

Evaluación De Técnicas De UNS (Unified Namespace) Para Una Plataforma De  
Entrenamiento IIOT a Través Del Protocolo MQTT Sparkplug B

Juliam Andrés Díaz Barrera

Tesis presentada para optar por el título de Magíster en Ingeniería de Telecomunicaciones

Director

Juan Manuel Rey López

Doctor en Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Maestría en Ingeniería de Telecomunicaciones

Bucaramanga

2025

## Índice

<b>Introducción</b>	<b>1</b>
Justificación . . . . .	5
Objetivo General . . . . .	5
Objetivos Específicos . . . . .	6
Esquema del trabajo de investigación . . . . .	6
<b>Capítulo 1: Transformación Digital</b>	<b>7</b>
Protocolo de Comunicación MQTT . . . . .	8
Integración de Sistemas Empresariales y de Control . . . . .	9
<b>Capítulo 2: Unified Namespace: Marco Conceptual</b>	<b>13</b>
¿Que es el UNS? . . . . .	14
¿Cuáles son las Ventajas de un UNS? . . . . .	15
¿Cómo funciona un UNS? . . . . .	17
<b>Capítulo 3: Técnicas de Diseño de UNS</b>	<b>20</b>
Método de Schultz . . . . .	20
Método de Parris . . . . .	21
<b>Capítulo 4: Implementación de Técnicas UNS en la Plataforma IIoT</b>	<b>24</b>
Descripción de la Plataforma IIoT . . . . .	24
Dispositivos y Nodos Industriales: . . . . .	24
Implementación del Método Schultz . . . . .	26
Arquitectura de Diseño . . . . .	27
Implementación en Ignition . . . . .	29
Implementación del Método Parris . . . . .	30
Arquitectura de Diseño . . . . .	30
Implementación en Ignition . . . . .	32

Aplicación Ignition Plataforma IIoT . . . . .	34
<b>Capítulo 5: Análisis Comparativo de las Técnicas Implementadas</b>	<b>37</b>
Métricas Cualitativas de Desempeño . . . . .	38
Integración . . . . .	38
Resiliencia . . . . .	39
Flexibilidad . . . . .	40
Seguridad . . . . .	40
Implementación . . . . .	41
Escalabilidad . . . . .	42
Métricas Cuantitativas de Desempeño . . . . .	42
Gestión de Tópicos y Sesiones MQTT . . . . .	44
Gestión de Autenticaciones y Autorizaciones . . . . .	46
Flujo de Datos . . . . .	48
Gestión de Clientes . . . . .	50
Gestión de Mensajes . . . . .	52
Actividad de Paquetes . . . . .	54
Síntesis y Recomendaciones de Implementación . . . . .	57
<b>Conclusiones</b>	<b>59</b>
<b>Trabajo futuro</b>	<b>61</b>
<b>Referencias</b>	<b>62</b>
<b>Anexo A: Integración de un Sensor de Espectro Radioeléctrico en la Plata- forma IIoT</b>	<b>68</b>
Características del Dispositivo Espectrómetro . . . . .	68
Actualización de la Arquitectura Método Parris . . . . .	68
Visualización del Espectro en Ignition . . . . .	70

Consideraciones Finales . . . . . 71

## Índice de figuras

1.	Arquitectura Stack basada en la norma ISA-95. . . . .	3
2.	Marco estructural para el protocolo de mensajería MQTT. . . . .	9
3.	Capas del Modelo ISA-95 . . . . .	10
4.	Arquitectura Tradicional y Central . . . . .	12
5.	Arquitectura Central UNS . . . . .	15
6.	Esquema de la estructura del método Schultz . . . . .	20
7.	Esquema de la estructura del método de Parris . . . . .	22
8.	Esquema de la Plataforma de Entrenamiento IIoT . . . . .	25
9.	Esquema de la Arquitectura UNS con el Método de Schultz en la Plataforma IIoT . . . . .	27
10.	UNS Schultz en Ignition . . . . .	29
11.	Esquema de la Arquitectura UNS con el Metodo de Parris en la Plataforma IIoT . . . . .	31
12.	Método de Parris en el Módulo MQTT Engine en Ignition . . . . .	32
13.	Método de Parris Estructurado en Ignition . . . . .	33
14.	Vista del Home en la aplicación de Ignition . . . . .	34
15.	Vista de Casiri en la aplicación de Ignition . . . . .	35
16.	Vista de la Microrred en la aplicación de Ignition . . . . .	35
17.	Vista del Vivero en la aplicación de Ignition . . . . .	36
18.	Arquitectura de Evaluación de Desempeño . . . . .	43
19.	Test de Gestión de Tópicos y Sesiones MQTT . . . . .	45
20.	Test de Gestión de Autenticaciones y Autorizaciones . . . . .	47
21.	Test de Flujo de Datos . . . . .	48
22.	Test de Gestión de Clientes . . . . .	51
23.	Test de Gestión de Mensajes . . . . .	53
24.	Test de Actividad de Paquetes . . . . .	56

25. Esquema de la Arquitectura UNS con el Metodo de Parris integrado el Es- pectrometro . . . . .	69
26. Vista de Espectrómetro de Casiri en la aplicación de Ignition . . . . .	70

## **Índice de tablas**

1.	Clasificación de Métricas de Desempeño . . . . .	37
2.	Comparación de Gestión de Tópicos y Sesiones MQTT . . . . .	45
3.	Comparación de Gestión de Autenticaciones y Autorizaciones . . . . .	47
4.	Comparación de Flujo de Datos . . . . .	49
5.	Comparación de Gestión de Clientes . . . . .	51
6.	Comparación de Gestión de Mensajes . . . . .	54
7.	Comparación de Actividad de Paquetes . . . . .	57
8.	Síntesis de Comparación entre Schultz y Parris . . . . .	58

## Resumen

La Cuarta Revolución Industrial, o Industria 4.0, está redefiniendo el sector manufacturero mediante la integración de tecnologías avanzadas como el Internet Industrial de las Cosas (IIoT), la inteligencia artificial (IA) y los sistemas ciberfísicos. Un elemento clave en esta transformación es el *Unified Namespace (UNS)*, una arquitectura que optimiza la gestión de datos en tiempo real, superando las limitaciones de integración y escalabilidad de las estructuras tradicionales. Sin embargo, su adopción generalizada se ha visto obstaculizada por la falta de documentación técnica detallada y metodologías de implementación bien definidas. Este estudio revisa los conceptos fundamentales del UNS, analiza sus metodologías de implementación y compara diferentes enfoques de diseño. Para validar su aplicación, se desarrolla un caso práctico basado en una plataforma de entrenamiento IIoT que utiliza el protocolo de comunicación MQTT Sparkplug B. Este trabajo no solo proporciona una referencia estructurada para la comunidad académica, sino que también ofrece lineamientos prácticos para la industria, facilitando la comprensión y adopción del UNS en entornos productivos.

## Abstract

The Fourth Industrial Revolution, or Industry 4.0, is transforming the manufacturing sector by integrating advanced technologies such as the Industrial Internet of Things (IIoT), artificial intelligence (AI), and cyber-physical systems. A key enabler of this transformation is the *Unified Namespace (UNS)*, an architecture that optimizes real-time data management, overcoming the integration and scalability challenges of traditional structures. However, its widespread adoption has been limited by the lack of comprehensive technical documentation and well-defined implementation methodologies. This study reviews the fundamental concepts of UNS, analyzes its implementation methodologies, and compares different design approaches. To validate its application, a case study is developed using an IIoT training platform based on the MQTT Sparkplug B communication protocol. This work not only provides a structured reference for the academic community but also offers practical guidelines for industry stakeholders, facilitating the understanding and adoption of UNS in manufacturing environments.

## **Introducción**

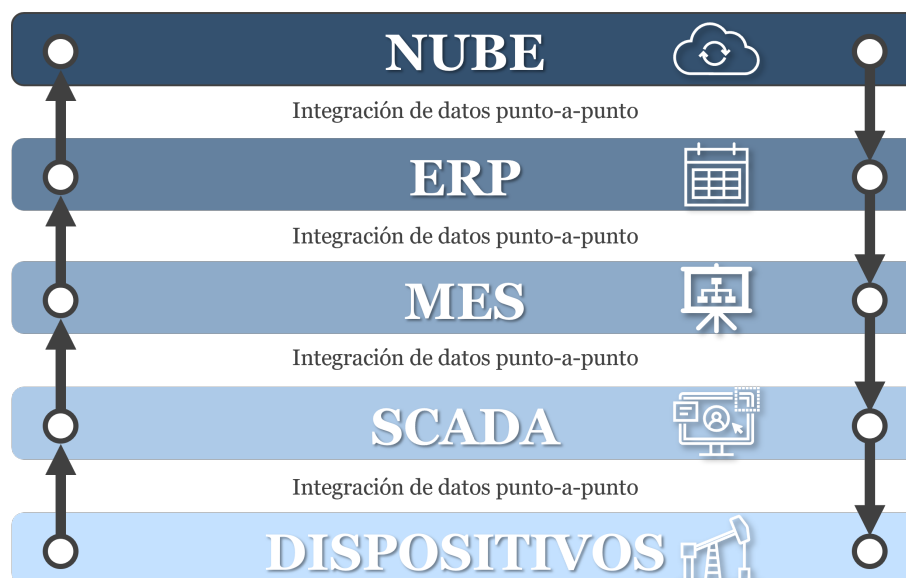
La cuarta revolución industrial marca un hito en la historia de la manufactura, representando una transformación disruptiva en cuanto a su alcance, velocidad e impacto sobre la productividad (Volf, Dohnal, Beranek, y Kyncl, 2024). La primera revolución industrial, a fines del siglo XVIII, introdujo la mecanización a través del agua y el vapor; la segunda revolución, en el siglo XIX, trajo la producción en masa con la electricidad; y la tercera revolución, a mediados del siglo XX, incorporó la automatización de procesos industriales y la creación del PLC para generar la data digital y las tecnologías informáticas (Stearns, 2018). Por su parte, la cuarta revolución industrial, o también llamada Industria 4.0, se caracteriza por la convergencia de las tecnologías digitales y físicas para transformar la forma en que vivimos y nos relacionamos, impulsando la productividad, la eficiencia y la innovación a través de la integración digital y generando información para brindar contexto a los datos recolectados y contribuir a la tomar decisiones. Las tecnologías clave que impulsan la Industria 4.0 incluyen los sistemas ciberfísicos (CPS), el Internet de las cosas (IoT), el cloud-computing, el big-data y la inteligencia artificial, entre otras (Srivastava, Sapra, Sinha, Anup, y Sinwar, 2023).

Para mejorar la productividad de los procesos de fabricación modernos, las industrias aprovechan las ventajas que ofrecen los sistemas automatizados, especializados e interconectados. Gracias a estos, es posible obtener un gran volumen de datos sobre los procesos y sus etapas, permitiendo el análisis para la mejora de la confiabilidad, la predicción de fallas y la optimización de las operaciones (Derhamy, Andersson, Eliasson, y Delsing, 2018). Debido a estas características, cada vez más fabricantes tradicionales han iniciado su viaje a la transformación digital, lo que supone enfrentar una serie de desafíos relacionados con el costo de inversión, la escalabilidad y la integración de nuevos dispositivos. En este contexto, el IoT juega un papel clave para la Industria 4.0, al permitir el desarrollo de soluciones confiables y escalables, que fácilmente pueden interconectarse e integrarse a diferentes tipos de plataformas, sistemas, procesos e industrias (Xu, He, y Li, 2014; Yu, Liu, Dillon, Rahayu, y Mostafa, 2022).

El objetivo final de la transformación digital en el sector manufacturero es la evolución hacia las *smart factories*. De acuerdo con la resolución ISO 114/2017, estas se definen como “organizaciones que optimizan su rendimiento mediante el uso integrado e inteligente de procesos y recursos dentro de entornos ciberfísicos y humanos, con el fin de diseñar y ofrecer productos y servicios en colaboración con otros dominios a lo largo de las cadenas de valor de la empresa” (Johnsson y Brandl, 2021). Estas se caracterizan por integrar un piso de producción altamente digitalizado que recolecta y comparte datos de manera continua a través de dispositivos y sistemas de producción conectados (Razooqi, Hassan, y Abdel Saheb, 2022). Para aprovechar al máximo las capacidades de las smart factories a través de las tecnologías de la información y la comunicación (TIC’s) y el IoT, es fundamental integrar cuatro pilares: la interconexión, la transparencia de la información, las decisiones descentralizadas y la asistencia técnica (Singh, Goyat, y Panwar, 2024; Software, 2019). En esta línea, aproximadamente hace 30 años, la Sociedad Internacional de Automatización (ISA) introdujo el estándar internacional ISA-95 con el fin de normalizar las prácticas de integración entre sistemas de empresa y de control dentro de una organización (Johnsson, 2004). Para esto, se propuso un modelo jerárquico que agrupa en niveles las diferentes actividades dentro de las organizaciones (Kannan y cols., 2017).

Actualmente, la ISA-95 ha sido ampliamente adoptada por industrias alrededor del mundo, mediante arquitecturas tradicionales como la que se muestra en la Fig. 1. Este tipo de arquitectura se caracteriza por la facilidad en su implementación y por permitir una visualización y monitorización práctica de los datos críticos en cada nivel (Theorin y cols., 2017). Sin embargo, este enfoque presenta desafíos en cuanto a la escalabilidad y la integración (Johnsson y Brandl, 2021). La arquitectura tradicional de stack de tecnología de automatización suele ser rígida, lo que dificulta la implementación ágil de nuevos dispositivos y sistemas, al requerir el ajuste de varios niveles de la estructura (Bai, Zhu, Zhang, y Han, 2024). Además, esta arquitectura presenta limitaciones para la integración eficaz de niveles no adyacentes, lo que genera brechas de comunicación y coordinación entre las áreas de producción y planificación.

Esta fragmentación dificulta la alineación estratégica y operativa, afectando la fluidez en la toma de decisiones y la optimización de recursos.



**Figura 1**

*Arquitectura Stack basada en la norma ISA-95.*

Lo anterior ha motivado la exploración de nuevas arquitecturas que se caracterizan por reducir las jerarquías tradicionales, facilitando una mayor agilidad organizativa (Johnson y Brandl, 2021). Entre estas, se destaca la arquitectura basada en el Unified Namespace (UNS). Este enfoque redefine el manejo de datos mediante la creación de un hub que actúa como una única fuente de verdad, permitiendo el control y monitoreo centralizado de todas las líneas de producción en un solo punto. El UNS ofrece ventajas como la mejora en la escalabilidad, pues permite crecer el número de dispositivos conectados sin comprometer el rendimiento; la mejora en interoperabilidad, al reducir las limitaciones de integración entre los niveles y sistemas; y un sólido control de acceso, garantizando la seguridad y confidencialidad de los datos críticos.

A pesar de estas ventajas operativas, la implementación masiva de sistemas con la arquitectura basada en UNS se ha visto limitada por la escasa documentación existente en la literatura especializada. De hecho, al realizar una búsqueda bibliográfica en bases de datos

científicas como IEEE, Scopus, ScienceDirect, MDPI, Springer and Taylor & Francis usando la ecuación de búsqueda “*Unified*” AND “*Namespace*” no se encuentran documentos que utilicen este término explícitamente en el título. Algunos trabajos como (Domíguez-Cid y cols., 2024; Koprov, Gadhwal, Walimbe, Fang, y Starly, 2023) presentan proyectos prácticos en los que se han implementado arquitecturas basadas en UNS. Sin embargo, estos trabajos buscan divulgar la experiencia de casos de aplicación más que profundizar formalmente en el concepto y alternativas de implementación de este tipo de arquitectura. Las aportaciones más relevantes actualmente en la construcción del concepto de UNS se han presentado en referencias como (Manditereza, 2022a, 2022b, 2022c; W. Reynolds, 2024b; Systems, 2024) que corresponden a páginas web y webinars. Recientemente, se publicó la tesis de maestría (Péter y Werner, 2024) en la que se discute algunos de los beneficios y desventajas del UNS frente a otras arquitecturas y se estudia cómo el concepto de UNS puede ser usado para abordar desafíos de la gestión de datos fragmentados en las organizaciones. Este trabajo es uno de los primeros documentos académicos en el área y se enfoca en solo uno de los métodos de diseño de arquitecturas UNS.

Por lo anterior, este documento busca abordar los retos mencionados anteriormente, presentando los principales conceptos y métodos de diseño propuestos hasta la fecha para la implementación del UNS. Para esto, se desarrolla el marco conceptual de la transformación digital, incluyendo conceptos como el de IoT, IIoT, el protocolo MQTT y el estándar ISA-95, además de describir la definición del UNS, haciendo énfasis en sus ventajas, funcionamiento e implementación mediante dos de los métodos de diseño más relevantes actualmente. Finalmente, con el objetivo de dar claridad, se ejemplifica su implementación mediante una plataforma de entrenamiento IIoT que se describe dentro del documento.

## Justificación

Como consecuencia de la cuarta revolución industrial, las empresas han llevado a cabo transformaciones digitales en sus procesos de fabricación. Su información, informes y bases de datos se reflejan en *data-lakes* que deben organizarse adecuadamente para convertirse en una empresa inteligente. Un protocolo de comunicación permite que varios dispositivos conectados en red interactúen entre sí independientemente de su diseño, estructura y procesos internos. Durante la ejecución del protocolo, tanto el hardware como el software permiten el intercambio de datos e información entre dispositivos, máquinas, sensores y actuadores del Internet Industrial de las Cosas (IIoT). En los últimos años, las empresas han implantado el estándar ISA-95, que establece una pirámide de 5 niveles de comunicación vertical entre los procesos. Aunque este estándar se ha implantado ampliamente, tiene el inconveniente de que limita la comunicación horizontal entre los sistemas del mismo nivel. Como alternativa, ha surgido el concepto de Unified Namespace (UNS), una arquitectura moderna para la organización estructurada de la información empresarial permitiendo una integración fluida vertical y horizontal entre procesos, sistemas y dispositivos en entornos de manufactura inteligente. Debido a lo reciente de la propuesta de arquitectura UNS, no existen comparaciones rigurosas de sus técnicas de diseño más relevantes. Por ello, en este trabajo se presenta una comparación y evaluación de las características operativas de diferentes técnicas de diseño del UNS. Como caso de aplicación, se utiliza una plataforma de entrenamiento IIoT basada en el protocolo de comunicación MQTT Sparkplug B para desarrollar pruebas de validación. Más allá de una contribución académica, este trabajo busca realizar un análisis que pueda ser útil en aplicaciones industriales del UNS.

## Objetivo General

Evaluar las características operativas de las técnicas UNS utilizando como caso de aplicación una plataforma de entrenamiento IIoT basada en el protocolo de comunicación MQTT Sparkplug B.

### **Objetivos Específicos**

1. Identificar y describir el estado del arte de las técnicas de diseño de UNS.
2. Definir métricas y criterios de evaluación para medir el desempeño operacional de las técnicas UNS a implementar.
3. Implementar las técnicas UNS en la plataforma Ignition.
4. Evaluar el desempeño operacional de las técnicas UNS en una plataforma de entrenamiento IIoT utilizando el protocolo de comunicación MQTT Sparkplug B.

### **Esquema del trabajo de investigación**

Este trabajo de investigación está organizado de la siguiente manera:

- **Capítulo 1:** Introducción a la transformación digital y su contribución al desarrollo de la industria 4.0 en diversos sectores tecnológicos.
- **Capítulo 2:** Descripción de la arquitectura centralizada Unified Namespace, incluyendo sus ventajas y consideraciones para su implementación.
- **Capítulo 3:** Presentación detallada de las técnicas de Schultz y Parris aplicadas a la implementación de una arquitectura UNS.
- **Capítulo 4:** Desarrollo de la implementación de las técnicas descritas dentro de una plataforma de entrenamiento IIoT.
- **Capítulo 5:** Definición de las métricas de rendimiento para el análisis comparativo de las técnicas seleccionadas.
- **Capítulo 6:** Exposición de las conclusiones más relevantes y propuesta de una línea futura de investigación.

## Capítulo 1: Transformación Digital

La transformación digital es eje central de la Industria 4.0, posicionando los datos como el insumo más valioso para las empresas modernas. Mediante la integración de sistemas físicos y digitales a través de las TIC y el análisis avanzado de datos, esta transformación redefine las operaciones industriales, la gestión de procesos y la toma de decisiones en tiempo real. En este contexto, las empresas deben reestructurar sus estrategias y operaciones para adaptarse a entornos cada vez más interconectados y flexibles, facilitando la transición de la Industria 3.0 a la Industria 4.0 (Kaiser, 2017).

Con la transformación digital es posible integrar tecnologías avanzadas en los procesos de fabricación, como la IA y el análisis de grandes volúmenes de datos (Bhadra, Chakraborty, y Saha, 2023). Para ello, es clave desplegar infraestructuras y sistemas de comunicaciones que favorezcan la monitorización y la integración, como el Internet Industrial de las Cosas (IIoT).

El IEEE define el IoT como *“una red de elementos, cada uno integrado con sensores, que están conectados a Internet”* (Minerva, Biru, y Rotondi, 2015). De manera más amplia, el IoT puede entenderse como una *“infraestructura de red global dinámica con capacidades de auto-configuración, basada en protocolos de comunicación estándar e interoperables, donde los ‘objetos’ físicos y virtuales poseen identidades, atributos físicos y personalidades virtuales, utilizan interfaces inteligentes y se integran de manera transparente en la red de información”* (Van Kranenburg, 2007; Xu y cols., 2014).

El IIoT es la aplicación del IoT a la interconexión de máquinas industriales, equipos, sensores, dispositivos y sistemas con plataformas avanzadas de análisis y desarrollo, capaces de procesar grandes volúmenes de datos, generalmente en tiempo real (Milić y Babić, 2020). Esta tecnología requiere el uso de arquitecturas más robustas para garantizar la alta disponibilidad exigida por los sectores industriales (Milić y Babić, 2020; Sisinni, Saifullah, Han, Jennehag, y Gidlund, 2018).

Para garantizar el intercambio seguro de información entre máquinas (M2M), es necesario

implementar protocolos de comunicación. Estos protocolos consisten en conjuntos de reglas o instrucciones que guían las acciones durante el intercambio de datos (Laghari, Manickam, y Karuppayah, 2021).

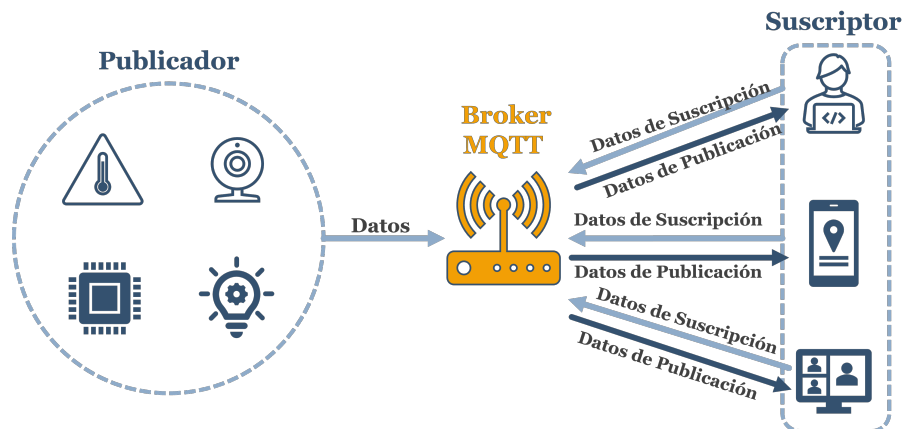
En aplicaciones de IIoT, los protocolos más utilizados incluyen CoAP (Constrained Application Protocol), AMQP (Advanced Message-Queuing Protocol), HTTP (Hypertext Transfer Protocol) y MQTT (Message-Queuing Telemetry Transport) (Hou y cols., 2016; Mantravadi, Srai, Brunoe, y Moller, 2020; Tripathi y Chaurasia, 2023). Entre ellos, MQTT ha adquirido una relevancia significativa en los últimos años debido a su eficiencia en el uso del ancho de banda y su bajo consumo de energía, lo que lo convierte en una opción ideal para garantizar la interoperabilidad entre dispositivos IoT e IIoT (Hintaw, Manickam, Karuppayah, y Aboalmaaly, 2019).

### Protocolo de Comunicación MQTT

MQTT es el principal protocolo de mensajería utilizado en IIoT, valorado por su ligereza, fiabilidad, escalabilidad y facilidad de implementación (Kurdi y Thayananthan, 2021). Fue diseñado para dispositivos con capacidades computacionales limitadas y conexiones de red inestables. Teóricamente, admite un número ilimitado de dispositivos intercambiando información sin requerir una conexión estable o síncrona (Kashyap, Sharma, y Gupta, 2018).

MQTT opera bajo un patrón de mensajería *publish/subscribe*, donde los publicadores y suscriptores de datos interactúan a través de *topics* mediante un servidor centralizado (broker MQTT) en tiempo real y de manera asíncrona. Esto permite una comunicación eficiente y escalable, además de un desacoplamiento entre clientes (Gruener, Koziolk, y Ruckert, 2021).

En el protocolo MQTT, existen dos entidades clave en la red: *el broker* y *el cliente*, siendo este último tanto publicador como suscriptor. Los clientes publicadores envían mensajes con un tema específico al *broker*, que actúa como intermediario, gestionando conexiones, almacenamiento y enviando los mensajes a los clientes suscriptores. Esta arquitectura permite que los publicadores no necesiten conocer a los suscriptores y viceversa. La Figura 2 muestra el esquema estructural del protocolo MQTT.



**Figura 2**

*Marco estructural para el protocolo de mensajería MQTT.*

Para estandarizar el lenguaje y la sintaxis de comunicación en aplicaciones industriales, se han propuesto varios estándares para el protocolo MQTT. Este es el caso de Sparkplug B, un estándar de software de código abierto diseñado para permitir un marco de integración de datos sin interrupciones entre dispositivos en sistemas IIoT. Mientras que MQTT proporciona el canal de comunicación, Sparkplug B añade una capa adicional para la definición de la carga útil (*payload*), organiza los temas de manera uniforme y establece un formato estandarizado de carga útil con la siguiente estructura (Foundation, 2016):

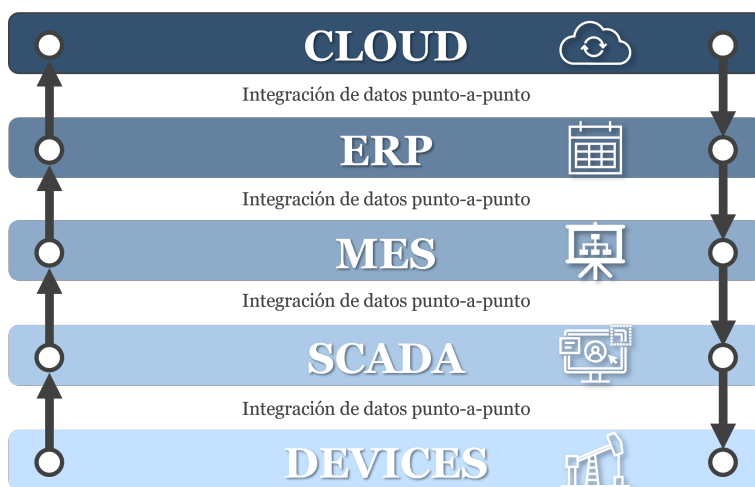
$$spBv1.0/GroupId/DData/NodeId/DeviceId$$

### **Integración de Sistemas Empresariales y de Control**

El estándar ISA-95 define las actividades específicas que una organización debe realizar sin especificar los sistemas que deben implementarlas o ejecutarlas. Por ejemplo, en el caso de la gestión de producción, ISA-95 establece que las actividades incluyen la planificación, ejecución, control y ajuste de la producción, pero no determina qué software debe llevar a cabo estas tareas (Yue, Niu, y Wang, 2019). Una empresa puede implementar estas funciones mediante un MES (Manufacturing Execution System), un ERP (Enterprise Resource Planning) o incluso mediante el registro manual de información y herramientas de automatización de

oficina (Moones y cols., 2015). Por lo tanto, aunque estas actividades siguen siendo fundamentales para la organización, los sistemas de soporte pueden evolucionar con el tiempo para cumplir con los requisitos de la transformación digital.

Actualmente, la mayoría de los sistemas industriales que implementan el estándar ISA-95 dependen de un modelo de integración discreta punto a punto (Apilioğulları, 2022), como se muestra en la Fig. 3. En muchas fábricas inteligentes, coexisten múltiples sistemas SCADA y ERP, lo que dificulta aún más la integración. Dentro de esta arquitectura, cada capa se comunica únicamente con la capa inmediatamente superior o inferior, lo que provoca que la transferencia de datos ocurra de manera secuencial, capa por capa, a través de conexiones punto a punto.



**Figura 3**

*Capas del Modelo ISA-95*

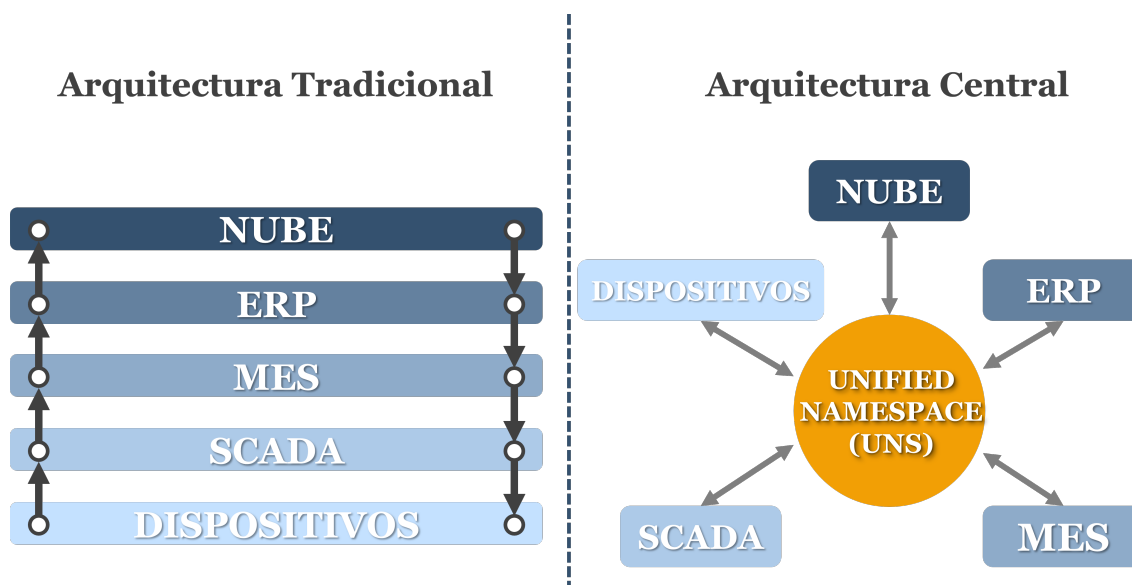
El modelo de integración vertical se caracteriza por su facilidad de implementación, permitiendo una integración empresarial efectiva al proporcionar visualización y monitoreo de datos críticos en cada nivel (Li, Mantravadi, y Moller, 2020). Sin embargo, presenta desafíos en términos de escalabilidad e interoperabilidad (Chen y cols., 2024). La rigidez de la arquitectura tradicional dificulta la incorporación ágil de nuevos dispositivos y sistemas, requiriendo con frecuencia una ingeniería especializada para adaptar los distintos niveles de

la estructura (Alguliyev, Imamverdiyev, y Sukhostat, 2018). Por ejemplo, la integración de nuevos sensores o PLCs para enviar datos a una plataforma en la nube basada en IA puede requerir una adaptación compleja para garantizar la compatibilidad en la extracción y transmisión de datos (Dotoli, Fay, Miśkowitz, y Seatzu, 2019). Asimismo, el modelo de integración vertical presenta limitaciones para la comunicación efectiva entre niveles no adyacentes, lo que genera brechas de coordinación entre las áreas de producción y planificación. A continuación, se presentan algunos de los principales desafíos y desventajas de implementar ISA-95 utilizando el modelo de integración vertical.

- **Costos:** La integración vertical con ISA-95 requiere inversiones significativas en infraestructura, software, hardware, licencias y herramientas de integración. Esto puede representar una barrera para empresas pequeñas o medianas (García-Domínguez, Marcos-Bárcena, Medina-Bulo, y Prades, 2014).
- **Complejidad:** El estándar ISA-95 estructura las operaciones en cinco niveles, lo que implica la integración de sistemas heterogéneos como PLCs, SCADA, MES, ERP y plataformas en la nube. En muchos casos, esta integración se realiza de forma manual mediante conexiones punto a punto, lo que dificulta la escalabilidad y la adaptación a cambios en la producción o expansión del negocio.
- **Sistemas heredados y dependencia de proveedores:** Muchas empresas dependen de sistemas heredados —que aún están en uso a pesar de estar obsoletos— o de sistemas que solo pueden integrarse con dispositivos específicos. La actualización o reemplazo de estos sistemas puede representar un desafío técnico desde el punto de vista de la ingeniería. La incompatibilidad con nuevas tecnologías dificulta la integración vertical.
- **Acceso limitado a los datos:** La cantidad de datos generados por sensores, controladores y dispositivos desde los niveles inferiores hasta los superiores puede ser enorme. Las arquitecturas tradicionales almacenan los datos de forma centralizada con acceso

limitado, lo que puede generar inconsistencias en la información y aumentar el riesgo de errores en la toma de decisiones a nivel empresarial (Listl y cols., 2023).

Basado en lo anterior, se puede concluir que el modelo de integración vertical puede limitar el desarrollo de fábricas inteligentes. Para abordar esta problemática, se ha propuesto una arquitectura centralizada en la que los usuarios de la red se comunican a través de un concentrador de datos central, conocido como Unified Namespace (UNS). La Fig. 4 muestra que el UNS es compatible con la estructura de ISA-95, pero en este caso, adapta cada uno de sus niveles a un modelo radial (Brim y cols., 2023). En la siguiente sección se presentan el marco conceptual y las características del UNS.



**Figura 4**

*Arquitectura Tradicional y Central*

## **Capítulo 2: Unified Namespace: Marco Conceptual**

El concepto de Unified Namespace (UNS) surgió entre 2003 y 2005, cuando Walker Reynolds enfrentó el desafío de desplegar una arquitectura capaz de integrar aproximadamente 3,000 etiquetas de datos provenientes de diversas áreas en una mina de sal en el estado de Nueva York, operando sobre múltiples protocolos de comunicación (W. D. Reynolds, 2023). Su objetivo era establecer una fuente única de verdad, que centralizara todos los datos y proporcionara visibilidad en tiempo real del estado de las operaciones. Para ello, diseñó un Control Logic Gateway (CLG) que permitiera la comunicación entre todos los dispositivos inteligentes dentro de la mina. Este concepto evolucionó con el tiempo, alcanzando un punto de inflexión en 2014, cuando el mismo Reynolds implementó una arquitectura basada en UNS en la plataforma Ignition, en un proyecto que abarcó 14,000 sites, 40,000 dispositivos, 11 millones de etiquetas de datos, 2 millones de alarmas diarias y 2,000 usuarios concurrentes (Trimax Systems, Inc., 2015). La magnitud e impacto de este desarrollo fueron tales que el proyecto recibió en el 2015 el prestigioso Firebrand Award de Inductive Automation (Automation, 2015). Desde entonces, Reynolds se ha consolidado como el principal promotor y pionero del UNS.

Actualmente, la información sobre el UNS proviene en su mayoría de fuentes industriales, como IIoT University y la comunidad 4.0 Solutions, ambas dirigidas por Reynolds. Sin embargo, el UNS sigue en constante desarrollo y aún no cuenta con una definición definitiva y universalmente aceptada. Las principales fuentes bibliográficas que proponen y analizan su marco conceptual incluyen blogs, seminarios web y publicaciones en línea, destacando las contribuciones de autores como David Schultz, Matthew Parris y Walker Reynolds (Manditereza, 2022a, 2022b, 2022c; W. Reynolds, 2024b; Systems, 2024). Recientemente, Péter y Werner presentaron una tesis de maestría bajo la supervisión de la Prof. Charlotta Johnsson en la Universidad de Lund (Suecia), en la que exploraron cómo el UNS podría abordar los desafíos asociados con la gestión fragmentada de datos dentro de las organizaciones, eva-

luando tanto sus beneficios como sus limitaciones (Péter y Werner, 2024).

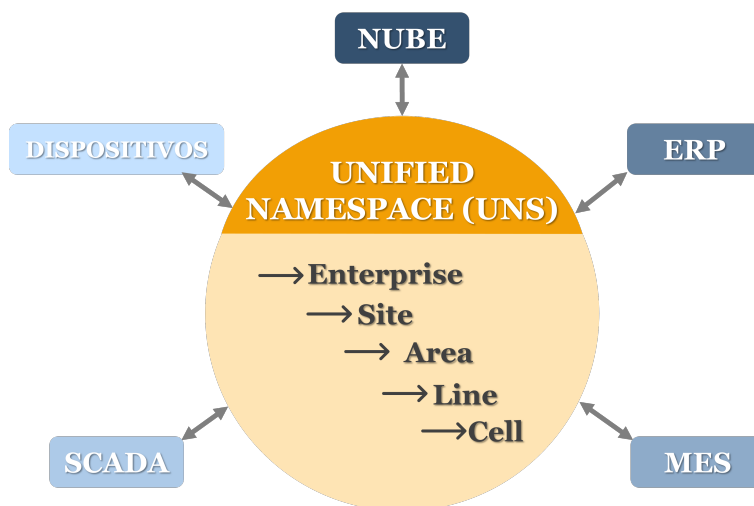
Además, experiencias prácticas documentadas en estudios como (Domínguez-Cid y cols., 2024; Koprov y cols., 2023; Mäule, Kutzler, Leich, Schäfer, y Richter, 2024) destacan implementaciones exitosas de arquitecturas basadas en UNS en entornos industriales reales, lo que refuerza su relevancia y aplicabilidad en la industria.

Las siguientes secciones presentarán el marco conceptual del UNS, abordando preguntas clave como: ¿Qué es el UNS?, ¿Cuáles son las ventajas de un UNS? y ¿Cómo funciona un UNS?

### **¿Que es el UNS?**

El Unified Namespace (UNS) es una arquitectura que organiza todas las entidades, recursos y dispositivos inteligentes dentro de un hub central o broker de datos estructurados. Este hub central se convierte en una fuente única de verdad para toda la información de la organización, proporcionando un marco integral que recopila, organiza y permite el acceso en tiempo real a los datos en todos los niveles empresariales, facilitando así la comunicación entre dispositivos inteligentes (Manditereza, 2022b). En esencia, el UNS representa tanto la estructura del negocio como todos sus eventos en tiempo real, actuando como el espacio donde reside el estado actual de la organización (W. Reynolds, 2024a).

El UNS puede traducir y contextualizar datos, facilitando la integración fluida entre diferentes sistemas y plataformas. Esto permite que los datos generados por dispositivos y sensores de diversas fuentes sean interpretados de manera uniforme. La arquitectura del UNS se basa en el patrón de comunicación publish-subscribe, lo que permite una integración sencilla mediante el protocolo MQTT. Esto posibilita que las organizaciones aprovechen las capacidades de comunicación M2M (Machine-to-Machine) y la gestión en tiempo real de grandes volúmenes de datos. Para integrar el UNS con la ISA-95, se recomienda utilizar el modelo de datos estandarizado (Enterprise/Site/Area/Line/Cell) para mapear la estructura organizacional dentro del data-lake, como se ilustra en la Fig. 5.



**Figura 5**

*Arquitectura Central UNS*

### ¿Cuáles son las Ventajas de un UNS?

La implementación de un Unified Namespace (UNS) ofrece múltiples beneficios para las organizaciones en proceso de transformación digital dentro del contexto de Industria 4.0. A continuación, se presentan las principales ventajas operativas derivadas de la adopción de esta arquitectura:

- **Escalabilidad:** El UNS elimina la necesidad de crear conexiones específicas entre diferentes sistemas, facilitando la integración fluida de nuevos dispositivos y sensores IoT. Esto mejora la escalabilidad del ecosistema de datos empresarial, impulsando la optimización de procesos y promoviendo la eficiencia e integración entre los equipos de trabajo gracias al acceso unificado a la información.
- **Normalización:** Una de las principales ventajas del UNS es su capacidad para normalizar datos provenientes de diversas fuentes y protocolos. Al estandarizar la estructura, el formato y la representación de los datos, el UNS garantiza coherencia y uniformidad en los distintos sistemas y aplicaciones de una arquitectura de integración industrial.

Esta estandarización simplifica el intercambio de datos y asegura que toda la información relevante pueda extraerse de las fuentes organizacionales, independientemente de su formato original.

- **Reducción del tiempo de valor:** El UNS minimiza el tiempo necesario para convertir los datos en información accesible en tiempo real para toda la organización. A diferencia de las arquitecturas tradicionales, donde los datos deben atravesar múltiples capas, el UNS proporciona acceso inmediato a cada nodo de información. Esto reduce los retrasos y permite a las empresas aprovechar oportunidades de manera más eficiente.
- **Seguridad:** El UNS incorpora mecanismos avanzados de control de acceso, garantizando tanto la seguridad de los datos como su pertinencia. Su sistema de permisos basado en roles restringe el acceso de los usuarios únicamente a la información necesaria para sus funciones específicas. Además, el UNS adopta un enfoque edge-driven, donde la comunicación se inicia desde el nodo mediante un único puerto de salida. Este diseño simplifica la gestión de conexiones, reduce los vectores de ataque y mejora significativamente la seguridad del sistema.
- **Agilidad:** Al ofrecer acceso en tiempo real a un flujo de datos unificado, el UNS mejora la agilidad operativa de las organizaciones. Los equipos pueden adaptarse rápidamente a condiciones cambiantes, tomar decisiones basadas en datos y realizar ajustes sin los retrasos asociados a los silos de datos tradicionales. Esta flexibilidad permite a las empresas responder con mayor rapidez a las demandas del mercado, optimizar procesos y mejorar continuamente su rendimiento operativo.

El UNS proporciona escalabilidad, estandarización, reducción del tiempo de valor, mayor seguridad y agilidad, permitiendo una gestión eficiente y segura de la información en diversos sistemas y dispositivos dentro del contexto de Industria 4.0 (Manditereza, 2022c). Estos beneficios favorecen la toma de decisiones basada en información en tiempo real, impactando positivamente la cadena de suministro y a los diferentes stakeholders. Al optimizar las

operaciones y facilitar decisiones basadas en datos, el UNS posiciona a las organizaciones para prosperar en un entorno industrial cada vez más digital e interconectado.

### **¿Cómo funciona un UNS?**

El UNS recopila datos de diversas fuentes, incluidos sensores, equipos, sistemas ERP, MES y SCADA, integrándolos en un marco unificado. Este entorno se basa en protocolos y estándares de comunicación, como MQTT y Sparkplug, respectivamente, para garantizar una transmisión de datos fiable (Manditereza, 2022a). Al mantener un entorno de datos en tiempo real, el UNS permite a las empresas monitorear, controlar y optimizar sus operaciones de manera continua.

Los dispositivos inteligentes dentro de una organización se comunican entre sí utilizando un protocolo de comunicación IIoT, cumpliendo con los requisitos técnicos mínimos, los cuales definen las condiciones esenciales que deben cumplir los dispositivos para garantizar una integración segura, eficiente y escalable dentro de una arquitectura industrial digital. Incluyen aspectos como la interoperabilidad, mediante el uso de protocolos estándar; la seguridad, a través de mecanismos de autenticación, cifrado y control de acceso; y la conectividad, asegurando compatibilidad con redes industriales modernas y tolerancia a fallos.

En el caso que un dispositivo no sea compatible con estos requisitos, debe integrarse mediante un **Gateway** que convierta el protocolo soportado por el dispositivo al protocolo adoptado por la organización que cumpla los requisitos mínimos (Systems, 2024). Este gateway publica las actualizaciones de datos en el UNS, utilizando tópicos predefinidos dentro del namespace, siguiendo una jerarquía semántica para garantizar un acceso intuitivo y estructurado a la información.

En protocolos como MQTT, la estructura jerárquica de los tópicos permite la implementación de un UNS, lo que optimiza la distribución y el acceso a los datos dentro de una organización. Para garantizar esta eficiencia, es fundamental mantener una estructura de tópicos coherente y predecible, alineada con modelos como ISA-95, que propone una jerarquía organizacional en distintos niveles (Manditereza, 2023) :

Enterprise > Site > Area > Line > Cell

Sin embargo, MQTT por sí solo no define un estándar sobre cómo deben estructurarse los tópicos, lo que puede generar inconsistencias e ineficiencias en la implementación del UNS. En este contexto, Sparkplug surge como una extensión que estandariza el namespace de tópicos en mensajerías MQTT, proporcionando una estructura lógica que facilita la identificación, acceso y gestión de dispositivos a través de la comunicación.

El marco de trabajo Sparkplug establece diferentes estándares para la transmisión de datos en aplicaciones que utilizan el protocolo MQTT. En su versión más reciente, Sparkplug B, se define una estructura lógica donde los nodos de borde (**Edge of Network - EoN**) se agrupan en un "*group\_id*", asignando un identificador único a cada nodo dentro del grupo. Además, cada nodo gestiona múltiples dispositivos individuales, identificados mediante un "*device\_id*". Esta estructura se representa de la siguiente manera:

```
spBv1.0/<group_id>/<message_type>/<edgenode_id>/<device_id>
```

La estructura de Sparkplug B inicia con el elemento raíz *spBv1.0*, seguido del *<groupid>*, que agrupa lógicamente los nodos EoN dentro del servidor MQTT. Luego, el elemento *<message\_type>* clasifica el tipo de mensaje, indicando cómo debe manejarse su carga útil (payload). El *<edgenode\_id>* identifica el nodo EoN dentro de la infraestructura, y finalmente, el *<device\_id>* representa el dispositivo conectado.

El formato de estos elementos, como *<group\_id>*, debe ser una cadena alfanumérica válida en UTF-8, con la restricción de no incluir los caracteres reservados '+' (signo de más), '/' (barra) y '#' (numeral). Además, para optimizar el uso del ancho de banda, se recomienda que los identificadores sean lo más descriptivos y compactos posible.

Sparkplug B limita la cantidad de elementos en la jerarquía de tópicos, lo que puede ser insuficiente para representar una estructura organizativa detallada basada en ISA-95. Para abordar esta limitación, la comunidad de Industria 4.0 ha desarrollado enfoques alternativos para diseñar UNS más flexibles, capaces de gestionar jerarquías más extensas y complejas.

A la fecha, los principales métodos de diseño para la implementación de UNS han sido propuestos por **David Schultz** y **Matthew Parris**. A continuación, se describen las principales características de estos enfoques.

### Capítulo 3: Técnicas de Diseño de UNS

En este capítulo se presentan y se definen los métodos de diseño UNS encontrados en el estado del arte, junto con sus aplicaciones y diagramas de conexión para su posterior implementación.

#### Método de Schultz

El método Schultz propone la construcción de un UNS sin depender de un único broker MQTT. En su lugar, se despliegan múltiples brokers en diferentes niveles de la jerarquía (*Enterprise, Site, Area, Line, Cell*) para facilitar la comunicación mediante UNS locales. Posteriormente, una plataforma IIoT puede agregar estas instancias de UNS locales y republicar los datos, formando así un UNS unificado a nivel empresarial (Manditereza, 2022b). Por ejemplo, en una organización, cada área tiene su propio broker, donde los dispositivos publican datos utilizando un namespace de tópicos predefinido. Luego, una plataforma IIoT consolida la información de cada área y la publica en el broker del nivel superior (por ejemplo, el nivel de sitio). Una representación esquemática de este método se muestra en la Fig. 6.

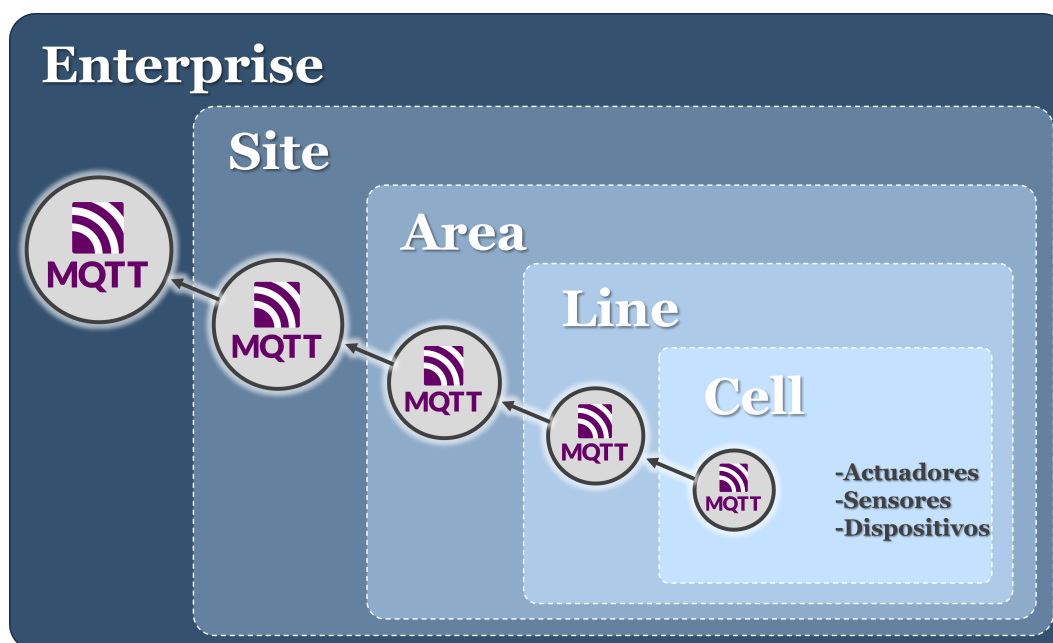


Figura 6

Esquema de la estructura del método Schultz

Una de las principales ventajas del método Schultz es que permite la creación de un UNS con una estructura jerárquica, sin las limitaciones impuestas por la cantidad de elementos en los tópicos de Sparkplug. Esta flexibilidad proporciona mayor versatilidad en el namespace, permitiendo la implementación de múltiples brokers en diferentes niveles. La conexión de los dispositivos a su broker local designado se simplifica, evitando conflictos con otros brokers y mejorando la eficiencia operativa.

La integración de diversas plataformas IIoT, mediante la interconexión de instancias locales, contribuye a una mayor comprensión de la estructura jerárquica del UNS. Esto facilita la identificación del nivel correspondiente a cada UNS local y optimiza la gestión del flujo de datos en función de los requerimientos específicos de cada línea de producción. Adicionalmente, la separación entre los UNS locales y el UNS a nivel empresarial proporciona una mayor resiliencia. En caso de una falla en el UNS empresarial (remoto), las operaciones pueden continuar sin interrupciones, ya que dependen de los UNS locales.

Sin embargo, aunque este enfoque elimina delimitadores y fomenta la flexibilidad en cada área de trabajo, también incrementa la complejidad de la integración de las plataformas IIoT. Conectar los UNS locales al UNS empresarial puede generar mayores costos y requerir recursos computacionales adicionales para gestionar eficazmente la arquitectura distribuida.

### **Método de Parris**

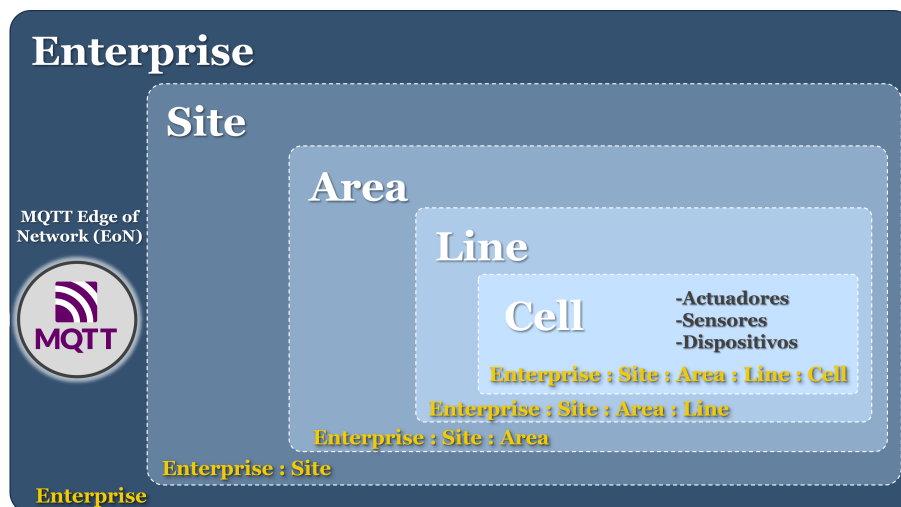
El método Parris consiste en agrupar toda la estructura empresarial dentro del elemento `<group_id>`, utilizando delimitadores para separar las categorías (Systems, 2024). Por ejemplo, si se emplea el carácter ":" como delimitador, el namespace de los tópicos en Sparkplug tendría el siguiente formato:

$$spBv1.0/Plant:Area:Line:Cell/DDATA/<edgenode\_id>$$

Donde "DDATA" corresponde al `<message_type>`. Como otra alternativa, el formato en el que `Line:Cell` corresponde a `<edgenode_id>` del namespace, se puede utilizar de la siguiente manera:

*spBv1.0/Plant:Area/DDATA/Line:Cell/<device\_id>*

Sin embargo, se recomienda evitar incluir el elemento *<edgenodeid>* dentro de la jerarquía ISA-95, ya que debe identificarse de manera única en el MQTT EoN (Edge of Network) dentro de la arquitectura para prevenir conflictos de redundancia. La Fig. 7 muestra un diagrama esquemático del funcionamiento del método Parris.



**Figura 7**

*Esquema de la estructura del método de Parris*

Este método garantiza un *"group\_id"* definido de manera única, lo que permite la transferencia de datos sin problemas entre diferentes brokers MQTT a través de un **Bridge** (mecanismo que permite interconectar dos brokers para reenviar mensajes entre ellos de manera automática) y así evitar conflictos en los topics. De este modo, se elimina la necesidad de una plataforma IIoT intermedia que decodifique los datos en niveles superiores de la jerarquía de temas (Systems, 2024).

Además, la incorporación de delimitadores en el *"group\_id"* mejora la estandarización, asegurando una interpretación coherente de los temas en toda la organización. Sin esta estandarización, cada aplicación requeriría programación personalizada para alinearse con la jerarquía de temas específica del broker al que está conectada, lo que aumentaría el esfuerzo de desarrollo y la complejidad de integración.

Otra ventaja clave de este método es la posibilidad de implementar un solo broker para todo el UNS, lo que simplifica el despliegue y reduce los costos operativos de software. Esto permite monitorear desde una única consola el estado de todos los mensajes enviados al broker central, facilitando la detección de fallos en la transmisión. La arquitectura centralizada consolida la información de todos los nodos y estructuras de manera estandarizada, permitiendo que un solo broker integre todos los niveles del UNS. Esta solución proporciona una vía de entrada rentable, ideal para organizaciones en crecimiento.

Aunque el método Parris es relativamente sencillo de implementar, presenta un desafío técnico cuando una plataforma IIoT recibe un mensaje Sparkplug al suscribirse a un tema general como "spBv1.0/#". En este caso, la plataforma debe extraer el GroupID y re-publicar el mensaje en temas MQTT específicos. Este proceso permite reconstruir la estructura original, convirtiendo la organización plana de los datos en una jerarquía más detallada.

A continuación, se presenta un ejemplo de cómo un mensaje Sparkplug se convierte en una estructura plana para implementar el método Parris:

$$spBv1.0/Plant:Area:Line:Cell/\# \rightarrow Plant:Area:Line:Cell/\#$$

En este caso, al suscribirse a un tema general como "spBv1.0/Plant:Area:Line:Cell/#", donde "#" actúa como un marcador global para recibir todos los sub-temas bajo el nivel especificado, la plataforma IIoT debe descomponerlo y re-publicarlo en temas específicos, como "Plant/Area/Line/Cell/#".

La implementación de estos métodos se ejemplificará mediante un caso de aplicación. El propósito de esto es proporcionar claridad sobre las consideraciones clave a tener en cuenta para su implementación práctica.

## Capítulo 4: Implementación de Técnicas UNS en la Plataforma IIoT

La implementación del UNS en la plataforma de entrenamiento IIoT se basa en la aplicación de técnicas de diseño previamente estudiadas en un entorno académico. Esta aplicación integra sistemas y dispositivos mediante el protocolo MQTT Sparkplug B, estableciendo un modelo jerárquico y estructurado de comunicación orientado a la gestión eficiente de datos. Este entorno de entrenamiento simula escenarios operacionales reales de la industria permitiendo a los usuarios interactuar directamente con dispositivos IIoT y validar los principios del UNS en tiempo real.

### Descripción de la Plataforma IIoT

La construcción de la plataforma IIoT para entrenamiento en tecnologías de la Industria 4.0 surge como resultado de dos proyectos de investigación estratégicos. El primero, desarrollado bajo la Convocatoria interna VIE en 2021, llevó por título "Plataforma IIoT para formación de profesionales en tecnologías de la cuarta revolución industrial" se ejecutó entre noviembre de 2021 y noviembre de 2023. Posteriormente, el proyecto "Diseño, desarrollo e implementación de una plataforma IIoT para formación de profesionales en tecnologías de la cuarta revolución industrial", financiado por la Convocatoria 890 de Minciencias, inició en abril de 2023 y se encuentra actualmente en desarrollo.

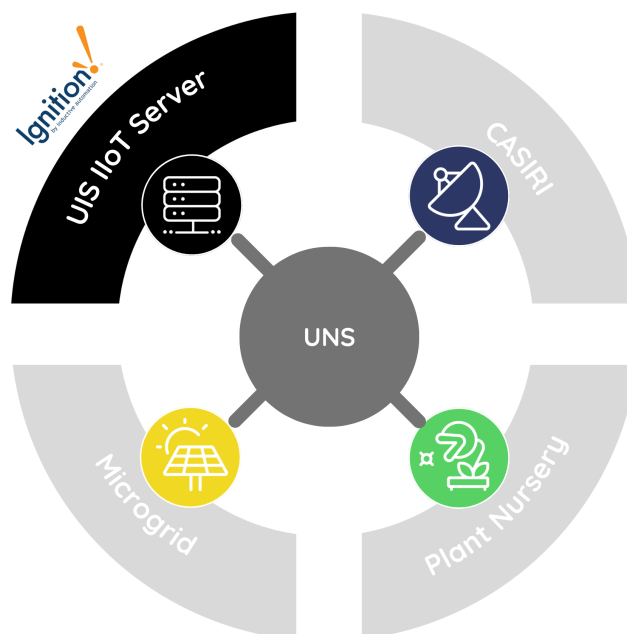
### *Dispositivos y Nodos Industriales:*

La plataforma se compone de tres casos de estudio, cada uno asociado a un grupo de investigación y con aplicaciones específicas en entornos industriales y académicos:

- **VIVERO:** Pertenece al Grupo de Investigación CENIVAM, donde se monitorean condiciones ambientales dentro de un invernadero mediante un PLC Opto22 - Groov RIO conectado a un termohigrómetro que captura datos de temperatura y humedad. Estos datos son procesados por un servidor Ignition Edge para ser transmitidos.
- **CASIRI:** Pertenece a los Grupos de Investigación RADIOGIS y CEMOS, consiste en una estación de radioastronomía equipada con sensores ambientales que miden tempe-

ratura, humedad, presión atmosférica, velocidad y dirección del viento; además de una cámara multi-espectral para captura de imágenes espectrales y un dispositivo de monitoreo del espectro radioeléctrico. Todos estos dispositivos están integrados mediante un servidor que envía los datos al servidor central.

- **MICRORRED:** Pertenece al Grupo de Investigación GISEL, ubicada en el Laboratorio de Investigación Eléctrica (LIE) en la sede UIS Guatiguara. Este sistema monitorea parámetros críticos de la red eléctrica como tensión RMS, corriente RMS, factor de potencia y diferentes tipos de potencia (aparente, activa y reactiva) mediante un PLC Opto22 - Groov EPIC que utiliza un servidor de Ignition Edge para prestar servicios a los usuarios.



**Figura 8**

*Esquema de la Plataforma de Entrenamiento IIoT*

Para centralizar la comunicación y visualización de datos en tiempo real de todos los dispositivos (casos de estudio) que actualmente se conectan de forma distribuida a través de

internet, se propone implementar un UNS que sea el nodo central del sistema representado en la Figura 8. Allí se debe integrar un broker MQTT escalable a nivel de Enterprise mediante un servidor IIoT, con el objetivo de normalizar los datos, unificar protocolos y gestionar usuarios, permitiendo así una plataforma única para agregar, procesar y visualizar toda la información de los casos de estudio.

A partir de la conexión y configuración de los casos de estudio, es posible diseñar un UNS empleando las diversas técnicas de diseño previamente estudiadas y analizadas. Para cada caso, se define un *namespace* estructurado, asegurando una organización eficiente de los datos y su interoperabilidad dentro del sistema.

Ambas metodologías se implementan utilizando el protocolo de comunicación MQTT Sparkplug B, lo que permite la integración de los casos de estudio al servidor principal de manera independiente. A continuación, se detalla el proceso de configuración en la plataforma.

### **Implementación del Método Schultz**

Para la implementación de la técnica de Schultz, se establecen los niveles jerárquicos del UNS, organizando la arquitectura del sistema en función de los *sites*, *areas*, *lines* y *cells* como se adquieren los datos. Esta estructura permite una gestión eficiente de la información para la integración de los dispositivos en la plataforma de entrenamiento IIoT.

El sistema está compuesto por dos *sites* principales: Central y Guatiguará. En el *site* Central, se encuentran dos *areas* diferenciadas. La primera, Cenivam, comprende la *line* Vivero y la *cell* GroovRio, donde este último es el PLC encargado de la adquisición de datos a través del termo-higrómetro. La segunda *area*, Radiogis, está conformada por la *line* Casiri, las *cell* Cámara, responsables de la captura de imágenes espectrales, y la *cell* Estación Ambiental, que recopila información sobre variables ambientales.

Por otro lado, en el *site* Guatiguará, se encuentra el *area* LIE (Laboratorio de Integración Energética), la cual alberga la *line* Microrred y la *cell* GroovEpic, un PLC encargado de la adquisición de datos eléctricos de la Microrred.

### Arquitectura de Diseño

Para implementar esta técnica en la aplicación de Ignition basada en MQTT Sparkplug B, se diseña la arquitectura mostrada en la Fig. 9, en donde se escala desde el nivel de *area*, al nivel de *site* y finalmente al nivel de *Enterprise*.

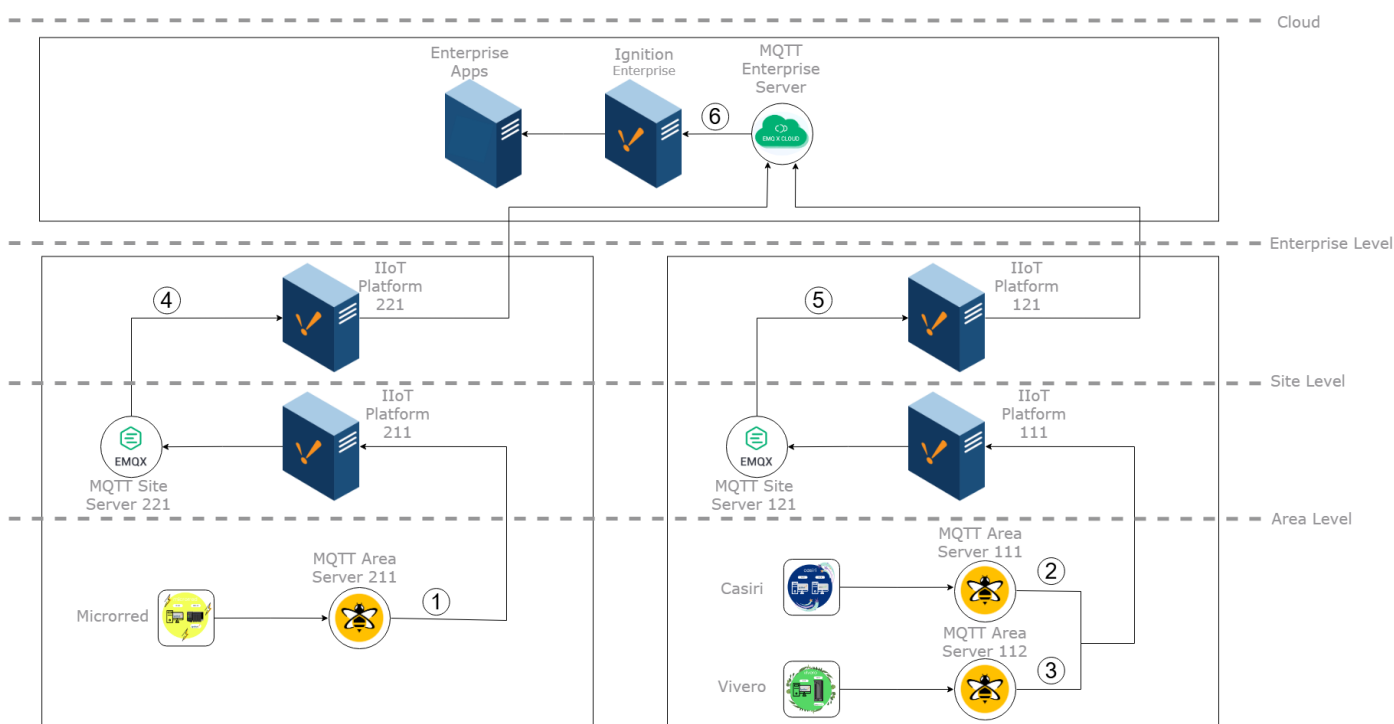


Figura 9

Esquema de la Arquitectura UNS con el Método de Schultz en la Plataforma IIoT

Los dispositivos IIoT de cada uno de los casos de estudio de la plataforma IIoT, se conectan a brokers de HiveMQ (MQTT Area Server 111, MQTT Area Server 112 y MQTT Site Server 211), que actúan como concentradores de datos para áreas específicas. Estos brokers recogen los datos y los envían a las Plataformas IIoT 111 y 211 para organizar los datos en un UNS de *area* para cada caso, con la siguiente estructura: (①, ② y ③ en Fig. 9)

```

Area: LIE
├─ Line: Microrred
│   └─ Cell: GroovEpic
    
```

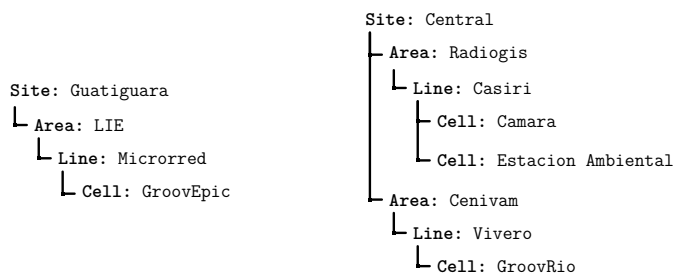
```

Area: Radiogis
├─ Line: Casiri
│   ├── Cell: Camara
│   └─ Cell: Estacion Ambiental
    
```

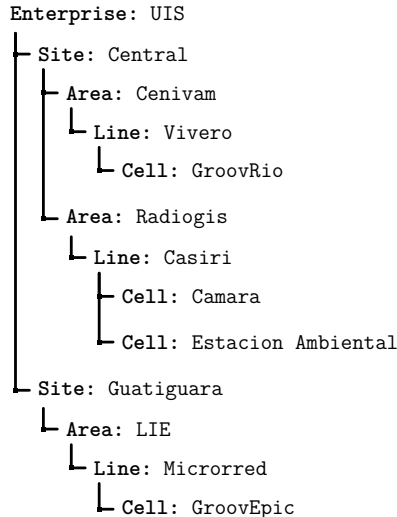
```

Area: Cenivam
├─ Line: Vivero
│   └─ Cell: GroovRio
    
```

Secuencialmente, los UNS de *area* son enviados a un broker local (MQTT Site Server 121 y 221) para recolectar la información por *site* y enviarlas a las Plataformas IIoT 121 y 221. Estas procesan los datos contextualizados y organizan los datos en un UNS de *site* con la siguiente estructura:(④ y ⑤ en Fig. 9))



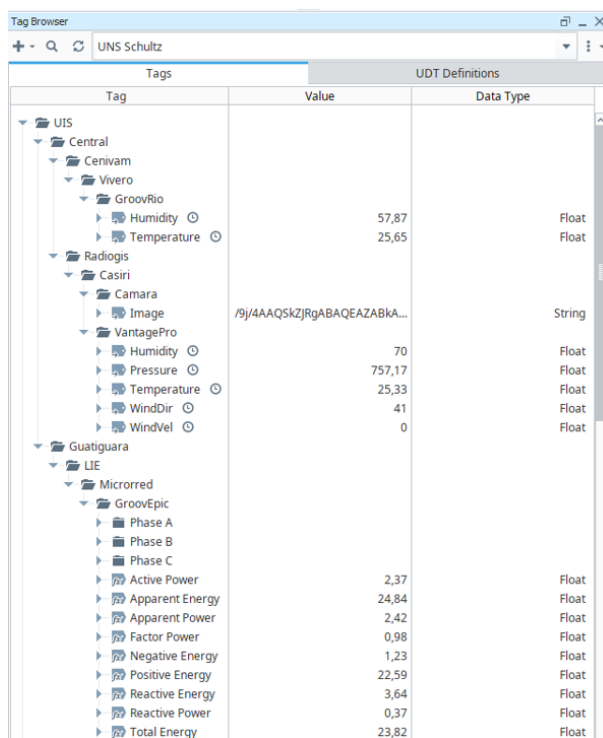
Finalmente, con la estructura definida para cada uno de los *site*, se consolidan todos los namespaces distribuidos en un broker en la nube (MQTT Enterprise Server). Este recibe todas las áreas de la infraestructura y organiza el UNS a nivel de *Enterprise* en el Ignition Enterprise, con el siguiente modelo jerárquico.(⑥ en Fig. 9)



La nube opera como la capa de integración global, permitiendo analítica avanzada y gestión remota de cada uno de los niveles de la jerarquía, además permite la conexión de diferentes aplicaciones al nivel de *Enterprise* para monitorear y controlar los datos. Esta estructura multi-capas mantiene la autonomía local, evita los diferentes silos de datos y permite la reconexión automática de los datos ante una falla en el sistema en cualquier nivel del UNS.

### Implementación en Ignition

Una vez diseñada e implementada la arquitectura del UNS utilizando el método de Schultz, es posible establecer conexión con el servidor Enterprise MQTT a través de la aplicación Ignition. Esta conexión permite visualizar de forma estructurada y en tiempo real toda la jerarquía del UNS previamente definida, facilitando el acceso a los datos organizados por *sites*, *areas*, *lines* y *cells*. La plataforma Ignition actúa como cliente MQTT compatible con Sparkplug B, lo que permite la exploración y supervisión directa del modelo UNS. A continuación, se presenta una visualización dentro del entorno de Ignition.



Tag	Value	Data Type
UIS		
Central		
Cenivam		
Vivero		
GroovRio		
Humidity	57,87	Float
Temperature	25,65	Float
Radiogis		
Casiri		
Camara		
Image	/9j/4AAQSKZJRgABAQEAZABKA...	String
VantagePro		
Humidity	70	Float
Pressure	757,17	Float
Temperature	25,33	Float
WindDir	41	Float
WindVel	0	Float
Guatiguara		
LIE		
Microrred		
GroovEpic		
Phase A		
Phase B		
Phase C		
Active Power	2,37	Float
Apparent Energy	24,84	Float
Apparent Power	2,42	Float
Factor Power	0,98	Float
Negative Energy	1,23	Float
Positive Energy	22,59	Float
Reactive Energy	3,64	Float
Reactive Power	0,37	Float
Total Energy	23,82	Float

Figura 10

#### UNS Schultz en Ignition

A través de esta estructura jerárquica es posible acceder a los datos de toda la infraestructura, desde el nivel *Enterprise* hasta el nivel de *cell*, reflejando con precisión los distintos niveles definidos en el UNS. Cada uno de estos niveles representa una instancia clara dentro de la jerarquía del sistema, lo que garantiza una organización coherente y uniforme de los dispositivos y sensores conectados en las diferentes áreas.

Esta estandarización en la representación y ubicación de los dispositivos permite lograr una alta escalabilidad e interoperabilidad sin necesidad de modificar la arquitectura base. De esta manera, cualquier nuevo caso de estudio o integración de un dispositivo adicional puede ser incorporado directamente en el área correspondiente, y automáticamente será reflejado en el UNS general, manteniendo la integridad y consistencia del sistema.

### **Implementación del Método Parris**

Para la aplicación de la técnica de Parris en la plataforma IIoT, se define la jerarquía del UNS replicando la estructura utilizada previamente, basada en los mismos *sites*, *areas*, *líneas* y *cells* de adquisición de datos de la plataforma. La arquitectura contempla nuevamente dos *sites* principales: Central, que incluye las *areas* Cenivam, con la *línea* Vivero y la *cell* GroovRio, además del Radiogis, con la *línea* Casiri y las *cells* Cámara y Estación Ambiental; por otro lado, se tiene el *site* Guatiguará, que incorpora el *area* LIE, compuesta por la *línea* Microrred y la *cell* GroovEpic.

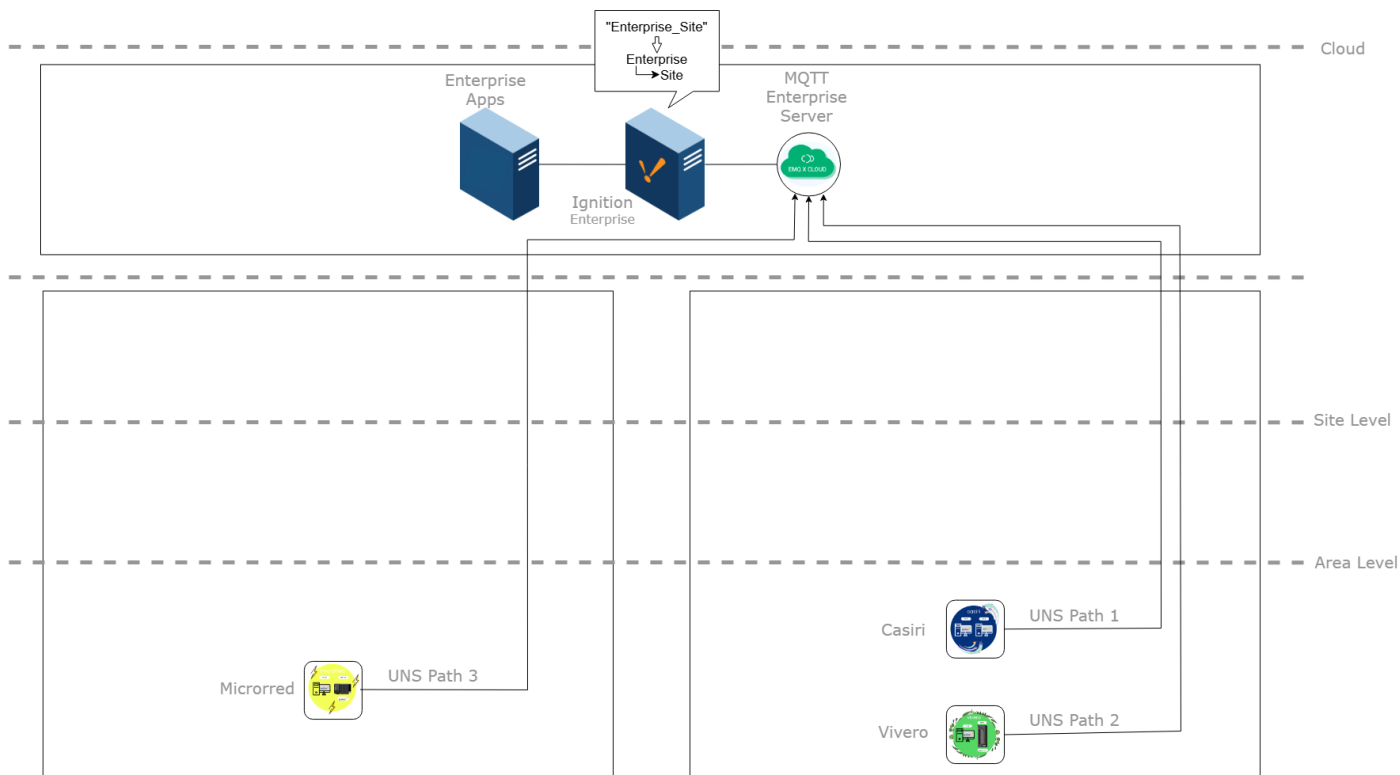
### **Arquitectura de Diseño**

La arquitectura basada en el método de Parris presenta un enfoque simplificado y estandarizado para implementar un UNS, utilizando un broker MQTT centralizado como columna vertebral de la comunicación. A diferencia del modelo jerárquico de Schultz, aquí todos los dispositivos (Sensor 1, Sensor 2, Sensor 3) de los 3 casos de estudios y las aplicaciones (como Ignition Enterprise) se conectan directamente al mismo MQTT Broker en la nube, eliminando capas intermedias de servidores locales. En la Fig. 11 se representa como cada dispositivo publica datos en rutas estandarizadas (UNS Path) para cada uno de los casos, que definen un namespace unificado y accesible para toda la organización. Algunos ejemplos de UNS Path utilizando el delimitador "\_" son los siguientes:

UNS Path 1: *UIS\_Central/Radiogis\_Casiri\_Estacion Ambiental/<device\_id>*

UNS Path 2: *UIS\_Central/Cenivam\_Vivero\_GroovRio/<device\_id>*

UNS Path 3: *UIS\_Guatiguara/LIE\_Microrred\_GroovEpic/<device\_id>*



**Figura 11**

*Esquema de la Arquitectura UNS con el Metodo de Parris en la Plataforma IIoT*

Las aplicaciones acceden directamente al broker central para visualizar, analizar y controlar los datos en tiempo real. De manera similar al método de Schultz, la jerarquía del UNS diseñada desde la plataforma Ignition Enterprise se representa con la siguiente estructura.

```

Enterprise: UIS
├── Site: Central
│   ├── Area: Cenivam
│   │   ├── Line: Vivero
│   │   │   └── Cell: GroovRio
│   │   └── Area: Radiogis
│   │       ├── Line: Casiri
│   │       │   ├── Cell: Camara
│   │       │   └── Cell: Estacion Ambiental
│   └── Site: Guatiguara
│       └── Area: LIE
│           ├── Line: Microrred
│           │   └── Cell: GroovEpic
    
```

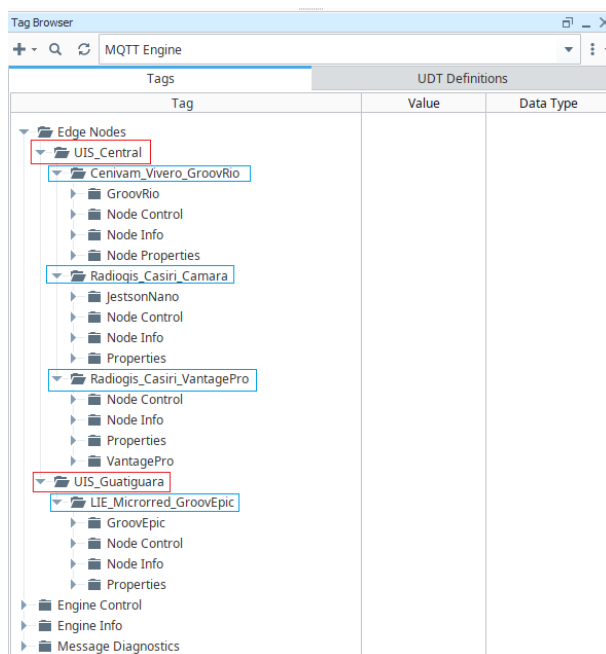
El método de Parris destaca por su simplicidad y flexibilidad, ideal para entornos donde la agilidad y la conectividad directa son prioritarias. Al eliminar servidores intermedios, se reduce el mantenimiento y se acelera la implementación, aunque requiere un broker MQTT robusto y una red confiable para manejar todo el tráfico de datos de manera eficiente.

### ***Implementación en Ignition***

Una vez implementada la arquitectura del UNS, se establece la conexión con el broker MQTT centralizado, el cual contiene la estructura completa del sistema. A través de la aplicación Ignition, es posible visualizar cada uno de los “Paths” correspondientes a los casos de estudio, estructurados según el delimitador definido. Esta visualización se realiza conforme a la especificación MQTT Sparkplug B, adoptando la siguiente organización jerárquica:

*<group\_id>: Enterprise\_Site*  
*<edgenode\_id>: Area\_Line\_Cell*

En la Fig 12 se puede observar como se reciben los topics en Ignition según la especificación del protocolo y la definición de cada uno de los nodos.



**Figura 12**

*Método de Parris en el Módulo MQTT Engine en Ignition*

La aplicación permite recibir y procesar los “topics” suscritos, y como se muestra en la Fig. 13, Ignition interpreta los delimitadores “\_” para dividirlos en sub-topics, construyendo así la jerarquía del UNS definida previamente.

Tags		UDT Definitions	
Tag	Value	Data Type	
UIS			
Central			
Cenivam			
Vivero			
GroovRio			
GroovRio			
OptoMMP			
Modules			
Channels			
Humedad Ambiente			
Value	57,98	Float	
Humedad en Sustrato 1			
Humedad en Sustrato 2			
Humedad en Sustrato 3			
Rele Valvula 1			
Rele Valvula 2			
Rele Valvula 3			
Temperatura Ambiente			
Value	25,78	Float	
Radiogis			
Casiri			
Camara			
JestsonNano			
Camera			
Images	/9j/4AAQSkzJRgA...	String	
Device Info			
Properties			
Node Control			
Node Info			
Properties			
VantagePro			
Node Control			
Node Info			
Properties			
VantagePro			
Device Info			
Properties			
Humidity	69	Float	
Pressure	756,59	Float	
Temperature	25,56	Float	
WindDir	42	Float	
WindVel	0	Float	
Guatiguara			
LIE			
Microrred			
GroovEpic			
GroovEpic			
Device Info			
Phase A			
Phase B			
Phase C			
Properties			
Active Power	2,11	Float	
Apparent Energy	26,35	Float	
Apparent Power	2,19	Float	
Factor Power	0,96	Float	
Negative Energy	1,25	Float	
Positive Energy	24,92	Float	
Reactive Energy	3,43	Float	
Reactive Power	0,42	Float	
Total Energy	26,17	Float	
Node Control			
Node Info			
Properties			

Figura 13

Método de Parris Estructurado en Ignition

## Aplicación Ignition Plataforma IIoT

La plataforma Ignition actúa como interfaz de visualización dentro de la arquitectura IIoT basada en Unified Namespace, proporcionando al usuario final acceso estructurado a los datos del sistema. En esta etapa, se detalla el diseño del *Front-End* de la aplicación, incluyendo la disposición de vistas, widgets e indicadores interactivos que permiten la supervisión, análisis y control operativo en tiempo real de forma eficiente e intuitiva.

La vista principal de la aplicación corresponde al Home, como se muestra en la Figura 14. Esta pantalla centraliza el acceso a los distintos módulos de monitoreo, permitiendo navegar de forma intuitiva entre los distintos sites y componentes del sistema.



Figura 14

*Vista del Home en la aplicación de Ignition*

La Figura 15 muestra la visualización correspondiente a Casiri, dentro del área Radiogis del site Central. En esta vista, se monitorean variables ambientales capturadas por la Estación Ambiental y se obtienen las imágenes capturadas por la Cámara espectral, presentando los datos en tiempo real mediante gráficos e indicadores analógicos y datos históricos.



Figura 15

Vista de Casiri en la aplicación de Ignition

En la Figura 16, se ilustra la vista de la Microrred, ubicada en el área LIE del site Guatiguará. Esta vista permite observar el comportamiento eléctrico de cada una de las fases de la Microrred que permite leer el PLC GroovEpic, representando variables claves del sistema.

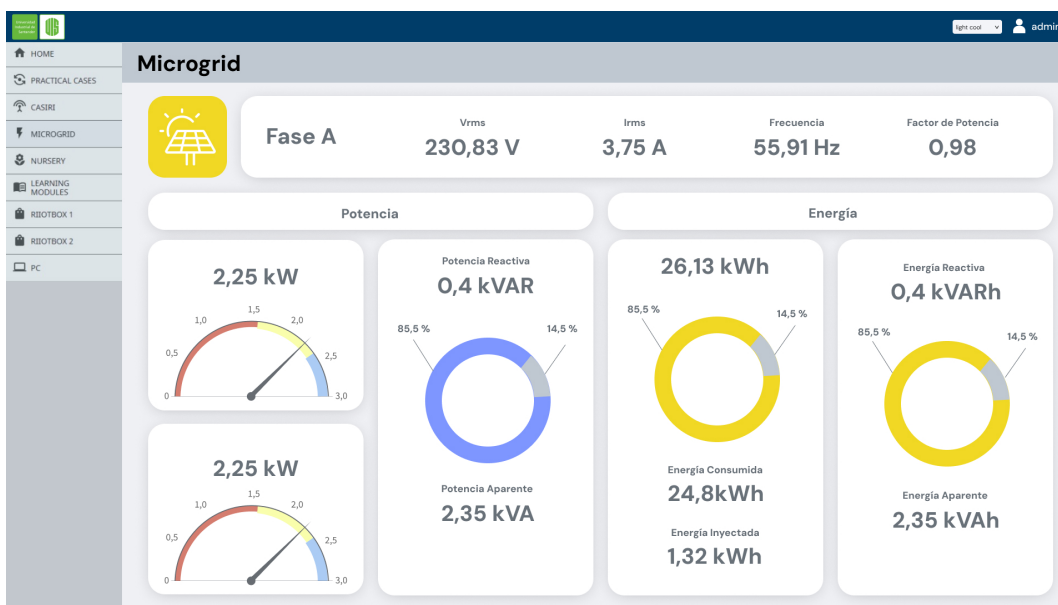
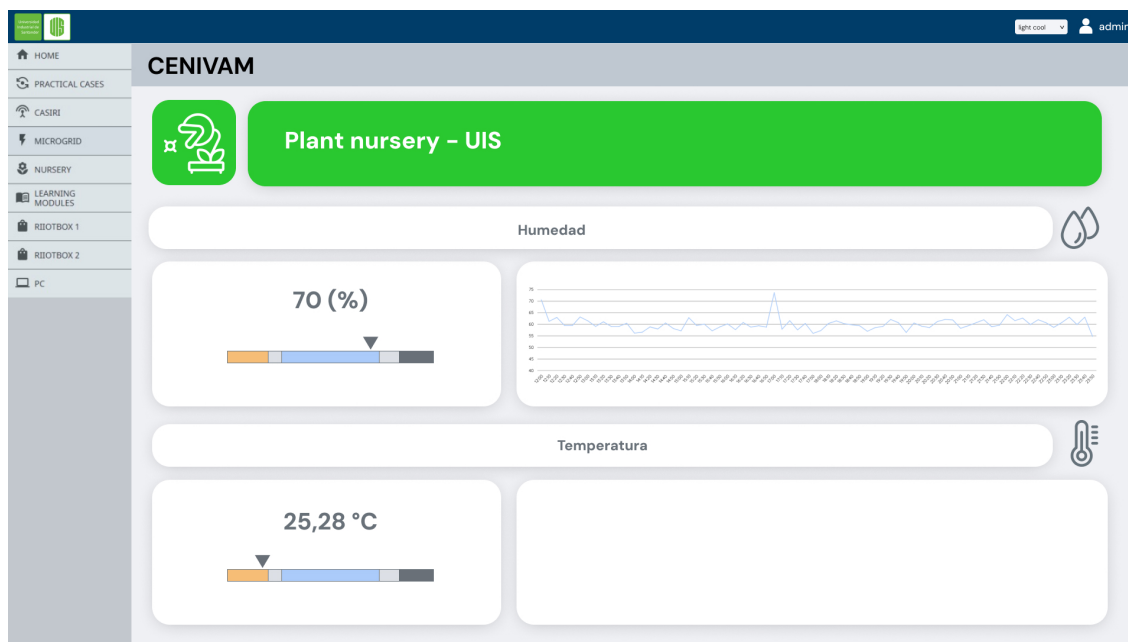


Figura 16

Vista de la Microrred en la aplicación de Ignition

Finalmente, la Figura 17 corresponde a la vista del Vivero, parte del área Cenivam del site Central. En esta interfaz se visualizan parámetros ambientales recolectados por el termohigrómetro a través del PLC GroovRio, permitiendo una supervisión detallada de las condiciones del entorno y un historial de cada una de ellas.



**Figura 17**

*Vista del Vivero en la aplicación de Ignition*

Estas interfaces gráficas están diseñadas para ofrecer una interacción eficiente y en tiempo real entre el usuario y el sistema, permitiendo la supervisión y análisis de los datos mediante una visualización estructurada según la jerarquía definida en el UNS. La presentación contextualiza a nivel general la información proveniente de cada uno de los dispositivos integrados en cada caso de estudio dentro de la plataforma de entrenamiento IIoT.

Adicionalmente, la aplicación incorpora un mecanismo de acceso seguro basado en credenciales, lo que permite gestionar permisos según el perfil de cada usuario. La navegación se organiza mediante un árbol jerárquico que facilita el ingreso personalizado a cada una de las vistas disponibles, asegurando una experiencia flexible, escalable y orientada al rol del operador dentro del entorno industrial.

## Capítulo 5: Análisis Comparativo de las Técnicas Implementadas

En esta sección se presenta un análisis comparativo de las técnicas de diseño de UNS implementadas en la plataforma de entrenamiento IIoT con los métodos propuestos por Schultz y Parris. Ambos enfoques han sido desarrollados bajo la misma plataforma IIoT, utilizando el protocolo MQTT Sparkplug B, lo que permite evaluar de manera objetiva las diferencias y ventajas que ofrece cada uno en términos de métricas de rendimiento dentro de un entorno industrial y académico que se definirán durante el desarrollo de esta sección.

El objetivo principal de este análisis es identificar las fortalezas y limitaciones de cada técnica en la representación de la arquitectura UNS, considerando métricas de desempeño cuantitativos y cualitativos como se muestra en la tabla 1. Allí se detallan métricas como cantidad de mensajes, autenticaciones, clientes, tópicos, además de características como la resiliencia de la arquitectura, la facilidad de integración de nuevos dispositivos y la flexibilidad para adaptarse a distintos sistemas y plataformas IIoT.

Tipo	Métrica	Tipo	Métrica
Cuantitativas	Gestión de Tópicos y Sesiones MQTT	Cualitativas	Integración
	Gestión de Autenticaciones y Autorizaciones		Resiliencia
	Flujo de Datos		Flexibilidad
	Gestión de Clientes		Seguridad
	Gestión de Mensajes		Implementación
	Actividad de Paquetes		Escalabilidad

**Tabla 1**

### *Clasificación de Métricas de Desempeño*

Las métricas de evaluación fueron seleccionadas con base en su capacidad de medición, su aplicabilidad dentro de la plataforma y el impacto que generan en ella. Además, se consideró la experiencia previa en su uso y el contexto en el que se implementan, lo que permitió identificar el enfoque más adecuado según los requerimientos específicos.

## **Métricas Cualitativas de Desempeño**

Las métricas cualitativas seleccionadas para el análisis comparativo responden a criterios fundamentales que determinan la viabilidad, robustez y sostenibilidad de una arquitectura IIoT en entornos industriales complejos. Estas métricas permiten evaluar el esfuerzo requerido en el proceso de desarrollo e integración, además de la adaptabilidad del sistema ante fallos, la facilidad para incorporar nuevos componentes, la compatibilidad con tecnologías existentes, el cumplimiento de requisitos de seguridad y la capacidad de crecimiento estructurado del sistema. En este sentido, estas dimensiones representan aspectos críticos que influyen directamente en la eficiencia operativa y en la toma de decisiones al momento de seleccionar una arquitectura IIoT adecuada para entornos reales de despliegue.

La elección de las métricas de capacidad de implementación, resiliencia de la arquitectura, flexibilidad, seguridad, facilidad de integración y escalabilidad obedece a su capacidad de capturar de manera integral las fortalezas y limitaciones de cada técnica desde una perspectiva técnica, operativa y estratégica. Cada una de estas dimensiones ofrece una visión complementaria sobre el comportamiento del sistema bajo distintas condiciones de operación, lo que permite establecer una comparación estructurada entre ambas técnicas. A continuación, se detalla el análisis individual de cada métrica, indicando para cada caso cuál de las dos técnicas presenta mayores ventajas.

### ***Integración***

La técnica de Parris se distingue por su enfoque directo en la construcción del UNS, donde cada uno de los Path correspondientes a la jerarquía definida se configura individualmente en los dispositivos de cada Plataforma IIoT. Todos los Path son publicados hacia un único broker MQTT centralizado, lo que permite que cualquier aplicación conectada al broker (sistemas SCADA, dashboards o bases de datos) pueda consumir los datos sin necesidad de procesos intermedios. Esta simplicidad en la arquitectura y la menor cantidad de capas intermedias facilitan la integración, reduciendo el esfuerzo en la configuración inicial y permitiendo una rápida puesta en marcha con conocimientos técnicos moderados.

En contraste, la técnica de Schultz exige una integración jerárquica estructurada que incluye la configuración de nodos UNS a nivel de *Area*, *Site* y *Enterprise*. Cada uno de estos niveles requiere instancias separadas que actúan como agregadores y modeladores de datos, lo que demanda una infraestructura de TI más robusta y mayor coordinación entre equipos de trabajo. Además, esta técnica implica un conocimiento avanzado sobre comunicación entre brokers, estructuras de tópicos intermedios y manejo de conexiones distribuidas, lo que eleva la complejidad general del despliegue. En consecuencia, la técnica de Parris se posiciona como la opción más sencilla y eficiente en términos de integración, especialmente en entornos donde se busca una solución rápida, flexible y de bajo costo de entrada.

### ***Resiliencia***

En el contexto de arquitecturas IIoT, la resiliencia se refiere a la capacidad del sistema para mantener la operatividad y disponibilidad de los servicios ante fallos, interrupciones o desconexiones parciales o totales. Esta métrica es crucial cuando se diseñan sistemas distribuidos para entornos industriales, donde la continuidad operativa es esencial y cualquier caída puede tener consecuencias significativas a nivel de producción o monitoreo.

La técnica de Schultz presenta una ventaja clara en este aspecto, ya que su diseño se basa en una arquitectura distribuida con múltiples niveles de brokers MQTT independientes, organizados por *Area*, *Site* y *Enterprise*. Esto permite que, en caso de una falla localizada en un broker de *Site* o broker de *Area*, el resto del sistema continúe funcionando sin afectación. Cada nivel actúa como una unidad autónoma de comunicación, lo que otorga tolerancia a fallos y segmentación natural del riesgo.

Por el contrario, la técnica de Parris depende de una estructura centralizada donde todos los dispositivos publican directamente hacia un único broker MQTT en la nube. Esta centralización implica que cualquier interrupción del broker o de la conectividad hacia la nube impacta inmediatamente a toda la infraestructura, afectando a todos los *Sites*, *Areas* y *Lines* conectadas. No existen mecanismos nativos de aislamiento ante fallos, lo que limita su resiliencia en escenarios de redes inestables o entornos críticos. En consecuencia, la técnica de Schultz es

claramente superior en términos de resiliencia, ya que ofrece una arquitectura distribuida y segmentada que proporciona mayor robustez operativa ante fallos de red o caídas puntuales en la infraestructura.

### ***Flexibilidad***

Esta métrica se conoce como la capacidad del sistema para adaptarse fácilmente a distintos entornos tecnológicos, aplicaciones externas y cambios en la infraestructura, permitiendo la interoperabilidad con diversas plataformas y facilitando el acceso a la información desde diferentes puntos del ecosistema industrial.

En este sentido, la técnica de Parris demuestra una mayor flexibilidad operativa. Su arquitectura centralizada permite que cualquier aplicación o sistema que soporte el protocolo MQTT, ya sea una plataforma IIoT, un sistema SCADA o un visualizador como Grafana, pueda conectarse directamente al broker para consumir la información, sin necesidad de intermediarios adicionales. Esta independencia amplía significativamente las posibilidades de integración con tecnologías de terceros en la nube y en entornos locales.

Por el contrario, la técnica de Schultz presenta limitaciones importantes en este aspecto. Su diseño jerárquico requiere la implementación de plataformas IIoT industriales como Ignition en cada uno de los niveles del UNS para consolidar y enviar los datos. Esto no solo implica una mayor inversión en infraestructura, sino que también restringe la compatibilidad con aplicaciones externas, ya que estas no pueden acceder directamente al flujo de datos sin pasar por las capas intermedias. Por lo tanto, en términos de interoperabilidad y adaptabilidad, la técnica de Parris resulta más flexible, al simplificar la conexión de nuevas aplicaciones y permitir un acceso más abierto y directo a la información industrial.

### ***Seguridad***

La capacidad para proteger la integridad, confidencialidad y disponibilidad de los datos a lo largo de toda la infraestructura define la seguridad del sistema. Esto incluye la gestión de accesos, la segmentación de redes, el control de usuarios y la protección contra amenazas tanto internas como externas.

Desde este enfoque, la técnica de Schultz ofrece una arquitectura inherentemente más robusta en términos de seguridad. Al implementar una jerarquía distribuida de brokers MQTT en los distintos niveles del UNS, es posible definir políticas de autenticación y autorización específicas para cada nodo de la red. Esto permite aplicar controles de acceso diferenciados en función del rol del usuario o del sistema cliente, así como segmentar los datos según su criticidad. Además, la descentralización facilita la contención de incidentes, es decir, una brecha de seguridad en un área no compromete al resto de la infraestructura, ya que los brokers funcionan de forma independiente.

Por el contrario, la técnica de Parris se basa en un modelo centralizado en el que toda la información fluye hacia un único broker MQTT. Aunque este broker puede protegerse mediante autenticación con credenciales, certificados y cifrado TLS, el punto único de acceso concentra todo el tráfico y la información del sistema, lo que lo convierte en un objetivo más vulnerable ante posibles ataques o fallos de configuración. Una vez establecida la conexión, los clientes autenticados tienen acceso a la totalidad del flujo de datos. En consecuencia, la técnica de Schultz representa una solución más segura, al permitir una segmentación más fina de los permisos, una gestión descentralizada de los accesos y una mayor capacidad de aislamiento ante incidentes.

### ***Implementación***

El nivel de esfuerzo técnico y operativo necesario para conectar distintas aplicaciones, plataformas y servicios dentro de una arquitectura IIoT es otro criterio a tener en cuenta. Se requiere asegurar que los datos puedan ser consumidos y comprendidos de manera estructurada y eficiente por los diferentes componentes del sistema.

En este aspecto, la técnica de Schultz presenta una implementación más directa y estructurada, ya que la segmentación de los niveles del UNS es gestionada automáticamente por la propia plataforma IIoT (Ignition). Esta jerarquía se construye conforme se interconectan los distintos brokers MQTT en cada nivel, lo que permite a los usuarios trabajar con rutas de datos ya organizadas y comprensibles desde el punto de vista de la arquitectura definida.

Por el contrario, en la técnica de Parris, a pesar de la centralización de la información en un único broker, la estructuración jerárquica del UNS debe realizarse de manera manual. Es necesario definir cada uno de los Path de publicación, estableciendo el uso de delimitadores (como el guion bajo “\_”) que posteriormente deberán ser interpretados por las aplicaciones consumidoras. Este enfoque implica un mayor conocimiento técnico en programación, manejo de estructuras de nodo y un entendimiento profundo de la lógica jerárquica del UNS, lo que puede dificultar la implementación en entornos con múltiples fuentes de datos y usuarios no especializados. Por estas razones, la técnica de Schultz ofrece una mayor facilidad de implementación, al delegar la segmentación jerárquica a la lógica propia de las plataformas IIoT utilizadas, reduciendo así la complejidad técnica para el usuario final.

### ***Escalabilidad***

La escalabilidad evalúa la facilidad con la que una arquitectura IIoT puede crecer incorporando nuevos dispositivos o casos de estudio sin re-configuraciones complejas. Esta métrica implica incorporar nuevos dispositivos, sensores, plataformas o casos de estudio, manteniendo siempre una lógica coherente y eficiente de integración de datos.

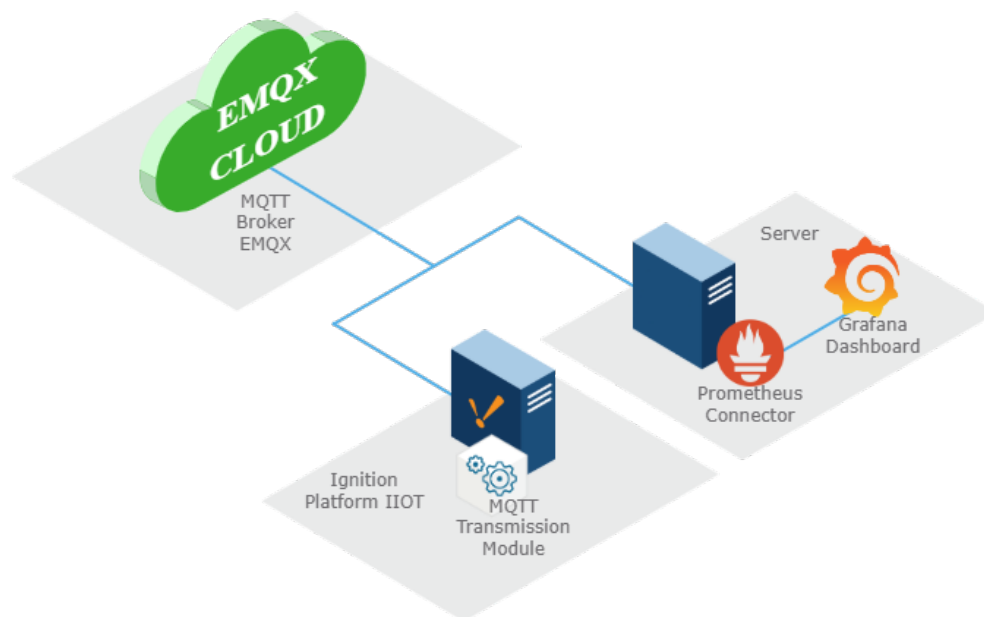
La técnica de Schultz se destaca al permitir que, al añadir nuevos elementos, estos se integren directamente al nivel jerárquico correspondiente del UNS, actualizando automáticamente toda la infraestructura sin necesidad de modificaciones manuales. Por el contrario, Parris requiere definir manualmente nuevos Path y adaptar la plataforma IIoT para nivel del UNS, lo que incrementa la complejidad y el riesgo de errores. En este sentido, Schultz ofrece una escalabilidad superior, siendo más eficiente y robusta frente al crecimiento del sistema.

### **Métricas Cuantitativas de Desempeño**

Para complementar el análisis comparativo entre las técnicas de diseño de UNS, se definieron un conjunto de métricas cuantitativas de desempeño orientadas a evaluar la eficiencia operativa y la carga generada sobre la infraestructura de comunicación. Las métricas seleccionadas incluyen: gestión de tópicos y sesiones, la gestión de autenticaciones, clientes y mensajes, el flujo de datos y finalmente la actividad de los paquetes enviados al **Broker Enterprise**.

Estas variables permiten cuantificar el comportamiento del sistema desde distintos ángulos, incluyendo el tráfico de red, el volumen de comunicación, el uso de recursos del broker MQTT y el nivel de concurrencia del sistema.

La razón principal para la elección de estas métricas radica en que son expuestas directamente por la API de monitoreo del broker EMQX, lo que permite su recopilación mediante Prometheus. La herramienta accede a los endpoints del broker mediante su API, almacenando la información en series temporales de forma confiable y sin intervención adicional. Esto garantiza una evaluación precisa, en tiempo real y desde la fuente, facilitando así un análisis comparativo consistente entre ambas técnicas.



**Figura 18**

*Arquitectura de Evaluación de Desempeño*

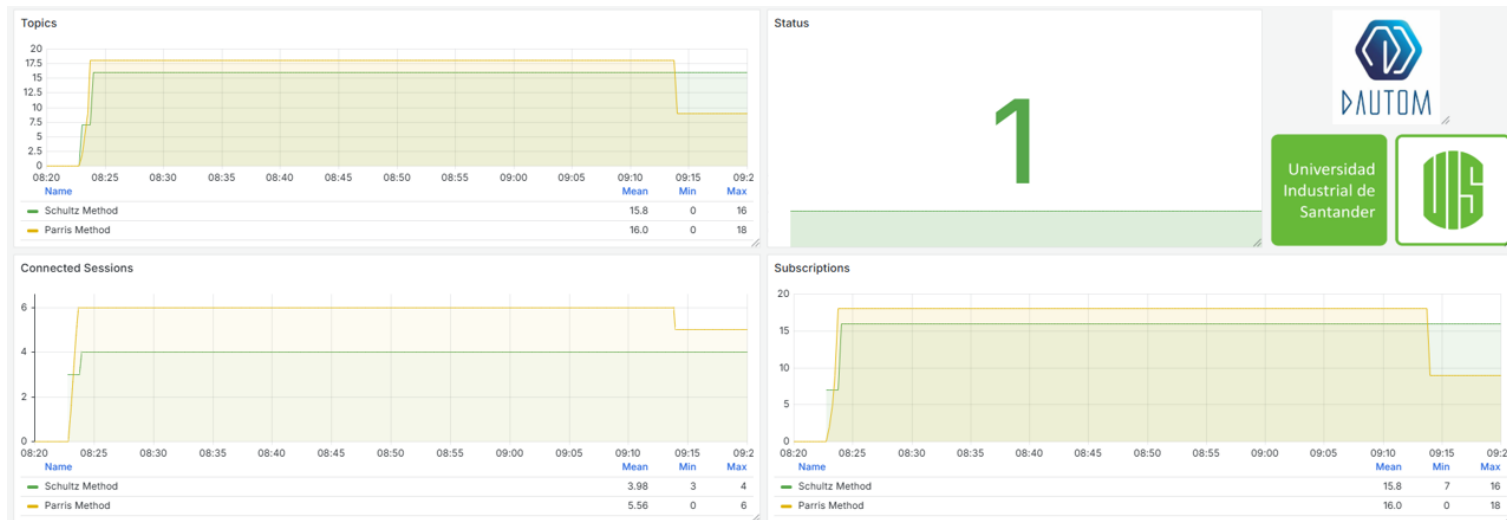
La Fig. 18 representa el diagrama de conexiones que permite la evaluación del desempeño de las métricas cualitativas previamente definidas. El flujo de información inicia en la plataforma IIoT Ignition al publicar información generada en la plataforma de entrenamiento hacia un broker MQTT, específicamente el servidor EMQX desplegado en la nube. Posteriormente, un servidor se encarga de consumir estos datos desde el broker y monitorear las métricas me-

dian­te su inter­faz. Estas mé­tricas son reco­lec­ta­das por Prometheus, un sistema de moni­to­reo que realiza con­sul­tas pe­ri­ó­di­cas a través de la API del broker, al­ma­ce­nan­do los re­sul­ta­dos en series tem­po­ra­les. Final­men­te, el vi­su­a­li­za­dor in­dus­trial Grafana, se con­ecta a la base de datos de Prometheus y per­mite con­struir pa­ne­les in­ter­ac­ti­vos para el aná­li­sis his­tó­rico, la su­per­vi­sión en tie­mpo real y la con­fi­gu­ra­ción de al­er­tas ba­sa­das en el com­por­ta­mien­to de las mé­tricas del sistema. Esta in­te­gra­ción ga­ran­ti­za una vi­su­a­li­za­ción ef­ci­ente del ren­di­mien­to dentro de la pla­ta­for­ma IIoT se­gún las mé­tricas que se des­cri­ben a con­ti­nuación.

### ***Gestión de Tópicos y Sesiones MQTT***

Este con­jun­to de mé­tricas evalúa la ef­ci­encia del sistema en cuanto a la ad­mi­nis­tra­ción de los tó­picos (temas) y las se­siones ac­ti­vas dentro del broker MQTT. Los tó­picos re­pre­sentan los ca­na­les por los cuales se pu­bli­ca o se sus­cri­be in­for­ma­ción, y las se­siones re­fle­jan las con­exio­nes ac­ti­vas de los clien­tes. Una ges­tión ef­ci­ente de estos ele­men­tos es fun­da­men­tal para ga­ran­ti­zar un sistema es­ca­la­ble, or­denado y con bajo ries­go de co­li­sio­nes o re­du­nan­cias en la co­mu­ni­ca­ción. Ad­e­más, un nú­mero op­ti­mi­zado de sus­cri­p­cio­nes in­di­ca una ar­qui­tec­tura me­jor es­truc­tu­rada y ali­neada con el mo­de­lo de datos je­rár­quico del UNS.

- **deployment\_emqx\_metrics\_topics\_count:** Mide la cantidad total de tópicos ac­ti­vos en el broker EMQX. Cada tó­pico re­pre­sent­a un ca­nal ló­gico por el que se pu­bli­can o se re­ci­ben men­sa­jes. Un sistema je­rár­quico­mente es­truc­tu­rado tie­nde a or­ganizar es­tos tó­picos de forma más or­denada y con­tro­lada, mi­ni­mi­zan­do re­du­nan­cias.
- **deployment\_emqx\_sessions\_count:** Corresponde al nú­mero de se­siones ac­ti­vas en el broker, es decir, las con­exio­nes clien­te-ser­vi­dor que están in­ter­can­biando datos. Esta mé­trica per­mite in­ferir el nú­mero de dis­po­si­ti­vos o ser­vi­cios que están in­ter­ac­tuando di­rec­ta­mente con el broker.
- **deployment\_emqx\_metrics\_subscriptions\_count:** In­di­ca cuántas sus­cri­p­cio­nes ac­ti­vas existen. Una sus­cri­p­ción ocu­rre cuando un clien­te se re­gis­tra para re­ci­bir los men­sa­jes pu­bli­ca­dos en un tó­pico es­pe­cí­fico.



**Figura 19**

*Test de Gestión de Tópicos y Sesiones MQTT*

En Grafana se presentan los paneles de visualización con las series temporales como se muestra en la Fig. 19. Allí se observan las métricas de gestión de tópicos y sesiones y se distinguen los resultados obtenidos mediante la técnica de Schultz (color verde) y los de la técnica de Parris (color amarillo). Además, se incluyen estadísticas relevantes derivadas del monitoreo continuo para contrastar el comportamiento de ambas arquitecturas.

Métrica	Schultz	Parris
Tópicos	16	18
Sesiones Conectadas	4	6
Suscripciones	16	18

**Tabla 2**

*Comparación de Gestión de Tópicos y Sesiones MQTT*

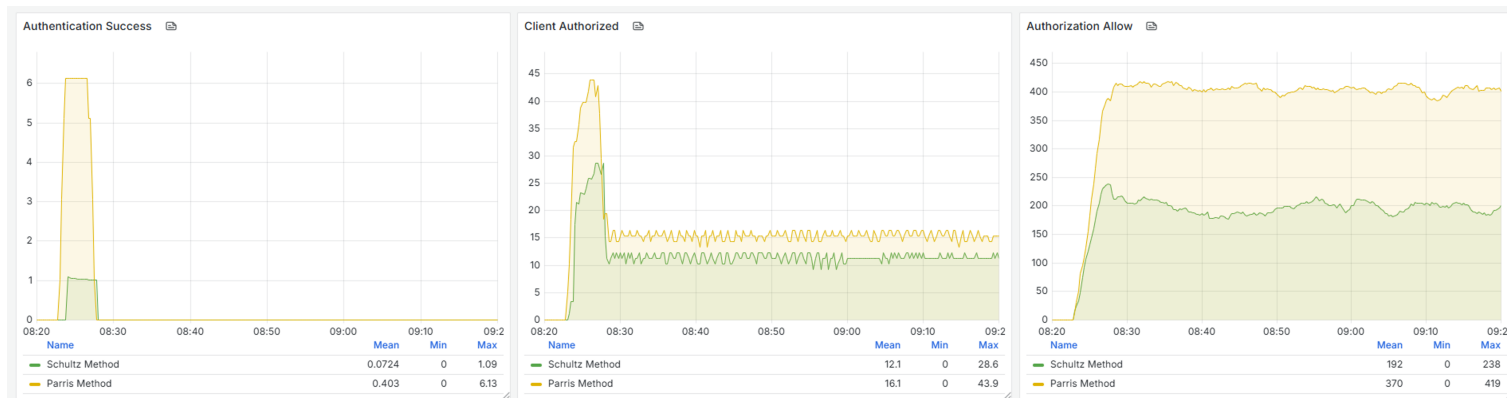
En la comparación de resultados entre las arquitecturas de Schultz y Parris, se observa un comportamiento consistente: Schultz presenta menores valores en las tres métricas como se evidencia en la Tabla 2. Estos resultados evidencian una arquitectura más contenida y organizada por parte de Schultz. Al estructurar el UNS en distintos niveles jerárquicos, Schultz

logra distribuir la responsabilidad de publicación y suscripción, reduciendo la concentración de carga en un único punto. En cambio, Parris, al centralizar toda la información en un solo broker, incrementa inevitablemente la cantidad de tópicos y sesiones necesarias para administrar la misma cantidad de datos, lo que puede dificultar la trazabilidad y escalabilidad del sistema.

### ***Gestión de Autenticaciones y Autorizaciones***

Este conjunto de métricas permite evaluar el comportamiento del sistema en relación con los procesos de autenticación y autorización de los clientes que interactúan con el broker MQTT. Estas operaciones son fundamentales para garantizar la seguridad, el control de acceso y la gestión eficiente de identidades dentro de arquitecturas IIoT. Analizar estas métricas proporciona una visión clara del volumen de autenticaciones realizadas, las solicitudes de autorización recibidas y las operaciones exitosas de control de acceso como se observa en la Fig.20, permitiendo detectar patrones o posibles vulnerabilidades.

- **deployment\_emqx\_metrics\_authentication\_success:** Representa el número total de autenticaciones exitosas realizadas por los clientes hacia el broker. Esta métrica es crítica para entender la cantidad de conexiones autenticadas correctamente durante el periodo de evaluación.
- **deployment\_emqx\_metrics\_client\_authorize:** Indica la cantidad de solicitudes de autorización recibidas por el broker por parte de los clientes autenticados. Cada vez que un cliente intenta publicar o suscribirse a un tópico, se genera una solicitud que debe ser autorizada, dependiendo de las políticas de control de acceso configuradas.
- **deployment\_emqx\_metrics\_authorization\_allow:** Refleja cuántas de esas solicitudes de autorización fueron permitidas según las reglas del sistema. Este valor debe correlacionarse estrechamente con la métrica anterior, y un alto valor sugiere que la mayoría de las operaciones solicitadas por los clientes fueron válidas y seguras.



**Figura 20**

### *Test de Gestión de Autenticaciones y Autorizaciones*

En la evaluación de las métricas relacionadas con autenticación y autorización, se evidencian diferencias notables entre las arquitecturas. La técnica de Parris presentó un mayor número de autenticaciones exitosas como se describe en la tabla 3. Este comportamiento está directamente relacionado con su arquitectura centralizada, donde todos los clientes deben autenticarse y autorizarse directamente contra un único broker. Esta concentración genera un volumen más alto de operaciones en el punto central, aumentando la complejidad de gestión y potenciales cuellos de botella.

Métrica	Schultz	Parris
Autenticaciones Exitosas	1	6
Cientes Autorizados	12	16
Autorizaciones Permitidas	192	370

**Tabla 3**

### *Comparación de Gestión de Autenticaciones y Autorizaciones*

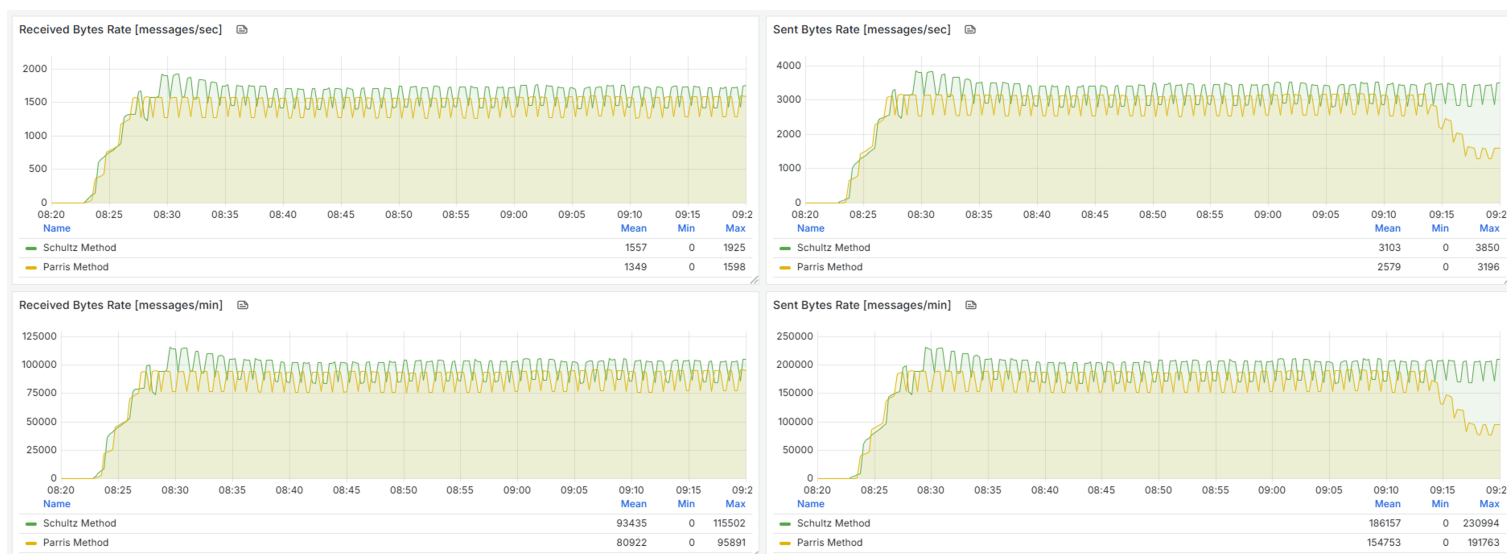
Por su parte, la técnica de Schultz, al implementar una arquitectura jerárquica con múltiples niveles de brokers distribuidos, logra desacoplar el tráfico de autenticación, permitiendo que muchas de estas operaciones ocurran de forma intermedia antes de llegar al broker principal. Esto reduce la cantidad de accesos directos y mejora la segmentación de seguridad. Aunque

sus métricas reflejan un menor volumen de eventos, este comportamiento implica una mayor eficiencia operativa y una arquitectura más segura y escalable, posicionando a Schultz como una mejor opción en entornos industriales que requieran control granular y tolerancia a fallos.

### **Flujo de Datos**

Estas métricas están orientadas a evaluar el volumen de datos que circula a través del broker MQTT durante el funcionamiento de cada arquitectura, permitiendo cuantificar tanto la entrada como la salida de información del sistema a partir de tasas de transferencia promedio por segundo y por minuto. La Fig.21 permite observar con mayor claridad el comportamiento del flujo de datos en escalas de tiempo relevantes para operaciones industriales.

Estas métricas son fundamentales para entender el rendimiento de red y la eficiencia en el intercambio de datos entre productores (Ignition) y consumidores (cliente). Un flujo constante y bien distribuido es indicativo de una arquitectura optimizada tanto para el tráfico entrante como saliente, aspecto esencial en entornos IIoT donde grandes volúmenes de datos deben ser procesados en tiempo real sin generar obstrucciones.



**Figura 21**

### *Test de Flujo de Datos*

- **deployment\_emqx\_metrics\_bytes\_received:** Mide la tasa de bytes recibidos

por el broker, calcula la cantidad promedio de bytes recibidos por segundo en ese lapso, proporcionando una visión precisa de la cantidad de datos que el broker recibe desde las plataformas IIoT o dispositivos conectados.

- **deployment\_emqx\_metrics\_bytes\_sent:** Similar a la métrica anterior, pero en este caso se mide la cantidad de bytes enviados por el broker a los clientes conectados para observar la salida de datos hacia las aplicaciones que consumen la información.
- **deployment\_emqx\_metrics\_bytes\_received\*60:** Esta métrica es similar a la primera, pero escalada a bytes por minuto. Multiplicando la tasa por 60 se obtiene la cantidad total de bytes recibidos en un minuto para facilitar la interpretación de la cantidad de datos en un tiempo mayor y así planificar capacidades de la red.
- **deployment\_emqx\_metrics\_bytes\_sent\*60:** Nuevamente, esta métrica mide la tasa de bytes enviados por el broker, pero escalada a bytes por minuto. Es útil para observar el volumen total de datos distribuidos hacia los clientes por minuto, brindando una visión más detallada del rendimiento en intervalos de tiempo más largos.

Los valores reportados para la tasa de datos recibidos indican que la técnica de Schultz alcanzó una media de 1.557 bytes/seg, frente a los 1.349 bytes/seg obtenidos por Parris como se compara en la tabla 4 lo que demuestra que Schultz maneja una mayor cantidad de información tanto de entrada como de salida.

Métrica	Schultz	Parris
Bytes Recibidos por Segundo	1557	1349
Bytes Enviados por Segundo	3103	2579
Bytes Recibidos por Minuto	93435	80922
Bytes Enviados por Minuto	186157	154753

**Tabla 4**

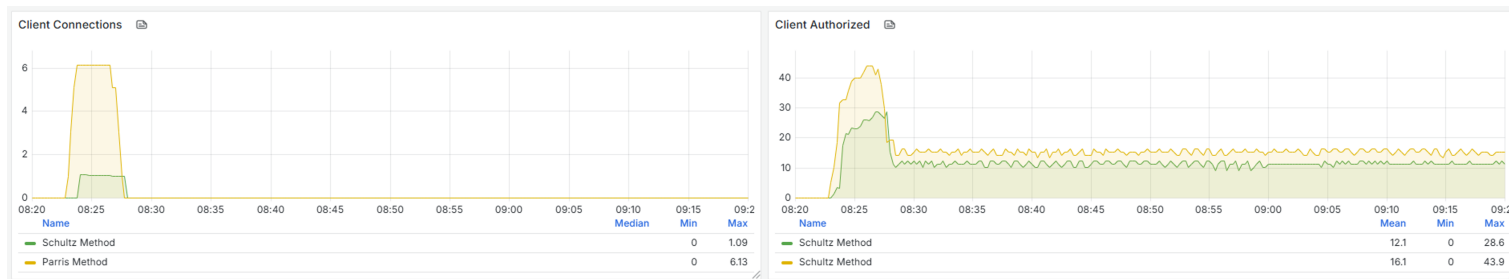
*Comparación de Flujo de Datos*

Esta diferencia refleja cómo la arquitectura jerárquica y segmentada de Schultz permite un procesamiento y redistribución de datos más robusto, especialmente al escalar el sistema. Pese a su mayor complejidad en integración, Schultz optimiza el flujo de datos a través de sus múltiples niveles, evitando cuellos de botella. En cambio, aunque la técnica de Parris es más sencilla de integrar, su enfoque centralizado limita el rendimiento general cuando el volumen de datos aumenta. Esto convierte a Schultz en la opción más sólida para escenarios donde se espera alta densidad de tráfico o expansión futura del sistema.

### *Gestión de Clientes*

Estas métricas se enfocan en la interacción de los clientes con el broker. Las métricas clave incluyen el número de conexiones establecidas por los clientes y la cantidad de autorizaciones que reciben para interactuar con el broker. Estos indicadores son fundamentales para medir la carga de trabajo del sistema, la eficiencia en la gestión de clientes y la efectividad de los mecanismos de autenticación y autorización. En términos generales, un análisis de estas métricas permite evaluar la capacidad de la infraestructura para manejar múltiples clientes y la velocidad con que se gestionan sus conexiones y autorizaciones. Los resultados se pueden evidenciar en la Fig.22.

- **deployment\_emqx\_metrics\_client\_connect:** Esta métrica mide el número de conexiones de clientes al broker durante el intervalo de tiempo. Representa la cantidad de clientes que logran establecer una conexión exitosa con el broker.
- **deployment\_emqx\_metrics\_client\_authorize:** Mide el número de autorizaciones realizadas para los clientes durante un intervalo de tiempo determinado. Cada vez que un cliente es autenticado y autorizado para acceder a los servicios del broker, esta métrica incrementa.

**Figura 22***Test de Gestión de Clientes*

Los resultados obtenidos en la tabla 5 revelan que la técnica de Schultz muestra una conexión significativamente menor con solo 1 cliente conectado durante el periodo de prueba en comparación con los 6 clientes de Parris. En términos de autorización, Schultz presenta 12 autorizaciones, mientras que Parris alcanza las 16, lo que sugiere que la técnica de Parris maneja una mayor cantidad de clientes conectados y autorizados dentro de los mismos intervalos de tiempo.

Métrica	Schultz	Parris
Cientes Conectados	1	6
Cientes Autorizados	12	16

**Tabla 5***Comparación de Gestión de Clientes*

En términos generales, estos resultados indican que la técnica de Parris logra gestionar un mayor número de clientes simultáneamente en comparación con Schultz, lo que podría estar relacionado con su arquitectura centralizada que permite una gestión más sencilla de las conexiones y autorizaciones de clientes. Sin embargo, la técnica de Schultz podría ser más adecuada para escenarios donde la gestión de clientes se realiza de manera más autónoma en distintas áreas, lo cual podría generar una carga menor en el broker central.

### *Gestión de Mensajes*

Este conjunto de métricas se centra en la actividad relacionada con el tráfico de mensajes dentro del broker. Evalúa tanto la recepción como el envío de mensajes, así como la frecuencia con la que los mensajes son publicados. Estas métricas son esenciales para entender el rendimiento de las arquitecturas IIoT en términos de comunicación, ya que permiten cuantificar la carga de datos que se transmite a través del sistema. Además, ayudan a identificar bloqueos en el flujo de información y permite evaluar la eficiencia del manejo del tráfico de mensajes bajo distintos esquemas de arquitectura.

- **deployment\_emqx\_received\_msg\_rate:** Tasa promedio de mensajes recibidos para indicar cuántos mensajes el broker recibe en promedio en cada segundo dentro de esa ventana temporal.
- **deployment\_emqx\_sent\_msg\_rate:** Tasa promedio de mensajes enviados para reflejar la frecuencia con la que el broker distribuye mensajes a los clientes suscritos.
- **deployment\_emqx\_received\_msg\_rate \* 60:** Proyección del total de mensajes recibidos por minuto. Se obtiene multiplicando la tasa por segundo por 60 para obtener una métrica más intuitiva a escala de minutos.
- **deployment\_emqx\_sent\_msg\_rate \* 60:** Tasa de mensajes enviados por minuto, derivada del mismo procedimiento. Proporciona una mejor apreciación de la actividad total en el sistema durante períodos más largos.
- **deployment\_emqx\_metrics\_messages\_publish:** Mide la tasa de publicación de mensajes, es decir, la cantidad de mensajes nuevos que son publicados al broker en promedio por segundo durante el intervalo de tiempo.

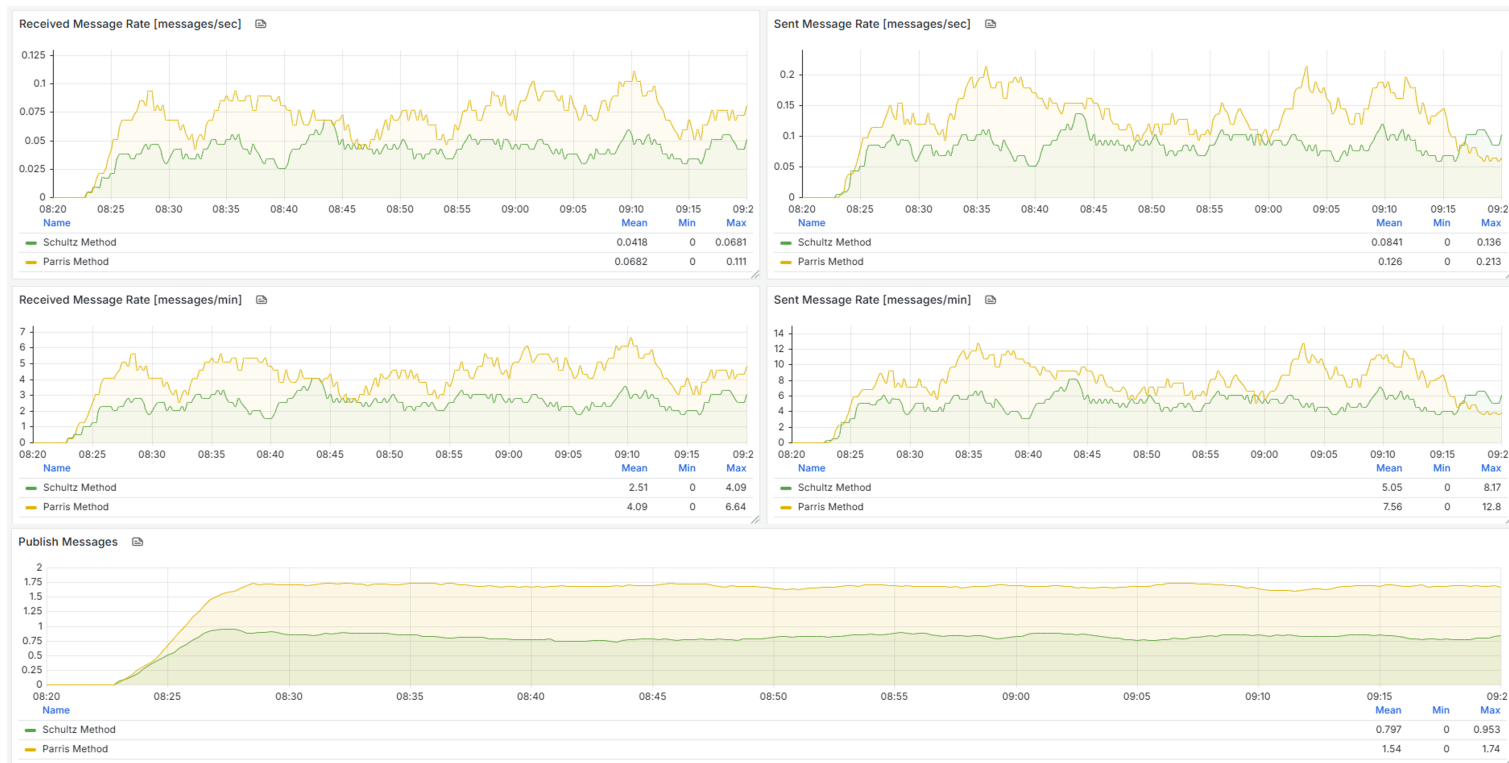


Figura 23

*Test de Gestión de Mensajes*

En la Fig. 23 se observan los resultado de la gestión de mensajes en cada una de las métricas definidas. Al comparar ambas arquitecturas, se observa que Parris mantiene una tasa superior tanto en recepción como en envío de mensajes. Los resultados de cada una de las tasas de mensajería se muestran en la tabla 6 a continuación.

<b>Métrica</b>	<b>Schultz</b>	<b>Parris</b>
Mensajes Recibidos por Segundo	0.0588	0.0763
Mensajes Enviados por Segundo	0.124	0.167
Mensajes Recibidos por Minuto	3.53	4.58
Mensajes Enviados por Minuto	7.45	10
Mensajes Publicados	0.797	1.54

**Tabla 6***Comparación de Gestión de Mensajes*

Estos resultados sugieren que Parris facilita una mayor actividad de mensajería dentro del mismo intervalo de tiempo. Esta diferencia se atribuye al enfoque centralizado de la arquitectura, que permite una difusión más rápida y directa de los mensajes, sin requerir pasos intermedios de agregación o procesamiento. En contraste, la arquitectura Schultz, al segmentar jerárquicamente los niveles de información, puede introducir latencias naturales que impactan la tasa de publicación y distribución de mensajes, aunque podría estar optimizada para casos donde se priorice robustez o filtrado local.

***Actividad de Paquetes***

Este último grupo comprende métricas orientadas a monitorear el volumen y comportamiento del tráfico de paquetes que circulan entre los clientes y el broker. A diferencia de las métricas de mensajes, que se enfocan en el contenido informativo que se transmite, las métricas de paquetes se centran en los distintos tipos de señales del protocolo MQTT que permiten establecer, mantener y finalizar las conexiones, así como realizar operaciones como suscripciones y publicaciones.

Estas métricas son fundamentales para caracterizar el nivel de interacción entre clientes y el servidor, evaluar el rendimiento del canal de comunicación y detectar posibles cuellos de botella en la negociación del protocolo.

- **rate(deployment\_emqx\_metrics\_packets\_received):** Tasa de recepción de paquetes por segundo. Refleja el flujo constante de datos entrantes hacia el broker, incluyendo todos los tipos de paquetes MQTT.
- **rate(deployment\_emqx\_metrics\_packets\_sent):** Tasa de paquetes enviados por segundo. Mide la capacidad del broker para responder a los clientes o reenviar datos, útil para analizar la carga saliente.
- **increase(deployment\_emqx\_metrics\_packets\_received):** Incremento total de paquetes recibidos durante el intervalo de análisis. Indica la acumulación neta de tráfico entrante.
- **increase(deployment\_emqx\_metrics\_packets\_sent):** Incremento total de paquetes enviados en el periodo observado. Complementa la métrica anterior para reflejar la magnitud de la actividad bidireccional.
- **deployment\_emqx\_metrics\_packets\_subscribe\_received:** Cantidad de paquetes SUBSCRIBE recibidos. Representa solicitudes de suscripción a tópicos por parte de los clientes.
- **deployment\_emqx\_metrics\_packets\_connect:** Número de paquetes CONNECT. Corresponde a los intentos de los clientes por establecer una conexión con el broker.
- **deployment\_emqx\_metrics\_packets\_pingreq\_received:** Cantidad de paquetes PINGREQ recibidos. Usados por los clientes para mantener viva la conexión, especialmente en períodos de inactividad de datos.

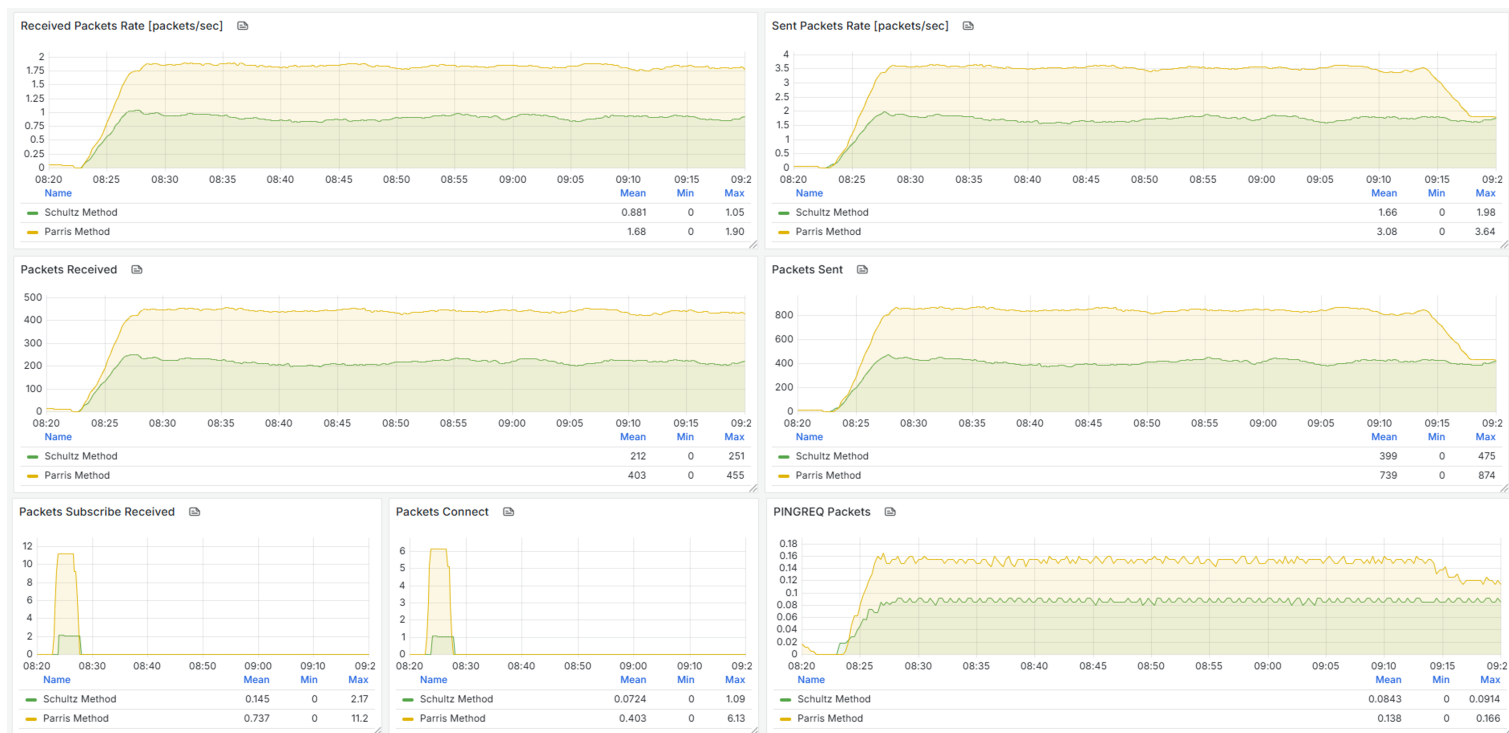


Figura 24

Test de Actividad de Paquetes

Los resultados presentados en la Tabla 7 muestran que la técnica Parris genera una actividad significativamente más intensiva en el intercambio de paquetes MQTT, con una mayor tasa de transmisión y recepción, así como un número más elevado de mensajes asociados a conexiones, suscripciones y mantenimiento de sesión. Este comportamiento indica una mayor frecuencia de interacción entre clientes y servidor, lo que puede traducirse en una respuesta más rápida ante eventos y cambios en el sistema. Tal dinámica resulta especialmente útil en entornos que requieren actualizaciones de estado en tiempo real o una sincronización continua entre múltiples dispositivos.

Métrica	Schultz	Parris
Paquetes Recibos por Segundo	0.881	1.68
Paquetes Enviados por Segundo	1.66	3.08
Paquetes Payload Recibidos	212	403
Paquetes Payload Enviados	399	739
Paquetes de Suscripción Recibidos	2	11
Paquetes de Conexión	1	6
Paquetes PINGREQ	0.0843	0.138

**Tabla 7***Comparación de Actividad de Paquetes*

En contraste, la técnica de Schultz genera un tráfico de paquetes más moderado con una comunicación con menor intensidad. Esta característica representa una ventaja en términos de eficiencia en el uso de la red y consumo de recursos, especialmente en entornos con conectividad limitada o donde se prioriza la estabilidad sobre la inmediatez. No obstante, considerando la mayor capacidad de respuesta y el flujo de información sostenido que ofrece la técnica Parris según estas métricas, sería recomendable su implementación en contextos donde la criticidad operativa y la necesidad de monitoreo continuo sean factores clave.

**Síntesis y Recomendaciones de Implementación**

El análisis comparativo entre las técnicas Schultz y Parris, sustentado en las métricas cualitativas y cuantitativas, permite establecer un panorama integral sobre su desempeño y aplicabilidad en distintos entornos industriales y académicos. La tabla 8 refleja que la técnica Schultz mostró un rendimiento más favorable en aspectos como la resiliencia, seguridad, facilidad de implementación, escalabilidad y eficiencia en la gestión de autenticaciones, clientes, tópicos y sesiones MQTT. Esto sugiere que Schultz es adecuada para entornos donde se prioriza la estabilidad operativa, la predictibilidad del comportamiento del sistema, y donde las capacidades de despliegue deben ser rápidas y confiables. En contraste, la técnica Parris

demonstró fortalezas en dimensiones como integración con otros servicios, flexibilidad para adaptarse a distintos escenarios de comunicación, y un desempeño más eficiente en la gestión dinámica de mensajes y paquetes. Estos resultados la posicionan como una alternativa más apropiada para arquitecturas en evolución, entornos distribuidos con alta interacción entre componentes, o soluciones que requieren una rápida adaptabilidad a cambios en el sistema.

Métrica	Schultz	Parris
Integración	✗	✓
Resiliencia	✓	✗
Flexibilidad	✗	✓
Seguridad	✓	✗
Implementación	✓	✗
Escalabilidad	✓	✗
Gestión de Tópicos y Sesiones MQTT	✓	✗
Gestión de Autenticaciones y Autorizaciones	✓	✗
Flujo de Datos	✓	✗
Gestión de Clientes	✓	✗
Gestión de Mensajes	✗	✓
Actividad de Paquetes	✗	✓

**Tabla 8**

*Síntesis de Comparación entre Schultz y Parris*

En síntesis, Schultz se presenta como la opción más robusta y sencilla para implementaciones en ambientes críticos, controlados o con restricciones de seguridad y recursos, mientras que Parris resulta ideal para proyectos que buscan flexibilidad, interacción con múltiples sistemas y optimización en el flujo de datos, particularmente en fases de expansión tecnológica o integración con plataformas externas. La elección entre ambas dependerá entonces de la etapa del proyecto, la infraestructura disponible y los objetivos operativos del sistema.

## Conclusiones

La revisión del estado del arte del concepto de UNS permitió comprender a profundidad su papel como pieza clave en la convergencia de tecnologías de información (IT) y operación (OT) dentro de los entornos industriales modernos. Se evidenció que el UNS no es simplemente una técnica de integración, sino una filosofía de diseño que reorganiza la forma en que los datos fluyen, se acceden y se gestionan en tiempo real, promoviendo la estandarización, la interoperabilidad y la contextualización a lo largo de toda la arquitectura digital industrial. Este enfoque rompe con los esquemas tradicionales basados en silos de información y abre paso a un modelo en el que todos los actores humanos, máquinas, software y procesos tienen una visión común y actualizada del sistema.

Además, al identificar y comparar las distintas interpretaciones, ventajas y limitaciones del UNS, se logró establecer una base conceptual sólida que sirvió como punto de partida para las siguientes etapas del estudio. El análisis crítico de métodos como los propuestos por David Schultz y Matthew Parris, permitió contrastar enfoques jerárquicos y funcionales, rescatando elementos clave para un diseño efectivo. Este ejercicio ayudó a delimitar el alcance de la investigación, seleccionando los métodos como referencias prácticas para la implementación experimental en la plataforma de entrenamiento IIoT.

La definición de métricas de evaluación cualitativas y cuantitativas fueron instrumentos de análisis para valorar el comportamiento de las arquitecturas UNS en entornos de simulación. Las métricas cualitativas como integración, resiliencia, flexibilidad, seguridad, implementación y escalabilidad permitieron evaluar dimensiones críticas del diseño, destacando la capacidad de las técnicas UNS para sostener flujos de datos robustos, adaptarse a cambios en tiempo real y facilitar la integración de nuevos elementos tecnológicos sin afectar la operación del sistema. Estas métricas ofrecieron una revisión del desempeño arquitectónico más allá de los simples indicadores técnicos.

Por otro lado, las métricas cuantitativas centradas en la estructura de topics, sesiones activas, autenticaciones, flujos de datos, gestión de clientes y actividad de mensajes MQTT permi-

tieron medir con precisión aspectos operativos relevantes de cada implementación. Gracias a estas métricas, se pudo observar el comportamiento del sistema ante distintos escenarios de carga, tráfico y número de dispositivos conectados. La obtención de estos datos aportó objetividad al análisis comparativo entre métodos UNS, posibilitando una toma de decisiones basada en evidencia técnica. Así, las métricas diseñadas se consolidaron como herramientas clave para evaluar y optimizar arquitecturas UNS futuras.

La implementación de los métodos Schultz y Parris en la plataforma Ignition permitió contrastar dos enfoques reconocidos para la estructuración del Unified Namespace. El método Schultz, al organizar jerárquicamente los datos en función del modelo físico y semántico de la planta, ofreció claridad en la navegación del namespace, facilitando la identificación de dispositivos, procesos y valores asociados. Su diseño promovió la estandarización del nombre de los tópicos, lo que redujo la complejidad operativa y mejoró la sostenibilidad del sistema a largo plazo. En la práctica, demostró ser altamente efectivo en escenarios donde la prioridad es la trazabilidad de activos y la consistencia de datos.

El método Parris, en cambio, ofreció una perspectiva más funcional, orientada a reflejar los flujos de proceso y las operaciones de negocio dentro del namespace. Su implementación permitió mapear el comportamiento de sistemas complejos de forma más dinámica y flexible, adaptándose mejor a entornos donde los procesos cambian con mayor frecuencia. Esta característica resultó valiosa para visualizar flujos de trabajo y asociar eventos operativos con su contexto de producción. El análisis comparativo entre ambos métodos evidenció que la elección del enfoque UNS debe responder a las necesidades específicas de cada planta, priorizando ya sea la estabilidad jerárquica o la agilidad funcional.

Mediante la incorporación de los dispositivos industriales de la plataforma de entrenamiento IIoT, conectados a través de un broker MQTT, se simuló condiciones reales de operación para evaluar la robustez de la infraestructura UNS. Se comprobó que el protocolo MQTT Sparkplug B ofrece ventajas notables en cuanto a gestión del estado de los dispositivos, retención de datos, supervisión de la conexión y estandarización del payload, lo cual facilitó

la integración de los datos en el namespace. Así, se validó la utilidad del protocolo como base sólida para entornos industriales modernos.

Finalmente, se logró evaluar el desempeño de cada implementación mediante pruebas controladas en la plataforma IIoT desarrollada. Estas pruebas incluyeron análisis en tiempo real de los datos publicados, verificación del cumplimiento de la estructura de nombres, evaluación de la carga del broker, y monitoreo del comportamiento ante eventos disruptivos. Se evidenció que ambas implementaciones UNS fueron funcionales y capaces de sostener flujos de información estables, pero cada una mostró ventajas particulares: mientras el método Schultz sobresalió en organización y claridad del namespace, el método Parris ofreció mayor flexibilidad para representar flujos operativos dinámicos.

Los resultados de las métricas cualitativas y cuantitativas se integraron para obtener una visión completa del rendimiento del sistema, destacando cómo el diseño del namespace influye directamente en la escalabilidad, trazabilidad y facilidad de uso de la plataforma. Esta evaluación permitió validar los beneficios del UNS como habilitador de arquitecturas IIoT, y al mismo tiempo identificar oportunidades de mejora para futuras implementaciones. En conjunto, la plataforma de entrenamiento sirvió como entorno de validación efectivo, permitiendo experimentar en condiciones controladas con resultados extrapolables a escenarios industriales reales. Así, se demostró que el uso de técnicas UNS, correctamente implementadas y evaluadas, puede acelerar la adopción de tecnologías digitales avanzadas en el sector industrial hacia la transformación digital.

### **Trabajo futuro**

La línea de este trabajo de investigación continúa con la implementación de un UNS en un entorno industrial real para validar su desempeño bajo condiciones operativas no simuladas. Además, se sugiere desarrollar una nueva técnica de diseño que combine las ventajas de los métodos Schultz y Parris, buscando un enfoque más equilibrado entre estructura jerárquica, escalabilidad, resiliencia y flexibilidad en la gestión de datos IIoT.

## Referencias

- Alguliyev, R., Imamverdiyev, Y., y Sukhostat, L. (2018). Cyber-physical systems and their security issues. *Computers in Industry*, 100, 212–223.
- Apilioğulları, L. (2022). Digital transformation in project-based manufacturing: Developing the ISA-95 model for vertical integration. *International Journal of Production Economics*, 245. (10.1016/j.ijpe.2022.108413)
- Automation, I. (2015). *Firebrand awards ceremony 2015*. <https://inductiveautomation.com/resources/icc/2015/firebrand-awards-ceremony-2015>.
- Bai, J., Zhu, W., Zhang, T., y Han, M. (2024). Research on the integration of UnifyFS and Slurm. *Proceedings of SPIE - The International Society for Optical Engineering*, 13181. (10.1117/12.3031392)
- Bhadra, P., Chakraborty, S., y Saha, S. (2023). Cognitive IoT meets robotic process automation: The unique convergence revolutionizing digital transformation in the Industry 4.0 era. *Smart Innovation, Systems and Technologies*, 335, 355 – 388. (10.1007/978-981-19-8296-5\_1)
- Brim, M., Moody, A., Lim, S.-H., Miller, R., Boehm, S., Stanavige, C., ... Oral, S. (2023). UnifyFS: a user-level shared file system for unified access to distributed local storage. *Proceedings - 2023 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2023*, 290-300. (10.1109/IPDPS54959.2023.00037)
- Chen, W., Yan-yi, L., Tie-zheng, G., Da-peng, L., Tao, H., Zhi, L., ... Ying-you, W. (2024). Systems engineering issues for industry applications of large language model. *Applied Soft Computing*, 151, 111165.
- Derhamy, H., Andersson, M., Eliasson, J., y Delsing, J. (2018). Workflow management for edge driven manufacturing systems. *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*, 774-779. (10.1109/ICPHYS.2018.8390805)
- Domínguez-Cid, S., Larios, D. F., Molina, F. J., Luque, J., Carrasco, A., y León, C. (2024). SSOT-Based laboratory on IIoT for Industry 4.0 architectures. *16th*

- Congreso de Tecnologia, Aprendizaje y Ensenanza de la Electronica, TAEE 2024.*  
(10.1109/TAEE59541.2024.10604981)
- Dotoli, M., Fay, A., Miśkowicz, M., y Seatzu, C. (2019). An overview of current technologies and emerging trends in factory automation. *International Journal of Production Research*, 57(15-16), 5047–5067.
- Foundation, E. (2016). Sparkplug™ specification [website]. Descargado de <https://sparkplug.eclipse.org/specification/version/2.2/documents/sparkplug-specification-2.2.pdf>
- García-Domínguez, A., Marcos-Bárcena, M., Medina-Bulo, I., y Prades, L. (2014). Towards an integrated SOA-based architecture for interoperable and responsive manufacturing systems using the ISA-95 object model. *Key Engineering Materials*, 615, 145 – 156.  
(10.4028/www.scientific.net/KEM.615.145)
- Gruener, S., Koziolk, H., y Ruckert, J. (2021). Towards resilient IoT messaging: An experience report analyzing MQTT brokers. *Proceedings - IEEE 18th International Conference on Software Architecture, ICSA 2021*, 69-79. (10.1109/ICSA51549.2021.00015)
- Hintaw, A. J., Manickam, S., Karuppayah, S., y Aboalmaaly, M. F. (2019). A brief review on MQTT's security issues within the internet of things (IoT). *Journal of Communications*, 14(6), 463 – 469. (10.12720/jcm.14.6.463-469)
- Hou, L., Zhao, S., Xiong, X., Zheng, K., Chatzimisios, P., Hossain, M. S., y Xiang, W. (2016). Internet of things cloud: Architecture and implementation. *IEEE Communications Magazine*, 54(12), 32-39. (10.1109/MCOM.2016.1600398CM)
- Johnsson, C. (2004). ISA 95-how and where can it be applied. *ISA Expo*, 1–10.
- Johnsson, C., y Brandl, D. (2021). Beyond the pyramid: using ISA95 for Industry 4.0/smart manufacturing. *Intech*, 14–20. Descargado de <https://www.isa.org/intech-home/2021/october-2021/features/beyond-the-pyramid-using-isa95-for-industry-4-0-an>
- Kaiser, M. (2017). Industry 4.0 meets IoT - logistics and supply chain in digital transfor-

- mation; [industrie 4.0 trifft IoT: Logistik und supply chain im digitalen wandel]. *VDI Berichte*, 2017(2305), 167 – 172.
- Kannan, S., Suri, K., Cadavid, J., Barosan, I., Brand, M., Alferez, M., y Gerard, S. (2017). Towards Industry 4.0: Gap analysis between current automotive MES and industry standards using model-based requirement engineering. *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 29-35. (10.1109/ICSAW.2017.53)
- Kashyap, M., Sharma, V., y Gupta, N. (2018). Taking MQTT and NodeMcu to IOT: Communication in internet of things. *Procedia Computer Science*, 132, 1611-1618. (10.1016/j.procs.2018.05.126)
- Koprov, P., Gadhwal, S., Walimbe, A., Fang, X., y Starly, B. (2023). Systems and methods for authenticating manufacturing machines through an unobservable fingerprinting system. *Manufacturing Letters*, 35, 1009 – 1018. (10.1016/j.mfglet.2023.08.051)
- Kurdi, H., y Thayananthan, V. (2021). Authentication mechanisms for IoT system based on distributed MQTT brokers: Review and challenges. *Procedia Computer Science*, 194, 132 – 139. (10.1016/j.procs.2021.10.066)
- Laghari, S. A., Manickam, S., y Karuppayah, S. (2021). A review on SECS/GEM: A machine-to-machine (M2M) communication protocol for Industry 4.0. *International Journal of Electrical and Electronic Engineering and Telecommunications*, 10(2), 105 – 114. (10.18178/ijeetc.10.2.105-114)
- Li, C., Mantravadi, S., y Moller, C. (2020). AAU open source MES architecture for smart factories-exploiting ISA 95. *IEEE International Conference on Industrial Informatics (INDIN), 2020-July*, 369-373. (10.1109/INDIN45582.2020.9442130)
- Listl, F. G., Fischer, J., Sohr, A., Dittler, D., Jazdi, N., y Weyrich, M. (2023). Utilizing ISA-95 in an industrial knowledge graph for material flow simulation - semantic model extensions and efficient data integration. En (Vol. 120, p. 1558 – 1563). (10.1016/j.procir.2023.09.214)

- Manditereza, K. (2022a). How does a unified namespace (UNS) work? [website]. Descargado de <https://www.hivemq.com/article/how-does-unified-namespace-uns-work-iiot-industry-40/>
- Manditereza, K. (2022b). Implementing unified namespace (UNS) with MQTT Sparkplug [website]. Descargado de <https://www.hivemq.com/article/implementing-unified-namespace-uns-mqtt-sparkplug/>
- Manditereza, K. (2022c). What is unified namespace (UNS) and why does it matter? [website]. Descargado de <https://www.hivemq.com/article/what-is-unified-namespace-uns-iiot-industry-40/>
- Manditereza, K. (2023). Designing your uns semantic information hierarchy [website]. Descargado de <https://www.hivemq.com/blog/designing-your-uns-semantic-information-hierarchy/>
- Mantravadi, S., Srail, J., Brunoe, T., y Moller, C. (2020). Exploring reconfigurability in manufacturing through IIoT connected MES/MOM. *IEEE International Conference on Industrial Engineering and Engineering Management, 2020-December*, 161-165. (10.1109/IEEM45057.2020.9309989)
- Milić, S. D., y Babić, B. M. (2020). Toward the future—upgrading existing remote monitoring concepts to IIoT concepts. *IEEE Internet of Things Journal*, 7(12), 11693-11700.
- Minerva, R., Biru, A., y Rotondi, D. (2015). Towards a definition of the internet of things (IoT). *IEEE Internet Initiative*, 1(1), 1–86.
- Moones, E., Vosgien, T., Kermad, L., Dafaoui, E. M., El Mhamedi, A., y Figay, N. (2015). PLM standards modelling for enterprise interoperability: A manufacturing case study for ERP and MES systems integration based on ISA-95. *Lecture Notes in Business Information Processing*, 213, 157 – 170. (10.1007/978-3-662-47157-9\_14)
- Mäule, J., Kutzler, T., Leich, T., Schäfer, D., y Richter, C. (2024). Unified namespace and asset administration shell: A winning combination for digital production. *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automa-*

- tion (ETFA), 1–7. doi: 10.1109/ETFA61755.2024.10710821
- Péter, Á., y Werner, S. (2024). The impact of Unified Namespace in Industry 4.0. (Department of Automatic Control, Lund University. Advisor: Prof. Charlotta Johnson)
- Razooqi, R. A., Hassan, H. J., y Abdel Saheb, G. M. (2022). Selection algorithm for reducing IoT service delay in the smart factory. *Journal Europeen des Systemes Automatises*, 55(3), 419 – 426. (10.18280/jesa.550315)
- Reynolds, W. (2024a). 4.0 Solutions: Free 1-hour unified namespace (UNS) training webinar [webinar]. Descargado de [https://www.youtube.com/watch?v=ogRy8\\_ut\\_AA](https://www.youtube.com/watch?v=ogRy8_ut_AA)
- Reynolds, W. (2024b). Digital factory mastermind: Methodology and architecture to deploy industry 4.0 solutions for manufacturers. Descargado de <https://www.iiot.university/digital-factory-mastermind>
- Reynolds, W. D. (2023). *Podcast interview with walker reynolds, chairman, intellic integration*. <https://goldenmean001.medium.com/podcast-interview-with-walker-reynolds-chairman-intellic-integration-a12cde3cc340>.
- Singh, M., Goyat, R., y Panwar, R. (2024). Fundamental pillars for industry 4.0 development: implementation framework and challenges in manufacturing environment. *The TQM Journal*, 36(1), 288-309.
- Sisinni, E., Saifullah, A., Han, S., Jennehag, U., y Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, 14(11), 4724–4734. (10.1109/TII.2018.2852491)
- Software, S. D. I. (2019). Four pillars of the industrial IoT [magazine]. Descargado de <https://www.ashb.com/wp-content/uploads/2020/11/IS-2020-152.pdf>
- Srivastava, G., Sapra, D., Sinha, A., Anup, M., y Sinwar, D. (2023). Artificial intelligence and IoT-assisted sustainable manufacturing for Industry 4.0. *Computational Intelligence based Optimization of Manufacturing Process for Sustainable Materials*, 15 – 34. (10.1201/9781003257714-2)

- Stearns, P. N. (2018). The industrial revolution in world history. , 1 – 318. (10.4324/9780429494475)
- Systems, C. (2024). Mapping unified namespaces to domain namespaces [website]. Descargado de <https://corsosystems.com/posts/mapping-unified-namespaces-to-domain-namespaces>
- Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., y Lennartson, B. (2017). An event-driven manufacturing information system architecture for Industry 4.0. *International journal of production research*, 55(5), 1297–1311.
- Trimax Systems, Inc. (2015). *Thousands of sites & millions of tags easily managed*. <https://icc.inductiveautomation.com/discover-gallery-detail/2015/Trimax>.
- Tripathi, S., y Chaurasia, B. (2023). Broker clustering enabled lightweight communication in IoT using MQTT. *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*. (10.1109/ISCON57294.2023.10112105)
- Van Kranenburg, R. (2007). *The internet of things. a critique of ambient technology and the all-seeing network of rfid* (Vol. 2). Institute of Network Cultures.
- Volf, L., Dohnal, G., Beranek, L., y Kyncl, J. (2024). Navigating the fourth industrial revolution: SBRI – a comprehensive digital maturity assessment tool and road to Industry 4.0 for small manufacturing enterprises. *Manufacturing Technology*, 24(4), 668 – 680. (10.21062/mft.2024.074)
- Xu, L. D., He, W., y Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233-2243. (10.1109/TII.2014.2300753)
- Yu, W., Liu, Y., Dillon, T., Rahayu, W., y Mostafa, F. (2022). An integrated framework for health state monitoring in a smart factory employing IoT and big data techniques. *IEEE Internet of Things Journal*, 9(3), 2443-2454. (10.1109/JIOT.2021.3096637)
- Yue, L., Niu, P., y Wang, Y. (2019). Guidelines for defining user requirement specifications (URS) of manufacturing execution system (MES) based on ISA-95 standard. *Journal of Physics: Conference Series*, 1168(3). (10.1088/1742-6596/1168/3/032065)

## Anexo A: Integración de un Sensor de Espectro Radioeléctrico en la Plataforma IIoT

Como parte del proceso de fortalecimiento de capacidades técnicas en la plataforma de entrenamiento IIoT, se integró dentro de la estación de radioastronomía CASIRI un dispositivo de monitoreo del espectro radioeléctrico destinado a la supervisión del entorno electromagnético. Esta integración tiene como finalidad complementar los datos industriales de CASIRI con información del comportamiento espectral del entorno, permitiendo identificar interferencias, disponibilidad de canal, congestión espectral u otros eventos relevantes para la transmisión inalámbrica de datos en contextos industriales.

### Características del Dispositivo Espectrómetro

El dispositivo de monitoreo está diseñado para realizar adquisiciones precisas en ubicaciones fijas, y sus mediciones están georreferenciadas mediante los siguientes parámetros:

- **Latitud y longitud:** Posición geográfica exacta.
- **Azimut:** Orientación horizontal de la antena receptora.
- **Elevación:** Ángulo vertical respecto al horizonte.
- **Altitud:** Altura sobre el nivel del mar o sobre el terreno de referencia.

Estas variables permiten una trazabilidad completa del entorno espectral observado y son registradas junto con la medición en cada adquisición.

### Actualización de la Arquitectura Método Parris

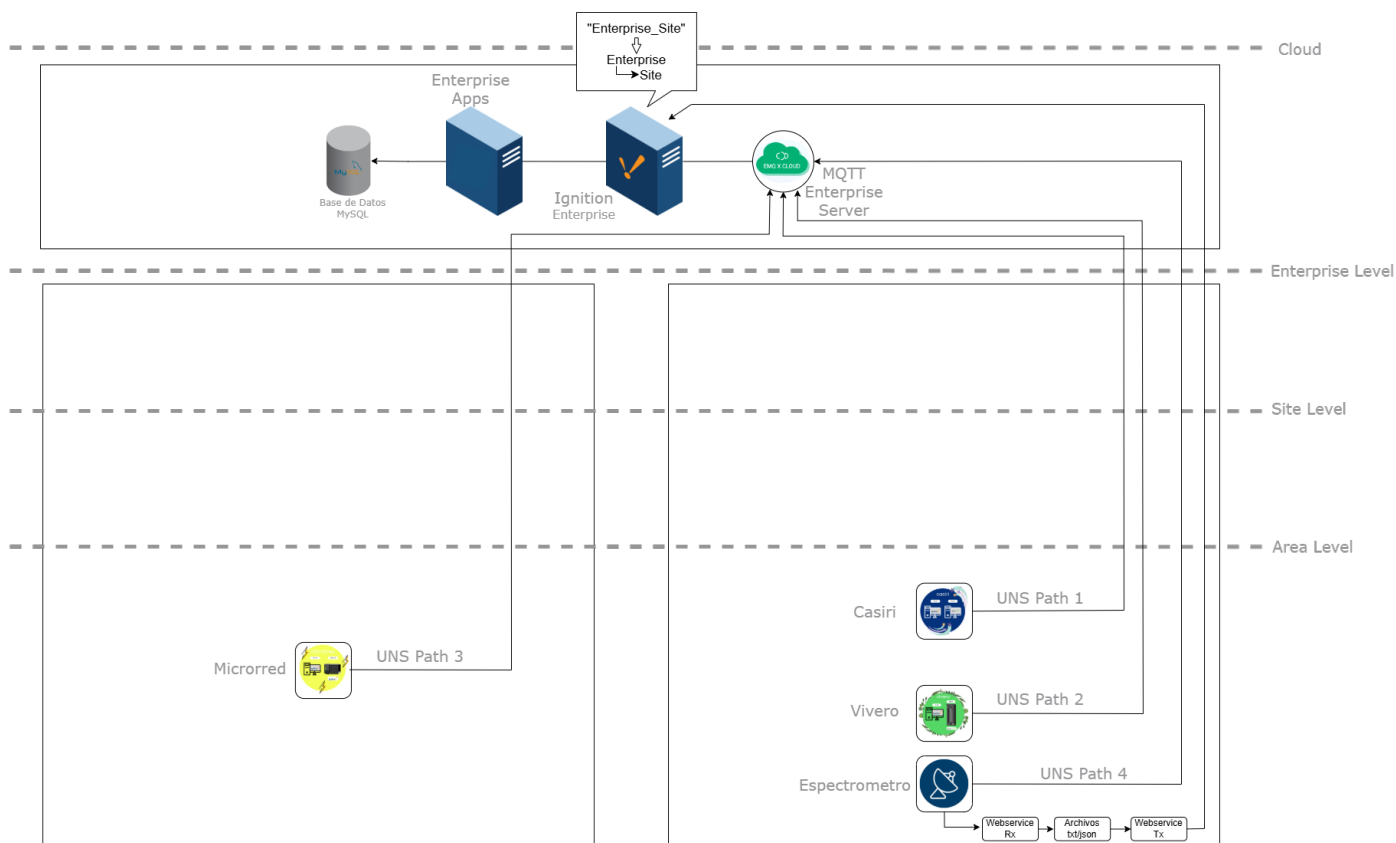
La integración del espectrómetro implicó una actualización en la arquitectura del método Parris implementado en la plataforma. Se añadió un nuevo nodo de adquisición que publica los datos del espectro al broker MQTT bajo una estructura de tópicos adicional, en coherencia con el modelo jerárquico del método como se muestra a continuación.

UNS Path 4: *UIS\_Central/Radiogis\_Casiri\_Sensor\_Espectro/<device\_id>*

Esta integración se diseñó considerando dos modos de operación:

**Modo On-line:** El espectrómetro transmite los datos directamente al broker MQTT mediante un cliente publicador. Esta opción permite la disponibilidad en tiempo real de los datos espectrales en la plataforma Ignition de nivel Enterprise, donde pueden ser procesados, visualizados y almacenados.

**Modo Off-line:** En escenarios donde no se cuenta con conectividad a internet, se activa un mecanismo de contingencia. El espectrómetro transmite los datos a un webservice receptor local que transforma las mediciones en archivos .txt o .json. Esta solución funciona como una base de datos local temporal. Una vez restablecida la conectividad, un webservice transmisor se encarga de sincronizar los datos acumulados hacia el servidor central de Ignition, asegurando que no se pierda ninguna adquisición.



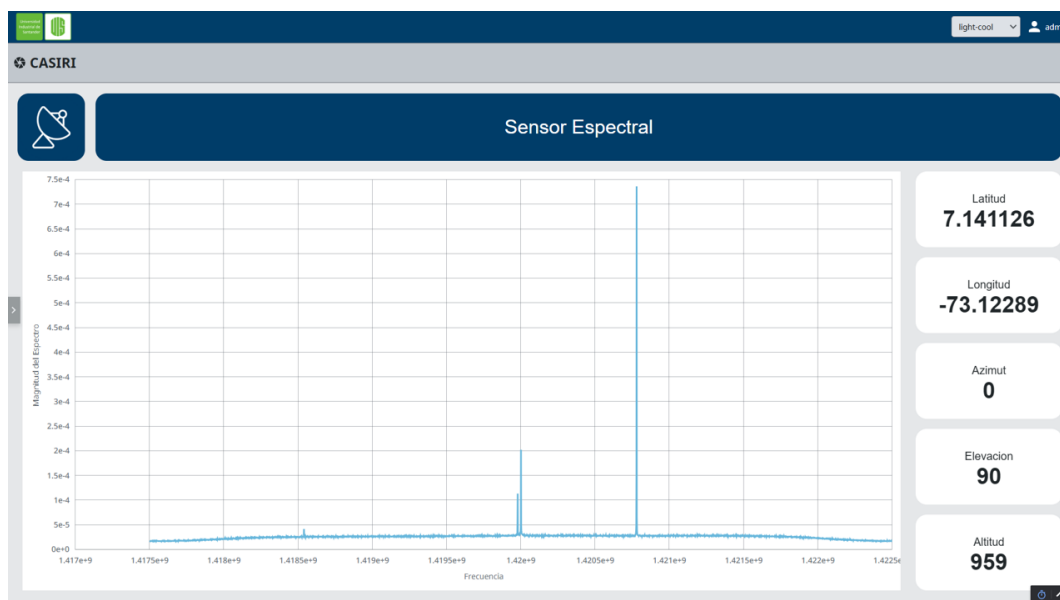
**Figura 25**

*Esquema de la Arquitectura UNS con el Metodo de Parris integrado el Espectrometro*

Un *Webservice* es un servicio web que permite la comunicación y el intercambio de datos entre diferentes aplicaciones o sistemas a través de internet o redes locales, utilizando protocolos estándar como HTTP. Funciona mediante solicitudes y respuestas: una aplicación cliente envía una solicitud al servidor que ofrece el servicio, y este responde con la información solicitada o ejecuta una acción determinada. Los datos intercambiados generalmente se estructuran en formatos TXT o JSON, lo que facilita su interpretación por distintos lenguajes de programación. En este contexto, en la arquitectura de la Fig. 25 el webservice actúa como un puente entre el dispositivo que genera los datos (el espectrómetro) y el servidor que los recibe, permitiendo una transferencia segura incluso en entornos distribuidos permitiendo que los datos críticos del espectro no se pierdan aún en condiciones de red adversas.

### Visualización del Espectro en Ignition

Con el objetivo de brindar una interfaz accesible para los operadores y usuarios de la plataforma, se añadió una nueva vista en la aplicación desarrollada en Ignition. Esta nueva vista muestra el espectro medido en tiempo real o histórico, (dependiendo de la disponibilidad de conexión a internet y del origen de los datos) y sus parámetros de adquisición.



**Figura 26**

*Vista de Espectrómetro de Casiri en la aplicación de Ignition*

Los usuarios pueden consultar espectros correspondientes a ubicaciones específicas, visualizar el comportamiento en diferentes bandas de frecuencia y acceder a la información almacenada en diferentes tiempos. Esto amplía la capacidad de análisis de la plataforma, integrando no solo variables del proceso, sino también condiciones del entorno de comunicación inalámbrica.

### **Consideraciones Finales**

La incorporación del sensor de espectro radioeléctrico fortalece la plataforma IIoT al ampliar el alcance de las variables monitoreadas por Casiri y permitir una evaluación más integral del entorno operativo. Desde la perspectiva del Unified Namespace, esta integración demuestra la flexibilidad del modelo para incorporar nuevos tipos de datos y adaptarse a condiciones variables de conectividad.

La arquitectura propuesta combina la jerarquía estructurada del método Parris con un enfoque práctico para la tolerancia a fallos, lo que refuerza su aplicabilidad en entornos industriales reales donde las condiciones de red no siempre son ideales.