realizar a través de Kura gracias a su gestor de despliegue de archivos.

De no disponer de un software tan versátil como el de Kura el desarrollo se

habría dificultado en este punto. En un SCADA tradicional se pueden abordar estas cuestiones pero las estrategias habituales pasan por crear un servicio ad hoc de integración y aumentar la complejidad y mantenimiento de la parte software.

La generación de alertas, por su parte, se implementó íntegramente en Kura. Aunque Kura dispone de algunas herramientas nativas para gestionar eventos o alertas, son difícilmente escalables ya que están pensadas para actuar variable a variable. Aunque en este caso de uso podría ser asumible, se desarrolló un componente nuevo que permitiera solucionar este problema para un amplio grupo de variables con configuraciones más simples. Así, se planteó la generación de alertas basadas en el tipo de dato, su nombre o una porción del mismo, comparando una referencia con el valor de la variable, en este caso para detectar si superaba un umbral superior. También se incorporó un margen de tiempo para que picos puntuales no fueran motivo de alerta necesariamente.

Los datos (temperaturas y alertas) obtenidos se debían exponer localmente y enviar remotamente. Para la integración local, se desplegó desde Kura un maestro Modbus al que el SCADA se conectó para obtener los datos a demanda y en tiempo real. El envío remoto, si bien similar a otras soluciones, se destaca que se realizó no solo de los datos numéricos sino también de la información termográfica empleada. Esto se hizo para completar de forma más robusta los datos ya analizados, de forma que se pudiera reprocesar en caso de ser necesario para afinar mejor el sistema de mantenimiento predictivo que se prevé desplegar.

Naturalmente, la instalación de estos equipos (cámara, sensor y PC), se ubicarán en torno a los embarrados, conforme a lo mostrado en la Figura 5.30. Para la captura de datos térmicos se empleó una cámara FLIR x95. En lo relativo al PC, se hacía necesario un equipo que cumpliera con margen los requisitos de Kura para poder alojar también Docker y el procesamiento de imágenes. Para ello se seleccionó el LEC-7230-M ([2]), un PC compacto de bajo coste que tiene características comparables a cualquier ordenador de sobremesa

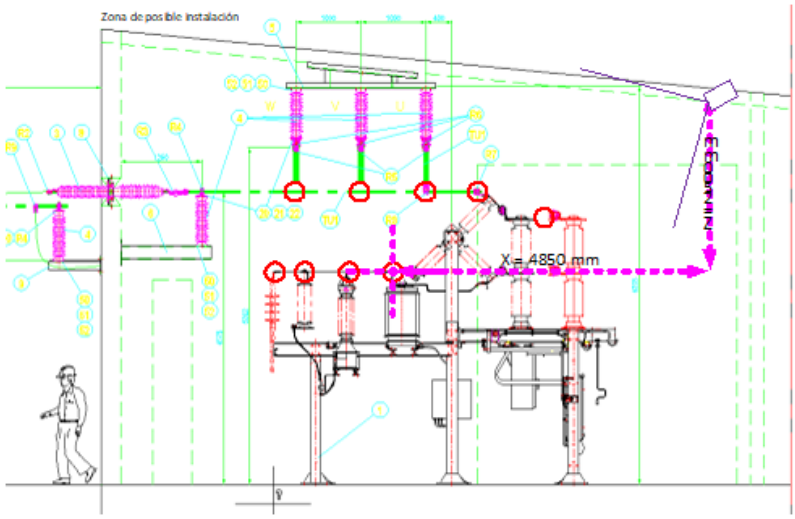


Figura 5.30. Plano de instalación de los equipos de Granadilla. Junto con la cámara se instaló un sensor de temperatura, el PC industrial y un módem.

con la ventaja de estar preparado para entornos industriales.

Puntos críticos

Durante la puesta en marcha y posterior operación, se encontraron algunas incidencias que tuvo que absorber la arquitectura IIoT. Se destacan las siguientes.

Cobertura deficiente Aunque la cobertura 4G en la zona de la subestación era relativamente buena, se trata de una región muy turística. Esto genera un exceso de carga en la red varios meses al año cuando hay más afluencia de gente. Tras la implantación, se descubrió que en estas épocas la tasa de transferencia bajaba mucho, sobre todo para la subida de datos, y la conexión se volvía mucho más inestable.

Para mejorar la eficiencia de la solución, se recurrió a la configuración remota y se modificó tanto el procesamiento integrado como la propia lógica de Kura, de forma que las imágenes se procesaran al mismo ritmo, pero sus datos y sobre todo las imágenes no se enviaran a SIS en cada procesamiento, sino con un periodo

mayor. De esta forma, se logró mantener un comportamiento más controlado y de tiempo real que si se intentara enviar constantemente muchos datos.

Se hace notar que todo esto no afectó la funcionalidad en la propia subestación, ya que el procesamiento ejecutado es autónomo y la integración local con el SCADA no necesita internet, por lo que el núcleo de la aplicación se mantuvo inalterado frente a esta situación.

Desalineamiento de las cámaras En cualquier procesamiento de visión artificial, es crucial mantener la condición de las cámaras instaladas en el momento de la calibración para el éxito del algoritmo de detección. En este caso, el parámetro fundamental para el resto del cálculo es la región en la que se deben detectar las temperaturas extremas. Estas regiones se delimitaron tras la instalación y anclaje de la cámara y se aseguró que no se pudiera mover por accidente.

Sin embargo, debido a una tarea de mantenimiento del tejado de la zona de operaciones dos años más tarde, la cámara tuvo que ser desinstalada y vuelta a instalar. Aunque se apuntó hacia el mismo sitio al que estaba apuntando antes, la nueva posición tenía un amplio desplazamiento hacia abajo respecto a la puesta en marcha y las regiones definidas dejaron de coincidir.

Al percibir que los datos obtenidos no se correspondían con valores habituales, el centro de control de la subestación notificó a TSK el problema sin advertir de la operación realizada. Por fortuna, aunque en un SCADA tradicional las imágenes procesadas no se estarían recogiendo (y de hecho el SCADA de la subestación no es capaz de procesarlas), su integración en la arquitectura IIoT hizo posible detectar el fallo. Se confirmó que efectivamente había habido manipulación de la cámara y se definieron las nuevas regiones sobre la nueva imagen obtenida sin necesidad de desplazarse a la instalación.

Orquestación de servicios locales El despliegue de varios procesamiento anexos al software principal del dispositivo edge, Kura, supone en general cierto aumento de la complejidad y sobre todo tiene impacto en el mantenimiento: actualizaciones, configuración, supervisión de los procesos, etc.

No obstante, el empleo de Kura permite que estos servicios puedan correr totalmente incorporados a la arquitectura IIoT. La capacidad de Kura para asimilar los contenedores de Docker, aprovisionar la configuración o incluso desplegar nuevas versiones o procesamientos hace que toda la gestión resulte mucho más transparente para el mantenimiento. Esto también significa que se pueden hacer estas tareas de forma remota aprovechando la conexión ya existente entre edge y la plataforma IIoT.

Desarrollo de componentes ad hoc Debido al caso de uso específico y la integración con el servicio *TA*, fue necesario desarrollar algunos componentes que a priori no son útiles más allá de esta aplicación. Sin embargo, gracias a la arquitectura de Kura y su capacidad para definir flujos de trabajo, se pueden orientar los desarrollos de forma que sean más reusables. Así, por ejemplo, sobre los desarrollos empleados se plantearon algunas mejoras que no eran estrictamente necesarias pero sí contribuyen a abarcar muchos más casos de uso con el mismo software:

- Gestión de archivos: era necesario acceder al sistema de archivos como una fuente de datos para las imágenes. Sin embargo, aprovechando el componente ya empleado para usar una base de datos como espacio de almacenamiento intermedio para los envíos, bastó con incorporar a esta lógica una definición de almacenamiento y dato más abstracta para que el sistema de archivos fuera solo un medio más y por tanto aprovechar el resto de las funcionalidades.
- Exposición de métricas localmente: si bien fue necesario desarrollar un

componente para exponer localmente métricas por API REST porque *TA* era estricto en el formato, el desarrollo se preparó para mantener cierta configurabilidad en la forma de exponer los datos. Así pues, el componente que debía ser específico se concibió como un acceso REST versátil para servir datos a terceros en otras aplicaciones, dando opción a modificar formato, URL o autenticación de acuerdo a las capacidades del cliente.

- Generación de alertas: se desarrolló un componente para generar alertas que debía advertir en dos niveles (según si era solo una advertencia o ya una alerta) sobre un umbral superior. Sobre este componente se incluyeron algunas configuraciones para adaptarse a otras situaciones, por ejemplo la capacidad para saltar solo al cabo de un tiempo, la opción de activar o desactivar un nivel de advertencia intermedio, el tipo de comparación a emplear para saber si se estaba o no en alerta, y el uso de otra variable en vez de un valor fijo para la referencia de la alerta.

Gracias a la modularidad de Kura todo el desarrollo incorporado en esta solución, se pudo incorporar al conjunto de herramientas para otros casos de uso. En líneas más generales, el uso de un software modular ayuda a que cualquier desarrollo realizado sume siempre a futuras aplicaciones.

5.5.3. Rendimiento de la solución

La medición en continuo implementada permite una monitorización en tiempo real mucho más eficaz para el mantenimiento de los embarrados y permite crear un histórico con el que se puede pasar de un mantenimiento basado en eventos o preventivo a uno predictivo. Gracias a Kura además se logró integrar un software de terceros evitando la necesidad de implementar un desarrollo más avanzado sin incrementar la complejidad y la manutención de la propia solución desplegada.

En mayor detalle las principales ventajas obtenidas serían:

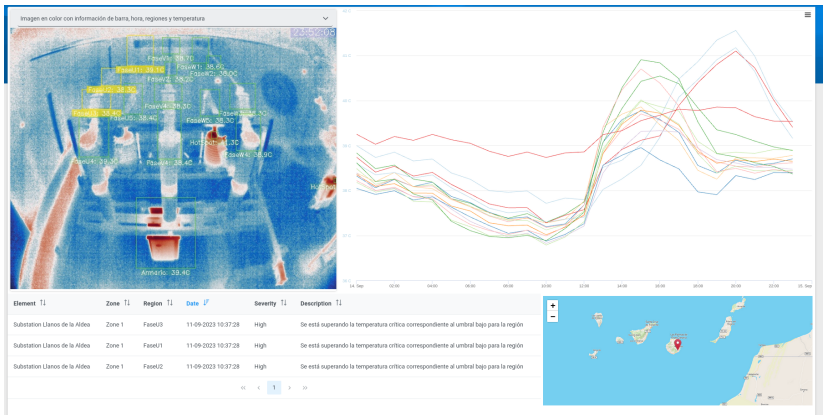


Figura 5.31. Ejemplo de visualización resumen de los datos de la planta de Granadilla en SIS. Se presenta en un mismo panel la imagen obtenida con el procesamiento solapado, los datos leídos y la ubicación de la planta.

- Integración horizontal: a través de Kura es posible realizar una integración local en planta y con un desarrollo externo sin incrementar la dificultad de mantener la solución.

- Fiabilidad: tanto la precisión como la completitud de los datos está garantizada para poder crear un registro histórico fiable del estado de los embarrados, fundamental para cualquier análisis sobre su duración y mantenimiento.

- Análisis avanzado: la composición de un histórico permite a posteriori el desarrollo e implementación de algoritmos para mejorar la duración y el procedimiento de mantenimiento de los embarrados de la subestación.

- Procesamiento edge: todo el procesamiento necesario para la solución de monitorización continua se realiza en el propio gateway IIoT, lo que aligera la plataforma IIoT, el tráfico de datos y sobre todo mejora mucho los tiempos de respuesta al dar una información más rápida. Además contribuye a la robustez de la solución frente a pérdidas de conexión.

- Desarrollo modular: la implementación de desarrollos nuevos considerados

en este caso de uso ha permitido incorporar todas las funcionalidades del algoritmo externo al mismo tiempo que se generan funcionalidades reaprovechables para otras soluciones.

- Evolucionabilidad: gracias a la supervisión remota de los procesamientos y resultados, es posible controlar desde SIS (Figura 5.31) el flujo de trabajo y mejorarlo continuamente.
- Gestión de eventos: la generación de alertas o eventos a partir de reglas fácilmente configurables permite a operarios y supervisores obtener una visión más clara y rápida del estado de la planta sin introducir complejidad.

En resumen la solución aporta no solo una mejora en el día a día del mantenimiento de la subestación, sino también abre el camino para la implementación de un sistema de mantenimiento integral más completo con datos para empezar a alimentar un modelo predictivo y así mejorar en la pirámide del mantenimiento (Figura 2.3).

5.6. Monitorización de plantas fotovoltaicas

Las plantas fotovoltaicas son plantas relativamente sencillas ya que la producción de energía eléctrica a partir del sol es directa mediante células fotovoltaicas y el resto de elementos adicionales se centran en la adecuación de esta energía eléctrica a la de la red. Una planta fotovoltaica típica consta de:

- Panel fotovoltaico: produce la energía eléctrica directamente a partir de la radiación solar en corriente continua.
- Seguidor: mecanismo utilizado en algunas plantas para rotar el panel y enfocarlo mejor hacia el sol. La rotación puede ser en el eje horizontal para adecuarse a la altura solar en distintos meses del año o (menos frecuente) en vertical para seguir al sol a lo largo de las horas del día.

- String: agregan la producción de varios paneles solares para alcanzar una tensión adecuada para su inversión
- Inversor: transforma la corriente continua saliente de los paneles en corriente alterna, para inyectarla a la red.
- Contador: mide la cantidad de energía eléctrica que se vierte a la red eléctrica. Son fundamentales para la cotización de la energía producida.
- Estación meteorológica: monitoriza las condiciones ambientales de la planta. Las variables más habituales son temperatura ambiente, temperatura del panel y radiación solar.
- Baterías: utilizadas para proporcionar una producción más estable y adecuarla a las necesidades y exigencias de la red.
- Controlador de carga: centralita que dirige la energía hacia o desde la batería en función de la producción.

La mayoría de los sistemas no tienen medios de actuación realmente. Es por eso que es un escenario relativamente sencillo desde el punto de vista de control industrial. Sin embargo, sí representa un reto en lo relativo a monitorización y comunicaciones.

Las plantas fotovoltaicas ocupan grandes extensiones de tierra y suelen estar en zonas aisladas y con mucha exposición solar. Al ser de gran longitud, la distribución de sensores en los puntos clave (como superficie de los paneles) o la medida de parámetros desde los equipos industriales requiere una inversión muy poco rentable en instalación y cableado. Las tecnologías inalámbricas de área local presentadas durante el Capítulo 3 pueden proporcionar los medios para recoger esta información. El aislamiento de la planta suele hacer poco práctica la monitorización in situ de la misma, sobre todo si se deben controlar varias plantas a la vez, una situación muy común con las de tamaño mediano o pequeño.

Sin embargo, la comunicación remota también puede suponer un reto al darse frecuentemente en sitios con mala conexión a Internet. La integración a Internet de IIoT puede proporcionar una capa segura y fiable de recolección de datos para su consulta remota.

Un punto adicional que se deberá tener en cuenta al diseñar la solución es la escalabilidad. Existen plantas fotovoltaicas de muchos tamaños y la solución debe ser igualmente rentable en plantas pequeñas sin comprometer su rendimiento para plantas grandes. Esto además fuerza un requisito de coste bajo para poder ser escalable y sobre todo porque la mayoría de las plantas fotovoltaicas son relativamente pequeñas.

Se destaca que en esta aplicación, dada la producción de energía de la planta y por tanto ubicuidad de electrificación, no es necesario disponer de funcionamiento por batería, más allá de un refuerzo ante caídas. Sin embargo, el requisito del bajo consumo estará igualmente presente porque un mayor consumo significa un impacto directo en la productividad de la planta.

Los requisitos que se valoran para esta solución serían:

- Ciberseguridad: al no haber mucho margen para el control en la planta, no existe un peligro inmediato una brecha de seguridad, pero sí estaría comprometida la producción y la confidencialidad.
- Versatilidad: aunque el formato de las plantas fotovoltaicas suele ser bastante similar, para ser escalable la solución a aplicar debe poder adaptarse a múltiples casos de uso a pesar de las incidencias que se puedan encontrar.
- Mantenimiento: las plantas fotovoltaicas suelen tener un mantenimiento bajo, por lo que es necesario que la solución de monitorización no suponga una carga adicional relevante.
- Integración: principalmente vertical, ya que la arquitectura IIoT debe poder

integrarse con los equipos nombrados aguas abajo para la supervisión de la planta y con sistemas de gestión remotos, pero en ninguno de los dos casos hay apenas bidireccionalidad.

- Conectividad: tanto por la topología de las plantas fotovoltaicas como por los protocolos de los equipos a integrar, es necesario tener una alta conectividad para adaptarse a los casos de uso y que la solución sea realmente escalable.
- Integridad: dado que en esta aplicación la solución IIoT sustituye por completo la función del SCADA, la veracidad y precisión de los datos es crítica.
- Rango industrial: los equipos desplegados deben ser capaces de soportar las condiciones industriales a las que se exponen. En este caso, la temperatura y el polvo es siempre un problema. También podría darse exposición a condiciones ambientales y precipitaciones.
- Coste: para resultar rentable en plantas de pequeño tamaño, el coste de la solución final debe ser bajo o por lo menos que se pueda reducir en casos concretos donde no sea necesarias ciertas características.

5.6.1. Caso de uso

Entre los servicios de operación y mantenimiento que ofrece TSK junto con sus obras está el de centro de control para supervisión de plantas fotovoltaicas. Este servicio se ofrece tanto a pequeños particulares como grandes productores con la idea de proporcionar una supervisión de la planta constante y una gestión de incidencias remota. Bajo este servicio, además se contemplan algunas plantas ubicadas en España (Figura 5.32) que son propiedad directa de TSK como parte de su diversificación de inversiones y también como base de trabajo para investigación y validaciones.

Figura 5.32. Ubicación de las plantas fotovoltaicas monitorizadas por TSK. La mayoría se encuentran en Castilla León, Extremadura y Andalucía (fuente [101]).

Con anterioridad al desarrollo de SIS, TSK subcontrató el servicio SolarWatch (Figura 5.33) de la empresa NextSolutions que desplegó en esta y otras plantas de características similares. Sin embargo, esta solución tenía bastantes ineficiencias y un alto coste, ambos debidos principalmente a la baja adaptabilidad de la solución. A continuación se detallan las características y deficiencias en tres de las plantas que pueden resultar más significativas en este sentido.

Hinojal Hinojal es una planta de 4 MW ubicada en la provincia de Cáceres con una extensión de algo más de 12 hectáreas (Figura 5.34). Está compuesta por un total de 192 strings, 32 inversores, 34 contadores (32 de Baja Tensión, uno de servicios auxiliares, y uno de Media Tensión) y una estación meteorológica.

Aunque la integración de los strings resultó directa con la solución de SolarWatch, para poder integrar los inversores fue necesario unificarlos en tres grandes buses de comunicación con un PLC en la cabecera de cada

Figura 5.33. Armario de monitorización de SolarWatch. Esta solución fue contratada inicialmente por TSK para la supervisión de las plantas fotovoltaicas desde el centro de control.

uno para traducir el protocolo del inversor (S7 de Siemens) a uno de los protocolos soportados, en este caso Modbus TCP. Esto impactó seriamente en la adquisición en tiempo real, retrasando mucho los tiempos de lectura. También añadió complejidad al esquema de monitorización, introduciendo un punto de fallo que además resultó ser bastante reincidente y difícil de solucionar en la lectura del PLC con los inversores.

Figura 5.34. Vista de pájaro de la planta fotovoltaica de Hinojal (fuente [102]). Con un total de 32 inversores tiene una capacidad de producción de hasta 4 MW.

La integración con los contadores y la estación meteorológica no fue posible realizarla con esta solución comercial. En el primer caso se debía a que el protocolo de telemedida establecido en España para los contadores, el *IEC 60870-5-102* (en adelante abreviado como IEC-102), no estaba disponible entre los conectores de SolarWatch. En el segundo caso porque la estación meteorológica tenía un protocolo propietario que no era posible incorporar a la solución. Dado que para ambos casos ya existían otras herramientas alternativas para realizar la monitorización (una aplicación desarrollada por los propios proveedores de la estación meteorológica y una general para telemedida), se renunció al desarrollo y

coste adicional que supondría tratar de integrar SolarWatch con estos elementos. Sin embargo, la falta de centralización de la información dificulta mucho la monitorización de la planta fotovoltaica, también complica los posibles análisis e incrementa los costes.

A pesar de existir una red ethernet que permite comunicar con todos los equipos de Hinojal desde un mismo punto, se instalaron un total de tres armarios completos de SolarWatch, cada uno con su PC y módem para cubrir toda la monitorización. Fue necesario hacerlo así dado que la capacidad de procesamiento y el tipo de licencia no permitía hacer toda la lectura con menos.

Iruela Iruela es una planta de 3,5 MW y una superficie de poco más de 8 hectáreas ubicada en la provincia de Jaén (Figura 5.35). Se compone de 18 instalaciones independientes conectadas a la red y compuestas a su vez por un inversor y un contador. También cuenta con una estación meteorológica común a toda la planta. Al contrario que en Hinojal, en Iruela los inversores no son todos del mismo fabricante, sino que coexisten algunos de la empresa Siemens y otros de Saft.

Al realizar la instalación de SolarWatch de nuevo fue necesario incorporar PLC para algunas de las comunicaciones. De nuevo se dio la incapacidad para integrar la estación meteorológica, ya que pertenecía al mismo fabricante que la anterior y los contadores.

En el caso de Iruela, se desplegó un total de seis armarios de monitorización, uno por caseta, debido a que la comunicación serie empleada para interactuar con los buses de comunicaciones no permitían cubrir las distancias entre las casetas. El sobrecoste que supuso incorporar esta solución en la planta resultó ser mucho mayor que en la anterior debido al recurrente a asociado a los equipos desplegados.

Jarandilla Jarandilla es una planta menor ubicada en Cáceres con 1,75 MW de potencia instalada y en torno a 6 hectáreas. De forma similar a Iruela, la planta

Figura 5.35. Vista de pájaro de la planta fotovoltaica de Iruela (fuente [102]). Construida en un terreno escarpado y con mucha heterogeneidad de marcas, esta planta puede producir hasta 3,5 MW.

se encuentra repartida en 8 instalaciones independientes con su propio contador. Tiene un total de 92 inversores de la marca Fronius repartidos desigualmente entre ellas, debido a que no todos los inversores trabajan a la misma potencia. Destacar que esta planta es la única de las tres que no dispone de una red de comunicaciones, sino solo la parte eléctrica.

A pesar de ser un protocolo propietario, SolarWatch disponía de integración con Fronius y estableció un total de ocho buses de comunicaciones serie agrupados por las distintas instalaciones aprovechando la cercanía entre los inversores de cada una. Esto, sin embargo, obligó a instalar ocho armarios de monitorización, incrementando mucho tanto el coste como el recurrente.

Un detalle diferenciador de esta planta es que los contadores no disponen de módem 3G como Hinojal o Iruela, sino que solo es posible comunicarse con ellos empleando el protocolo GSM. Esta comunicación es comparable a una llamada telefónica sobre la que se envían datos digitales traducidos en lugar de voz.

Este tipo de comunicaciones están cada vez más en desuso, lo que incrementa la dificultad de crear una solución.

Como puede verse, la solución de SolarWatch tiene un rendimiento deficiente para la topología de plantas debido a algunas peculiaridades y a que tiene una adaptabilidad muy baja.

5.6.2. Diseño e implantación

Para mejorar este sistema de monitorización, se realizó una migración a la arquitectura IIoT de SIS, tratando de mejorar la eficiencia de las plantas con la flexibilidad de esta nueva solución. A este respecto es importante apuntar que la función de esta arquitectura va a ser, al contrario que en los casos de uso anteriores, una sustitución completa del SCADA, por lo que la solución ha de ser completa.

Al incorporar una capa edge adaptable, se buscaba sobre todo que un solo equipo fuera capaz de cubrir la supervisión de una planta completa. Esto no significa que no se puedan recurrir a otros equipos de la capa de percepción que permitan centralizar la información en este dispositivo. En la Figura 5.36 se puede ver un planteamiento conceptual de esta propuesta.

El flujo de trabajo, tal como se propone, sería el mismo para las tres instalaciones y también que el planteado en la Sección 5.1.2, y la mayor parte de la complejidad se traslada a la integración de los distintos equipos presentados. En estos se plantea por un lado la cuestión de los protocolos disponibles y por otro el medio de comunicación. Se empezarán analizando los protocolos:

1. Strings: la única planta donde se monitorizan es en Hinojal, donde están accesibles por Modbus RTU.

2. Inversores: los inversores de las instalaciones citadas emplean tres protocolos distintos, a saber Modbus (tanto TCP como RTU), Siemens S7 y Fronius

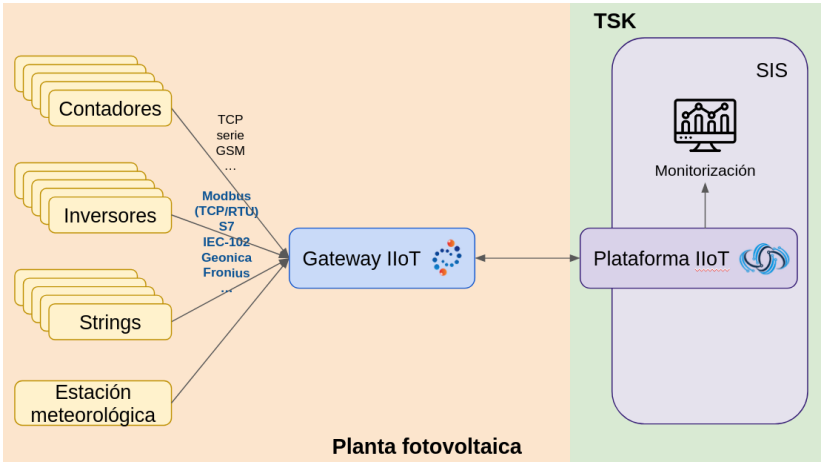


Figura 5.36. Diagrama funcional de las plantas fotovoltaicas. Cada instalación tendrá particularidades pero se mantendrá este esquema en líneas generales.

(propietario).

3. Contadores: todos los contadores se comunican a través del protocolo IEC-102.

4. Estación meteorológica: las dos estaciones meteorológicas a monitorizar se pueden leer mediante el protocolo propietario de Geonica Meteodata.

De los protocolos citados, solo Modbus y Siemens S7 están disponibles en el Marketplace oficial de Kura. De los otros, aunque se tiene documentación del fabricante (o el estándar en el caso del IEC-102), no se encontró ninguna implementación. En consecuencia, fue necesario implementarlos específicamente para estos casos de uso. Destacar que en el caso del IEC-102 aunque no existía ninguna librería para Java, sí se identificaron publicadas en otros lenguajes, lo que facilita la creación de una propia y las pruebas asociadas.

En lo relativo a los medios de conexión, se han analizado cuatro escenarios distintos que serían: buses serie, conexiones por 3G, conexiones por GSM y planta sin red local. Se detalla a continuación en qué puntos se dan y cómo se abordaron para habilitar la comunicación desde el dispositivo edge:

- Bus serie: existentes en los strings de Hinojal, en los inversores de Iruela y en algunos de los inversores de Jarandilla. En estos casos se buscará trasladar esta conexión a TCP para habilitarla en la red local de la planta. Para ello se emplearán conversores de RS a TCP como Circutor TCP2RS+, el Moxa NPort 5110 o el USR-TCP232. Todos ellos son equivalentes y traducen cualquier mensaje en RS-232 o RS-485 a TCP/IP. Incluso incluyen una traducción más específica para pasar de ModbusRTU a ModbusTCP, que facilita la integración para algunas implementaciones de Modbus (ya que no es trivial leer por TCP/IP con mensajes de ModbusRTU). En particular fue necesario introducir uno de estos en Hinojal y siete en Iruela; el caso de Jarandilla se analiza más abajo pues entraña una dificultad adicional.

- Conexiones por 3G: presentes en los contadores de Hinojal e Iruela. La conexión a estos equipos es directa al estar disponibles para cualquier equipo conectado a Internet, como es el caso de la capa edge implementada.

- Conexiones por GSM: se dan en los contadores de Jarandilla. Aunque están disponibles remotamente, no sirve una conexión estándar a Internet como en el caso anterior, sino que es necesario un módem específico y una conexión única con él. Para lograr esta lectura se desplegó un solo equipo remoto (en las oficinas de TSK) equipado con el módem adecuado. Este equipo instalado inicialmente para Jarandilla, pudo escalar su uso para otras plantas conectadas posteriormente que también funcionaban por GSM.

- Sin red local: este es el caso de la planta completa de Jarandilla, en la que no hay ninguna conexión ethernet cableada. Esto plantea dos problemas, por un lado la conexión a Internet para el envío de datos y por otro la conexión a los distintos equipos de la planta, en particular los inversores. La conexión a Internet se pudo paliar fácilmente con un módem 4G conectado al dispositivo edge. Para cubrir la conexión entre los equipos locales, se

desplegó una solución LoRa. De esta manera, el equipo edge actuando de gateway LoRa podía acceder inalámbricamente a los buses serie en los que estaban los distintos inversores. Dado que el equipo edge se instaló conectado a uno de los buses serie, fue necesario adquirir siete nodos LoRa para integrar el resto de los buses.

En base a estas consideraciones, se pueden construir ya los tres casos de uso propuestos. En las Figuras 5.37, 5.38 y 5.39 se presentan las topologías concretas, que mantienen igualmente el flujo presentado en la Sección 5.1.2 y el concepto planteado en la Figura 5.36

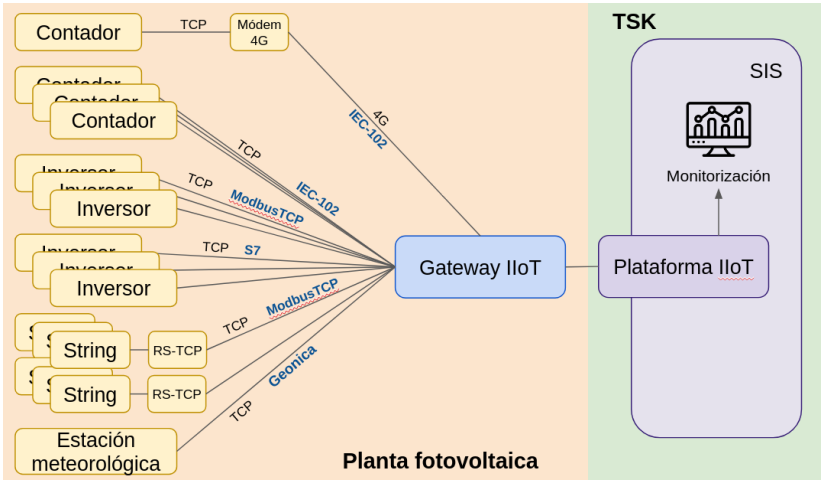


Figura 5.37. Diagrama funcional de la planta de Hinojal. El principal reto de esta planta es la gran cantidad de equipos a monitorizar a través de la red de planta.

Como se puede ver, en las plantas de Hinojal y Iruela, se desplegó simplemente un PC industrial de bajo consumo y mantenimiento como el Moxa UC-2112 ya presentado. Sin embargo, en Jarandilla fue necesario emplear un equipo que pudiera actuar como gateway LoRa y un router. El PC seleccionado fue el Multitech DT Conduit y como router un Sierra LS-300. En la Figura 5.40 se puede ver un armario tipo de los empleados para las soluciones fotovoltaicas

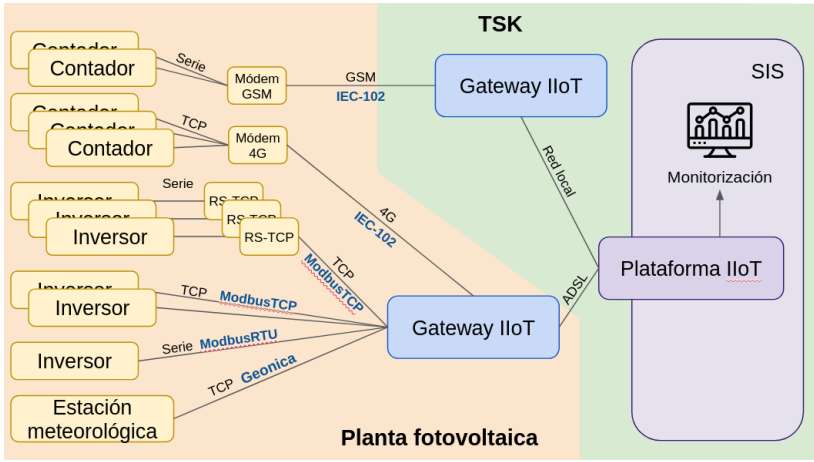


Figura 5.38. Diagrama funcional de la planta de Iruela. El principal reto de esta planta es la variedad de los equipos a monitorizar y sus medios de conexión.

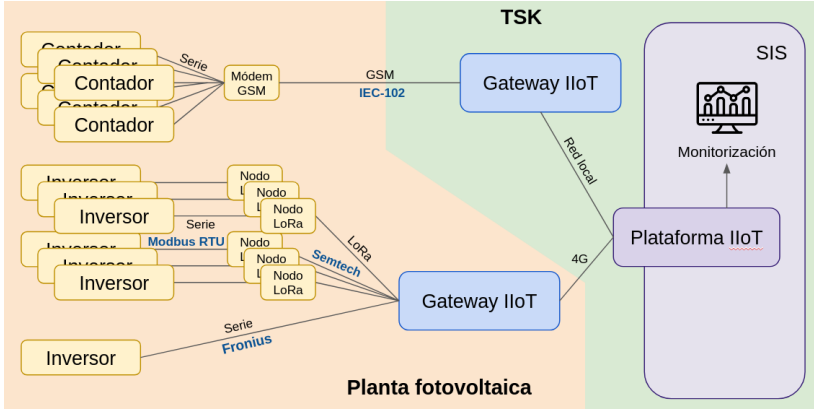


Figura 5.39. Diagrama funcional de la planta de Jarandilla. El principal reto de esta planta es la ausencia total de red ethernet local y la necesidad de conecatar inalámbricamente los distintos elementos.

sobre el que se plantearon las modificaciones pertinentes como la incorporación del Multitech o la retirada del módem.

Puntos críticos

Ya se ha visto que la estructura en cada planta fotovoltaica requería cierto estudio específico. Además de las cuestiones ya presentadas, se encontraron algunas incidencias en el curso de la validación.

Variedad de fabricantes y protocolos En plantas medianas como las planteadas aquí no es extraño que se encuentren mezclados modelos y fabricantes. De hecho no solo se dan inversores y contadores de distinta marca, sino también se da con frecuencia que haya incluso inversores dispares (véase el caso de Iruela). Esto no supone un problema a nivel eléctrico ya que el modo de funcionamiento es muy compatible y en una planta mediana puede suponer una ventaja ya que no siempre es posible acceder a descuentos por el lote de compra, como sí puede pasar en las grandes instalaciones.

Esto supone en primer lugar una gran variedad de protocolos de comunicación, algunos incluso propietarios, y en segundo lugar un mapa de datos no siempre compatible entre los distintos equipos. Sobre esto último, Kura proporciona una capa de abstracción, ya mencionada en la Sección 5.1.2, que permite interactuar con los datos independientemente del modelo, protocolo o proceso de lectura empleado a bajo nivel. En cuanto a los protocolos, a pesar del amplio catálogo de protocolos disponibles para Kura, no fue posible realizar todas las lecturas necesarias con ellos. No obstante, al estar abierto a ampliaciones particulares, se desarrollaron drivers específicos que se conectasen a través de la capa de abstracción de Kura. Estos desarrollos lograron ampliar la base de equipos integrados en la solución respecto a los disponibles en la de SolarWatch, que no era posible ampliar a demanda.

Figura 5.40. Armario de monitorización para las plantas fotovoltaicas. Además del gateway IIoT, dispone de alimentación, batería de refuerzo, módem y un módulo de entradas y salidas auxiliar para usos varios.

Lectura contadores GSM En la actualidad, la mayor parte de la telemedida de contadores en España se realizar a través de la red 3G/4G, sin embargo, hay muchos equipos que emplean la red 2G con conexión GSM, que se puede comparar a una conexión de datos codificada por llamada de voz. Este tipo de conexiones no se pueden hacer con una conexión normal a Internet ya que es necesario tener control único sobre el módem desde el que se inicia la comunicación y establecer una conexión punto a punto.

Dado que este tipo de lecturas resulta más compleja, menos escalable y está cada vez más en desuso muchos fabricantes o proveedores de soluciones fotovoltaicas no tienen interés en desarrollarlas. Dado que Kura permite acceder al hardware con pleno control y desarrollar componentes propios, es posible afrontar esta problemática.

De cara a simplificar la parte hardware, se preparó un único punto de lectura GSM para todos los contadores, ya que en esta monitorización la cercanía al contador no proporciona ninguna ventaja. Este punto se ubicó en las instalaciones de TSK y se usa para acceder a todos los contadores de este tipo abarcados por la solución SIS. De nuevo gracias a la abstracción de drivers y assets proporcionada por Kura, no fue necesario más que resolver la comunicación serie entre el PC y el módem que abre la conexión y se puede mantener como funcionalidad dentro del resto de la solución sin rehacer el protocolo IEC-102 ni la definición de las variables leídas de los contadores.

Ausencia de red Ethernet Lo que aquí se ejemplifica a través de la planta de Jarandilla, no es una situación extraña en una planta fotovoltaica. Muchas plantas pequeñas y medianas se crean únicamente en torno a la producción hacia el contador, por lo que establecer una red de comunicación resulta innecesario y simplemente se provee de conexión a esa capa final que es la fundamental para la monetización de la inyección de energía en la red. Esto supone un reto para

los SCADA tradicionales ya que la base de muchas soluciones es la conexión cableada, preferentemente por Ethernet.

La flexibilidad de una arquitectura IIoT habilita la creación de sistemas de supervisión y control sin necesidad de recurrir al cableado y sin tener que integrar un vertical cerrado. En este caso, se establece una red LoRa dado que no se tiene mucha frecuencia de lectura ni mucha densidad de datos. Para gestionar esta red, se despliega un hardware que incorpora la conectividad y se adhiere la funcionalidad necesaria a través de un servicio nuevo de Kura. Estos equipos pasan así a tener una conexión directa dentro de la arquitectura IIoT, permitiendo un control completo de ellos. Como también se prepara la integración de los datos mediante un estándar de facto de la tecnología LoRa, el desarrollo se puede exportar a otras soluciones como parte del servicio integral de monitorización.

Instalación en Multitech Tal como se ha planteado en el anterior punto, para el control de la red LoRa fue necesario adquirir un nuevo equipo sobre el que no se había validado aún Kura ni los procesamientos empleados. Sin embargo, dado que los requisitos de Kura no son muy estrictos al ser solo necesario que sea un equipo Linux y algunas otras cosas comunes en la mayor parte los Linux, fue bastante sencillo encontrar entre los productos comerciales y de uso habitual uno que fuera válido, el Multitech.

Una vez considerados estos primeros requisitos, la instalación y configuración específica de Kura para este equipo resultó bastante trivial. Al poder desarrollarse también un componente para la gestión de la conexión LoRa a través de las utilidades de Multitech, el control de Kura sobre el software y hardware del equipo permitió como en otros casos una actuación sobre el equipo como si de un gemelo digital se tratase. Este nivel de control e integración manifiesta las capacidades de una solución IIoT como la planteada.

Caídas frecuentes de conexión Las plantas fotovoltaicas en España se sitúan principalmente en entornos rurales. La conexión a Internet en estas zonas es muy variable siendo en la mayor parte de los casos poco robusta. Incluso si la conexión habitualmente es aceptable en velocidad, sigue siendo muy vulnerable a eventos meteorológicos o sociales. Lo primero se debe a que las conexiones se llevan por aire en lugar de soterradas y muchas veces se llevan hasta zonas de difícil acceso. Lo segundo es la afectación que sufren estas redes por sobrecargas temporales como las que puede haber en una zona vacacional. También se establecen algunas por conexión móvil, pero en esos casos se sufre la falta de cobertura.

Debido a esto, es muy importante que el sistema propuesto sea robusto frente a estas caídas de conexión. En los casos planteados, al disponerse una lectura local desde la propia red de la planta, la obtención de los datos no se ve interrumpida, sino únicamente su envío a SIS. No obstante, la lógica de envío por defecto en Kura ya aborda esta cuestión al disponerse un espacio de almacenamiento en caso de fallo durante la entrega de los datos al servidor remoto. Al ir todos los datos etiquetados con la marca de tiempo en la que se obtuvieron, se pueden paliar los efectos del envío diferido desde la plataforma.

5.6.3. Rendimiento de la solución

La arquitectura IIoT se adaptó a la perfección al entorno fotovoltaico como medio para asumir el SCADA completo de la planta con un coste relativamente bajo y una adaptabilidad plena a cada uno de los escenarios. De hecho, se exportó esta solución a otras muchas plantas de tamaño pequeño y mediano gestionadas por TSK en España como San Miguel (León), Reliegos (León), Carpio (Valladolid), Roces (Gijón), Logrosán (Cáceres), etc.

A pesar de la gran cantidad de datos obtenidos, para la visualización web (presentada en las Figuras 5.41 y 5.42) se buscó la simplificación en la vista general, permitiendo de un solo vistazo comprobar el estado de todas la plantas.

Esta facilidad en la monitorización ayuda al centro de control de TSK a detectar incidencias y hasta diagnosticarlas para saber exactamente qué acciones llevar a cabo. Por ejemplo, para saber si una planta está funcionando bien en general, un primer dato que puede ayudar a discernir una situación ordinaria de una potencial incidencia es el ratio de producción respecto a la irradiancia recibida del sol.

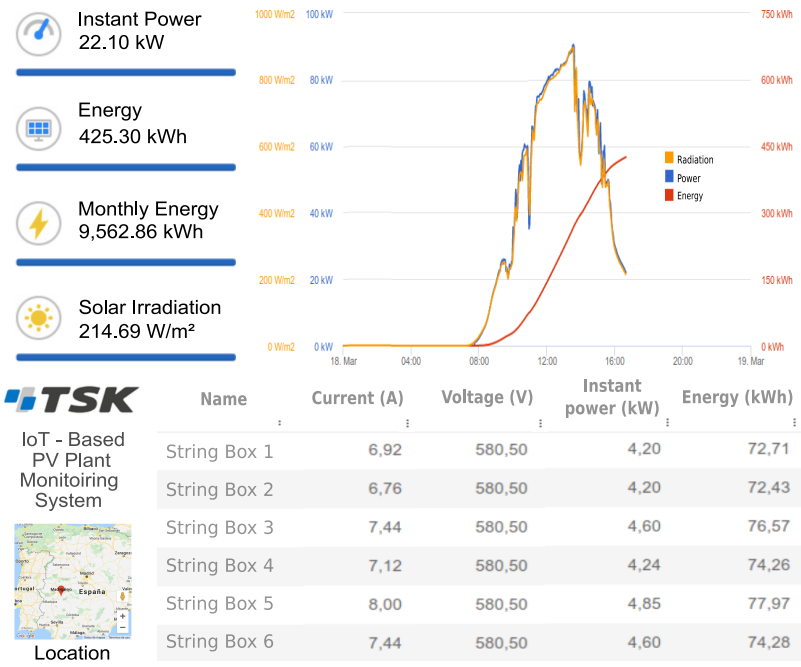
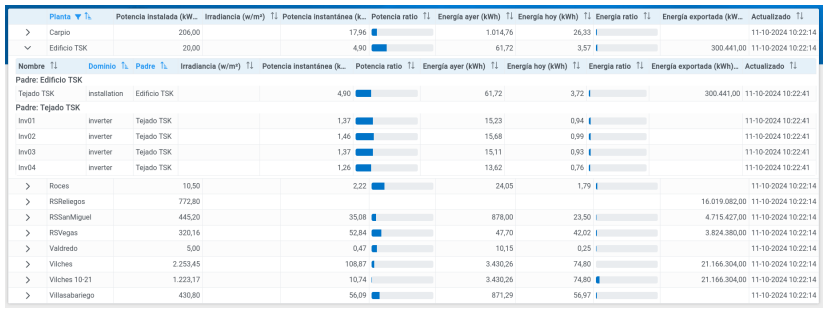


Figura 5.41. Panel de monitorización de planta fotovoltaica. A partir de los datos meteorológicos y de producción es posible conocer la eficiencia de la planta en todo momento.

Las principales ventajas aportadas al caso de uso y en particular en contraste con la anterior solución serían:

- Flexibilidad: la arquitectura IIoT ha mostrado ser capaz de adaptarse a cada escenario con la misma facilidad como si hubiera sido concebida específicamente para esa planta, manteniendo la reusabilidad de la solución.



Planta	Potencia instalada (kW...	Irradiancia (w/m²)	Potencia instantánea (k...	Potencia ratio	Energía ayer (kWh)	Energía hoy (kWh)	Energía ratio	Energía exportada (kW...	Actualizado
Carpio	206,00		17,96		1.014,76	26,33			11-10-2024 10:22:14
Edificio TSK	20,00		4,90		61,72	3,57		300.441,00	11-10-2024 10:22:14

Nombre	Dominio	Padre	Irradiancia (w/m²)	Potencia instantánea (k...	Potencia ratio	Energía ayer (kWh)	Energía hoy (kWh)	Energía ratio	Energía exportada (kWh)...	Actualizado
Padre: Edificio TSK										
Tejado TSK	installation	Edificio TSK		4,90		61,72	3,72		300.441,00	11-10-2024 10:22:41
Padre: Tejado TSK										
Inv01	inverter	Tejado TSK		1,37		15,23	0,94			11-10-2024 10:22:41
Inv02	inverter	Tejado TSK		1,46		15,68	0,99			11-10-2024 10:22:41
Inv03	inverter	Tejado TSK		1,37		15,11	0,93			11-10-2024 10:22:41
Inv04	inverter	Tejado TSK		1,26		13,62	0,76			11-10-2024 10:22:41

Roces	10,50		2,22		24,05	1,79			11-10-2024 10:22:14
RSReliegos	772,80							16.019.082,00	11-10-2024 10:22:14
RSSanMiguel	445,20		35,08		878,00	23,50		4.715.427,00	11-10-2024 10:22:14
RSVegas	320,16		52,84		47,70	42,02		3.824.380,00	11-10-2024 10:22:14
Valdredo	5,00		0,47		10,15	0,25			11-10-2024 10:22:14
Vilches	2.253,45		108,87		3.430,26	74,80		21.166.304,00	11-10-2024 10:22:14
Vilches 10-21	1.223,17		10,74		3.430,26	74,80		21.166.304,00	11-10-2024 10:22:14
Villasabariego	430,80		56,09		871,29	56,97			11-10-2024 10:22:14

Figura 5.42. Tabla resumen del estado de las plantas fotovoltaicas. Se puede ver de un vistazo rápido las plantas que puedan tener problemas e identificar el elemento concreto que lo está causando.

- Conectividad: gracias a los desarrollos ya disponibles y la posibilidad de incorporar otros, la solución planteada ha podido comunicarse con todos los elementos de la planta de manera homogénea respecto al resto de las aplicaciones planteadas en este capítulo.

- Integración: tanto vertical (plataforma, equipos de planta) como horizontal (red de comunicaciones inalámbrica), dando un control completo de la planta desde la solución.

- Mantenibilidad: al simplificar la integración con los equipos, se reducen también los puntos de fallo de toda el servicio de supervisión de la planta. Además la actuación y control remoto disponibles a través del gemelo digital del gateway IIoT, facilita labores rutinarias de actualización y auditoría.

- Supervisión unificada: al disponer de todos los datos en SIS, es posible obtener una visión de toda la planta mucha más completa y realizar una supervisión más eficaz. También permite simplificar las visualizaciones al poder cruzar datos entre los distintos equipos. Por ejemplo, el cálculo planteado más arriba del ratio de producción contra la irradiancia no sería posible con la anterior solución al no estar integrados estos datos en la misma.

- Mantenimiento: una ventaja directa del anterior punto es que se pueden detectar y diagnosticar mejor las incidencias, mejorando las labores de mantenimiento y permitiendo ganar conocimiento para avanzar hacia un mantenimiento predictivo al tener datos desde los cuales se consigue detectar la incidencia.

- Integridad: la adquisición y manejo de datos se mantiene robusta para evitar su pérdida o corrupción mediante la lógica de almacenamiento y envío diferido. En consecuencia, también es posible reconstruir la situación de la planta a posteriori. Además de contribuir a la supervisión, esto es fundamental para la reclamación de cobros frente a la compañía eléctrica y garantizar la rentabilidad de la producción.

- Coste: además de ser una solución de coste relativamente bajo al no tener asociadas licencias ni equipos específicos, la simplificación de la solución en las tres plantas presentadas reduce drásticamente los costes de mantenimiento a medio plazo.

Además de estas aportaciones, a nivel particular también se dieron algunas ventajas más específicas. Se destacan a continuación en los tres casos presentados, aunque en las otras plantas que se han citado también se dieron algunas de estas mejoras u otras similares.

- Hinojal: se simplificó la arquitectura de supervisión y se aumentó la robustez al eliminar intermediarios entre el equipo edge y los inversores y strings. La protección frente a caídas de conexión contribuyó mucho al tratarse de una planta que tenía cortes de red con mucha frecuencia por una mala instalación de Internet que se tardó varios años en arreglar.

- Iruela: se redujeron mucho los costes al eliminar todos los equipos edge excepto uno. También se obtuvo una supervisión más homogénea ayudando

mucho a mejorar la detección de incidencias.

- Jarandilla: la reducción de costes al centralizar las conexiones y quitar la conexión móvil que se estaba haciendo desde cada instalación supuso una mejora inmediata sin necesidad de recurrir a levantamiento de tierras ni obra mayor. También, la integración de los datos obtenidos en esta planta con los de Hinojal, permitió exportar los conocimientos de la estación meteorológica de esta última para poder estimar mejor el rendimiento de Jarandilla.

Como se puede ver en todos los casos se simplificó y mejoró la arquitectura de monitorización y se pudieron aprovechar las características de una solución tan versátil y adaptable para mejorar la situación previa.

5.7. Integración domótica de autoconsumo

En los últimos años, en particular con la reducción del coste de las soluciones fotovoltaicas, es cada vez más común encontrarse con productores de energía domésticos. La facilidad de uso y gestión de estos equipos ha llevado a que muchos particulares realicen instalaciones de paneles para rebajar su consumo desde la red eléctrica. Si bien no se trata de un escenario estrictamente industrial, tampoco es un simple escenario doméstico.

Como ejemplo de supervisión de plantas fotovoltaicas, se afronta una situación más sencilla que en el apartado anterior, sin embargo, es más compleja en general por las características domésticas. Se identifican tres factoras como base de este incremento de complejidad.

En primer lugar, el productor de una estación fotovoltaica industrial no debe atender apenas a consumo interno, sino vender toda la energía producida. En estos casos se busca optimizar la entrega de energía en momentos en los que el coste de la misma proporcione mayor beneficio. En cambio, el usuario doméstico consume energía además de producirla. Considerando que el consumo de esa

energía siempre es mayor en coste por kWh que la producción, se busca minimizar el consumo desde la red eléctrica en vez de una mayor entrega a la misma.

Esta optimización es mucho más compleja que en el escenario industrial porque a los parámetros de esa, se añade ahora el consumo del usuario que no atiende a un ciclo periódico o un modelo meteorológico como puede ser la producción de energía solar sino que sigue patrones más caóticos e impredecibles. A este problema se suma el aumento de electrificación en los hogares, debido a una tendencia general al uso de energía eléctrica que se ve especialmente incrementada con el vehículo eléctrico.

Para minimizar el consumo de la red eléctrica, la solución pasa por el uso de baterías que permitan acumular energía en horas de menor consumo o bajo coste, pero es necesario disponer de un análisis avanzado en tiempo real para determinar si ante un consumo en un momento dado es más adecuado recurrir a la energía de esta batería o solicitarla a la red.

En segundo lugar, las redes industriales tienen unas características de determinismo y control especializado que no se dan en el ámbito doméstico. Este entorno poco especializado puede inutilizar una solución que haya sido probada con mucho éxito en un entorno industrial o puede invalidar alguna de sus características de rendimiento, fiabilidad y ciberseguridad.

Por último, la capacidad de inversión de un particular es muy limitada en comparación con la de una empresa. Por tanto, la solución a plantear debe ajustar mucho el precio final para que sea ejecutable y rentable a escala doméstica, lo que supone una restricción de coste de instalación y mantenimiento mucho mayor que en los anteriores casos de uso.

Estos tres factores presentados, si bien supondrían un inconveniente serio para un SCADA tradicional, son mucho más sencillos de afrontar para una arquitectura IIoT por su bajo coste, escalabilidad, y capacidades de procesamiento e integración. En este último punto se destaca la necesidad

adicional de esta solución de integrarse con los sistemas domóticos del hogar. Además de resultar más natural y usable para el cliente, esta integración podría proporcionar información de gran valor en la creación de patrones de consumo para optimizar la entrega y solicitud de energía a la red.

Por tanto, se plantean los siguientes requisitos para el sistema:

- Ciberseguridad: el bien más sensible que se expone al aplicar la solución es la privacidad e información personal del usuario. Esto debe quedar bien protegido, considerando que la arquitectura se va a integrar en la red doméstica. También es vulnerable a un uso no cualificado por parte del usuario, ante lo cual debe estar protegida.

- Escalabilidad: para poder reducir las características a un nivel de particular.

- Mantenimiento: a pesar de que se pueden proporcionar instrucciones de mantenimiento al usuario, es preferible que desde su punto de vista no sea necesario hacer nada, ya que no es posible garantizar estas operaciones de la misma manera que se hace en la industria.

- Conectividad: tanto por la variedad de protocolos que se pueden encontrar en los distintos hogares como por la diversidad de topologías de red y conectividad que pueden componer la red doméstica.

- Integración: la solución debe ser capaz de integrarse con servicios locales para dar una mejor experiencia al usuario y facilitarle el uso de la solución en lo posible.

- Procesamiento edge: por un lado para ofrecer un control de inyección y consumo de energía respecto a la red, y por otro para proporcionar al usuario, desde su producción y consumo, una buena estimación de la deuda contraída con la compañía eléctrica.

- Bajo coste: para que un particular pueda permitirse la puesta en marcha de esta aplicación.
- Bajo consumo: para no generar sobrecostes en la factura del usuario.

Como se ve, la adaptación de la solución industrial del la Sección 5.6 a un entorno doméstico y de autoconsumo, no resulta trivial y se requiere una solución muy adaptable para lograrlo.

5.7.1. Caso de uso

Se proponen para la validación dos hogares asturianos (Figura 5.43). Por un lado habría una casa de la familia López, en el pueblo de Limpias, y por otro la de la familia Molina ubicada en Montemayor. En ambos casos existe una instalación de 5 kW que se encuentra conectada primeramente al hogar y luego el hogar a la red eléctrica. Destacar que se trata de plantas pequeñas y que en ninguno de los dos casos incurrieron los usuarios en el uso de baterías para gestionar el consumo propio contra el de la red. En la Figura 5.44 se puede ver en foto aérea una de estas instalaciones, y se puede apreciar que su tamaño es bastante pequeño, sobre todo en comparación con las demás soluciones planteadas en este capítulo.

Para tener un control completo del flujo de energía entre el hogar, los paneles y la red eléctrica, se propusieron dos puntos de lectura: la entrada y salida de energía en la conexión a red (mediante un contador) y la producción de energía fotovoltaica (mediante el inversor). Solo con estos dos puntos es posible conocer la energía eléctrica que circula en todas las direcciones, es decir:

- Energía inyectada en la red: obtenida directamente desde el contador
- Energía consumida de la red: obtenida directamente desde el contador
- Energía producida por los paneles: obtenida directamente desde el inversor

Figura 5.43. Ubicación de las instalaciones de autoconsumo (fuente [101]).

Figura 5.44. Vista de satélite de la casa de la familia López. La instalación fotovoltaica tiene una producción pico teórica de 5 kW.

- Energía consumida desde los paneles: calculada a partir de la resta entre la producida y la inyectada
- Energía consumida en el hogar: calculada a partir de la anterior más la energía consumida de la red.

Las dos primeras energías ya permiten el cálculo de la factura de la luz mensual, ya que se limita a la interacción con la red. Las restantes sirven para conocer el patrón de consumo del hogar y poder realizar recomendaciones a los usuarios. También se pueden explotar los datos de cara a la instalación de una batería que permitiera optimizar el tráfico de energía para reducir la inyectada y la consumida desde la red.

Al ser ambos hogares similares en lo esencial, se propone el mismo planteamiento en las dos, aunque esto no significa que no tengan diferencias como se verá al diseñar la solución en mayor detalle.

5.7.2. Diseño e implantación

Para poder reducir el coste total de la solución el primer paso es adecuar el hardware a un presupuesto más ajustado. Así, donde en otras soluciones se empleaban mini PC industriales, en este caso no es necesario recurrir a ese tipo de equipos y se puede entrar en gamas más sencillas. Se propone, en cambio, el empleo de placas como la Raspberry Pi. La Raspberry Pi 4B es un modelo con una capacidad más que de sobra para correr Eclipse Kura y mucha versatilidad para los entornos inalámbricos como será el caso de esta solución.

Para facilitar las tareas de mantenimiento y mejorar la durabilidad, se colocará el PC en el armario eléctrico de las casas donde hay espacio más que suficiente para este tipo de equipos y facilitará la lectura de contador. Además de la Raspberry Pi 4B, será necesario colocar otros elementos para obtener los datos propuestos:

- Contador bidireccional: ubicado en la cabecera de la casa para monitorizar

la energía entrante y saliente hacia la red. Se usará el modelo Circutor CEM-C6-MID. Expone los datos por ModbusRTU

- Inversor: ya presente a la salida de los paneles fotovoltaicos. Además dispone de una tarjeta de comunicación por Modbus (RTU en Limpias, TCP en Montemayor).
- Conversor RS-USB: necesario para conectar el contador a la Raspberry Pi. Se empleará uno básico basado en el integrado *SP485EEN-L/TR*.
- Conversor RS-TCP: necesario en Limpias para traducir la señal serie de la tarjeta del inversor a TCP. Al estar en el exterior, se buscará un rango industrial y se empleará el USR-TCP232 ya mencionado en la sección anterior y que tiene un coste muy ajustado

Dispuestos de la manera en que se presenta en la Figura 5.45, se obtiene una monitorización completa de la producción y consumo de energía. Destacar que la conexión a Internet en este caso no está proporcionada por una red móvil propia (por reducir costes) ni por una red industrial. El equipo edge está plenamente integrado en la red doméstica. Según el caso es posible hacerlo por WiFi o Ethernet, ambos comunes en un hogar y el equipo se configura para no admitir ninguna conexión externa, lo cual es posible gracias a la gestión remota desde la plataforma IIoT.

Respecto a los servicios hacia el usuario, ya se ha comentado que no disponen de baterías por lo que no es necesario gestionar la inyección de energía a la red. Lo que sí se incorporó es la estimación de la factura eléctrica mensual. Para realizarla es necesario leer del contador instalado la energía inyectada y consumida de la red cada hora. A partir de estos datos, se obtiene la energía neta producida/consumida en esa hora y se le aplica el precio correspondiente. Hay que considerar también impuestos y que la factura incluye algunas partes fijas. En conclusión, se realiza la siguiente operación:

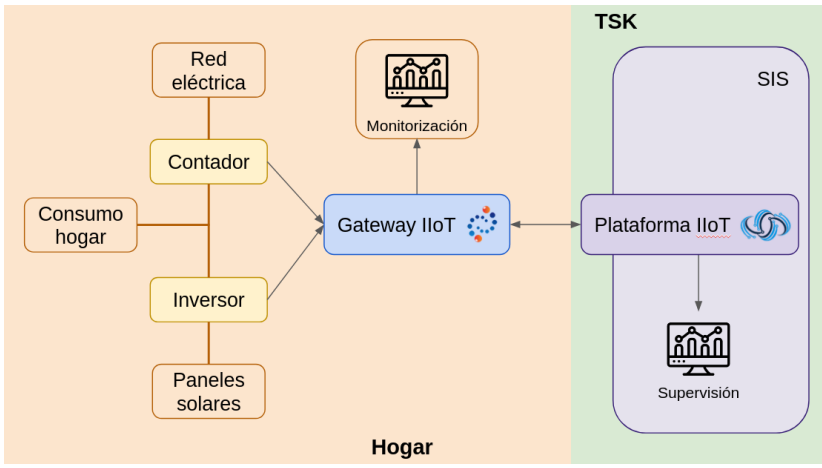


Figura 5.45. Diagrama funcional de la monitorización de Montemayor. Con solo un contador además del inversor es posible conocer variables de consumo interno además de exportación a la red.

1. Obtener energía consumida neta de la hora: se hace por resta simple entre la energía consumida y la energía inyectada. Se podría hacer el cálculo a la inversa para conocer la producción neta, pero el resto de los términos de la factura pasarían a estar invertidos también.

2. Aplicar precio horario de la energía: destacar que es distinto el precio en producción que en consumo (siempre es mayor el de consumo).

3. Añadir término de potencia: se calcula a partir de la potencia contratada y su precio. El precio se puede obtener del contrato, habitualmente expresado en kW/día o kW/año. Para obtener su valor horario basta con prorratearlo entre 365 días y 24 horas.

4. Aplicar impuesto eléctrico: un porcentaje fijo aplicado sobre la suma de lo anterior. Hay que considerar que en los últimos años precisamente ha variado bastante por las medidas gubernamentales para paliar el aumento de precio de la electricidad.

5. Añadir alquiler de equipos de medida: es un precio fijo según el punto de

medida instalado con precios regulados por mes. Para poder prorratearlo a días y horas, es necesario multiplicarlo antes por 12 para llevarlo a coste anual y luego ya dividir entre 365 y 24.

6. Añadir IVA: sobre la suma de todo lo anterior. Es un porcentaje fijo de 21 %.

Se hace notar que la familia López tenía pactado con la compañía un precio de compra y venta fijo. Sin embargo, en el contrato de la familia Molina, estos precios se fijaban por el mercado regulado en el que se determina el precio para cada hora d antemano a lo largo del día anterior. En el cálculo implementado se permitió elegir entre ambos mercados, según el caso. De la secuencia presentada solo cambia el origen del dato utilizado en el paso 2. El resto se calcula de la misma manera.

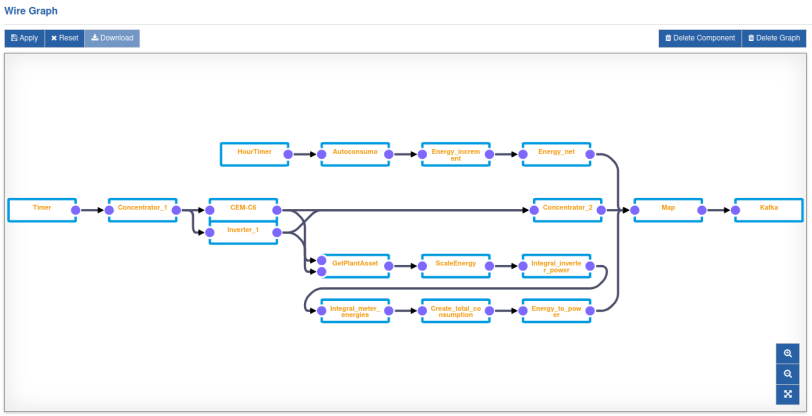


Figura 5.46. Flujo de trabajo empleado en Limpias. Contiene tres líneas de procesamiento: datos de facturación, datos de producción y datos de autoconsumo.

En la Figura 5.46 se pueden apreciar los tres flujos de trabajo descritos para la aplicación. En primer lugar se observa una lectura a las horas exactas (flujo de parte superior de la figura) para poder calcular el gasto o beneficio de la energía. Justo debajo se encuentra la lectura periódica ordinaria de los datos brutos de

contador e inversor para análisis posteriores y visualizaciones básicas. Y el último de los flujos, en la parte inferior, corresponde a los necesarios para obtener los parámetros de autoconsumo propuestos en la presentación del caso de uso. Estos cálculos parten de los datos leídos del segundo flujo de trabajo y se procesan para obtener las variables derivadas.

Puntos críticos

Como se puede suponer, a pesar de ser un escenario aparentemente sencillo, se dieron bastantes complicaciones derivadas de la infraestructura, que en una industria no se darían.

Cambio de operador En una de las plantas, al poco de la puesta en marcha se modificó el operador de Internet de la casa. Esto supuso en primer lugar un cambio del router y un corte durante el mismo. Al haberse planteado la conexión del equipo edge como uno más de la red doméstica, esto no supuso ningún cambio ya que se limitó a reconectarse a la nueva red por DHCP.

Sin embargo, surgió una complicación adicional, al poco de instalarse el nuevo router, se cayó y pasó a estar defectuoso. Con esto, el equipo IIoT perdió acceso a la red local y a Internet. Lo segundo solo significó la acumulación de datos pendientes de enviar. Sin embargo, la pérdida de la red local le hizo perder acceso al inversor. Al haberse ubicado el equipo en serie con el contador principal, la contabilidad principal de energía se pudo mantener y el cálculo de la factura mensual permaneció inalterado.

Red no industrial Existen dos desventajas mayores derivadas de no tener una red industrial. La primera es que la red no es accesible remotamente. En muchas plantas industriales se dispone un sistema de acceso remoto ya sea por IP pública (ya sea dinámica o estática) o por una VPN intermedia. A veces se emplean ambas cosas como equilibrio entre la usabilidad y la seguridad. En todo caso, la

red aquí propuesta no era posible acceder a ella una vez instalado el equipo. Es por eso que se planteó una configuración de red sencilla a nivel doméstico con el equipo conectado por DHCP, asumiendo la falta de gobierno sobre la red. Tras instalar el equipo y acceder este a Internet, se conectó a la plataforma IIoT y se pudieron retocar detalles de la puesta en marcha.

La segunda es la falta de seguridad. Si bien las redes domésticas son cada vez más seguras, no dejan de ser bastante más vulnerables que la industriales en general. Esto no se debe solo a la existencia de conexiones inalámbricas, sino principalmente a la falta de cualificación de los usuarios y el empleo de configuraciones por defecto en los elementos de red. Esta vulnerabilidad hace que cualquier equipo conectado a la red deba contar con protecciones propias. Kura por fortuna ayuda a gestionar un firewall en el equipo y puede desactivar su interfaz web para mayor seguridad.

Cambio de tarifa En junio de 2021 se cambió el modo de calcular la factura de la luz. Una de las principales novedades de la nueva tarifa, denominada 2.0 TD, fue la introducción de franjas horarias para el coste de la energía y en algunos casos de la potencia contratada. El precio de la energía pasó a depender de estos momentos según si se trata del periodo punta, con un alto consumo general y por tanto un precio más caro, periodo valle, con menos consumo y precio, y periodo supervalle, donde existe muy poca demanda de energía y por tanto el precio es aún menor.

Gracias a la arquitectura modular y gestión remota del equipo edge, fue posible aplicar estas modificaciones sobre el componente que se ocupaba del cálculo de tarifa y desplegarlo con la configuración del nuevo contrato establecido con la compañía. Durante un periodo de validación anterior al inicio de esta tarifa, ambas se calcularon en paralelo para ayudar a validar la lógica implementada.

Cambios de impuestos Desde otoño de 2021, se han aprobado en España múltiples medidas para combatir la subida de precios de la energía eléctrica. Entre estas medidas estaban las reducciones a los impuestos tanto eléctrico como IVA. Estos dos parámetros son fundamentales para el cálculo de la factura mensual descrito más arriba. Haciendo uso de las configuración remota de Kura, fue posible actualizar estos parámetros conforme se iban aplicando, tanto al empezar el descuento como al terminar.

Normalización de señales Para poder realizar cálculos entre las distintas señales, en particular obtener la energía consumida en el hogar, es necesario que tanto inversor como contador expresen la información de una forma común. Para esto, se parte de las energías inyectada y consumida en el contador, pero no se puede encontrar una información comparable en el inversor. El inversor dispone de conteo de energía pero no tiene apenas precisión. La potencia, en cambio, sí resulta muy fiable. El contador por su parte da una cuenta de energía total que no comparte punto de referencia con el inversor.

Para completar esta normalización se requieren las siguientes operaciones:

1. Alineamiento de timestamps entre las dos lecturas para que puedan operarse realmente. Destacar que aquí se introduce siempre un error de milisegundos igualmente.

2. Integral de la potencia para poder obtener la energía. Esta integral incluye también escalado para pasar de los kW de potencia a kWh de energía.

3. Incremento sobre las energías para obtener fracciones de producción comparables.

4. Sumatorio de las energías para obtener una referencia común, en este caso el inicio de cada día, aunque se podría definir otro criterio.

5. Operaciones binarias para obtener las energías descritas más arriba, como la consumida total del hogar.

6. Derivada de las energías obtenidas para poder tener también su potencia correspondiente.

Esta consecución de cálculos no resulta tan extraña cuando los datos a calcular provienen de equipos distintos. Gracias a los flujos de trabajo de Kura y la modularidad de los componentes, en este caso el de las operaciones, fue posible configurar este conjunto de cálculos y sobre todo retocarlo para garantizar su fiabilidad y robustez. Se plantea incluso el desarrollo de un módulo que contenga toda esta adecuación para otros casos de uso con problemáticas similares.

5.7.3. Rendimiento de la solución

Gracias a la arquitectura IIoT se ha podido escalar la solución presentada en la Sección 5.6 a las posibilidades y dimensiones de un escenario doméstico y se ha logrado adaptar la lógica también para atender a las necesidades de un servicio de autoconsumo en vez de producción pura. Además gracias a SIS, el usuario puede acceder de forma sencilla a los datos de su propio hogar con paneles resumen como el presentado en la Figura 5.47.

Se destacan las siguientes ventajas de la solución aplicada:

- Coste reducido: al poder desplegar el software en un equipo como la Raspberry Pi ya con prestaciones sobradas, se ha podido trasladar a un entorno doméstico una solución con la fiabilidad de un entorno industrial.

- Ciberseguridad: las herramientas de firewall y configurabilidad de seguridad en Kura se han adaptado a la red doméstica para conservar una alta protección frente a ataques.

Figura 5.47. Vista resumen de la producción de Montemayor. Con esta visualización la familia Molina puede conocer en tiempo real el beneficio y estado de su instalación de autoconsumo.

- Conectividad: la capacidad de incorporarse en cualquier equipo y agregar hardware y software, ha permitido que la solución IIoT se adapte a las necesidades de conectividad (WiFi principalmente) de la red de un hogar.
- Integración horizontal: a pesar de mostrarse como un vertical en esta aplicación, el uso de Kura permite dotar al conjunto de integración horizontal para facilitar la conexión con otros servicios que se incorporen a futuro en el hogar, sin necesidad de recurrir a la conexión a Internet para esta funcionalidad.
- Modularidad: la capacidad de reconstruir los módulos funcionales de Kura permite que se reajusten a futuro los cálculos conforme a nuevos intereses.

La solución dota al usuario doméstico de una monitorización fiable por un lado y de una vía de ampliar sus servicios domésticos a través de la integración por Kura. De esta manera, se valida que la solución IIoT industrial también puede contribuir a la mejora en un entorno domótico al escalar con facilidad a casos de uso mucho menores.

Capítulo 6

Conclusiones

La implantación de las tecnologías IIoT en la industria ha marcado un hito en la forma de concebir la automatización y control de las plantas. Si bien esta implantación plantea algunas dificultades, la construcción de una arquitectura bien definida que aborde los retos de un entorno productivo puede resultar muy beneficiosa y proporcionar mucha flexibilidad y robustez.

En este documento se detallan los pasos para identificar los pilares de una arquitectura IIoT y diseñar a partir de ella una solución que aborda todos los retos a los que se enfrenta la industria (Hipótesis 1). Además de definir las tecnologías más aptas del estado del arte para esta arquitectura, se han planteado los principios de diseño y características generales para adaptar las implementaciones a los nuevos avances tecnológicos que se sucedan.

Estos principios parten de la selección de una infraestructura en tres capas: Percepción, Edge y Plataforma. Sobre ellos se abordan cada una de los retos encontrados para describir en mayor detalle las características que deben tener, tales como el requisito de despliegue de aplicaciones en la capa de Plataforma, la modularidad en la capa Edge o la capacidad de soportar condiciones ambientales adversa en la capa de Percepción. A partir de estas características se identifica el software, hardware, protocolos y redes más adecuados para abarcar todas en cada

capa. Así, se define de principio a fin todo el proceso necesario para establecer una arquitectura IIoT plenamente adaptada a los desafíos de la industria.

También se ha puesto a prueba una solución IIoT desarrollada con estos principios en múltiples entornos productivos (Hipótesis 2). En estos entornos se ha buscado también validar que dicha solución es capaz de adaptarse a aplicaciones dispares o que puede implementarse con distintos niveles de penetración en los sistemas de control industrial.

Para ello, los casos de uso seleccionados han sido:

- Supervisión de una planta termosolar: habilitando la monitorización remota, la asistencia experta y ampliando las posibilidades de investigación aplicada.
- Gestión de la infraestructura de red industrial: contribuyendo a mejorar la robustez de la red y la ciberseguridad de la planta.
- Digitalización de instalaciones analógicas: ampliando las posibilidades de integración de una planta con automatización muy obsoleta sin comprometer la estabilidad de la misma.
- Mantenimiento de subestaciones eléctricas: reduciendo costes de mantenimiento y ayudando a construir un sistema más avanzado de gestión de mantenimiento.
- Monitorización de plantas fotovoltaicas: reemplazando por completo el SCADA existente, reduciendo gastos de licencias y mantenimiento.
- Integración domótica de autoconsumo: proporcionando información al pequeño productor para gestionar su propio autoconsumo con mayor conocimiento.

Destacar que todos los casos de uso han sido aplicados en al menos dos

instalaciones (en algunos casos muchas más), dando una mejor perspectiva de los potenciales problemas para una validación más minuciosa.

Contrastando con los objetivos propuestos en el Capítulo 1, se describe a continuación en qué medida se han logrado completar:

- **Modernización de las redes industriales**: la arquitectura IIoT diseñada ha aportado nuevas vías de comunicación, gestión e integración a las redes industriales, haciéndolas más versátiles y capaces de adaptarse a futuros retos. Además estas mejoras han venido de la mano de un incremento en la fiabilidad, robustez y ciberseguridad de las redes.

- **Optimización de los sistemas de comunicación industriales**: además de aprovechar las comunicaciones ya existentes, la arquitectura IIoT propuesta ha introducido otras vías de comunicación como las redes inalámbricas locales. También ha mejorado la eficiencia de las comunicaciones remotas al aplicarse sobre protocolos ligeros como es el caso de MQTT.

- **Aumento de la capacidad de integración**: la solución desarrollada ha incorporado vías de integración a todos los niveles de la red industrial y ha facilitado las integraciones horizontales, que suponen una menor complejidad.

- **Optimización del coste de las soluciones finales**: el uso de la arquitectura IIoT propuesta ha reducido costes en todos los casos de uso. El origen de esta reducción se debe a múltiples causas. Entre otras, un coste de equipamiento bajo, poco mantenimiento, eliminación de las licencias o la mitigación de las caídas.

- **Validación de las arquitecturas IIoT en entornos productivos**: tras implementar la solución propuesta, se ha puesto a prueba en gran

cantidad de casos de uso, todos ellos casos reales, entornos productivos y de naturaleza industrial. Esto confirma la Hipótesis 2 y demuestra que, independientemente del grado de madurez de tecnologías concretas, las arquitecturas IIoT están listas para entrar en producción.

- **Perfeccionamiento de las arquitecturas IIoT**: se ha diseñado una infraestructura IIoT agnóstica y plenamente adaptada a la industria. Además se ha definido a múltiples niveles, desde los retos de partida, capas de abstracción, requisitos a abordar, características capa a capa, su topología, tecnologías disponibles y más adecuadas y finalmente la implementación concreta. Esto permite incorporar nuevos elementos en cualquier etapa del proceso de diseño para definir una arquitectura alternativa.

- **Diseño de una arquitectura IIoT versátil**: la implementación de la solución IIoT propuesta se ha puesto a prueba en seis aplicaciones distintas. Además de plantear dominios de trabajo distintos, cada escenario implicó también un grado de integración en la plata distinto. Por último, dentro de cada aplicación se llevaron a cabo dos o más casos de uso que permitieran contrastar la aplicación a mayor nivel.

En conjunto, se puede ver que la arquitectura IIoT se ha adaptado a los distintos escenarios a la perfección y ha proporcionado varias vías de optimización a los entornos industriales. También se han validado tecnologías y protocolos específicos como aptos para entornos productivos y se han establecido los pasos para la construcción de nuevas soluciones basadas en IIoT.

6.1. Propuesta de trabajo futuro

En esta tesis se ha validado el proceso de construcción de una arquitectura IIoT y su capacidad para emplearse en entornos productivos. A partir del estudio realizado y los casos de uso afrontados, se ratifica la arquitectura IIoT como una herramienta clave para mejorar la competitividad de los sistemas de automatización industriales, aumentando su interoperabilidad y eficiencia.

En lo que se refiere a tecnologías concretas, no todo lo que se engloba en IIoT presenta el mismo grado de madurez. Sin embargo, esta investigación revela que las arquitecturas en su conjunto sí tienen madurez suficiente para un despliegue masivo, considerando que una de las fortalezas de las mismas es poder hacer un reemplazo o actualización de algunas de las tecnologías implicadas si otras alcanzan la madurez suficiente y resultan más adecuadas.

En este contexto, la línea de trabajo principal planteada es impulsar su desarrollo en la industria para su aplicación generalizada para la monitorización y control. Para llevar a cabo este proceso de industrialización en masa, será necesario realizar una considerable labor de implantación, ampliando la base de variedad dentro de los casos de uso ya validados, explorando la aplicación en nuevos escenarios y sectores o redefiniendo las tecnologías seleccionadas de acuerdo a últimos avances. Según se revela en esta tesis, los dos primeros puntos suponen un trabajo complejo y de larga duración por la inercia de los sistemas industriales actuales.

En lo relativo a la redefinición de tecnologías, se debe considerar que algunos de los avances que se están realizando en la actualidad representarán también nuevos retos u oportunidades para las arquitecturas IIoT del futuro. Por ejemplo, la llegada de las inteligencias artificiales generativas dan una nueva dimensión a la integración con la nube, y abre el campo a investigar su posible implicación para equipos de menores prestaciones como los de la capa edge. Otras tecnologías como

la computación cuántica suponen un reto para la comunicación, y significarán una transformación en el ámbito de la ciberseguridad respecto a cómo se concibe hoy en día.

En consecuencia, hay todavía mucho campo de investigación y desarrollo hasta establecer la industrialización completa de las arquitecturas IIoT, sus fundamentos y funcionalidades, para poder aprovechar al máximo sus posibilidades.

Referencias

1. de Arquer Fernández, P., Fernández Fernández, M. Á., Carús Candás, J. L. y Arboleya Arboleya, P. An IoT open source platform for photovoltaic plants supervision. *International Journal of Electrical Power and Energy Systems* **125,** 106540. ISSN: 01420615 (feb. de 2021).

2. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. Determining Operational Constraints for IoT-Based Advanced Metering Infrastructure. *2021 IEEE Madrid PowerTech,* 1-6 (jun. de 2021).

3. de Arquer Fernández, P., Carús Candás, J. L. y Arboleya Arboleya, P. en *Encyclopedia of Electrical and Electronic Power Engineering* (Elsevier Ltd, 2022).

4. Fernández Villán, A., Fernández Fernández, M. Á., Carús Candás, J. L., de Arquer Fernández, P., Arias Linacero, N. y Usamentiaga Fernández, R. *Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography* en *2018 2nd International Research Conference on Sustainable Energy, Engineering, Materials and Environment (IRCSEEME2018)* (jul. de 2018), 206-207.

5. Naeem, G., Asif, M. y Khalid, M. Industry 4.0 digital technologies for the advancement of renewable energy: Functions, applications, potential and challenges. *Energy Conversion and Management: X,* 100779. ISSN: 2590-1745 (oct. de 2024).

6. Adeyemi, O. A., Pinto, P. M. G., Sunmola, F., Aibinu, A. M., Okesola, J. . O. y Adeyemi, E. . O. Towards the Adoption of Industry 4.0 Technologies in the Digitalization of Manufacturing Supply Chain. *Procedia Computer Science. 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023)* **232,** 337-347. ISSN: 1877-0509 (ene. de 2024).

7. Lucizano, C., de Andrade, A. A., Blumetti Facó, J. F. y de Freitas, A. G. *Revisiting the Automation Pyramid for the Industry 4.0* en *2023 15th IEEE International Conference on Industry Applications (INDUSCON)* (nov. de 2023), 1195-1198.

8. Bueno, A., Godinho Filho, M. y Frank, A. G. Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering* **149,** 106774. ISSN: 0360-8352 (nov. de 2020).

9. Bilad, A., Zaim, M. y Zaim, F. *Industry 4.0 tools in the industrial sector: A Systematic Literature Review* en *2022 14th International Colloquium of Logistics and Supply Chain Management (LOGISTIQUA)* (mayo de 2022), 1-6.

10. Forcina, A. y Falcone, D. The role of Industry 4.0 enabling technologies for safety management: A systematic literature review. *Procedia Computer Science* **180,** 436-445. ISSN: 1877-0509 (ene. de 2021).

11. Attaran, S., Attaran, M. y Celik, B. G. Digital Twins and Industrial Internet of Things: Uncovering operational intelligence in industry 4.0. *Decision Analytics Journal* **10,** 100398. ISSN: 2772-6622 (mar. de 2024).

12. Pawar, P., TarunKumar, M. y Vittal K., P. An IoT based Intelligent Smart Energy Management System with accurate forecasting and load

strategy for renewable generation. *Measurement* **152,** 107187. ISSN: 0263-2241 (feb. de 2020).

13. Sioshansi, F. en *Behind and Beyond the Meter* 47-82 (Elsevier, ene. de 2020). ISBN: 978-0-12-819951-0.

14. Mostafa, N., Ramadan, H. S. M. y Elfarouk, O. Renewable energy management in smart grids by using big data analytics and machine learning. *Machine Learning with Applications* **9,** 100363. ISSN: 2666-8270 (sep. de 2022).

15. Ijemaru, G. K., Ang, L. M. y Seng, K. P. Transformation from IoT to IoV for waste management in smart cities. *Journal of Network and Computer Applications* **204,** 103393. ISSN: 10848045 (ago. de 2022).

16. Kampa, T., Müller, C. K. y Großmann, D. Interlocking IT/OT security for edge cloud-enabled manufacturing. *Ad Hoc Networks* **154,** 103384. ISSN: 1570-8705 (mar. de 2024).

17. Garimella, P. K. *IT-OT Integration Challenges in Utilities* en *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (oct. de 2018), 199-204.

18. Sharma, V., Sharma, K. y Kumar, A. *From Theory to Practice: A Systematic Review of Digital Twin Implementations Across Industry 4.0* en *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (jul. de 2023), 1-7.

19. Maleh, Y. IT/OT convergence and cyber security. *Computer Fraud & Security* **2021,** 13-16. ISSN: 1361-3723 (dic. de 2021).

20. Ehie, I. C. y Chilton, M. A. Understanding the influence of IT/OT Convergence on the adoption of Internet of Things (IoT) in manufacturing organizations: An empirical investigation. *Computers in Industry* **115,** 103166. ISSN: 0166-3615 (feb. de 2020).

21. Zhu, H. *y col.* Key technologies for smart energy systems: Recent developments, challenges, and research opportunities in the context of carbon neutrality. *Journal of Cleaner Production* **331,** 129809. ISSN: 0959-6526 (ene. de 2022).

22. Sverko, M., Grbac, T. G. y Mikuc, M. SCADA Systems With Focus on Continuous Manufacturing and Steel Industry: A Survey on Architectures, Standards, Challenges and Industry 5.0. *IEEE Access* **10,** 109395-109430. ISSN: 2169-3536 (2022).

23. Shahzad, Y., Javed, H., Farman, H., Ahmad, J., Jan, B. y Zubair, M. Internet of Energy: Opportunities, applications, architectures and challenges in smart industries. *Computers & Electrical Engineering* **86,** 106739. ISSN: 0045-7906 (sep. de 2020).

24. Falco, G., Caldera, C. y Shrobe, H. IIoT Cybersecurity Risk Modeling for SCADA Systems. *IEEE Internet of Things Journal,* 1. ISSN: 2327-4662 (2018).

25. Ghosh, S. y Sampalli, S. A Survey of Security in SCADA Networks: Current Issues and Future Challenges. *IEEE Access* **7,** 135812-135831. ISSN: 2169-3536 (2019).

26. Mullet, V., Sondi, P. y Ramat, E. A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0. *IEEE Access* **9,** 23235-23263. ISSN: 2169-3536 (2021).

27. Corallo, A., Lazoi, M., Lezzi, M. y Luperto, A. Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review. *Computers in Industry* **137,** 103614. ISSN: 0166-3615 (mayo de 2022).

28. Cesar, E. L., Fernandes, G. S., Kagami, M. T. N. y Calisto, T. N. *Technological Obsolescence Management of Electrical Equipment and*

Automation Systems en *2019 IEEE Petroleum and Chemical Industry Committee Conference (PCIC)* (sep. de 2019), 303-310.

29. Cesar, E. L., Fernandes, G. S., Kagami, M. T. y Calisto, T. N. Technological Obsolescence Management: Monitoring Electrical Equipment and Automation Systems. *IEEE Industry Applications Magazine* **26,** 82-87. ISSN: 1558-0598 (jul. de 2020).

30. Newman, R. *Designing hypermedia documentation for safety critical applications* en *Proceedings International Conference on Information Technology: Coding and Computing (Cat. No.PR00540)* (mar. de 2000), 247-252.

31. Doyen, L. y Gaudoin, O. Modeling and Assessment of Aging and Efficiency of Corrective and Planned Preventive Maintenance. *IEEE Transactions on Reliability* **60,** 759-769. ISSN: 1558-1721 (dic. de 2011).

32. Albouq, S. S., Sen, A. A. A., Almashf, N., Yamin, M., Alshanqiti, A. y Bahbouh, N. M. A Survey of Interoperability Challenges and Solutions for Dealing With Them in IoT Environment. *IEEE Access* **10,** 36416-36428. ISSN: 2169-3536 (2022).

33. Amjad, A., Azam, F., Anwar, M. W. y Butt, W. H. A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT. *IEEE Access* **9,** 96528-96545. ISSN: 2169-3536 (2021).

34. Lee, E., Seo, Y.-D., Oh, S.-R. y Kim, Y.-G. A Survey on Standards for Interoperability and Security in the Internet of Things. *IEEE Communications Surveys & Tutorials* **23,** 1020-1047. ISSN: 1553-877X (2021).

35. Alam, S. y Khan, M. F. Enhancing AI-Human Collaborative Decision-Making in Industry 4.0 Management Practices. *IEEE Access* **12,** 119433-119444. ISSN: 2169-3536 (2024).

36. Islam, K., Kim, D. y Abu-Siada, A. A review on adaptive power system protection schemes for future smart and micro grids, challenges and opportunities. *Electric Power Systems Research* **230,** 110241. ISSN: 0378-7796 (mayo de 2024).

37. Byabazaire, J., O'Hare, G. M. y Delaney, D. T. End-to-End Data Quality Assessment Using Trust for Data Shared IoT Deployments. *IEEE Sensors Journal* **22,** 19995-20009. ISSN: 1558-1748 (oct. de 2022).

38. Béres, R., van der Wel, A., Fattahi, A. y van den Broek, M. The impact of national policies on Europe-wide power system transition towards net-zero 2050. *Energy* **310,** 133216. ISSN: 0360-5442 (nov. de 2024).

39. Hrga, A., Capuder, T. y Žarko, I. P. *Decentralized IoT Platform for Flexibility Service Providers in Power Systems* en *2021 IEEE International Conference on Blockchain (Blockchain)* (dic. de 2021), 1-7.

40. Bento, P. M. R., Mariano, S. J. P. S., Pombo, J. A. N. y Calado, M. R. A. Large-scale penetration of renewables in the Iberian power system: Evolution, challenges and flexibility options. *Renewable and Sustainable Energy Reviews* **204,** 114794. ISSN: 1364-0321 (oct. de 2024).

41. Tom, R. J., Sankaranarayanan, S. y Rodrigues, J. J. Agent negotiation in an IoT-Fog based power distribution system for demand reduction. *Sustainable Energy Technologies and Assessments* **38,** 100653. ISSN: 2213-1388 (abr. de 2020).

42. Wang, J., Zheng, W. y Li, Z. Detection and estimation of behind-the-meter photovoltaic generation based on smart meter data analytics. *The Electricity Journal* **35,** 107132. ISSN: 10406190 (jun. de 2022).

43. Aheleroff, S. *y col.* IoT-enabled smart appliances under industry 4.0: A case study. *Advanced Engineering Informatics* **43,** 101043. ISSN: 1474-0346 (ene. de 2020).

44. Goswami, S. A., Padhya, B. P. y Patel, K. D. *Internet of Things: Applications, Challenges and Research Issues* en *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (dic. de 2019), 47-50.

45. Abowd, G. D. Beyond Weiser: From Ubiquitous to Collective Computing. *Computer* **49,** 17-23. ISSN: 1558-0814 (ene. de 2016).

46. Khaleel, T. A., Mustafa, F. A. y Khattab, M. F. *Applications of Sensor Networks and Remote Sensing in Environmental Sustainability: A Review* en *2022 International Conference on Engineering & MIS (ICEMIS)* (jul. de 2022), 1-3.

47. Wambua, R. N. *Systematic Review of the Influence of Internet of Things (IoT) on the Education of Students with Disabilities* en *2023 IST-Africa Conference (IST-Africa)* (mayo de 2023), 1-8.

48. Boddu, R. S. K. *Internet of Things (IoT): Accelerating the Digital transformation of Healthcare system* en *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* **1** (mar. de 2021), 1716-1720.

49. Agarwal, K., Agarwal, A. y Misra, G. *Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT* en *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (dic. de 2019), 629-633.

50. Bansal, M., Nanda, M. y Husain, M. N. *Security and privacy Aspects for Internet of Things (IoT)* en *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (ene. de 2021), 199-204.

51. Karmakar, K. K., Varadharajan, V., Nepal, S. y Tupakula, U. SDN-Enabled Secure IoT Architecture. *IEEE Internet of Things Journal* **8,** 6549-6564. ISSN: 2327-4662 (abr. de 2021).

52. Kutseva, M. *Adaptation of Seven-Layered IoT Architecture for Energy Efficiency Management in Smart House* en *2022 10th International Scientific Conference on Computer Science (COMSCI)* (mayo de 2022), 1-5.

53. Swamy, S. N. y Kota, S. R. An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access* **8,** 188082-188134. ISSN: 2169-3536 (2020).

54. Krčo, S., Pokrić, B. y Carrez, F. *Designing IoT architecture(s): A European perspective* en *2014 IEEE World Forum on Internet of Things (WF-IoT)* (mar. de 2014), 79-84.

55. Bouaouad, A.-E., Cherradi, A., Assoul, S. y Souissi, N. *The key layers of IoT architecture* en *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)* (nov. de 2020), 1-4.

56. Singh, D., Tripathi, G. y Jara, A. *Secure layers based architecture for Internet of Things* en *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (dic. de 2015), 321-326.

57. Wu, Y., Wu, Y., Guerrero, J. M. y Vasquez, J. C. Digitalization and decentralization driving transactive energy Internet: Key technologies and infrastructures. *International Journal of Electrical Power & Energy Systems* **126,** 106593. ISSN: 01420615 (mar. de 2021).

58. Mohd Aman, A. H., Yadegaridehkordi, E., Attarbashi, Z. S., Hassan, R. y Park, Y. J. *A Survey on Trend and Classification of Internet of Things Reviews* 2020.

59. Alsubaei, F., Abuhussein, A. y Shiva, S. en *Internet of Things A to Z* 77-112 (John Wiley & Sons, Inc., mayo de 2018).

60. Swamy, S. N. y Kota, S. R. An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access* **8,** 188082-188134. ISSN: 2169-3536 (oct. de 2020).

61. Al-Awami, S. H., Mahfud Al-Aty, M. y Al-Najar, M. F. *Comparison of IoT Architectures Based on the Seven Essential Characteristics* en *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)* (mayo de 2023), 305-310.

62. Calderoni, L., Magnani, A. y Maio, D. *IoT Manager: a Case Study of the Design and Implementation of an Open Source IoT Platform* en *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (abr. de 2019), 749-754.

63. Qiu, R., Yu, J., Zheng, F., Liang, L. y Li, Y. *Electric IoT Perception Layer Data Privacy-preserving Using Multi-identity-based Fully Homomorphic Encryption* en *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)* (nov. de 2020), 30-34.

64. Lin, G. *y col. Research on IoT Perception Technology of Renewable Energy Based on Edge Computing* en *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)* (ago. de 2022), 36-40.

65. Xiao, Z., He, S., He, X., Guo, X. y Li, C. *Research on Perception Layer Architecture In Differentiated Application Scenarios for Power IoT:Evidence from Hunan Province* en *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)* (oct. de 2021), 3380-3385.

66. Ejaz, M., Kumar, T., Ylianttila, M. y Harjula, E. *Performance and Efficiency Optimization of Multi-layer IoT Edge Architecture* en *2020 2nd 6G Wireless Summit (6G SUMMIT)* (mar. de 2020), 1-5.

67. López Peña, M. A. y Muñoz Fernández, I. *SAT-IoT: An Architectural Model for a High-Performance Fog/Edge/Cloud IoT Platform* en *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (abr. de 2019), 633-638.

68. Mocrii, D., Chen, Y. y Musilek, P. IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*. ISSN: 25426605 (2018).

69. Labs, S. *AN1138: Zigbee Mesh Network Performance* inf. téc. (Silicon Labs, 2021). `https://www.silabs.com/documents/login/application-notes/an1138-zigbee-mesh-network-performance.pdf`.

70. Labs, S. *AN1142: Mesh Network Performance Comparison* inf. téc. (Silicon Labs, 2021). `https://www.silabs.com/documents/public/application-notes/an1142-mesh-network-performance-comparison.pdf`.

71. Labs, S. *AN1137: Bluetooth® Mesh Network Performance* inf. téc. (Silicon Labs, 2021). `https://www.silabs.com/documents/public/application-notes/an1137-bluetooth-mesh-network-performance.pdf`.

72. Labs, S. *AN1141: Thread Mesh Network Performance* inf. téc. (Silicon Labs, 2021). `https://www.silabs.com/documents/login/application-notes/an1141-thread-mesh-network-performance.pdf`.

73. Kalalas, C., Thrybom, L. y Alonso-Zarate, J. Cellular Communications for Smart Grid Neighborhood Area Networks: A Survey. *IEEE Access* **4,** 1469-1493. ISSN: 2169-3536 (2016).

74. Tang, Q., Ermis, O., Nguyen, C. D., Oliveira, A. D. e Hirtzig, A. A Systematic Analysis of 5G Networks With a Focus on 5G Core Security. *IEEE Access* **10,** 18298-18319. ISSN: 2169-3536 (2022).

75. Raza, U., Kulkarni, P. y Sooriyabandara, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials* **19,** 855-873. ISSN: 1553-877X (2017).

76. Sanchez-Gomez, J. *y col.* Integrating LPWAN Technologies in the 5G Ecosystem: A Survey on Security Challenges and Solutions. *IEEE Access* **8,** 216437-216460. ISSN: 2169-3536 (2020).

77. Khalifeh, A., Aldahdouh, K. A., Darabkh, K. A. y Al-Sit, W. *A Survey of 5G Emerging Wireless Technologies Featuring LoRaWAN, Sigfox, NB-IoT and LTE-M* en *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)* (mar. de 2019), 561-566.

78. Pitu, F. y Gaitan, N. C. *Surveillance of SigFox technology integrated with environmental monitoring* en *2020 International Conference on Development and Application Systems (DAS)* (mayo de 2020), 69-72.

79. Chen, M., Miao, Y., Hao, Y. y Hwang, K. Narrow Band Internet of Things. *IEEE Access* **5,** 20557-20577. ISSN: 2169-3536 (2017).

80. Faber, M. J., van der Zwaag, K. M., dos Santos, W. G. V., Rocha, H. R. d. O., Segatto, M. E. V. y Silva, J. A. L. A Theoretical and Experimental Evaluation on the Performance of LoRa Technology. *IEEE Sensors Journal* **20,** 9480-9489. ISSN: 1558-1748 (ago. de 2020).

81. Tomar, R. y Gemein, O.-G. *LoRa network for cities Private and complete secured by design* en *2018 Global Internet of Things Summit (GIoTS)* (jun. de 2018), 1-5.

82. Astrain, J., Falcone, F., Lopez, A., Sanchis, P., Villadangos, J. y Matias, I. *Monitoring of Electric Buses within an Urban Smart City Environment* en *2020 IEEE SENSORS* (oct. de 2020), 1-4.

83. Vicente, G. y Marques, G. *Air Quality Monitoring through LoRa Technologies: A Literature Review* en *2020 International Conference on Decision Aid Sciences and Application (DASA)* (nov. de 2020), 350-354.

84. Kanakaraja, P., Kotamraju, S. K., Nagulmeera, S., Reddy, Y. D. y Divya, A. *LoRA based Indoor Localization using LPWAN Gateway and BLE Beacons* en *2022 International Conference on Electronics and Renewable Systems (ICEARS)* (mar. de 2022), 683-687.

85. Khan, F. U., Awais, M., Rasheed, M. B., Masood, B. y Ghadi, Y. A Comparison of Wireless Standards in IoT for Indoor Localization Using LoPy. *IEEE Access* **9,** 65925-65933. ISSN: 2169-3536 (2021).

86. Amjad, A., Azam, F., Anwar, M. W. y Butt, W. H. A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT. *IEEE Access* **9,** 96528-96545. ISSN: 2169-3536 (2021).

87. Shah, P., Arora, M. y Adhvaryu, K. *Lightweight Cryptography Algorithms in IoT - A Study* en *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (oct. de 2020), 332-336.

88. Karie, N. M., Sahri, N. M., Yang, W., Valli, C. y Kebande, V. R. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* **9,** 121975-121995. ISSN: 2169-3536 (2021).

89. Hemram, S., Kathrine, G. J. W., Palmer, G. M. y Ewards, S. V. *Firmware Vulnerability Detection in Embedded Systems and Internet of Things* en *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)* (nov. de 2022), 1161-1167.

90. Sindhwani, M. y Srikanthan, T. *Framework for Automated Application-Specific Optimization of Embedded Real-Time Operating Systems* en *2005*

5th International Conference on Information Communications & Signal Processing (dic. de 2005), 1416-1420.

91. Anisetti, M., Ardagna, C. A., Bena, N. y Bondaruc, R. *Towards an Assurance Framework for Edge and IoT Systems* en *2021 IEEE International Conference on Edge Computing (EDGE)* (sep. de 2021), 41-43.

92. Pandey, M. y Kwon, Y.-W. *Middleware for Edge Devices in Mobile Edge Computing* en *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)* (jun. de 2021), 1-4.

93. Ferreira, L. C. B. C. *y col.* Edge Computing and Microservices Middleware for Home Energy Management Systems. *IEEE Access* **10,** 109663-109676. ISSN: 2169-3536 (2022).

94. Ray, P. P. A survey of IoT cloud platforms. *Future Computing and Informatics Journal* **1,** 35-46. ISSN: 23147288 (dic. de 2016).

95. Singh, K. J. y Kapoor, D. S. Create Your Own Internet of Things: A survey of IoT platforms. *IEEE Consumer Electronics Magazine* **6,** 57-68. ISSN: 2162-2256 (2017).

96. Zdravković, M. *y col. Survey of Internet-of-Things platforms* inf. téc. (HAL, 2016), 216-220.

97. Marosi, A. C. *y col.* Toward Reference Architectures: A Cloud-Agnostic Data Analytics Platform Empowering Autonomous Systems. *IEEE Access* **10,** 60658-60673. ISSN: 2169-3536 (2022).

98. Lanza, J. *y col.* Experimentation as a Service Over Semantically Interoperable Internet of Things Testbeds. *IEEE Access* **6,** 51607-51625. ISSN: 2169-3536 (2018).

99. *Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1* Standard (International Organization for Standardization, Geneva, CH, jun. de 2016).

100. *Sparkplug 3.0.0: Sparkplug Specification* Standard (Eclipse Foundation, nov. de 2022).

101. Wikimedia. *Wikipedia* Wikimedia. `https://www.wikipedia.org/` (2024).

102. TSK, G. *Grupo TSK* Grupo TSK. `https://www.grupotsk.com/` (2024).

103. Ecoener. *Ecoener* Ecoener. `https://www.ecoener.es/` (2024).

Apéndice A

An IoT Open Source Platform for Photovoltaic Plants Supervision

Electrical Power and Energy Systems 125 (2021) 106540

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

An IoT open source platform for photovoltaic plants supervision

Pedro de Arquer Fernández [a,*,1], Miguel Ángel Fernández Fernández [a], Juan Luis Carús Candás [a,1], Pablo Arboleya Arboleya [b,1]

[a] R&D&i Department at TSK Electrónica y Electricidad S.A., Technological Park of Gijón, 33211, Spain
[b] Department of Electrical Engineering, University of Oviedo, Campus de Gijón, 33204, Spain

ARTICLE INFO

Keywords:
Internet of Things
Photovoltaic
SmartGrid
Industry 4.0
Eclipse Kapua
Eclipse Kura
Real-time monitoring

ABSTRACT

In the present work, the authors propose an IoT solution for photovoltaic plants monitoring based entirely on Open Source software. The described solution is implemented and deployed in a real plant of approximately 3 MW with a total number of 24 inverters and 156 string boxes. All details about software and hardware architecture are provided, proving that it is possible to develop a flexible, versatile and competitive monitoring system just using available Open Source tools. In addition, the authors make a detailed comparison between the proposed IoT solution and conventional SCADA-based monitoring systems, describing all benefits and drawbacks of conventional SCADAs and modern IoT systems and proposing solutions in order to overcome the identified weak points of the newest IoT-based monitoring systems.

1. Introduction

Historically, the Supervisory Control And Data Acquisition (SCADA) systems are the core of industry automation as shown in Fig. 1. Downstream in the automation pyramid, the SCADA is the system responsible for monitoring, control and coordination of work procedures among all the devices in the industry. Upstream, the SCADA acts as an interface for human supervision and Manufacturing Execution Systems (MES) easing decision making and the control of the industrial process.

The general structure of a SCADA system contains multiple sensors and actuators, controlled and communicated through Remote Terminal Units (RTU). It also includes Human Machine Interfaces (HMI) so that human actions can be introduced in the system based on production and safety requisites, allowing the supervision of the process.

This structure has become more a more complex along the years as the business logic requires more control and information on the production process. The processes require more adaptability to the market [1] and the product itself must be continuously improved to be competitive. These requirements have made the SCADA model unable to adapt fast enough to the requirements in multiple industrial areas, resulting in an over-complexity of the industrial network and interaction between multiple vertical systems [2,3]. This is the case of photovoltaic industry, where recent technology developments are pushing the traditional industrial systems to face new challenges.

The Industry 4.0 is an emerging paradigm supported by multiple Key Enabling Technologies (KET) -such as advanced robotics, 3D printing, or Big Data Analytics- that provides new opportunities for SCADA systems to be redefined for a more adapted industry model. Usually, these technologies provide solutions that are easier to maintain, have better integration with third party applications, use more efficient communication channels and can be achieved more cost efficiently [4]. These improvements are critical to enhance the SCADA model to a more future-proof solution.

One of the most promising technologies included in the Industry 4.0 paradigm is the Internet of Things (IoT) [5]. IoT is a network architecture motivated initially by wireless and low consumption communication technologies. In the following years IoT turned to a complete stack of technologies for field-to-cloud communication layer [6–8].

The IoT technologies make possible the communication of shop floor components with the consumer of such component (whether the consumer is a human, another machine, a business software or a service allocated in the cloud) in a structured and secure way. Such pattern eases also both horizontal and vertical integration of third party solutions, providing a more expandable model of production.

* Corresponding author.
E-mail addresses: pedro.arquer@grupotsk.com (P. de Arquer Fernández), miguelangel.fernandez@grupotsk.com (M.Á. Fernández Fernández), juanluis.carus@grupotsk.com (J.L. Carús Candás), arboleyapablo@uniovi.es (P. Arboleya Arboleya).
[1] This work was partially funded with the Industrial PhD Program form the Ministry of Science and Innovation, Gonvernment of Spain under the grant DIN2019-010585 / AEI / 10.13039/501100011033.

https://doi.org/10.1016/j.ijepes.2020.106540
Received 30 March 2020; Received in revised form 10 September 2020; Accepted 18 September 2020
Available online 9 October 2020

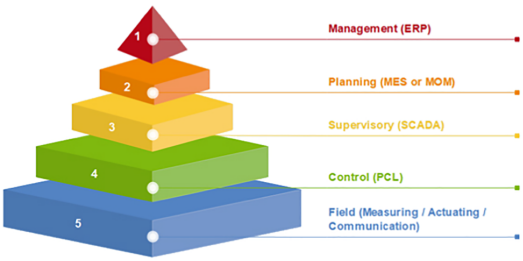


Fig. 1. Automation Pyramid. Traditionally, the levels are only connected with the previous or next level.

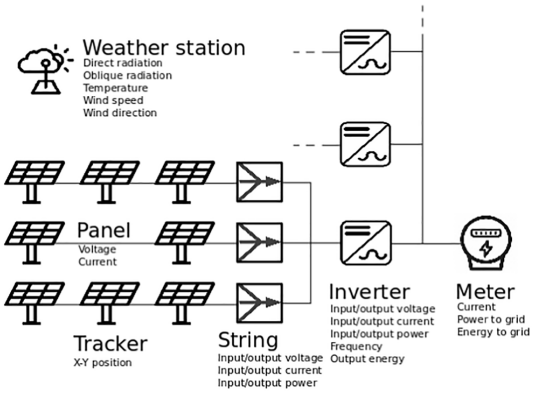


Fig. 2. Typical composition of a photovoltaic installation. A plant may be composed by multiple installations sharing the weather station.

IoT as a monitoring paradigm can be applied to multiple industries. Among them, photovoltaic plants are some of the most interesting. This is due to the increase in installed power every year and the constant evolution that this industry will suffer in the following years to adapt to new models of consumption and generation: electric vehicles, self-generation, data transmission, etc.

In this article, a solution based on IoT technology and Open Source (OS) software is described and deployed in a photovoltaic plant. The improvement over a previous solution is quantified, while the plant is upgraded to a new monitoring system needed to tackle the monitoring challenges.

In the next section, the photovoltaic domain will be described along with its requisites and challenges. Later, in Section 3, the IoT architectures and contributions will be discussed. Out of the photovoltaic requirements and IoT strengths, a solution will be developed in Section 4 trying to meet the challenges of both domains. Then, in Section 5, the use case and results will be discussed prior to obtaining the conclusions and the lines for future research that this article puts forward.

2. Photovoltaic scenario

Nowadays, the world production of photovoltaic (PV) energy increases every year due to the environmental benefits and the advantages it provides to the energy industry when compared with the rest of the renewable sources [9,10]: low maintenance and simple production. In the past years, a total 512 GW of PV power has been installed [11] and it is the most promising renewable energy in terms of installed power per year.

As the impact on the grid increases, the control on the production process must be higher to be more reliable, profitable and competitive [1]. The technology used in PV plants is currently being upgraded as the solar panels produce energy more efficiently, the auxiliary systems are more reliable and the communications become more critical. It is particularly relevant in applications related with the smartgrid trend [12–14]. The monitoring systems for such plants are supporting more complex scenarios with every new installed plant.

For a full understanding of the PV monitoring, a brief review follows on the components of PV plants and the key variables that must be measured and monitored. Next, the limitations will be remarked and the current challenges on the field will be described.

2.1. PV plant composition

A PV plant is mostly composed of solar panels to produce the electricity, and a set of electric elements that adapt such electricity to the rating required on the grid. A general scheme of a PV power plant is depicted in Fig. 2. The key elements for such production are:

- Photovoltaic panels: which capture the energy from sun radiation and convert it into Direct Current (DC) electricity.
- Trackers: used sometimes to support the solar panel and rotate it (on horizontal or both the horizontal and vertical axis) to achieve the most perpendicular orientation with the solar beams.
- String: formed by the connection of several panels to provide enough voltage for inverter input.
- Inverter: which converts the input DC to Alternate Current (AC).
- Meter: which measures the final power delivered to the grid.
- Weather station: they are equipped with multiple sensors (temperature, radiance, humidity, pressure, wind speed, etc.) to provide additional information to the SCADA system about the production conditions.
- Batteries: Used sometimes to store the excess of production or provide extra power when the production declines.

The variables provided by these elements must be retrieved by an infrastructure layer devoted to monitoring and communication [15]. Such variables are required to monitor the performance of the PV plant and control the maintenance of the devices. The most significant variables considered are, according to S. R. Madeti et al. [1]:

- PV current and voltage: to create the I-V curve and evaluate the performance of the plant.
- Power to utility grid: needed to evaluate in Real-Time the efficiency of the PV plant and the production injected to the grid.
- Energy to utility grid: used to monetize the production within a period.
- Consumption from the grid: used to feed some auxiliary systems and when there is no radiation.
- Voltage, current, power and energy: measured in different elements to monitor losses.
- Solar radiation: which sets a reference measure of the expected power to be generated in the solar panel.
- Ambient temperature: it has an impact in the productivity of the PV plant and the maintenance of the devices.

The status of the plant, its production and some other Key Performance Indicators (KPIs) are calculated from these variables to complete the monitoring requirements and ease the understanding of the data. On one hand, with a proper calibration and monitoring system, some of the variables may be used to provide a very precise measure of the losses in the production, such as the difference between the produced power and the injected power or the difference between the solar radiation that fall upon the PV panels and the power produced by them. It allows a very specific actuation of maintenance or optimization. On the other hand, a bad monitoring system would be unable to point to the critical element with accuracy leading to severe economical losses.

Moreover, the monitoring system must be able to provide a set of alarms or events related to the devices and the plant, such as: production stop, performance under expected, disconnection of a device, faults on the monitoring system, etc.

2.2. Monitoring challenges

To perform an efficient monitoring, the PV plant must be provided not only with a well calibrated equipment, but also a set of auxiliary devices. Such auxiliary systems must integrate with measuring components and get the most of them. Both systems would compose the SCADA of the plant. The SCADA systems must face challenges that are often hard to solve with traditional solutions. Such challenges are [1,10]:

- Interoperability: the PV plants have an increasing need for both horizontal and vertical integration and good interoperability among their systems. The monitoring infrastructure contains a wide catalogue of equipment and manufacturers. They are specifically calibrated for a set of measures and provide communication with different protocols (sometimes even required by law). The vertical integration is mostly required on cases where multiple plants must be monitored even if their SCADA differs due to different technology or installing company.
- Management: the upgrade and monitoring of the industrial devices or the infrastructure can simplify the maintenance and allow a more active action instead of reactive fixes. For this purpose, good communication systems are also needed because the plants are usually large and isolated which means that all the monitoring must be done remotely. The communications are also required for video surveillance and typically for remote access by energy companies to the meter of the plant. The whole network load is quite intense and the quality of the connections tend to be inadequate due to the isolation of the plant.
- Security: as communications and remote connection require internet access from the SCADA, the system must provide cybersecurity measures and avoid direct exposure of devices that are not ready to support internet cyber attacks.
- Processing: the systems in the SCADA and the monitoring infrastructure must be capable to provide reliable data. Data must be collected up to a frequency of 1 s for a good Real-Time analysis. The data could also be processed to generate relevant metrics for the monitoring system like the power ratio of the plant or the accumulated energy produced if the monitoring devices do not provide this information.
- Scalability: the size of the plants is very variable as it is possible to create a PV installation with a few solar panels for self-consumption or a large plant with hundreds of MW of installed power. A PV monitoring solution must adapt to these use cases without oversizing. Highly related with interoperability challenge, the infrastructure must be also future-proof to allow legacy and new systems interaction or the scalability will be also compromised.
- Cost: the cost is always considered in industrial deployments, both initial and maintenance costs. The maintenance cost must include also the endurance of the devices as the PV scenarios are typically in harsh and isolated environments and most monitoring devices must be outdoor exposed to sun and dust. There is also the need for low power systems. This need is not set for autonomy reasons but because the consumption affects directly on the overall production of the PV plant.

These challenges would most likely imply a specific and hardly extensible development. This is the case of some of the most recent monitoring systems available [1]. In these systems, the data is collected locally by industrial PCs and accessed through applications developed by the providers. The whole platform is proprietary and does not provide integration with third party systems. They may also require specific components and sensors for measurement and the visualization tools are sometimes rudimentary or hardly adaptable.

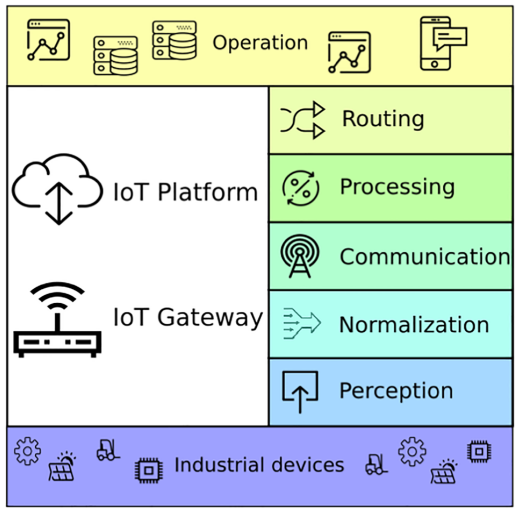


Fig. 3. IoT architecture layers. Depending on the implementation, the functionality is distributed between Platform and Gateway.

However, in the context of IoT architectures, it is possible to create a more generic solution even if it is customized for a specific scenario. They can leverage the IoT technologies to fulfill some of the exposed challenges such as, for example, the low consumption restriction that can be achieved using single board computers without losing the high level processing required for an optimum solution. More advantages will be described below.

3. IoT infrastructure

Besides the components cited above, a PV plant always has a networking layer to allow monitoring and communications with remote systems. The devices involved in this layer are mostly industrial PCs, routers and switches. Networking layer is a very promising field of application for IoT technologies due to their communications and processing capabilities.

The IoT are a set of technologies that come together to provide very versatile and functional solutions in the industry. These protocols and devices can be customized for an optimal functionality in very heterogeneous cases. For this reason it is needed to review the architecture in an IoT solution prior to analyze its contribution to industrial monitoring and specifically the PV sector.

3.1. Architecture layers

The IoT solutions are intrinsically heterogeneous and their structure is also variable. There is no consensus on a single IoT architecture nor IoT solutions, as shown in [16]. However, in all the approaches prevail a general set of functionalities and challenges that any architecture must tackle to develop the full potential of the infrastructure [17]. As shown in Fig. 3, the functionalities are:

- Perception: it is devoted to communication with industrial devices both for sending and receiving data.
- Normalization: data need to be standardized before being used across the architecture and sometimes filtered or compressed for a better communication.

- Communication: one of the greatest contributions of IoT architectures is the variety of strategies for communication depending on the environment: lack of wiring, bad coverage, distributed networks, etc.
- Processing: the acquired data must be processed to provide new metrics, to set alarms or to perform a critical reaction.
- Routing: once the data is in the architecture it must be easily provided to other tools for operation (monitoring and visualization, further processing or management).

These functionalities are provided by one or more IoT gateways which usually manages the perception and normalization of data and a single IoT platform that processes and routes the information for proper exploitation. However, the distribution of the functionalities may vary among solutions depending on the requirements set by users and the conditions of the industry. The main challenges that the IoT gateways typically face are the heterogeneity of the devices and protocols found, while the IoT platform must solve management and usability issues.

3.1.1. IoT gateway

There is a vastly heterogeneous range of IoT gateways that can be used for lower layers of the architecture. There is also a wide variety of IoT middleware that can be executed in the gateways. The characteristics of both gateway and middleware are very different depending on their capabilities [18], the connection requirements or architectural approach among others. However, some of the most typical [19] and most interesting features for industry monitoring can be enumerated:

- Data management: acquire and process data. It is a must-have feature in both IoT and industrial applications.
- Event management: generate events based on conditions. Important on industry as the processes are usually event-driven.
- Timeliness: provide Real-Time response. Critical on multiple industry systems.
- Scalability: allow horizontal growth. To provide a flexible business model.
- Reliability: the basic functionality must always be provided. It is required for both production and safety reasons.
- Security: reject unauthorized access. Also required for both production and safety reasons.

Some additional features can be found on multiple IoT middleware, like resources discovery or code management. Those are not strictly needed for an industrial process. They can be considered additional benefits when present, but not requirements.

3.1.2. IoT platform

The IoT platforms have appeared as a necessity while the catalogue of IoT devices got increased and their heterogeneity was no longer manageable with vertical solutions. The IoT platforms cover a large set of features taking advantage of the capabilities of the IoT devices. Some of the most usual features are:

- Software management: download, upgrade and provide version control of the software in the IoT devices.
- Monitoring: track the status of the IoT devices and some of the device metrics, such as memory usage, CPU, storage, execution state, etc.
- Broker: centralize the data from the IoT devices and offer them an optimized communication interface (using IoT protocols).
- Processing: execute calculations over the data received from the IoT devices.
- Routing: provide the routing interface for the data, allowing a centralized model of the data flow to increase the control over it.
- Security: authenticate and encrypt the communication to the IoT devices.

The list of features of an IoT platform can vary from one to another depending on the application, and some could provide extra capabilities, such as programming interfaces, operation dashboards or on-premise distribution.

Depending on the applications, IoT platforms can be found as a single horizontal platform or as multiple vertical IoT solutions with their platform. A natural evolution from this conception are the platforms of platforms that lately have restructured themselves as IoT platforms with high integration capabilities against other platforms, in case vertical solutions must be integrated.

3.2. Industry applications

The application of IoT technology in the industry has increased as new use cases are being found and the companies have seen the potential in the creation of industry solutions based on IoT [20,17,21].

At first, the IoT focused companies, such as Carriots or ThingWorx, started with generic IoT solutions for home automation, health care or smart cities. As the IoT paradigm became more mature, their solutions were more often used for industrial testbeds and these companies have increased their efforts to provide the services and features required by the industrial sector. Some additional examples of cases like this are Xively, ThingSpeak, ThingStream or Libelium.

Following the trend of the IoT technologies, leader software companies as Microsoft or IBM have created products and solutions for IoT and specifically focused on industrial environments. Their development and commercial efforts have increased the attention on the topic and increased the global investment on the area. Their contribution would be to provide tools or guidelines for the IoT platform or cloud architectures. Some of the solutions provided by these companies are Microsoft Azure, Google Cloud, IBM Watson or Oracle.

As the solutions are increasingly demanded from industrial sectors, the classical industrial providers (Siemens, Schneider or ABB) have also created solutions and integrated their systems with IoT technologies stack. The main value of these companies is the provision of IoT devices capable of working on industrial and harsh environments. There is also an effort of these companies to provide improved development and management platforms (Mindsphere, Wonderware, etc.) that could be resembled as vertical IoT platforms.

Additionally, since the first appearance of IoT solutions, lots of companies have increasingly invested to create products or solutions of IoT from their own field of application, such as communications (Vodafone, Samsung, Huawei), management software (Salesforce, SAP), automation (Bosch) or services (Amazon).

Finally, there are multiple projects of IoT products created from the community, particular developers or relinquished from proprietary solutions. This is the case of Kepware, Eurotech or Evrythng, whose solutions have been at some point shared with the OS community although they keep working on a more advanced solution under a proprietary product. Other solutions are currently supported by a community of developers who keep working on the project like Kaa, Eclipse Kura and Kapua, Apache Camel, etc.

Based on these solutions, several use cases have been successfully deployed in the industry in very different scenarios, making the most of the multiple advantages of the IoT technology stack. Some of the most meaningful of them are:

- Scalability has allowed Algae Lab Systems [22] to implement a solution to keep their current management platform among multiple plants.
- Edge processing could be used to provide cybersecurity features to a SCADA, such as traffic analysis [23] or improve functionality in wide spread scenarios [24].
- Semantic improvement [25] of industrial processes allows a better Artificial Intelligence integration.

Table 1
Advantages and disadvantages of IoT solutions in industry.

Feature	Purpose	Advantages	Disadvantages
Interoperability	Integrate heterogeneous data and solutions	Abstracts downstream devices functionality	Lack of well established standards Loss of functionality due to abstraction
	Enable vertical integration (third party solutions)	Wide catalog of devices and protocols available	Architectures become complex
Management	Monitor applications and devices	Provides monitoring tools (via Iot Platform) Provides detailed diagnostics information	Requires a specific IoT platform
	Easy update of hardware and software	Provides provisioning tools (via IoT Platform) Low hardware requisites	
	Ease the user experience	Typically provides visualization tools	Permission management becomes complex
Security	Ensure the availability of the infrastructure	Versatile communication technologies	Internet may expose the industrial network Wireless communications can be intercepted
	Guarantee confidentiality	Capacity to process encryption algorithms	Increases communication complexity
	Authenticate devices	Capacity to store and manage certificates	Certificate management increases complexity
Processing	Avoid process interruptions	Can integrate high-reliability devices	Lack of stress validation of multiple solutions
	Provide Real-Time data	Can integrate high-performance devices	High rates increases consumption
	Verify data integrity	Enables Real-Time validation	Rules management increases complexity
	Industrial process optimization	Provides advanced processing tools	Processing management increases complexity
Scalability	Ease horizontal expansion	Allows dynamic integration of devices Can integrate data from different scopes	Lack of validation in wide deployments Data management is typically decentralized
	Decrease start-up delay	Can be adapted to very different solutions	Initial steep development curve
Cost	Minimize overall cost	Low consumption devices	Low consumption compromises functionality
		Low requisites devices	Software expenses gets increased

- Indexing nodes [26] of a mixed network with ordinary and IoT devices would improve the quality of the communications and create a more scalable infrastructure.
- Online monitoring [27] would allow safety yet accurate monitoring of a continuous steel casting process to improve the quality of the final product.
- Advanced image analysis [28] on infrared data of electrical substations allows the monitoring of its condition and decreases maintenance costs.
- Remote monitoring of photovoltaic production can be used to provide a unified architecture to integrate multiple PV plants [29].

Depending on the scenario, IoT technology is used to reduce maintenance, increase the quality of the product, follow the life cycle of the product, improve the efficiency of the process or develop an scalable monitoring platform. There are also some usages of IoT technology for cybersecurity or industry control [30,31].

3.3. Discussion

The usage of IoT technology for monitoring brings multiple advantages to the industrial process and SCADA, but also implies some inconveniences. The Table 1 [32–35] summarizes some of the most relevant advantages and disadvantages related to the deployment of IoT solutions on industrial environments.

On one hand, most of the difficulties related to the application of IoT are caused by bad implementations and lack of knowledge about the technologies, which also provokes long time development efforts. The result is typically disappointing, insecure and low in performance. However, there are some issues that remains in well implemented solutions: the lack of IoT standards (which does not guarantee the solution is future-proof), the increase of management and complexity (as new systems are introduced in the environment), and the difficulties of merging the Information Technologies (IT) and Operational Technologies (OT) environments (specially in partial deployments).

On the other hand, if the solution is properly set up, it increases the efficiency of the monitoring solution and can be easily replicated as the components implied are more configurable than typical monitoring systems. The time and cost effort for future implementations decreases greatly.

It also provides extra functionality not directly related with the process such as allowing the management of non-IoT devices (PLCs, RTUs or dataloggers for instance), provide cybersecurity tools in SCADA networks, advanced processing on remote devices (such as computer vision or machine learning) and horizontal architecture for future implementations. Even if the used tools become obsoletes, the connectivity possibilities remain and the solution can be recycled.

The solution proposed in this paper has been designed considering the most critical disadvantages and the most interesting advantages presented in this section.

4. Solution proposed

In the IoT solutions studied previously the disadvantages could be summarized as: lack of standards, inefficient end-devices, complexity in mixed implementations (both IoT and legacy SCADA systems) and an intensive initial effort. The system described next is focused on these issues to provide a solution that resolves them.

Additionally, the most challenging difficulties in the PV industry (cited in Section 2.2) are targeted: communications, cybersecurity, heterogeneity of integrated solutions, low power requirements and integration capabilities.

4.1. Architecture

To tackle these challenges, an architecture adaptable to different applications is required. The needs of the different applications are often similar, but the challenges and scenarios are very heterogeneous. To integrate such needs, the following structure is proposed:

- Industrial devices: those already present in the industrial environment provided by (or integrated with) sensors, typically capable to

communicate via serial or TCP interfaces and in a wide variety of protocols, whether standard (Modbus, OPC, ProfiNET, IEC-60870-5, etc.) or proprietary. Due to this layer, the final solution takes advantage of well tested devices and their good characteristics for the industrial process.
- IoT Gateway: provided with the functionality to collect, process, store and export the data from the industrial devices. It must be very adaptable to multiple scenarios and normalize the data to simplify upstream management.
- IoT Platform: to manage the IoT devices and centralize the communications and data. It must be scalable and provide full integration with the IoT Gateway.
- Operation: to execute specific business related processing and human monitoring. It sets the integration requirements for Gateway and Platform software.

In the PV use case, the industrial endpoint contains a set of protocols that must be developed (particularly when the protocol is proprietary), but most of them are common to other industries. The tool at the top of the architecture (operation) is a monitoring dashboard that requires Real-Time access to the data in a standard format. It must provide the most interesting metrics for the workers that monitors it. Both integration points (industrial devices on the bottom of the stack and monitoring interface on the top) may vary among projects, but the Gateway, Platform and data flow remain unaltered no matter the application or use case.

4.2. Software

The search of an adaptable, high performance and cost effective solution led to Eclipse Kura, which is an OSGi (Open Services Gateway initiative) based software developed under an Open Source (OS) license. OSGi systems set a basic framework where software can be deployed as plugins and serve their functionality to the rest of the plugins. It makes the system highly adaptable. Kura framework also allows the creation of a work flow on graphic format to ease the configuration. The exchange of information among the pieces of software can be used for both Real-Time and batch processing tasks providing an environment valid for high performance solutions. Finally, OS projects can be developed by anyone and freely distributed, although they can also be distributed as part of a proprietary solution. This kind of projects allows a permanent update of the software based on the collaboration of the users. Besides, the democratic approach of the software in IoT (where no standards have yet stand out) is also considered as a strength to create de facto standards.

The main advantages of this approach match the main issues and requirements identified for both photovoltaic and IoT systems in the beginning of this section, such as:

- Standard: as an OS project the software is developed attending to a community of actual users and the solution is more globally valid.
- Efficiency: the software can be deployed in multiple devices and it can integrate other systems to maximize the efficiency of the solution.
- Mixed implementations and heterogeneity: both can be solved taking advantage of the plugin architecture of the framework.
- Initial effort: deployment and learning process is eased by the presence of a community and a set of already developed plugins.
- Communications: the framework integrates multiple communication technologies (WiFi, Ethernet, Bluetooth, serial).
- Cybersecurity: it is considered from the base of the project in the management of certificates, the encryption of configurations and the web interface. The provision of frequent updates improves also the security of the tool.
- Integration: it can also be faced thanks to the modularity of the framework and it is specifically enabled for MQTT and REST integration.
- Low power: the framework can be used in minimal devices whose consumption do not exceed 1 W if needed (e.g. Raspberry Pi).

Additionally, the framework provides support for advanced functions such as operations or even the integration of an external solution in the software. The software can also be monitored and managed through an MQTT connection to deploy software, configure, request data or configurations, reboot the system, etc. Thanks to MQTT protocol, the communications are lightened and it can be easily integrated with other IoT solutions.

On the other hand, despite the poor requirements Kura holds, it is needed a minimum of processing capabilities in the device and an Operating System. It is more conceived for IoT gateways than minimal IoT nodes. Besides, the software is still being tested on multiple devices and some stability issues are still being faced.

As a complementary tool for Kura framework, it is also used Eclipse Kapua. Kapua is an IoT platform conceived specifically for Kura but whose management tools are published to allow the inclusion of third party software. It uses a MQTT communication for both transmission of data and management of the devices. It also provides a REST API and a console (web interface) for the use of the data. As Kapua is an OS project, it also holds the advantages of the Kura framework regarding this feature.

Kapua platform is composed of the following blocks:

- Main database: to store the data related with the management of the devices, the users and roles and the structural information of the broker, web and REST API. A H2 database is deployed for this purpose.
- Messages database: to persist the data received from the devices. Data is stored in an ElasticSearch database.
- Events broker: to allow the communication and transmission of events among the services presented in this list. Events are managed by an ArtemisMQ broker.
- Broker: to communicate with the devices and provide the data to the REST API and console, it is the most important service in the framework as it processes the permissions, transmit the messages and basically performs all the most important part of the functionality of Kapua.
- Console: to provide a human interface to manage Kapua. It allows to create roles, users and accounts (separated workspaces), it can be used to manage the devices, execute tasks on them or read the data received.
- REST API: to provide integration tools for Kapua management. Its functionality is mostly (but not entirely) replicated with the web interface.

4.3. Hardware

The hardware required to hold the software described above would imply two parts: an IoT gateway to hold Kura framework and a server to deploy the Kapua platform.

More specifically, the IoT gateway requisites are a Linux Operating System (no specific distribution but some functionality could be lost depending on which), 1 GHz of CPU, 500 MB of RAM memory and 100 MB of storage. These requisites can be achieved by lots of commercial devices. Some of the possible devices that has already been evaluated are:

- Raspberry Pi 2 B+: capable to run the framework properly (it is specifically maintained by the developers) but it launched slowly.
- Artila Matrix 700: capable to hold the framework and run it as a stable tool but it lacks of performance when web interface is used.

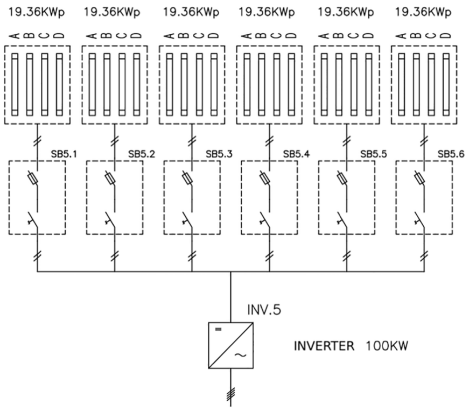
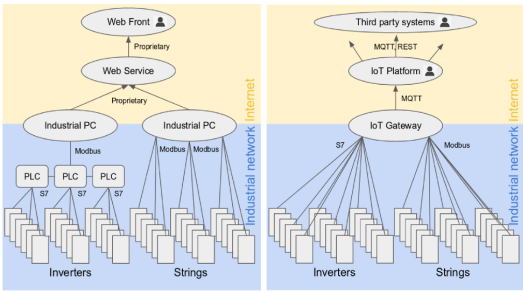


Fig. 4. Blueprint of one inverter of the plant. All the inverters replicate the same pattern.



(a) Previous solution (b) Final solution

Fig. 5. Monitoring system in the plant before (a) and after (b) the IoT integration. Previous solution lacked of integration capabilities and the communication network was more complicated.

- Moxa UC-2112-LX: which runs Kura with plenty functionality and performance but is in the limit of the minimum requisites and may not be good enough for some applications. On simple use cases it is optimum.
- Lanner LEC-7230M: that is capable to run without limitations the Kura framework and could cover very complex use cases and processing requirements. It results, however, a bit oversized for most of the use cases.

Kapua requisites are not so restrictive, as it is intended to be used in a Data Center or an industrial server. It is available to be launched over container technologies like OpenShift or Docker but it can also be installed directly on the system. The specific hardware requirements depend on the scale desired, but it has been achieved some minimum for a proper deployment of 4 GB or RAM and nearly 10 GB of storage. It was used an ordinary CPU of a computer, more specifically it was a 64 bits CPU with around 2.7 GHz and 4 cores.

5. Implementation

The purpose of the implementation is to improve the quality of the solution, reduce deployment and maintenance costs, provide an estimation of the performance of the software tested and obtain a series of best practices for future works.

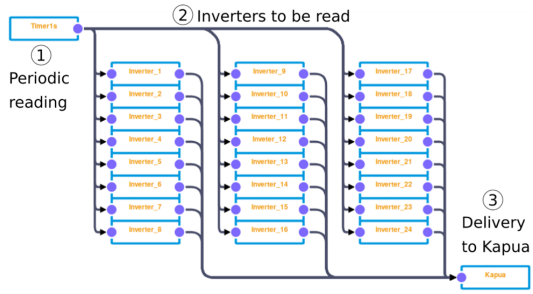


Fig. 6. Work flow of Kura Framework (inverters). Strings workflow includes more devices with and identical schema.

5.1. Case of use

The use case selected for the solution was a medium size photovoltaic plant in Spain with approximately 3 MW of installed power. The plant is composed of strings, inverters, meters and a weather station. The solution was targeted to monitor the inverters and string boxes present in the plant. There are a total of 24 inverters and 156 string boxes connected hierarchically, as the Fig. 4 shows. These devices produces more than 500 variables. Both inverters and strings were previously monitored with a commercial solution. Such solution required an additional set of three Siemens PLCs, shown in Fig. 5a, a, to handle the communication with the inverters and expose their metrics in Modbus protocol, because the solution was unable to perform a direct communication. Additionally the SCADA includes several communication modules to convert to digital value the strings measurement.

To deploy the solution, it was used a Moxa UC-2112 PC with Kura installed as IoT Gateway. The device connects directly with the strings communication modules and the inverters. With this implementation, the read speed was increased and it allowed also the possibility of make parallel readings as every communication was done independently. The PLC is no longer needed in the infrastructure as shown in Fig. 5b.

It was also deployed a Kapua instance which was configured to enable a secure connection from Kura clients by MQTT protocol using SSL certificates managed by Kura environment. The device connected was configured to read the data from the industrial devices periodically and send the data to Kapua server. The Fig. 6 shows the work flow defined in the Kura framework for the inverters reading (strings are configured equally with more devices). The data could be extracted from Kapua with the same MQTT connection or via REST API, but also viewed in its web console using a simple data explorer. An advanced display tool should be provided independently.

Both Modbus and S7 protocols and the MQTT connection with Kapua were already implemented and supported for the Kura framework so there was no need of software development but only the configuration of Kura and Kapua platforms, reducing greatly the costs and time required. Additionally, the communication with the Kapua platform enables a configuration interface via MQTT as cited in 4.2. This connection allows remote management of the device: monitoring of the status of Kura and the device, configuration of the services in Kura, reading of the industrial devices and upgrade of the software components installed.

5.2. Performance

The monitoring system provided by the solution led to multiple advantages:

- Improved refresh period: the refresh period of the data was improved from 10 min to 5 s. The data can be now processed for Real-Time analysis and the irregularities of the I-V curve can be detected.

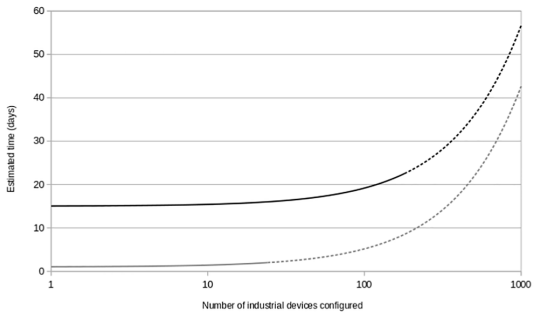


Fig. 7. Days required to deploy Kura initially (black line) and the second time (grey). Discontinuity represents estimation.

- Reduced costs: due to the Open Source software, development requires no inversion. Besides, the software does not require licences to be paid periodically. Considering that the medium life of a PV plant is 30 years, it has a great impact in long term expenses. The concepts are broken down in Table 3 and totalled in a short period of 5 years.
- Increased reliability: the system has not gone into failure for at least 6 months. As the data is transferred via MQTT, the data is transmitted even in bad communication conditions. Besides, due to local storage, the data is recovered after a connection failure. The Fig. 8 shows the data read during an issue with the network provider. It was stored during the failure and sent after the connection was reestablished. Now the data can be queried from Kapua database as if no connection issue had ever happened.
- Security: the previous solution was not properly secured against cyber attacks. Both Kura and Kapua platforms requires authentication and handle certificates for an encrypted communication. The

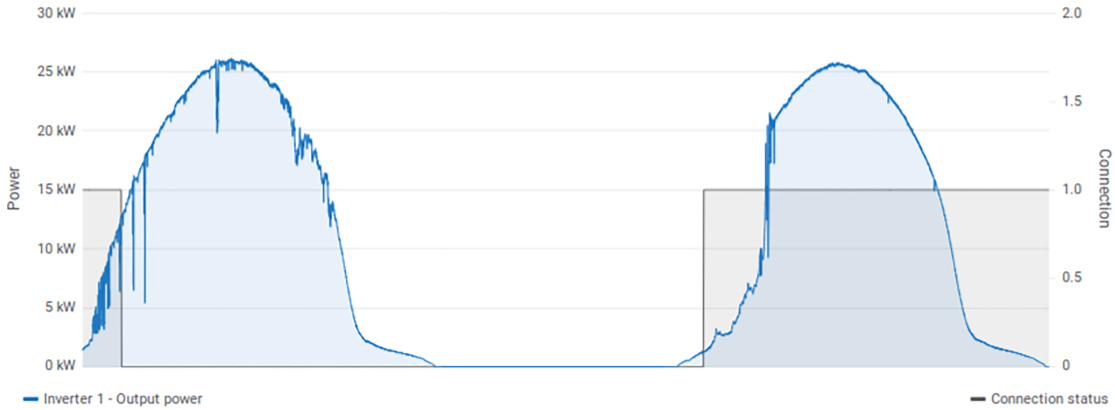


Fig. 8. Example of coverage failure. The loss of connection is not reflected in the integrity of the data once the connection is reestablished.

Fig. 9. Monitoring dashboard of an inverter on Grafana. Data is extracted from Kapua database to Grafana graphics to improve the quality of the monitoring.

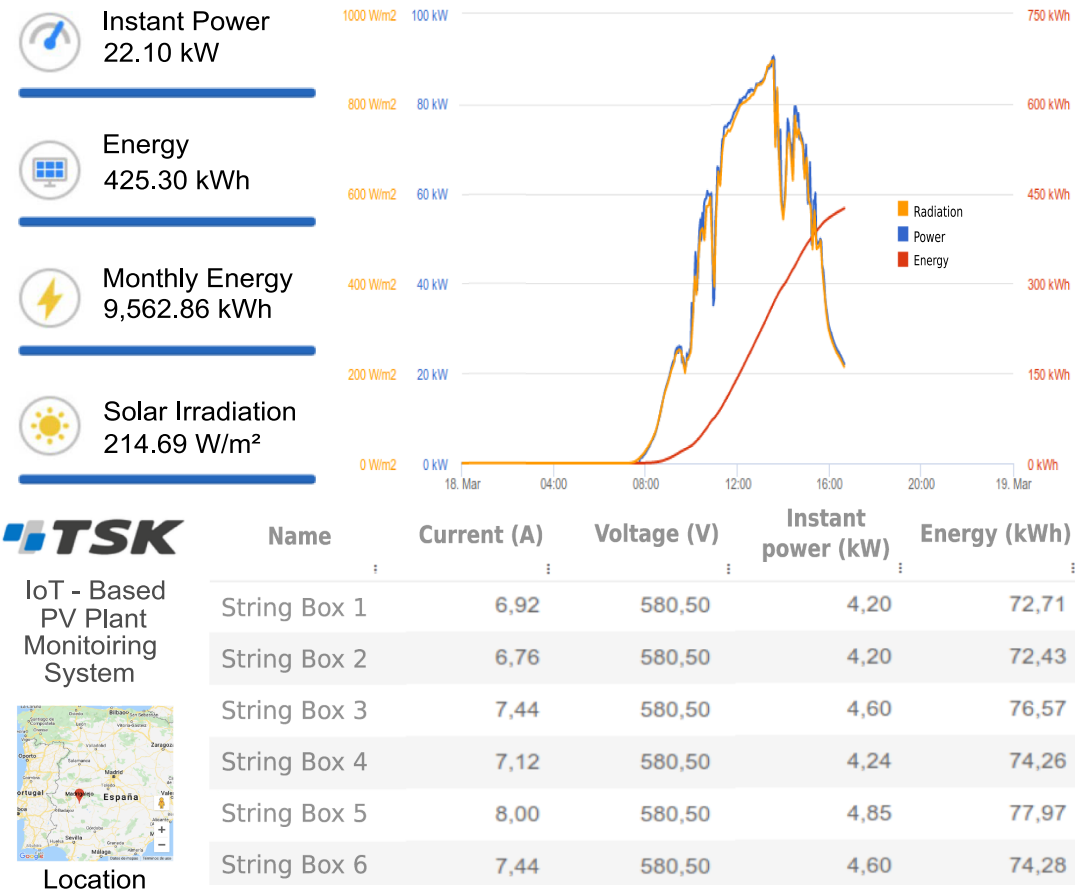


Fig. 10. Custom monitoring dashboard of an inverter based on the integration with Kapua.

configuration of Kura (with critical information about the plant infrastructure) and the certificates stored in the device are encrypted in case the gateway is somehow accessed. The traffic encryption with the industrial devices is available in the protocols where it is possible, although this was not validated as the use case only required Modbus and S7. There is also a feature to manage the gateway firewall in order to prevent security breaches.

- Decreased time investment: the development time of the solution was relatively low as the software was already developed and it was only needed to install and configure it. In Fig. 7 is shown the required and estimated time for the setup of the solution. Note that the initial time is very low. However, the increase of time by industrial device is almost linear as it must be configured one by one. It is also shown an estimation of the time required to deploy a whole new installation, the initial reduction of the time is significant as the knowledge is already gained and some configurations can be loaded from a template.
- Integration: unlike previous solution, Kura/Kapua based solution allows multiple integration ways. The main connections available are:
  - ElasticSearch: direct connection with Kapua database, which can easily be done with Grafana. The Fig. 9 is an example of such Grafana integration, but it was also used, for instance, to create Fig. 8.
  - MQTT broker: connection to Kapua MQTT broker to show Real-Time data and analysis. There are multiple connectors for MQTT available in IoT market to integrate with other tools.
  - Kapua API: the Kapua API can be accessed to retrieve both Real-Time or historical data. The Fig. 10 is an example of a front-end connected with it.
  - Kura API: only for Real-Time data, the Kura API can be queried to check instant values on the industrial devices
  - Camel routes: both in Kapua and Kura, the data can be send by a Camel route to any supported endpoint.
  - Other: due to the plugin architecture of Kura, it can be developed additional connectors based on specific needs.
- Power consumption: the monitoring devices used in the previous solution required about 30 W while current monitoring solution has a maximum consumption of 4 W.

Furthermore, some of the typical issues relating to IoT deployments are also mostly covered such as security. Some additional strengths (or weaknesses) are detailed in Table 2.

Despite of the results assessed, some considerations must be taken also into account. Not all of them are related with the Kura software but they will be typically found in all the deployments of this monitoring system:

Table 2
Eclipse Kura contributions identified.

Feature	Purpose	Matching Kura features	Contributions assessed
Interoperability	Integrate heterogeneous data and solutions	Multiple plugins available	Assessed: Modbus, OPC, S7, etc.
		Wide purpose functionality	Assessed: Ethernet, WiFi, serial, Bluetooth
		Multiple data format integration	Assessed: Numeric, text, array, raw binary
	Enable vertical integration (third party solutions)	Routing capabilities	Assessed: integration with Apache Camel [36] and REST interface
Management	Monitor applications and devices	Device monitoring through MQTT subscription	Assessed: only requires an MQTT client
		Detailed monitoring with Kapua platform	Not assessed
	Ease update of hardware and software	Software provisioning via MQTT (Kapua)	Assessed
	Ease the user experience	Management available via web interface	Assessed: monitored and managed with Kapua
Security	Avoid the most typical cyber attacks	User authentication	Assessed
		Encrypted configuration	Assessed: but encryption method is reversible
		Network management	Assessed: includes firewall capabilities
	Avoid interception of data	Native management of certificates	Assessed: simple certificate management
	Verify devices connected	Identification via certificate	Not assessed
Processing	Avoid process interruptions	Self-monitoring of critical processes	Assessed: by default for Kapua connection
	Provide Real-Time data	Parallel workflows	Assessed: available multiple rates
	Verify data integrity	Operations on workflows	Not assessed
	Optimize industrial process	Batch and real-time processing options	Not assessed
Scalability	Ease horizontal expansion	Integration in architecture via self-register	Assessed: no configuration needed in platform
		Easy integration of new industrial devices	Assessed: intuitive but tedious configuration
		Data aggregated through Kapua	Assessed: data structure in platform is independent of its physical origin
		Dynamic workflow design	Assessed: multipurpose and reusable modules
Cost	Minimize overall cost	Free Open Source software	Assessed: Moxa UC-2112, Lanner LEC 7230-M, Artila 700
		Low hardware requisites	Assessed: available on low-consumption or inexpensive devices

Table 3
Cost structure of previous and current solutions.

Concept	Previous	Kura
Infrastructure	€15215.28	€13165.10
Development	€6005.00	€0.00
Licences (annual)	€6000.00	€0.00
Total (5 years)	€51220.28	€13165.10

- Lack of documentation: there was little documentation about the communication with the industrial device. It was necessary to test by trial-error different configurations to establish some connections.
- Large-scale configuration: Kura is configured via a web interface which makes it easy but a bit repetitive and slow. Several hours were required to create the 24 inverters and 156 strings and program the reading. Large plants would require proportional time. There are tools that could help like the possibility to export/import configurations or Kapua Jobs. Both of them requires anyway a specific configuration file.
- Reading period: the estimated performance of the gateway and Kura software tested allowed up to hundreds of variables per second and a theoretical period of around 500 ms in this use case. It should have satisfied the period of 1 s required by the system. However, this period was increased to 5 s mostly due to the delays on the network and sequential reading of some devices that are implemented over a FieldBus (the strings).

6. Conclusions

The IoT solution proposed has taken advantage of most of the benefits of an IoT solution and even cover some of the common disadvantages. Regarding the requirements set from the industrial scope, the IoT solution has been capable to improve the previous solution and reached most of the targets exposed in 2.2 except for the monitoring interval:

- Interoperability: the solution has proven good adaptation to several different industrial protocols and communication systems. The integration can also be easily expanded thanks to the Kapua and Kura camel routes and the multiple ways to integrate them with third party systems as shown.
- Management: IoT devices are monitored by a centralized platform that allows also maintenance tasks such as reconfiguration and upgrade.
- Security: the communications are now encrypted. The access to Kapua and Kura is protected by authentication and managed with users and roles for authorization. The external access to the plant is not required directly by the monitoring system because it is the local software which pushes the data instead of allowing remote access to the plant.
- Processing: the 5 min interval for historical data has been easily reached. However, the challenge of 1 s for Real-Time has not been properly fulfilled but only a 5 s period. It is noted that the delay lays mostly on the communication infrastructure of the plant
- Scalability: the data is centralized in a single platform and the aggregation of new devices is done dynamically (the devices self-register in the platform). The system can be reconfigured to adapt to new scenarios as the environment is modified.
- Cost: the solution diminished mostly the maintenance costs due to the Open Source components used, but also the overall consumption of the devices is decreased both because the devices are reduced and their individual consumption is lower. Considering the harsh environment, the device used to host Kura software is prepared for basic industrial environment resistance. Due to the little requirements of the system, if it were faced a worse environment, it could be used a more fitted device.

The IoT solution developed has increased the most direct indicators of improvement such as monitoring period and cost of the solution, but it also has increased the scalability, security and management capacity of the monitoring system. The complexity is also diminished as intermediary devices have become unnecessary.

There are, however, two additional challenges that has been identified from the problems and possibilities on the current solution and should be taken into account for further work. On one hand, the configuration has required a non-scalable amount of work and it should be simplified and automatized to allow better deployment times. On the other hand, the functionality of the Kura framework can be extended with plugins to enable its integration with other platforms and more industrial devices. There is also interest in create more plugins to activate some processing capabilities or new workflows.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Pedro de Arquer Fernández: Methodology, Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Miguel Ángel Fernández Fernández:** Methodology, Software, Resources. **Juan Luis Carús Candás:** Conceptualization, Validation, Resources, Writing - review & editing, Supervision, Project administration. **Pablo Arboleya Arboleya:** Conceptualization, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ijepes.2020.106540.

References

[1] Madeti SR, Singh S. Monitoring system for photovoltaic plants: a review. Renew Sustain Energy Rev 2017;67:1180–207.
[2] Rezai A, Keshavarzi P, Moravej Z. Key Management issue in SCADA networks: a review. Eng Sci Technol Int J 2017;20(1):354–63.
[3] Upadhyay D, Sampalli S. SCADA (supervisory control and data acquisition) systems: vulnerability assessment and security recommendations. Comput Secur 2020;89:101666.
[4] Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. J Clean Prod 2020;252:119869.
[5] Sisinni E, Saifullah A, Han S, Jennehag U, Gidlund M. Industrial Internet of things: challenges, opportunities, and directions. IEEE Trans Industr Inf 2018;14(11):4724–34.
[6] Chae BK. The evolution of the Internet of Things (IoT): a computational text analysis. Telecommun Policy 2019;43(10).
[7] Gerber DL, Liou R, Brown R. Energy-saving opportunities of direct-DC loads in buildings. Appl Energy 2019;248:274–87.
[8] Wang Y, Xu Y, Tang Y. Distributed aggregation control of grid-interactive smart buildings for power system frequency support. Appl Energy 2019;251:113371.
[9] Kang J-N, Wei Y-M, Liu L-C, Han R, Yu B-Y, Wang J-W. Energy systems for climate change mitigation: a systematic review. Appl Energy 2020;263:114602.
[10] Singh G. Solar power generation by PV (photovoltaic) technology: a review. Energy 2013;53:1–13.
[11] Solar Power Europe. Global Market Outlook: 2019–2023. Global Market Outlook; 2019. p. 92.
[12] Manur A, Venkataramanan G, Sehloff D. Simple electric utility platform: a hardware/software solution for operating emergent microgrids. Appl Energy 2018;210:748–63.
[13] O'Dwyer E, Pan I, Acha S, Shah N. Smart energy systems for sustainable smart cities: current developments, trends and future directions. Appl Energy 2019;237:581–97.
[14] Png E, Srinivasan S, Bekiroglu K, Chaoyang J, Su R, Poolla K. An Internet of Things upgrade for smart and scalable heating, ventilation and air-conditioning control in commercial buildings. Appl Energy 2019;239:408–24.
[15] Sayed K, Gabbar H. SCADA and smart energy grid control automation. In: Gabbar HA, editor. Smart energy grid engineering. Academic Press; 2017. p. 481–514 [chapter 18].
[16] Spring MB. The future of standardization: are we destined to repeat history? Computer 2016;49(01):99–101.
[17] Martín-Lopo MM, Boal J, Sánchez-Miralles Á. A literature review of IoT energy platforms aimed at end users. Comput Netw 2020;171:107101.
[18] Taivalsaari A, Mikkonen T. A taxonomy of IoT client architectures. IEEE Softw 2018;35(3):83–8.
[19] Razzaque MA, Milojevic-Jevric M, Palade A, Clarke S. Middleware for Internet of Things: a survey. IEEE Internet Things J 2016;3(1):70–95.
[20] Ray PP. A survey of IoT cloud platforms. Future Comput Informat J 2016;1(1–2):35–46.
[21] Ammar M, Russello G, Crispo B. Internet of Things: a survey on the security of IoT frameworks. J Inform Secur Appl 2018;38:8–27.
[22] Geng H. SCADA fundamentals and applications in the IoT. In: Internet of Things and Data Analytics Handbook. Wiley; 2017.
[23] Sajid A, Abbas H, Saleem K. Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. IEEE Access 2016;4:1375–84.
[24] Alonso RS, Sittón-Candanedo I, García Ó, Prieto J, Rodríguez-González S. An intelligent edge-IoT platform for monitoring livestock and crops in a dairy farming scenario. Ad Hoc Netw 2020;98.
[25] Patel P, Ali MI, Sheth A. From raw data to smart manufacturing: AI and semantic web of things for industry 4.0. IEEE Intell Syst 2018;33(4):79–86.
[26] Miao D, Liu L, Xu R, Panneerselvam J, Wu Y, Xu W. An Efficient Indexing Model for the Fog Layer of Industrial Internet of Things. IEEE Trans Industr Inf 2018;14(10):4487–96.
[27] Zhang F, Liu M, Zhou Z, Shen W. An IoT-based online monitoring system for continuous steel casting. IEEE Internet Things J 2016;3(6):1355–63.
[28] Usamentiaga R, Fernandez MA, Villan AF, Carus JL. Temperature monitoring for electrical substations using infrared thermography: architecture for industrial Internet of Things. IEEE Trans Industr Inf 2018;14(12):5667–77.
[29] Pereira RI, Jucá SC, Carvalho PC. IoT embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants. Meas J Int Meas Confeder 142 (2019) 195–212.
[30] Abu Waraga O, Bettayeb M, Nasir Q, Abu Talib M. Design and implementation of automated IoT security testbed. Comput Secur 2020;88.
[31] Chen YQ, Zhou B, Zhang M, Chen CM. Using IoT technology for computer-integrated manufacturing systems in the semiconductor industry. Appl Soft Comput J 2020;89.
[32] Khan WZ, Rehman MH, Zangoti HM, Afzal MK, Armi N, Salah K. Industrial Internet of Things: recent advances, enabling technologies and open challenges. Comput Electr Eng 2020;81:106522.
[33] Brous P, Janssen M, Herder P. The dual effects of the Internet of Things (IoT): a systematic review of the benefits and risks of IoT adoption by organizations. Int J Inf Manage 2019;51:101952.
[34] Fafoutis X, Elsts A, Piechocki R, Craddock I. Experiences and lessons learned from making IoT sensing platforms for large-scale deployments. IEEE Access 2018;6:3140–8.
[35] Da Cruz MA, Rodrigues JJP, Al-Muhtadi J, Korotaev VV, De Albuquerque VHC. A reference model for internet of things middleware. IEEE Internet Things J 2018;5(2):871–83.
[36] "Home - Apache Camel." [Online]. Available: https://camel.apache.org/ .

Apéndice B

Determining Operational Constrains for IoT-Based Advance Metering Infrastructure

Determining Operational Constraints for IoT-Based Advanced Metering Infrastructure

Pedro de Arquer Fernández
Dept. of R&D
TSK Electrónica y Electricidad
Gijón, SPAIN
pedro.arquer@grupotsk.com

Pablo Arboleya Arboleya
LEMUR Research Group
University of Oviedo
Gijón, SPAIN
arboleyapablo@uniovi.es

Juan Luis Carús Candás
Dept. of R&D
TSK Electrónica y Electricidad
Gijón, SPAIN
juanluis.carus@grupotsk.com

2021 IEEE Madrid PowerTech | 978-1-6654-3597-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/PowerTech46648.2021.9494783

***Abstract*—In the present paper, different devices and configurations for the implementation and deployment of an advanced metering infrastructure based on Internet of Things in electrical distribution systems are proposed. They are evaluated as an alternative for the conventional PLC-based smart-metering infrastructure. The IoT solutions are described from the point of view of their architecture, software and operational procedure for obtaining the data. After running a battery of tests, the proposed alternatives are compared in terms of performance. The conclusions that can be extracted from this study will allow the researchers to determine whether or not a given configuration is applicable for a specific application since conventional smart metering infrastructure does not provide the necessary bandwidth to be used in some cases like for instance real time flexibility management in low-voltage networks.**

***Index Terms*—Advanced metering infrastructure, IoT, behind the meter, flexibility management.**

I. Introduction

The process of decarbonisation and electrification is an unstoppable reality underpinned not only by an increase in social environmental awareness, but also by a reduction in the costs of core technologies that enable this process to be implemented on a large scale. Moreover, in the current context, regulation at European level (i.e. the EU Directive 2019/944) [1] is providing a strong stimulus for the Member States to promote laws in this direction.

In addition to the strong development of core technologies such as those related to renewable distributed generation (DG), residential photovoltaic (PV) generation, energy storage systems (ESS), electric vehicles (EV) or heat pumps (HP) [2], there are also a series of enabling technologies that allow for the optimal management of these resources embedded in electrical distribution networks. These technologies are related to Big Data techniques [3], Artificial Intelligence [4], IoT [5], advanced measurement infrastructure [6], new communication protocols [7], etc.

In general, all researchers agree that the above-mentioned decarbonising technologies will provoke a change in paradigm with numerous advantages in the mid and long term [8]. Their management within distribution networks is proving to be a complex challenge for distributors and operators due to the following factors [9]; (i) the increase in demand due to new charges on the network; (ii) the need for Real-Time or quasi-Real-Time management of many of the devices connected to the distribution network, not only at the smart meter level, but also at the so-called behind-the-meter level.

Certainly, the electrical load on the terminal distribution networks will increase greatly. However, this type of load will be flexible in most cases (e.g., PV generation systems with ESS, EV charging systems, space heating systems based on HPs, etc.). Real-Time management of this flexibility will be a critical task in order to accommodate the millions of new generation and consumption devices on the network [10], [11].

On the other hand, this Real-Time management implies the deployment and use of advanced measurement infrastructure that allows data to be obtained with sufficiently low latency [12]. In most cases, the intelligent metering infrastructure deployed by the electricity companies, even the most advanced ones, do not meet the necessary specifications (see, for example, the metering infrastructure that uses PLC technology installed in most European utilities) [13], [14].

IoT technology applied to obtaining electrical variables in Real-Time is destined to be one of the game changers in the sector as it allows a large number of measurements to be obtained at a very low cost and with relatively low latencies [15], [16].

This article studies a possible solution and configuration based on IoT technology. It is validated in different alternatives of hardware for its application in advanced measurement infrastructure systems in electrical distribution networks. The conclusions obtained will be very useful for researchers who intend to carry out studies on techniques in order to estimate the state of the network in Real-Time, systems for managing flexibility in Real-Time or network operation and maintenance systems.

To perform this study, the solution to be tested is described in Section II matching the most common structure in these kind of applications. After that, in Section III is settled the conditions for the test performed and the devices to be evaluated. Then, Section IV presents the results of the testing and a discussion to understand the data obtained in order to take the most of the devices evaluated depending on the

This work was partially funded with the Industrial PhD Program form the Ministry of Science and Innovation, Gonvernment of Spain under the grant DIN2019-010585 / AEI / 10.13039/501100011033

application. Finally Section V presents the conclusions of the whole study performed.

II. Solution proposed

The solution proposed is based on the common use of Internet of Things (IoT) technologies, their architecture and functionality expected. Its scope of application is valid for large deployments as much as for minimal solutions.

A. Architecture

The architecture and layers of functionality in an IoT solution can vary greatly depending on the application. Nevertheless, most of the IoT solutions would have the following components [17] [18]:

- Perception layer: a layer capable of measuring and digitise physical magnitudes or perform a physical action. In industrial environments this layer is typically composed of specific and accurate devices with little or none IoT capabilities, but in domestic applications they usually provide such technology
- Gateway layer: it collects the data of the sensing layer in order to process and/or send it. This layer would implement IoT protocols and edge processing technologies.
- Management layer: in this level the devices of the previous layers are managed to maintain their software, monitor the status, aggregate the information, etc. It is typically supported by an IoT Platform or integrated with the operation layer presented below.
- Operation layer: whether it is a web interface, a mobile application or a virtual reality environment, the data must be represented to the user to monitor it, allow manual operations or simply provide the service. The interface is not necessarily visual but also by voice recognition, for example.

The layers could be overlapped if a device can cover multiple functionality, such as a device that integrates sensing capabilities but also direct integration with the management layer. Some features could be omitted like the processing of data or the management of software on devices. In the present solution, this architecture is respected and all the functionality is provided.

B. Software

The software running in the gateway layer (middleware) is expected to provide most of the functionality for an IoT solution allowing the connection with local devices, processing capabilities and integration with multiple services both on-premise or cloud. When a middleware is selected it is highly recommended that it has independence from the hardware used in order to diversify devices, functionalities and providers. It must be also considered that it will be used in multiple targets so the cost must be reduced as much as possible. An Open Source Software solution could be of great advantage considering the last issue.

As described in [18], an IoT solution can be created based entirely on Open Source software. The system is used to monitor photovoltaic plants but the software can be recycled into other environments such as self-generation and home automation applications. The middleware used in this solution is Eclipse Kura.

Kura is a gateway-oriented framework with the capability to communicate via industrial protocols and support background services for advanced processing. The system is developed in modules which allow fine grained installations and software management. Besides, the more generic the modules are, the more easily they can be reused. For example, a timer module, or a database storage module can be reused for different purposes independently of the scope. This capability would allow the change of application to provide the features required in advanced metering. Some useful plugins are already provided in the Eclipse Marketplace [19].

Kura would represent only the gateway layer of the architecture. A cloud environment is also required. Cloud allow data for long-term persistence, cloud processing and particularly Big Data analysis. In order to ensure the integration with IoT solutions, the cloud is expected to be available with a light protocol of communication. The most common protocols in IoT are also by publication/subscription commonly to ensure the stability during disconnections.

C. Operational Procedure

The solutions process the data as expected for most of the applications in metering. The data is collected from the devices, processed and further sent to the cloud with reliability. To provide this functionality, the stages to be executed are:

- Reading data from an industrial device
- Mapping the data to a different format
- Storing the information in a local database
- Sending the data to a cloud service

During the reading task the gateway interacts with local devices to collect data from them via an industrial protocol. It could be standard (Modbus, OPC, for instance) or proprietary. Kura already provide some drivers in its own marketplace. Once data is integrated in the framework, it can be processed in the next stages.

The mapping task could include simple operations to reduce the volume of data, or transform it to a more compact format for the further processing. The sending task would take advantage of the simplification in order to enlighten the communications. In the present solutions such mapping is done to create independent variables with an unique identifier but it also scales their value to provide it in a standard unit.

The storage provide temporal persistence of the data prior to the delivery. It is done in the file system with a database. The standard database in Kura framework is H2. The data stored is deleted after the delivery to avoid overload of the internal memory of the gateway. Without storage, the data could be lost if, during the delivery, there were a connection failure. Keep the data locally stored guarantees that it reaches at least once. In this stage, last values of each variable are also stored

for a local usage. It could be used to provide integration with local devices that require the data.

Finally, the delivery to the cloud is required for a long-term persistence. The cloud also provide advanced processing, remote access to the data and additional functionality. The direct connection to internet services from gateways is one of the most promising features of IoT technology. However, it is very dependent on the Internet Service Provider (ISP) and the environmental conditions, more than the solution or gateway itself.

The payload sent to the cloud service was composed by an identifier of the variable, a timestamp when it was acquired and the value itself. Such data was codified with Google Protocol Buffers format to ensure its lightness. Despite this reduction, the communication was done under a certificate authentication and the data was encrypted for a secure transference. This security overlay means additional data load but it is required in any application to avoid most common cyberattacks.

III. Testing conditions

A. Evaluated devices

The solution presented in the previous section requires a gateway device to run Kura. The requirements for Kura are very low and multiple devices are ready to execute it. It has been stated by the Kura project developers that the empirical minimum requirements are a Linux Operating System (OS), 700 MHz of CPU and 256 MB of RAM [20]. Under these specifications, the software may run but the performance is low enough to make it unusable.

Based on these requirements six devices were selected, three of them have industrial capabilities and the other three are different versions on Raspberry Pi (RPi) boards. The industrial devices provide reliability under stressing conditions, while the RPis are cheaper. The whole collection of devices would allow any scale of solution: from single device to be monitored to hundreds of elements and variables; from domestic environment to industrial solutions. The devices selected were: Lanner LEC 7230-M [21], Moxa UC-8112A-LX [22], Artila Matrix 700 [23], RPi 4 [24], RPi 2B [25] and RPi Zero [26]. The specifications of each device are shown in Table I.

Artila, Moxa and LEC devices are specifically designed for industrial applications. Artila is under the stated requirements in CPU and will serve as reference. Moxa is expected to be good enough for most industrial applications with a more adjusted cost. LEC on the contrary have oversized specifications for simple solutions but should be capable of running heavy processing or load of data. Besides, LEC is a configurable PC whose characteristics could be enhanced by the replacement of RAM, storage or even processor. The characteristics selected were based on the average needs for an industrial PC with middle processing capabilities.

Both 2B and Zero RPis have similar characteristics but the RPi Zero has a more compact format and wireless features. RPi 2B has slightly better specifications and is better suited for wired communications. RPi 4 posses high processing capabilities and the specifications are very promising. All RPis share the same supported OS (Rasbian), a video interface and a set of digital inputs and outputs. Such digital IO can be used to expanse the functionality via sensors, communication or the use of extension boards (hardware to be connected through their interface pins). RPi 2B could be also representative of most of the Single Board Computers (SBC) for its average characteristics, even though those systems are acquiring more quality and processing capabilities each year.

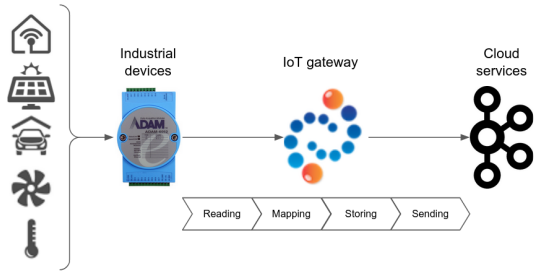


Fig. 1. Schema of the test executed. Eclipse Kura will be tested as the software in the IoT Gateway.

B. Simulated environment

The devices performed the four tasks presented in II-C: reading, mapping, storing and sending. Mapping and storing has already been sufficiently described in II-C and depends uniquely in the devices itself. Reading and sending on the other side are more dependent on the environmental conditions. The whole schema is shown in Fig. 1. Some significant testing conditions were created so that they could be representative of real performance. For a proper simulation of the reading, an industrial device was used. Specifically Adam 6052 was selected due to:

- The response of the device was fast enough to not introduce its own delay in the measurement.
- The communication is done via Modbus TCP, which is one of the most extended protocol in industry
- The device serves 110 variables, 40 of them are coils (single bit variables) and the rest are holding registers (2 bytes variables). This amount of variables could represent the monitoring of a small plant.

The driver required to make the reading was developed by Eurotech and is available on the Eclipse Marketplace.

As stated before, the connection is more depending on the ISP than the gateway or cloud itself most of the times. In order to measure the performance of the device both with this handicap and without it, two scenarios were created: a wired 1 Gbps Ethernet connection in the same local network of the cloud service, and a 4G connection using modem Airlink Sierra LX40 for remote access. The modem was used with optimum signal strength (-45 RSSI). In remote installations, the RSSI could fall and the rate of acquisition would be lower.

TABLE I
SPECIFICATIONS OF THE DEVICES SELECTED

Feature	RPi Zero	RPi 2B	RPi 4	Artila	Moxa	LEC
Temperature	0 to 50 ºC	0 to 50 ºC	0 to 50 ºC	0 to 70 ºC	-40 to 85 ºC	-20 to 70 ºC
Consumption	2 W	4 W	8 W	3 W	4 W	19 W
Power supply	5 VDC	5 VDC	5 VDC	9-48 VDC	9-48 VDC	12 VDC
Dimensions (mm)	30x65x5	56.5x85.6x11	56.5x85.6x11	77x111x25.5	141x125.6x33	144.8x198x42
Weight	9 g	46 g	46 g	290 g	550 g	1 kg
OS	Raspbian	Raspbian	Raspbian	Yocto	Debian	Linux, Win
Debug access	Serial	Serial	Serial	Serial	Serial	BIOS
Cost	€10.00	€30.00	€70.00	€300.00	€400.00	€550.00
CPU	1 GHz	0.9 GHz	1 GHz	536 MHz	1 GHz	High
RAM	0.5 GB	1 GB	4 GB	0.5 GB	1 GB	4 GB
Storage	SD card	SD card	SD card	Internal flash	Internal flash	HDD
External	USB	USB	USB	SD card	SD card	USB
WiFi	Yes	No	Yes	No	No	No
Bluetooth	Yes	No	Yes	No	No	No
3G / 4G	No	No	No	No	No	No
Ethernet	1	1	1	2	2	2
USB ports	2 (micro)	4	4	2	0	3
Video	Micro HDMI	HDMI	Micro HDMI	-	-	VGA/HDMI
Serial	3x 232	3x 232	3x 232	4x 232/485	2x 232/485	2x 232/485

The performance of each stage was measured using a component created to monitor the time in milliseconds (ms) of each step. This component registers the beginning and ending of each stage to measure the specific duration of such stage. Time under 1 ms is not relevant in most of the Linux distributions as they are not Real-Time OS. Consider values under 1 ms could lead to inaccurate conclusions [27]. For a better accuracy on the results and discard as much as possible the interference of the OS, the data will be measured over a cycle of 10,000 readings.

IV. PERFORMANCE

A. Results

After executing the test on the six devices in both of the sending conditions settled, times could be extracted for the four stages. Such times are shown in the Table II. The rate column represents the approximate rate of variables processed per second in the overall cycle. There is also a comparative graph of each step in Fig. 2 where the differences among the devices in each of the stages can be appreciated. Artila has been excluded from this graph as its performance was hardly comparable with the others and it deformed the output of the figure.

The first appreciation to be done is the slight difference among the measurements in local network and 4G network in the first three steps. This is due to the not exclusive dedication of the OS to Kura, which provokes occasional delays on the processing. Thanks to the measurement of 10,000 readings, the differences are very low near 1 ms in most cases or at a small proportion compared with the measurement in the case of Artila.

The results of the tests show the limitations of the devices and some of their strengths. But mainly, the delivery of the data to the cloud service deserves attention. Sending data is the slowest step in most of the devices. It is noted that the delivery through 4G connection takes 80 to 100 ms more than local, so the increase seems similar independently of the device. As a general consideration, this delay should

TABLE II
PERFORMANCE OF THE DEVICES BY STAGE. RATE REPRESENTS THE NUMBER OF VARIABLES READ PER SECOND.

	Local network			4G network		
Step	**Time (ms)**	**%**	**Rate**	**Time (ms)**	**%**	**Rate**
	Artila Matrix 700					
Reading	338	6.55		322	6.06	
Mapping	528	10.23		517	9.75	
Storing	2403	46.61		2181	41.13	
Sending	1887	36.60		2283	43.06	
Total	**5156**		**21**	**5303**		**21**
	Moxa UC-8112A-LX					
Reading	9	5.89		9	4.00	
Mapping	20	12.65		19	8.07	
Storing	45	28.71		45	18.85	
Sending	82	52.75		164	69.08	
Total	**156**		**704**	**237**		**465**
	LEC 7230-M					
Reading	4	4.34		4	2.05	
Mapping	2	2.84		3	1.38	
Storing	14	16.51		14	7.69	
Sending	67	76.31		165	88.87	
Total	**87**		**1260**	**186**		**591**
	Raspberry Pi Zero					
Reading	23	7.73		22	5.61	
Mapping	42	14.44		44	11.40	
Storing	110	37.52		114	29.79	
Sending	118	40.32		204	53.21	
Total	**294**		**375**	**384**		**286**
	Raspberry Pi 2B					
Reading	7	3.58		7	2.48	
Mapping	13	7.34		14	4.94	
Storing	88	48.25		92	31.58	
Sending	75	40.84		178	61.01	
Total	**183**		**600**	**292**		**377**
	Raspberry Pi 4					
Reading	3	2.50		3	1.30	
Mapping	5	4.80		5	2.53	
Storing	27	26.97		31	14.80	
Sending	66	65.73		169	81.37	
Total	**100**		**1095**	**207**		**530**

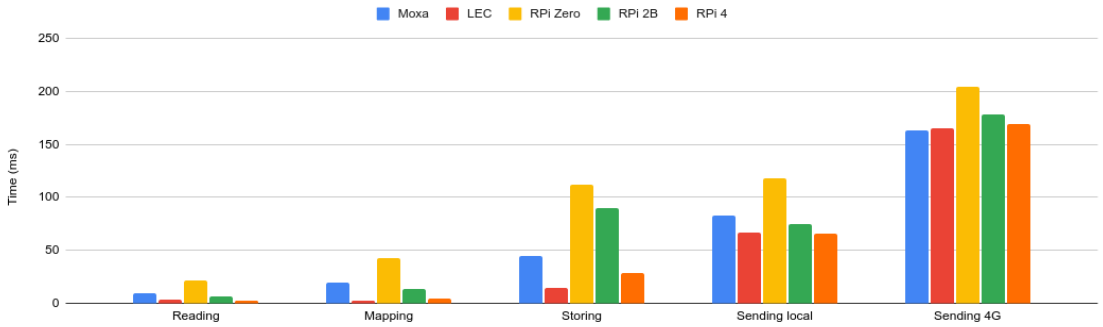


Fig. 2. Comparative bar graph of the stages.

be expected in any industrial-cloud application, as the Real-Time could be compromised. Besides, internet delay could be variable depending on the conditions and load of the network. To analyse the behaviour of the performance depending on the network conditions, a deeper study should be performed testing different strategies to overcome the low rate of variables per second it provokes.

Analyzing the results by device, the Artila Matrix 700 has the lowest stats of performance. Its characteristics were already under the minimum stated in CPU and the performance supports this requirement. It requires more time just reading than any other in the whole cycle. The most time consuming stage is the storage, even with the delivery through 4G this stage remains relevant. The delay could be explained by the low quality flash storage that the device use as internal memory. A possible solution could be to provide an external memory in case that is the actual reason. The slowness in the delivery to cloud is also explained as a consequence of the processing capabilities as the difference between local and 4G performance is insignificant and it should be higher.

Moxa UC-8112A-LX has an average performance compared with the rest of the devices. The whole time required is acceptable in terms of Real-Time local response as it only takes 30 ms to read and process the data. The delivery to the cloud is the most consuming task in both local and 4G connections so the network delays seems to be the most limiting issue.

The Lanner LEC 7230-M and RPi 4 have the highest speed in all tests and steps with some difference, except for the similar 4G connection on Moxa. Their processing time is much faster than the rest and the only differentiate in a slight better mapping performance in LEC and a large discrepancy in the storing stage in benefit of LEC. Probably this is due to the usage of a Hard Disk Drive (HDD). A Solid-State Drive (SSD) would be much faster, but the HDD is good enough to overcome not only RPi 4 but all the devices tested. The rest of the devices have storage based in flash memories. The network connection in both LEC and RPi 4 are by far the most requiring stage taking from 65 to 90 % of the time depending on the case.

Regarding the other two RPis, 2B has a performance similar to Moxa but the storage is a clear burden to the process as it takes most of the time in the local connection and still a significant part in 4G delivery. Zero, on the other side, has and overall lower performance. Storage is still one of the main problems, but even the reading takes more time than any other (excluding Artila). The network slowness could be explained by the fact that the Ethernet connector is in an external board. It would affect also the reading.

B. Discussion

Based on the results of the test, the devices could be pointed to different scopes of application based on their performance and characteristics.

Due to the endurance of the devices, none of the RPis could be of use in industrial environment, but only Artila, Moxa and LEC. The times measured have pointed to the LEC as the appropriate gateway for an application with high performance requirements. For instance, it could be the need for multiple services or large volume of data. The resources in the device can manage a much greater load than that of the tests. Moxa would be most suitable device where the environment conditions are more demanding but the processing or volume of data is not. It also offers a cheaper and more compact format than the LEC. Both devices are equally capable of communicating via serial and Ethernet interfaces, which are the most common in industry. Unfortunately, Artila has proved to be inadequate for any use of Kura framework.

The RPis have good enough performance compared with the rest of the devices. They could be very valid still for other purposes such as domestic or office scope. Their lower cost and promising features make them perfect for massive usage of the devices. Their application would depend on the most important issue on the solution. Firstly, RPi 4 should be applied on large solutions were the number of variables or processing capabilities require good performance. RPi 2B would be the basic solution when the solution is smaller but still demanding. Zero should be considered only when the cost

is a heavy limit or its wireless communication capabilities are forcefully required. RPi 2B would be of no use in these cases despite its better performance.

However, if the devices tested are not well suited enough for the application, some requirements are extracted out of the tests to help with the selection of a new device. This candidate to IoT gateway should fulfil the following minimum and recommended (rec.) requirements: 1 GHz CPU (1.8 GHz rec.), 0.5 GB RAM (2 GB rec.), flash memory (HDD rec.), Linux OS (standard distribution rec.). The connections (Ethernet, serial, WiFi, Bluetooth) could be chosen based on the specific application.

V. Conclusions

In this study, it was presented an IoT solution based on Open Source software Eclipse Kura for advanced metering. The solution was tested in six different devices to measure the performance on each of them. All six performed the same tasks of a common IoT solution. The resulting data was analysed to define the best application of each of them, covering multiple scenarios depending on the case. The selection allowed to identify devices for industry, home automation, large scale deployments, minimal cost solutions, etc.

A general conclusion extracted is the need for a deeper analysis on the sending stage to lower the impact on the whole process of the data. There are multiple protocols to enlighten the communications and also package options for the message itself. The aggregation of more variables is also a possible way to process faster the data as the creation of the connection requires some additional time in each delivery. However, none of this possible solutions could be considered until a proper test is done to validate such strategies.

Finally, the devices selected have proved that the solution proposed can be applied independently of the hardware used. It means that it is prepared to adapt to future devices and that the device could be upgraded to a more interesting gateway without losing functionality.

References

[1] Council of European Union, "Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU," 2019, http://data.europa.eu/eli/dir/2019/944/oj.

[2] N. G. Paterakis, O. Erdinç, I. N. Pappi, A. G. Bakirtzis, and J. P. S. Catalão, "Coordinated operation of a neighborhood of smart households comprising electric vehicles, energy storage and distributed generation," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2736–2747, 2016.

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2019.

[4] A. C. Şerban and M. D. Lytras, "Artificial intelligence for smart renewable energy sector in europe—smart energy infrastructures for next generation smart cities," *IEEE Access*, vol. 8, pp. 77 364–77 377, 2020.

[5] R. Morello, C. De Capua, G. Fulco, and S. C. Mukhopadhyay, "A smart power meter to monitor energy flow in smart grids: The role of advanced sensing and iot in the electric grid of the future," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7828–7837, 2017.

[6] A. Ghosal and M. Conti, "Key management systems for smart grid advanced metering infrastructure: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2831–2848, 2019.

[7] R. Prior, D. E. Lucani, Y. Phulpin, M. Nistor, and J. Barros, "Network coding protocols for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1523–1531, 2014.

[8] Council of European Union, "Clean Energy for all Europeans Package," 2019, https://ec.europa.eu/energy/topics/energy-strategy/clean-energy-all-europeans_en.

[9] S. Jupe, S. Hoda, A. Park, M. Wright, and S. Hodgson, "Active management of generation in low-voltage networks," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 916–919, 2017.

[10] O. Alrumayh and K. Bhattacharya, "Flexibility of residential loads for demand response provisions in smart grid," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6284–6297, 2019.

[11] K. Oikonomou, M. Parvania, and R. Khatami, "Deliverable energy flexibility scheduling for active distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 655–664, 2020.

[12] A. Gomez-Exposito, L. Mili, and W. Wu, "Guest editorial: State estimation for future cyber-physical power and energy systems: Challenges and solutions," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 4, pp. 1–2, 2020.

[13] G. López, J. Matanza, D. De La Vega, M. Castro, A. Arrinda, J. I. Moreno, and A. Sendin, "The role of power line communications in the smart grid revisited: Applications, challenges, and research initiatives," *IEEE Access*, vol. 7, pp. 117 346–117 368, 2019.

[14] P. Arboleya, "State estimation in low voltage networks using smart meters: Statistical analysis of the errors," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, 2018, pp. 1–5.

[15] L. Barbierato, A. Estebsari, E. Pons, M. Pau, F. Salassa, M. Ghirardi, and E. Patti, "A distributed iot infrastructure to test and deploy real-time demand response in smart grids," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1136–1146, 2019.

[16] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1934–1944, 2017.

[17] E. Bances, U. Schneider, J. Siegert, and T. Bauernhansl, "Exoskeletons Towards Industrie 4.0: Benefits and Challenges of the IoT Communication Architecture," *Procedia Manufacturing*, vol. 42, pp. 49–56, 2020.

[18] P. de Arquer Fernández, M. Ángel Fernández Fernández, J. L. Carús Candás, and P. Arboleya Arboleya, "An iot open source platform for photovoltaic plants supervision," *International Journal of Electrical Power and Energy Systems*, vol. 125, p. 106540, 2021.

[19] "Home - Eclipse MarketPlace," accessed: 2020-11-10. [Online]. Available: https://marketplace.eclipse.org/

[20] "Eclipse kura support forum," accessed: 2021-03-24. [Online]. Available: https://www.eclipse.org/forums/index.php/t/1077162/

[21] "Lec 7230m datasheet," accessed: 2021-03-25. [Online]. Available: http://www.lannerinc.com/support/download-center/datasheets/category/35-iot-appliances?download=175:lec-7230-datasheet

[22] "Moxa uc8112a specifications," accessed: 2021-03-25. [Online]. Available: https://www.moxa.com/en/products/industrial-computing/arm-based-computers/uc-8100a-me-t-series#specifications

[23] "Artila 700 datasheet," accessed: 2021-03-25. [Online]. Available: https://www.artila.com/docs/Matrix-700/Matrix-700_Datasheet.pdf

[24] "Raspberry pi 4 specifications," accessed: 2021-03-25. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

[25] "Raspberry pi 2b specifications," accessed: 2021-03-25. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

[26] "Raspberry pi zero specifications," accessed: 2021-03-25. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-zero/

[27] J. Arm, Z. Bradac, and V. Kaczmarczyk, "Real-time capabilities of Linux RTAI," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 401–406, 2016.

Apéndice C

Internet of Things (IoT) for Power System Applications

Chapter 1
Internet of Things (IoT) for Power System Applications

Pedro de Arquer Fernández
TSK Electrónica y Electricidad, Technological Park of Gijón, 33211, Spain, e-mail: pedro.arquer@grupotsk.com

Juan Luis Carús Candás
TSK Electrónica y Electricidad, Technological Park of Gijón, 33211, Spain, e-mail: juanluis.carus@grupotsk.com

Pablo Arboleya Arboleya
University of Oviedo, , Campus de Gijón, 33204, Spain, e-mail: arboleya.pablo@uniovi.es

Abstract: Internet of Things is nowadays one of the leading technology in digital transformation for industry. The Application on Power Systems has enabled the concepts of Smartgrids, Distributed Demand Response or Advanced Metering Infrastructure. The term not only refers to a simple technology but a more abstract conception of the architecture of the solution, with an extensive technology stack supporting it. This chapter presents the architectural concepts, a brief collection over the technology stack and a description on how those are arranged in some examples of Power Systems Applications.

Keywords: Demand Response, Edge computing, Internet of Things, IoT architecture, IoT Platform, LPWAN, Mesh network, Power Systems, Smart

1 Definition

The term Internet of Things (IoT) has evolved over the years since its first appearance in 1999. The core idea of IoT was the connection of physical things to the internet to collect its data, extend its functionality and enable its interaction with the rest of the things in the network. This definition is still valid but the technological advances on Information and Communications Technologies (ICT) allows today much more scope of application than in the years this concept was conceived. Smartphones, wireless technologies, Augmented Realily, Virtual Reality or Unmanned Aerial Vehicle (UAV) are only a small part of nowadays technological landscape and IoT concept make the most of all of them to achieve the goal of connecting every object to internet and increase their value.

In the first years, even before the term was officially referenced, IoT was technologically supported by the already present technologies, such as Ethernet wiring, HTTP protocol or mechanical interfaces. In time, the IoT gained relevance by itself and started to define the requisites and roadmap for some technologies. In order to be adaptable to every thing, an IoT device must be low-cost, portable, versatile and small-sized. The technology has evolved to provide wireless connectivity (WiFi, Bluetooth, ZigBee, 6LowPAN, NB-IoT, etc.) lightweight communication protocols (MQTT, CoAP, REST, etc.), affordable yet good processing capabilities (cheap solutions like Raspberry Pi, Arduino or ESP32) and self-discovery features (PnP, WoT).

Soon with the appearance of IoT devices, it was also required some extra abstraction layer and reusable tools. Some platforms were developed to manage the wide range of IoT devices that could not be coordinated with the tools already present. The concept of IoT platform as a service in a data center was created to solve the management of an heterogeneous and large set of IoT devices. The descent of the processing capabilities cost and its minituarization led afterwards to the use of flexible IoT frameworks allocated in local devices. This advance also provided the capability to run applications with high processing requisites in local devices, or edge

computing.

As the technology became more promising, the concept of IoT moved from simple proofs of concepts and isolated solutions to different fields of application. The earliest were the environmental and educational solutions, but soon home automation and healthcare applications were widely created. With the refining of the initial challenges, industry has started to accept the IoT solutions in the recent years, becoming one of the most important element in the digital transformation of industries. However, industrial applications require a whole new approach on some of the principles already stated in IoT. The main reason is that reliability and security of a solution are much more critical features than the solution itself. Such redefinition has even led to the specific concept of Industrial IoT (IIoT).

In the present chapter, the architecture of IoT solutions will be analysed (Section 2) and there will be presented a brief catalogue (Section 3) of some of the most typical technologies found in IoT solutions. Then, some Power Systems applications solved via IoT will be described in Section 4.

2 Architecture

Despite of the years of evolution, research, and deploy of applications there is no standard architecture for an IoT solution up to now. As the solutions are increasing every year, there is not a stable reference to state a standard. The technologies are applied to very different scopes in new ways every year. The standards are continuously updated, the functionality is extended and the architecture is restructured according to the new features and applications.

Despite this inconvenience, some efforts have been made to define a reference architecture and the layers to be expected in all solutions [1, 2, 3, 4]. There are approaches based on the functionality, the technology or the physical devices among others. The elements described are similar but the structure differ among them. Considering a detailed analysis, there is a diffuse layer separation that can be perceived in most of these works. Even though it is not universal, the layers proposed are:

- IoT layer: this is the first layer of the architecture. In this layer the data is converted to digital format by sensors and integrated in the system.

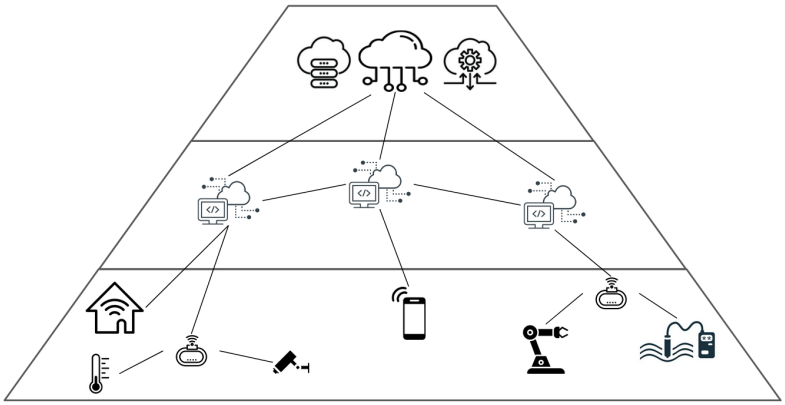


Fig. 1 Typical layers of an IoT architecture: IoT layer at the bottom with heterogeneous devices, Cloud layer at the top providing advanced functionality and persistent storage and Edge layer in between offering integration tools and intermediate services.

There are an immense quantity of devices in this layer depending on the manufacturer and the application. In this layer the data must be normalized to a more standard format so that the following layer can digest them. This layer also allows the actuation over the field and the interaction with physical elements. The challenges in this layer are the integration of a wide variety of devices and the communication with low range wireless technologies. The classical concept of IoT device is usually referred to those found in this layer: cheap, low consumption, wireless, quotidian and small.

- Edge layer: data from the IoT layer must be delivered to high-level systems through the Edge layer. This layer also processes the data to make the most of the information and provide local functionality to the architecture. It allows a quick response to the events and fault tolerance of the network as the processing is decentralized and it does not depend on remote centralized servers. The challenges in this layer are the wide area wireless

communication technologies and the algorithms and processing capabilities. It could also hold third-party application integration. The kind of equipment in this layer answers better to the concept of IoT gateway, which may be a small PC, a router or a PLC (Programmable Logic Controller).

- Application/Cloud layer: for a full use of the data, remote monitoring and long time persistence, data must arrive centralized servers in a remote location (cloud). There, data is processed with high performance algorithms and an improved interface can be achieved. This layer also holds most of the integration with third party applications. Finally it must be capable to manage the whole system, which means monitor, provision and manage the IoT devices and gateways and the data. The challenges in this layer are storage of data, management of devices, advanced processing and third party integration. Heavy PCs and clusters are typically found in this final layer.

As the IoT technology and devices are in constant evolution, the boundaries among the presented layers may vary from one application to another. For instance, an Edge functionality can be found in IoT layer if the devices in that layer have enough resources to provide it. However, this architecture is clear enough to help with the comprehension of the concepts described in this text and will be used as a reference for the technologies stack and the explanation of the use cases.

3 Technology catalogue

There are a vast list of IoT related technology, standards and devices. Part of this stack of technologies is not even specific for IoT applications but an old system reused or redefined in IoT applications. In this chapter there are only a brief selection of them standing out the most important and their features.

The technologies described comprise wireless communication standards, lightweight communication protocols and IoT specific software. Note that both wireless standards and communication protocols are required for the communication but they differ in the level where it is used. The first group are more related with the physical layer of transmission while the second

group cover the message sent over such layer. Some wireless standards includes also the messages description but it is not common.

3.1 Wireless communications

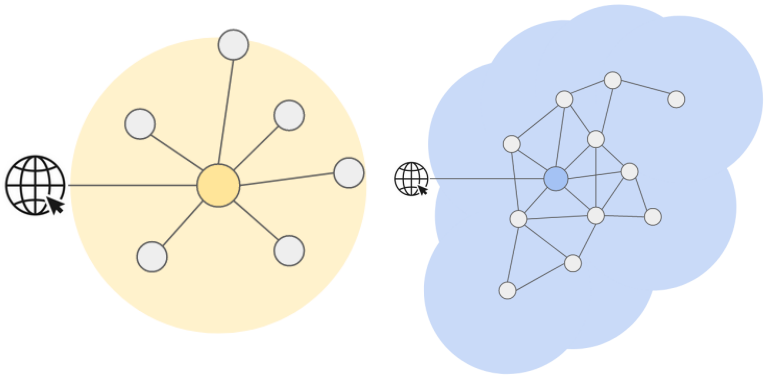
The appearance and expansion of IoT has come hand in hand with the emerging of wireless communication technologies. They can be divided by their range in local or wide area. Local area communication covers up to some hundred of meters range. It is used to reduce the cost and power consumption of each unit. This kind of technology is found in communications inside IoT layer or between IoT and Edge layers. Wide area communication would be from one km to world wide range. It is mostly used to get internet connection or cover vast areas. The consumption may also be low but in this case it compromises the bandwidth. It covers communication from Edge layer to Application layer.

In the Tables 1 and 2 are summarized the technical characteristics of all the technologies cited in order to simplify the next discussion without all the figures relating range, number of devices, data transmission, etc. The data was obtained from [5, 6, 7, 8, 9]

3.1.1 Local area communication

Two of the most common and well known technologies for local area communication are Bluetooth (and Bluetooth Low Energy) and WiFi. Most IoT solutions are based in any of these technologies as they are very extended. However, some architectures are limited by the range and number of devices connected at the same time, particularly in Bluetooth. To tackle with these issues, there are efforts to redefine them as mesh networks. A mesh network is a network where any element can communicate directly with any other element in the network, in contrast with traditional star network whose communication is always through a central element as shown in Figure 2a. In a mesh network, a message hops from one node to another until it finally finds the destination node. This structure makes the network more reliable in case a node fails and extends the range of the network to the range of the most external nodes as in Figure 2b. Both of them need a gateway for

internet access but in mesh networks, the gateway can be substituted if it fails. It also increases potentially the number of devices allowed. The star network is limited by the capacity of the central node, but in mesh networks it is limited mostly by the capacity of the technology to address them.



(a) Star network and its range.

(b) Mesh network and its range.

Fig. 2 Comparison between star (2a) and mesh (2b) networks. Star network depends only on central node for communication while mesh network allow inter node communication. Internet connection is provided through a gateway which, in mesh network, can be substituted in case of failure.

In addition, there are other IoT technologies which provide this feature by design. ZigBee is an example of a mesh network with high scalability as a single network can hold up to 65535 devices with a range of 10-20 meters among each of them. The main advantage of ZigBee over previous WiFi and Bluetooth is the low energy consumption while keeping a good enough range and rate of data transmission. There are other similar alternatives depending on the application, desired transmission rate, range, energy consumption, etc. They are all based on the same wireless technology but they differ in their approach. The most extended are:

- Z-Wave: while ZigBee can be freely implemented, Z-Wave is a proprietary standard so a device must be certified before prior to be used in a Z-

Wave network. The advantage is that, once it is certified, there is full guarantee of the compatibility with other Z-Wave devices (latest ZigBee 3.0 specification save this gap of compatibility). The one-to-one range of Z-Wave it higher than ZigBee (70-100 meters) but it only supports 232 devices in the network.

- 6LowPAN: in order to take advantage of current standards and protocols of communication over internet (HTTP, MQTT, FTP, SSH, etc.), 6LowPAN has been based in IPv6 protocol. IPv6 is the extended protocol of IPv4, which is the most extended protocol communication over internet. By using this protocol, 6LowPAN can be directly integrated with other internet connections or even WiFi, instead of requiring a gateway to make the conversion among protocols as in ZigBee or Z-Wave. In simplest terms, this standard would allow a device to navigate through internet almost directly. Compared with WiFi, the range is much lower (30 meters), although it provides mesh capabilities and the consumption is drastically lower.

- Thread: on top of 6LowPAN standard, Thread provide some security and reliability features and simplifies the interaction with non-6LowPAN networks. One of the most interesting features is the re-election of an edge node (node used to interact with external networks). This avoids the failure of the whole network if the edge node fails. For an easier development on Thread, Google has created an open source implementation called OpenThread, which simplifies even more the access to the different features of the network in any hardware platform. The use of Thread is nowadays more extended than the raw communication through 6LowPAN as the learning curve is clearly lower and the features offered are very aligned with the typical usage of the technology.

In a different scale, there are other local area communication technologies which can be used for very low power consumption if the application only require simple data transmission. Radio Frequency Identification (RFID) is a technology where a device (designated as active) can emit a message that forces a remote device response even if the remote device (designated as passive) is not connected to any power source. This kind of communication

Table 1 Local area wireless technologies summary. Some of the values are obtained from theoretical performance, not actual technological measures. The data is only valid for a basic understanding of the technologies, more than a deep analysis of their limitations.

Technology	Data rate	Range	Max nodes	Topology
WiFi	450 Mbps	10 - 100 m	250	Star
WiFi Mesh	450 Mbps	10 - 100 m	-	Mesh
Bluetooth	2 Mbps	15 - 20 m	8	Star
BLEMesh	1 Mbps	15 - 20 m	65535	Mesh
ZigBee	250 kbps	10 - 100 m	65535	Mesh
Z-Wave	100 kbps	30 - 50 m	232	Mesh
6LowPAN	250 kbps	10 - 100 m	250	Star / Mesh
Thread	100 kbps	10 - 100 m	250	Star / Mesh

has low range (less than 2 meters) but the power consumption saving in the passive device makes it very practical for a wide kind of applications. There is also a similar alternative called Near Field Communication (NFC). The main difference between them is the range which makes RFID useful to detect approximate presence of an element (useful for inventory and retail) while NFC requires a deliberate approximation (very useful for personal cards). Other short-range radio communications were used for long time but mostly under proprietary protocols.

3.1.2 Wide area communication

In wide area communication the most commonly used technologies in the world are 3G and 4G. They are both based on cellular communication and they are well known to most of the people due to their use in smartphones to dial and grant internet access. However, the need for low consumption applications led to the creation of different protocols with more optimized features.

The main traditional wide area communication technology is 3G. 3G, launched for the first time in 1998, enhanced the internet capabilities of 2G.

Table 2 Wide area wireless technologies summary. Some of the values are obtained from theoretical performance, not actual technological measures. The data is only valid for a basic understanding of the technologies, more than a deep analysis of their limitations.

Technology	Data rate	Range	Consumption	Coverage
2G	40 kbps	-	Medium	Worldwide
3G	15 Mbps	-	High	Worldwide
4G	300 Mbps	15 km	High	Worldwide
5G	30 Gbps	0.5 km	Adaptable	Europe, Asia, USA
SigFox	600 bps	10 - 40 km	Very low	Central-West Europe
NB-IoT	200 kbps	1 - 10 km	Low	Worldwide except Africa
LTE-M	1 Mbps	1 - 10 km	Low	Europe, America, Oceania
LoRaWAN	50 kbps	5 - 20 km	Very low	Private

It provoked a greater use of internet and powered the daily use of internet in smartphones. Nowadays, 3G has been mostly overcame by 4G communications which basically offers the same features with higher characteristics. At this moment there are plans in multiple countries to shutdown 2G or 3G. Note that, despite being less advanced, some applications may prefer 2G due to the low consumption, so there are countries which has more devices connected to 2G than to 3G.

However, each of these technologies have some drawbacks for their use in IoT. The most significant are the power consumption and the traffic symmetry. The power consumption is clearly referred to the need for the communication module to allow a long duration of a battery. To achieve this, it is necessary to make an optimized transmission: modify the frequency of the communication, reduce the overhead of the payload sent, etc.

Regarding the traffic symmetry, the domestic or mobile use of 3G/4G has been initially asymmetric towards downloading. This means that it has better performance for downloading data than for uploading, which is very reasonable as most users consume the information instead of producing it. In fact, 3G and 4G are theoretically symmetric but the infrastructure is undersized for uploading transference, expecting less traffic. In IoT applications, the data transmission is the opposite as the devices mostly produce data and

they rarely consume it.

A final characteristic to be considered in the protocols presented in this subsection is the coverage. Most of them are very recent and their future use is very uncertain so most of the companies involved in these services are cautious in the deployment of the infrastructure required to provide the communication. Prior to select a technology it must be checked if it will be available in the area where it will be used.

One of the first and most significant protocols is SigFox. SigFox is a proprietary network with a very limited data transmissions rate but with the lowest power consumption specifications in wide range. The user must register the connection prior to use it and must use a certified device. The data transmission is only up to some bytes by day for uploading and much less for downloading. The SigFox network is already available in most of Central and West Europe. There are areas in other countries but mostly for specific test-beds and proofs of concepts. The need for the infrastructure may represent an issue particularly as SigFox coverage is only provided by SigFox. Unlike the rest of the technologies cited here, no other company can provide the service. The lack of coverage may represent a scalability issue.

Despite SigFox consumption performance, the need for higher data transmission rates led to the development of two similar technologies based on current 3G/4G infrastructure. The technologies are LTE-M (Long Term Evolution for Machines) and NB-IoT (Narrowband IoT). The two of them are optimized for low energy applications and lots of devices.

NB-IoT, on the one side, allows more devices with the same infrastructure. It is also theoretically capable of better indoor and inland communication performance. The cost by terminal is estimated to be lower than with LTE-M. The infrastructure is already available in all Europe and most of Asia, America and Oceania.

On the other side, LTE-M has better data transmission rate. It can be used for voice communication, which makes it suitable for telephone communications. This protocols is optimized to fit mobility applications with better performance than NB-IoT.The available coverage is lower than in NB-IoT as it is not deployed in Asia and Eastern Europe.

The upcoming arrival of 5G is intended to include the features of both previous protocols. The technology supports under the same protocol all

the previous features with optimization via software. The technology is expected to support intense data transmissions from vast quantity of devices. However, there are at the moment only proofs of concepts where 5G is really put to the limit of its capabilities. The infrastructure is deployed in USA, Europe, and some countries in Asia and Oceania but even where it is deployed there are great areas not yet covered.

Completely different to the last protocols, LoRaWAN (or just LoRa) is a mesh communication technology. Similarly to some local area communications presented, the nodes can retransmit messages. This is the only technology cited in this subsection that does not require a service provider. The network is private and anyone can deploy it using own infrastructure. Some providers also offer the infrastructure but most solutions use private networks. It means that coverage is not a factor as in the previous solutions, which may represent an advantandge for some use cases.

The initial conception for the protocol was covering wide areas thanks to the virtual range of the mesh network. However, in the recent years, the technological features of LoRa are making possible the application for different scenarios. The most appreciated are good range, low cost, private infrastructure and mesh capabilities. There are solutions that leverage those characteristics for indoor communication or for assets/personnel location. It is expected to be found additional use cases in the next years.

3.2 IoT protocols

Due to the low consumption and wireless communication requisites, IoT systems have required the creation of new communication protocols. These protocols are intended to be functional in a context of non-deterministic connections and low data rate, which are typical given the communication technologies previously presented. The protocols must rely on verification in the endpoints to guarantee the reception of data and be ready for the temporary disconnection of a remote device. The most significant protocol in the communication of IoT systems is MQTT.

Message Queuing Telemetry Transport (MQTT) is a communication protocol that allows the delivery and reception of data via publication / subscription pattern. On one side, the data producer publishes the information

to a given topic. A topic is a channel where data is available at real time for everyone that is connected to the topic as a subscriber. The MQTT broker (the server that manages the messages published) typically provides functionality to guarantee that the delivery is executed from any publisher to every subscriber. Due to the publication/subscription feature, the devices can save power by reducing the communications while there is no data exchange. It also allows the concentration of data from multiple sources in the same topic or the distribution of the information of a topic to multiple subscribers. This simplifies the escalation of the architectures making it easier to include an additional device, that simply must connect to the broker instead of connect to all the devices one by one.

There are other protocols with similar features developed for the same purpose. AMQP (Advanced Message Queuing Protocol) is based on the same concept but with some pros and cons. For instance, the implementation is a bit heavier but it adds some extra security measurements. In general terms both protocols are mostly alike.

CoAP (Constrained Application Protocol) is also comparable to MQTT. The main difference is that CoAP is decentralized as it does not require a broker. Any device can act as requester or requested depending on the configuration and capabilities. The three of them (MQTT, AMQP and CoAP) are equally intended for a scope were connection may be unstable and with low rate.

However, the IoT environment has additional needs such as the discovery of features. In order to distribute the processing and to extend the functionality to any place were a device can be connected, the IoT systems must have a procedure to describe their own functionality to other devices and discover the functionality of nearby ones. This discovery will allow the use of distributed systems instead of access to a centralized remote cloud.

Web of Things (WoT) is a protocol developed to match this need. Each device must have a description of himself called Thing Description and the description must be available by a standard communication protocol such as HTTP or CoAP. The description provides the list of properties (data accessible by the device), actions (executions available through the device) and events (alerts to be subscribed in case they are triggered).

The description allows anyone to take advantage of the features in the

device without additional configuration. It also provides horizontal scalability as any new device can provide his functionality without specific configuration and use nearby devices to improve its own features.

A protocol with a similar approach is LWM2M, which provides the functionality of an element locally so that others can directly communicate with it, enabling the distributed processing.

3.3 IoT Software

There are countless software solutions developed for IoT, however, some of them are particularly remarkable due to their versatility for multiple use cases and the extended use of them. IoT software is found in the three layers: firmware for embedded devices (IoT layer), middleware (Edge layer) or management platforms (Application layer).

Prior to describing the different softwares, some of them will be presented as open source projects. An open source project is an increasing trend in software development and frameworks where the source code of the tool is published and available for anyone to inspect and propose modifications. The open source projects are driven by voluntary developers or companies interested in the enhancement of the project. Given the availability of the source, it is intended to be used freely (but referencing the origin). The project gets benefit from the users experience, suggestions and work while it provides any user with a free tool and some support. Due to the free access, an open source project has a clear competitive advantage in front of proprietary developments. However, such projects are not covered by any guarantee neither of the functionality nor of the compromise for future development. A good example of a successful open source project is Android, which is sponsored by Google.

3.3.1 Embedded software

Embedded devices are hardware solutions with limited resources that require very specific firmware solutions and hardly reusable. This is the case of Microcontroller Units (MCU). However, the current trend is towards abstraction in this area. There are some firmware frameworks that provides an standard

layer that can be applied to multiple models or even manufacturers. An example of this firmware is FreeRTOS. FreeRTOS (Free Real-Time Operating System) is an open source project for an Operating System (OS) driven by Amazon that can be executed in most MCUs with Real-Time response guaranteed. The support of FreeRTOS on multiple MCUs provides an standard layer to access the functionality despite of the manufacturer or the model. Some of the features supported are the possibility of pseudo-parallel processing, the management of events or the internal memory management.

FreeRTOS is supported by most of the manufacturers, as its simplicity makes the integration easy. However, the features are limited and the user must still develop most of the application. In case the user needs specific features of the MCU, FreeRTOS starts losing the portability expected. A more advanced project, yet not as much extended, is Zephyr. Zephyr provides similar basic tools as FreeRTOS but also incorporates support for communication features, which makes more types of application fully portable. It includes, for instance, bluetooth connectivity, ciphering tools, sound tools or Modbus protocol.

There are other alternatives to the cited OS. One of them is Contiki. Contiki, for instance, provides better support for some wireless applications but is less extended than FreeRTOS.

3.3.2 Middleware

The devices on the Edge layer, and some in IoT layer, have enough resources to run high performance software. A typical middleware for such devices is NodeRED. NodeRED is an open source project developed originally by IBM which consists in a web tool configured as a block diagram. The blocks available can be expanded by the installation of new packages. The functionality is easily modified and configured while specific features can be developed and applied through the modular feature. Its simplicity makes NodeRED powerful for quick developments, proof of concepts and small pieces of logic. Particularly the logic related to the routing of the data and some minor processing/modification. The software has native support in multiple devices as a way of integration with third party applications.

A similar tool is Eclipse Kura. Kura is also an open source project

evolved from the release of the proprietary tool Eurotech Software Framework. Similarly to NodeRED, Kura provide a configurable workflow based also in block diagrams. However, Kura is also specifically adapted for industrial purposes. The software provides specific tools to integrate industrial drivers, cloud connection, background processes, processing capabilities and security management. Additionally, the framework can be fully monitored and controlled remotely with commands sent over MQTT. This feature is particularly integrated with a platform presented in 3.3.3 (Eclipse Kapua). The software is much more powerful than NodeRED but the learning curve is also higher and requires more processing capabilities on the device.

There are similar tools both in proprietary and open source side. Some examples are Cisco Kinetic, Azure IoT or Flogo. The fundamental tools are similar: integration with cloud and local devices and configurable workflow. The security and remote configuration are the most distinctive features in all of them as the proprietary solutions are trying to differentiate their solution.

The middleware presented must be supported by an OS. In IoT gateways, the most typical Operating System is Linux based. Some of them are Raspbian, Debian, Ubuntu (particularly Ubuntu Core) or Android. There is also a modular option called Yocto which is also used in those devices that lacks features or must adjust the performance. Out of Linux environment the most relevant system is Windows 10 IoT.

3.3.3 IoT platforms

With the increasing appearance of devices, protocols and middlewares, the IoT solutions are increasingly needing a management platform. The platform should monitor the devices, provision the software, control its actions and configure the workflow. The traditional tools for such operations have been overcame by the variety and features of the new IoT technologies. The IoT platforms appeared as a natural solution for this issue.

The main idea of an IoT platform is to provide a central hub for all the IoT connections and devices. Despite of the cited functionality, the IoT platforms typically provide tools for storing the data, execute advanced processing or applications and visualize the information collected from lower layers. However, the most critical feature of an IoT platform is the integration

capabilities. The integration must be provided both for devices in a lower level of the architecture and other systems in the Application/Cloud layer. An additional feature available in some platforms is the capability to run isolated from the cloud (on-premise) to provide local services in case the internet connection gets deteriorated.

Table 3 IoT platforms summary. The features are evaluated qualitatively according to literature and reviews. Some deeper analysis can be found in studies as [10, 11].

Platform	Management	Integration	Analysis	Visualization	Agnostic	Open Source
Microsoft Azure	Advanced	Medium	Advanced	Advanced	No	No
Amazon Web Services	Medium	Advanced	Advanced	Medium	Yes	No
Oracle IoT	Basic	Medium	Medium	Medium	-	No
Eclipse Kapua	Advanced	Basic	Basic	Basic	Yes	Yes
Carriots	Advanced	Medium	Advanced	Advanced	No	No
Evrythng	Advanced	Basic	Advanced	Medium	Yes	No
ThingWorx	Medium	Basic	Medium	Advanced	No	No
ThingSpeak	Basic	Basic	Advanced	Basic	No	No
Kaa	Basic	Advanced	Advanced	Basic	Yes	Yes

There are a wide number of IoT Platforms even considering only the most relevant: Microsoft Azure, Amazon Web Services, Oracle IoT, Eclipse Kapua, Carriots, Kaa, Thingworx, ThingSpeak, etc. These platforms differs in the approach for lots of possible functionalities. In general terms, the features are:

- Device management: such as device provisioning (procedure to register a device in the platform), remote management, protocol for loss of connection, software updates, status monitoring, etc.
- Integration: routing of incoming data, access endpoints (MQTT broker, REST API, etc.) or events notifications.
- Analysis: like data mapping (for example format conversion), advanced processing (Artificial Intelligence, Machine Learning), alerts generation or the capacity to run additional applications.
- Visualization: historical data, charts and dashboards.

In the Table 3 some platforms are compared according to the cited parameters, but only qualitatively. The criteria to select one of them for a given solution is mostly based on the needs for specific features, the acceptable cost and the integration with current technologies if needed.

There are two additional comparison features to be taken into account in Table 3. On one side data agnosticism is the characteristic that allows an IoT platform to ignore the meaning of the incoming data. In fact, most of the platforms can handle unidentified data. However, if data has not been clearly identified, some features may not be available. On the other side, as stated in Subsecion 3.3, Open Source projects have some advantages to be taken into account in a selection.

Due to the vast presence of platforms some of them has even restructured themselves to be used as a platform of platforms. Such feature could be used to integrate several independent solutions (verticals) in the same tool in order to consolidate the whole system. This feature is particularly useful in complex industries where multiple specific solutions must coexist in the same environment.

4 Power System Applications

There are multiple examples with the use of IoT in power system applications. Depending on the solution, IoT technology is used only as a specific component of the system or all the solution is structured as an IoT architecture. Four of these examples have been selected for a detailed description. The architecture will be distributed according to section 2 for a better understanding.

4.1 Demand Response and Transactive Energy Management

The electricity market is becoming increasingly distributed with the appearance of small producers and the self-consumption solutions. In this landscape, [12] propose an IoT architecture capable to provide real time data to consumers and producers in order to balance the demand and provide efficient customer costs.

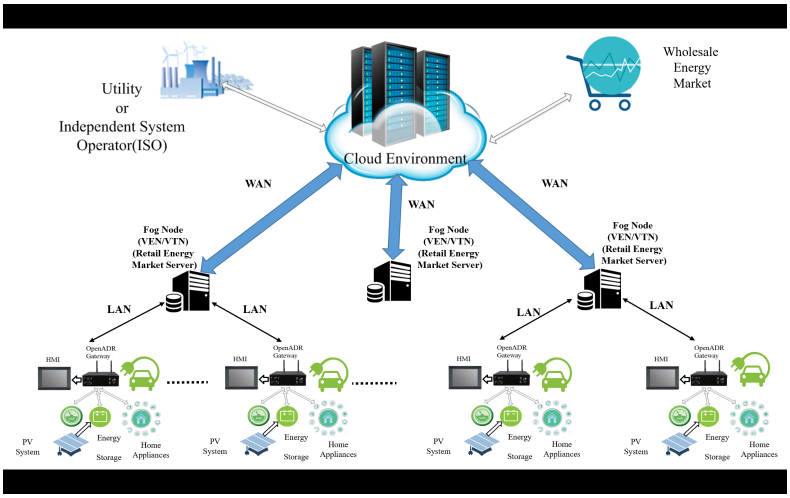


Fig. 3 Architecture of the application for Transactive Energy Management. The image has been obtained from [12].

The Figure 3 shows the architecture according to the authors. The structure matches precisely the three layers presented in 2, with the following functionality:

1. IoT layer: in this layer, a gateway is placed to collect the energy consumption information of each home. Such gateway must be capable to integrate with multiple consumer devices and grid analyzers and process the data to a standard format so that the next layer can process it without complexity. The data is sent to the network.

2. Edge layer: some nodes are located near the gateways in order to provide fast response to local devices of the IoT layer and interact with the external environment. The nodes collect the data of the IoT layer and pre-process it before sending a reduced, aggregated and more significant data to the cloud. The initial data is temporarily available in the node for local IoT

devices and gateways. The node is also responsible to act as interface among producers and consumers when local balance can be achieved.

3. Application/Cloud layer: remote servers are used for permanent and reliable data storage. In this layer is also allocated batch processing of the data to improve the model running in edge layer. It also provides the connection to the wholesale energy market and independent operators whose data is not directly available locally for the nodes.

The communication between the IoT and the Edge layers is based on REST protocol both by HTTP or CoAP. The first offers high compatibility and the second offers a lightweight communication channel in those devices that allow or require it. A specific protocol, OpenADR (Open Automated Demand Response), is used on top of the communication to provide a stable compatible interface despite the communication channel.

The solution proposed reduces drastically the delay in response to the variations of electrical consumption. The edge layer is much faster than the cloud connection, providing a good timing while keeping some local general view of the consumption, which is critical for demand response applications. The edge layer also reduces the consumption of processing capabilities in the cloud, which allows better performance of the overall solution while it avoids the congestion of the devices involved.

As shown in [13] by the same authors, the system can also be applied to optimize a Distributed Demand Response system. The use case is only considered theoretically over a very similar architecture. The use of intermediary gateways provide a faster response while the application layer provide stability in the long term.

Multiple use cases like the ones presented can be found in the literature as IoT is offering multiple applications in smartgrids area. Some of them are [14], [15] or [16].

4.2 Battery Management

The power generation is intermittent in some renewable plants, for instance those based on wind or solar energy. This unpredictable factor require the

presence of batteries to provide the energy when production falls. The management of such batteries is an increasing challenge in power systems. The authors of [17] have validated a monitoring solution based on IoT.

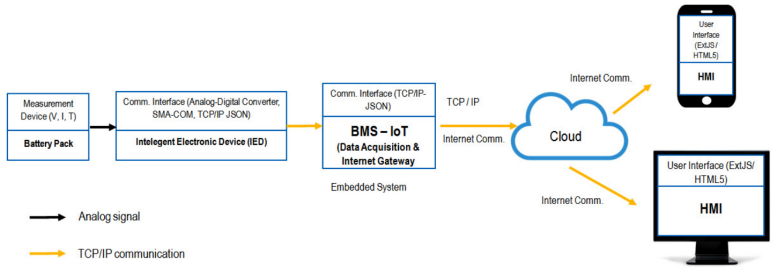


Fig. 4 Architecture of the application for Battery Management. The image has been obtained from [17].

As shown in Figure 4, this application is simpler than the previous one as the processing is performed in the IoT device, providing it with extra capabilities, instead of requiring a whole edge layer. The structure is:

- IoT layer: an IoT device is used to collect data from commercial BMS (Battery Management System) and process that data. The information collected is processed and delivered to a remote platform. It also holds the interaction with the options in the BMS in order to control the consumption and charging rules.

- Application/Cloud layer: the application in this case provides an remote management interface. The user can access to the tools for actuation or monitoring via web or a smartphone app.

Battery management and smart charging applications can be also found in mobility applications related with electric car such as [18] or [19].

4.3 Substation Maintenance

As the substation are critical infrastructures, the correct maintenance and prediction of failure of its systems is a very valuable application. The authors of [20] have created a solution to monitor the electrical busbars of substations with thermographic images. The system detects abnormal temperatures in the busbars and alerts the operators of the issue, recommending a repair in the system.

The architecture layers of the solution are slightly different to the layers presented in this chapter. However, attending to the functionality, it could be structured in the following layers:

- IoT layer: a gateway device is used to collect images from an industrial thermographic camera and local sensors that allow a better analysis of the image.
- Edge layer: the thermographic image is used, with the sensors data collected, to feed a computer vision algorithm that calculates the temperature of some areas in the image and generate alerts based on defined thresholds. This layer also integrates with local SCADA to provide the data to previous monitoring systems. Although the functionality is clearly different from the previous layer, they are located in the same gateway.
- Application/Cloud layer: the data is sent to a remote cloud where it is stored and allows the visualization in a web interface.

Similarly to the first example, this is an application that includes edge processing to enhance the value of the data collected and avoid the delivery of high volume of data to the cloud.

Advanced processing in edge nodes is an increasing trending. Particularly those use cases related with Artificial Intelligence algorithms or computer vision. Some additional examples are [21], [22] or [23].

4.4 Renewable Plants Supervision

The low maintenance cost and the scalability of the photovoltaic plants, the installed power of such renewable energy has increased in the last years. Even

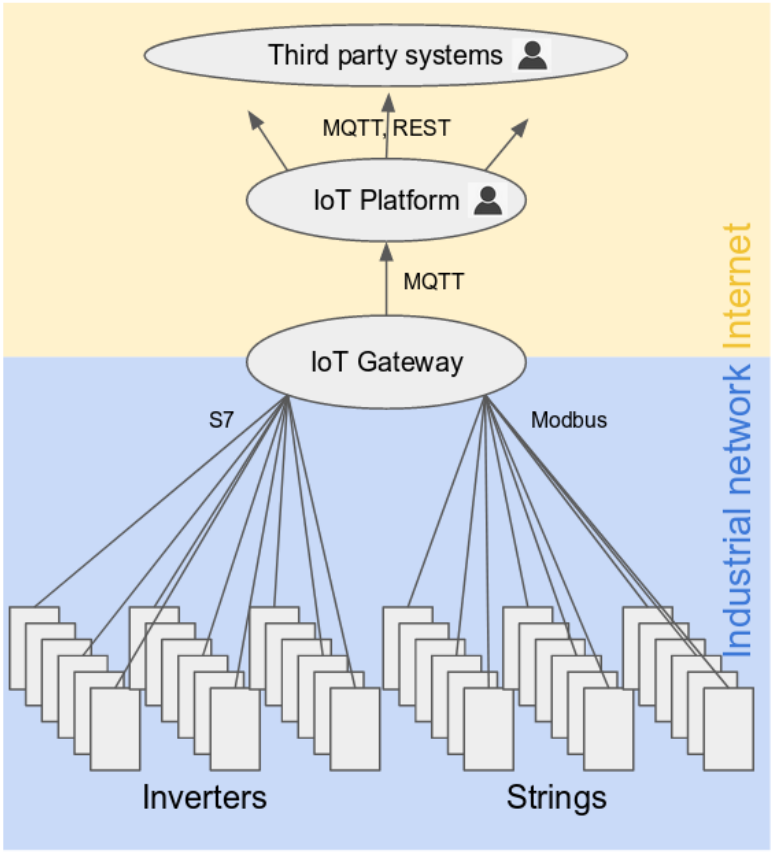


Fig. 5 Schema of the photovoltaic plants supervision system based on open source. The image has been obtained from [24].

domestic environment have integrated the deployment of solar panels. There is an increasing need for solutions that can be used to monitor the systems involved. The authors of [24] have developed a IoT solution completely

based on open source for photovoltaic plants supervision.

The Figure 5 shows the architecture according to the authors. In this case, IoT and Edge layer are presented together in the IoT gateway. The structure could be described as:

- IoT layer: the IoT device (called IoT gateway in the Figure) integrates the role of both layers as it is capable to collect the data from all of the devices but it transforms the data directly into the required format for the upper layer. As such abstraction is already performed and no processing is required, there is no need for an Edge layer in the system.

- Application/Cloud layer: the IoT platform centralizes the acquisition of data and enables the actuation over the IoT devices for direct management or third party integration. The data is also available for such external tools. The platform is not intended to be used for direct monitoring of the data but instead provide tools for integration with other systems. Some examples are shown in the solution

The communication under the IoT layers is purely industrial protocol communications. IoT-Cloud communication is made based on MQTT not only for the data delivery but also to hold the management commands. The data is collected in various formats but it is transformed into a standard format for the incoming in the IoT platform. The data can be exported from the solution at both IoT and Application layers by multiple protocols: MQTT, REST, direct integration with database, or even a new protocol as both layers allow extensible routing tools.

Other applications on renewable plants supervision based on IoT can be found in [25] or [26].

References

[1] Y. Wu, Y. Wu, J. M. Guerrero, and J. C. Vasquez, "Digitalization and decentralization driving transactive energy Internet: Key technologies and infrastructures," *International Journal of Electrical Power & Energy Systems*, vol. 126, p. 106593, mar 2021.

[2] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y. J. Park, "A Survey on Trend and Classification of Internet of Things Reviews," pp. 111 763–111 782, 2020.

[3] F. Alsubaei, A. Abuhussein, and S. Shiva, "An Overview of Enabling Technologies for the Internet of Things," in *Internet of Things A to Z.* John Wiley & Sons, Inc., may 2018, pp. 77–112.

[4] S. N. Swamy and S. R. Kota, "An Empirical Study on System Level Aspects of Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 188 082–188 134, oct 2020.

[5] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet of Things*, 2018.

[6] S. Labs, "AN1138: Zigbee Mesh Network Performance," Silicon Labs, Tech. Rep., 2021. [Online]. Available: https://www.silabs.com/documents/login/application-notes/an1138-zigbee-mesh-network-performance.pdf

[7] ——, "AN1142: Mesh Network Performance Comparison," Silicon Labs, Tech. Rep., 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1142-mesh-network-performance-comparison.pdf

[8] ——, "AN1137: Bluetooth® Mesh Network Performance," Silicon Labs, Tech. Rep., 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1137-bluetooth-mesh-network-performance.pdf

[9] ——, "AN1141: Thread Mesh Network Performance," Silicon Labs, Tech. Rep., 2021. [Online]. Available: https://www.silabs.com/documents/login/application-notes/an1141-thread-mesh-network-performance.pdf

[10] K. J. Singh and D. S. Kapoor, "Create Your Own Internet of Things: A survey of IoT platforms." *IEEE Consumer Electronics Magazine*, vol. 6, no. 2, pp. 57–68, 2017.

[11] M. Zdravković, M. Trajanović, J. Sarraipa, R. Jardim-Gonçalves, M. Lezoche, A. Aubry, H. Panetto, and al Survey, "Survey of Internet-of-Things platforms," HAL, Tech. Rep., 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01298141

[12] M. H. Yaghmaee Moghaddam and A. Leon-Garcia, "A fog-based internet of energy architecture for transactive energy management systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1055–1069, 2018.

[13] M. H. Yaghmaee, A. Leon-Garcia, and M. Moghaddassian, "On the Performance of Distributed and Cloud-Based Demand Response in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5403–5417, sep 2018.

[14] G. Radhakrishnan and V. Gopalakrishnan, "Applications of internet of things (IOT) to improve the stability of a grid connected power system using interline power flow controller," *Microprocessors and Microsystems*, vol. 76, p. 103038, jul 2020.

[15] R. J. Tom, S. Sankaranarayanan, and J. J. Rodrigues, "Agent negotiation in an IoT-Fog based power distribution system for demand reduction," *Sustainable Energy Technologies and Assessments*, vol. 38, p. 100653, apr 2020.

[16] P. Pawar, M. TarunKumar, and P. Vittal K., "An IoT based Intelligent Smart Energy Management System with accurate forecasting and load strategy for renewable generation," *Measurement: Journal of the International Measurement Confederation*, vol. 152, p. 107187, feb 2020.

[17] K. Friansa, I. N. Haq, B. M. Santi, D. Kurniadi, E. Leksono, and B. Yuliarto, "Development of Battery Monitoring System in Smart Microgrid Based on Internet of Things (IoT)," in *Procedia Engineering*, vol. 170. Elsevier Ltd, jan 2017, pp. 482–487.

[18] A. D'Elia, F. Viola, F. Montori, P. Azzoni, and M. Maiero, "Electro Mobility automation through the Arrowhead Framework," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, oct 2016, pp. 5246–5252.

[19] C. Garrido-Hidalgo, F. J. Ramirez, T. Olivares, and L. Roda-Sanchez, "The adoption of internet of things in a circular supply chain framework for the recovery of WEEE: the case of lithium-ion electric vehicle battery packs," *Waste Management*, vol. 103, pp. 32–44, feb 2020.

[20] R. Usamentiaga, M. A. Fernandez, A. F. Villan, and J. L. Carus, "Temperature Monitoring for Electrical Substations Using Infrared Thermography: Architecture for Industrial Internet of Things," *IEEE Transactions on*

Industrial Informatics, vol. 14, no. 12, pp. 5667–5677, dec 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8453868/

[21] C. Zheng, S. Wang, Y. Zhang, P. Zhang, and Y. Zhao, "A robust and automatic recognition system of analog instruments in power system by using computer vision," *Measurement*, vol. 92, pp. 413–420, oct 2016.

[22] M. Dorokhova, Y. Martinson, C. Ballif, and N. Wyrsch, "Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation," *Applied Energy*, vol. 301, p. 117504, nov 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261921008874

[23] A. Mellit, A. M. Pavan, and V. Lughi, "Deep learning neural networks for short-term photovoltaic power forecasting," *Renewable Energy*, vol. 172, pp. 276–288, jul 2021.

[24] P. de Arquer Fernández, M. Á. Fernández Fernández, J. L. Carús Candás, and P. Arboleya Arboleya, "An IoT open source platform for photovoltaic plants supervision," *International Journal of Electrical Power and Energy Systems*, vol. 125, p. 106540, feb 2021.

[25] R. I. Pereira, S. C. Jucá, and P. C. Carvalho, "IoT embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants," *Measurement: Journal of the International Measurement Confederation*, vol. 142, pp. 195–212, aug 2019.

[26] D. Prasanna Rani, D. Suresh, P. Rao Kapula, C. Mohammad Akram, N. Hemalatha, and P. Kumar Soni, "IoT based smart solar energy monitoring systems," *Materials Today: Proceedings*, jul 2021.

Apéndice D

Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography

Temperature Monitoring of Electrical Substation Equipment by Infrared Thermography

A. Fernández Villán(1) 1*, M. A. Fernández Fernández(1) 2, JL. Carús Candás(1) 3, P. De Arquer Fernández(1) 4, N. Arias Linacero(1) 5, R. Usamentiaga Fernández(2) [1]

(1) TSK, c/ Ada Byron, 220, Gijón, Asturias (Spain)
{[1]alberto.fernandez, [2]miguelangel.fernandez, [3]juanluis.carus, [4]pedro.arquer, [5]nabila.arias}@grupotsk.com
(2) Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques 33204 Gijon, Asturias, Spain
[1]rusamentiaga@uniovi.es

** Corresponding Author e-mail: alberto.fernandez@grupotsk.com*

Topic/s: Performance Analysis, Sustainable Energy Research and Applications for Industries, Safety and Security, Computer Engineering and Technology

Keywords: Temperature measurement, Infrared Inspection, Electrical substations monitoring

In recent years, the economic expansion of countries has led to a growth in the demand for electrical energy, requiring larger and more complex power systems. This complexity shows a direct consequence in the inspection and maintenance tasks, and the occurrence of blackouts in different electrical networks worldwide has exhibited the vulnerability of power systems [1]. In this sense, the systematic inspection of electrical substations is a key factor to predict failures. For example, small and undetected electrical problems can lead to serious consequences if left unchecked. These failures provoke unnecessary energy losses and additionally, they can also result in: a) costly unplanned outages [1], b) severe injuries in technicians or c) a fire [2], among others. Therefore, frequent inspections are essential to determine the integrity, safety and reliability of this type of industrial installations ensuring safe long-term operation [3].

Faults in electrical power systems can be classified into a few categories, such as poor electrical connections, short or open circuits, overloads, loads imbalance and improper equipments installation [4]. In high voltage installations, heat plays a critical role. Many sources of failures can be detected based on the heat of the components [5]. When the temperature increases, thermal expansion causes the connection to lose even more strength [6]. According to an IRT survey conducted during the period of 1999–2005 [7], it was found that most of the thermal problems were found in conductor connection accessories and bolted connections. This kind of problem can be recognized by inspecting the heat of the components via infrared thermography (IRT) where the highest temperature point indicates the source and location of the problem.

Temperature monitoring using IRT has become a mature and widely accepted technology, having many advantages over other types of sensors and technologies [8,9]. Nevertheless, IRT requires adequate setup and testing procedures, depending on the operator skill. The quantitative temperature measurements can only be performed if adequate configuration

is carried out. Since IRT is a complex and expensive procedure, it is only performed periodically, decreasing the ability of this technique to detect early failures.

The solution proposed in this paper is based on the deployment of an automatic temperature monitoring system. This system is installed in the electrical substation inspecting the required components continuously. The main contribution of this work is the definition of a comprehensive architecture for the temperature monitoring of electrical substations using IRT. This software and hardware architecture includes: 1) the selection of the appropriate hardware components and 2) the definition of the software design and its implementation.

On the one hand, the selection of the hardware components is carried out taking into account main requirements to be achieved and the characteristics of the substation to be monitored. On the other hand, this software design encompass key aspects to achieve an integral monitoring of this kind of installations, from the acquisition of the sensor signals to be processed to the friendly visualization of the aggregated information.

[1] Menendez, O., Cheein, F. A. A., Perez, M., & Kouro, S. (2017). Robotics in Power Systems: Enabling a More Reliable and Safe Grid. IEEE Industrial Electronics Magazine, 11(2), 22-34.
[2] Babrauskas, V. (2001, January). How do electrical wiring faults lead to structure ignitions. In Proc. Fire and Materials 2001 Conf (pp. 39-51).
[3] Shull, P. J. (2016). Nondestructive evaluation: theory, techniques, and applications. CRC press.
[4] Jadin, M. S., & Taib, S. (2012). Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography. Infrared Physics & Technology, 55(4), 236-245.
[5] M. Braunovic, N. K. Myshkin, and V. V. Konchits, Electrical contacts: fundamentals, applications and technology. CRC press, 2006.
[6] A. N. Huda and S. Taib, “Application of infrared thermography for predictive/preventive maintenance of thermal defect in electrical equipment,” Applied Thermal Engineering, vol. 61, no. 2, pp. 220–227, 2013.
[7] J. Martínez, R. Lagioia, Experience performing infrared thermography in the maintenance of a distribution utility, in: Proceedings of the International Conference on Electricity, Distribution, 2007, pp. 1–4.
[8] R. Usamentiaga, P. Venegas, J. Guerediaga, L. Vega, J. Molleda, and F. G. Bulnes, “Infrared thermography for temperature measurement and non-destructive testing,” Sensors, vol. 14, no. 7, pp. 12 305–12 348, 2014.
[9] Ullah, I., Yang, F., Khan, R., Liu, L., Yang, H., Gao, B., & Sun, K. (2017). Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach. Energies, 10(12), 1987.