Módulo ciber-físico IoT para la simulación de la operación de sistemas de distribución con múltiples microrredes

Jhon Héctor Sandoval Manrique

Trabajo de Grado para optar al título de Magíster en Ingeniería Eléctrica

Director

Juan Manuel Rey López

PhD. en Ingeniería Electrónica

Codirector

César Antonio Duarte Gualdrón

PhD. en Ingeniería Eléctrica e Informática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Maestría en Ingeniería Eléctrica

Bucaramanga

2026

Dedicatoria

Quiero expresar mi profundo agradecimiento y dedicatoria primero a Dios y a la Virgencita del Milagro por las bendiciones y fortaleza que me brindaron a lo largo de este camino. Su guía y protección han sido fundamentales en cada paso que he dado. Agradezco a los profesores Juan Manuel Rey, Iván Serna y Cesar Duarte por el apoyo y la confianza con este proyecto, espero haber cumplido sus expectativas. Fue muy interesante. A mis queridos padres Mariela y Héctor, les agradezco de corazón por su apoyo incondicional y su amor inquebrantable. Han sido mi motor, mi inspiración y mi apoyo constante durante toda mi vida. Sin ustedes, este logro no habría sido posible. Gracias por creer en mí y por alentarme a seguir adelante en todo momento. Este logro es para ustedes y por ustedes. A mis hermanos Vane, Fercho y Sergio, les agradezco por los mensajes de aliento y el amor que me han demostrado a lo largo de este proceso, sus palabras de ánimo y su presencia han sido un gran impulso para mí. A mis angelitos en el cielo por nunca dejarme solo, gracias por estar siempre a mi lado. A mi novia Natalia B, quiero agradecerle por sentirse orgullosa de mi y por recordarme siempre que todo el esfuerzo y dedicación valdría la pena. Su apoyo incondicional y su amor me han dado la fuerza necesaria para superar cualquier desafío. Gracias por ser mi compañera de vida. Finalmente, a mis abuelos, tíos, primos, sobrino y demás familiares, por su aliento, por estar presentes en mi vida. Sus palabras y su apoyo han sido un estímulo adicional en mi camino hacia el éxito. De corazón, este trabajo de grado no habría sido posible sin el amor y el apoyo de todas las personas mencionadas anteriormente. Estoy profundamente agradecido por su presencia en mi vida y por haber sido una parte integral de este importante logro.

Agradecimientos

Deseo expresar mi más sincero agradecimiento, en primer lugar, a la Universidad Industrial de Santander por brindarme la oportunidad de realizar mis estudios de posgrado. Agradezco especialmente a los profesores y al grupo de investigación que hicieron posible el desarrollo de este trabajo. Este proyecto fue dirigido por el profesor Juan Manuel Rey y codirigido por el profesor César Duarte, cuyo acompañamiento, orientación y apoyo fueron fundamentales durante todo el proceso de investigación. Asimismo, extiendo mi agradecimiento a los profesores del Grupo de Investigación en Sistemas de Energía Eléctrica (GISEL) de la universidad, en particular al profesor Iván Serna, por su apoyo académico y sus valiosos aportes al desarrollo de este trabajo. De igual manera, agradezco la colaboración internacional del grupo de investigación LIREI de la Université du Québec à Trois-Rivières (Canadá), por su disposición y apoyo en el marco de este proyecto. Agradezco especialmente la oportunidad de haber realizado una pasantía de investigación, la cual representó una experiencia académica y profesional profundamente enriquecedora.

Financiación

El presente trabajo de investigación de Maestría en Ingeniería Eléctrica fue financiado con recursos de MinCiencias y la Universidad Industrial de Santander bajo el proyecto titulado ”Diseño, desarrollo e implementación de una plataforma IIoT para formación de profesionales en tecnologías de la cuarta revolución industrial” código Interno UIS: 8290-3776 - código Externo: 82273 Cto. 2022-0719. Este proyecto fue financiado mediante la Convocatoria MinCiencias 890-2020.

Tabla de Contenido

Lista de Figuras

Lista de Tablas

Lista de Apéndices

Glosario

ADMM (Alternating Direction Method of Multipliers): Algoritmo de optimización distribuida utilizado para resolver problemas complejos dividiéndolos en subproblemas que se resuelven de forma iterativa por diferentes agentes.

Broker MQTT: Servidor intermediario responsable de recibir los mensajes publicados por los clientes y distribuirlos a los dispositivos suscritos a los temas correspondientes.

CPS (Cyber-Physical System): Sistema que integra componentes físicos, computacionales y de comunicación para supervisar y controlar procesos físicos en tiempo real mediante la interacción entre hardware, software y redes de comunicación.

DER (Distributed Energy Resources): Recursos energéticos distribuidos conectados al sistema de distribución, tales como generación fotovoltaica, turbinas eólicas, almacenamiento en baterías o sistemas de generación local.

DSO (Distribution System Operator): Entidad responsable de la operación, mantenimiento y gestión de la red de distribución eléctrica, garantizando el suministro seguro y confiable de energía.

Gemelo digital (Digital Twin): Representación virtual de un sistema físico que permite supervisar, analizar y simular su comportamiento en tiempo real.

IoT (Internet of Things): Red de dispositivos físicos interconectados que incorporan sensores, actuadores y capacidades de comunicación para recopilar e intercambiar datos a través de Internet.

Microrred (Microgrid): Sistema eléctrico a pequeña escala que integra generación distribuida, almacenamiento de energía y cargas locales, capaz de operar de forma conectada a la red principal o de manera aislada.

MQTT (Message Queuing Telemetry Transport): Protocolo de mensajería ligero basado en el modelo publicación-suscripción, diseñado para comunicaciones eficientes entre dispositivos en entornos con recursos limitados o redes de baja capacidad.

OpenADR (Open Automated Demand Response): Protocolo de comunicación estándar utilizado para automatizar programas de respuesta a la demanda, permitiendo el intercambio de señales entre operadores del sistema eléctrico y recursos energéticos distribuidos.

Optimización distribuida: Enfoque de optimización en el cual múltiples agentes o nodos resuelven partes de un problema global de manera coordinada mediante el intercambio de información.

Publicador (Publisher): Dispositivo o aplicación que envía mensajes a un broker MQTT bajo un tema específico.

Suscriptor (Subscriber): Dispositivo o aplicación que recibe mensajes de un broker MQTT al estar suscrito a uno o varios temas.

VEN (Virtual End Node): Entidad cliente en el protocolo OpenADR que recibe señales del VTN y ejecuta acciones de control o respuesta energética.

VTN (Virtual Top Node): Servidor central en la arquitectura OpenADR que gestiona los eventos de respuesta a la demanda y coordina la comunicación con los nodos clientes.

Resumen

Título: Módulo ciber-físico IoT para la simulación de la operación de sistemas de distribución con múltiples microredes [*]

Autor: Jhon Héctor Sandoval Manrique [**]

Palabras Clave: IoT, Sistema Ciber-físico, OpenaADR, MQTT, Microrredes eléctricas.

Descripción: La creciente integración de recursos energéticos distribuidos y el avance de las tecnologías de la información y la comunicación han impulsado la evolución de las redes eléctricas hacia arquitecturas más flexibles, distribuidas e inteligentes. En este contexto, los sistemas ciber-físicos y el Internet de las Cosas (IoT) ofrecen herramientas adecuadas para la supervisión, control y optimización de microrredes. Este trabajo propone el desarrollo de un módulo ciber-físico basado en IoT para la simulación distribuida de múltiples microrredes, utilizando dispositivos embebidos y un esquema híbrido de comunicación basado en MQTT y OpenADR. La arquitectura desarrollada permite la interacción entre un nodo central y múltiples tarjetas que representan microrredes, facilitando el intercambio de información en tiempo real y la ejecución de procesos de optimización distribuida. Para validar el módulo se implementaron diferentes estrategias experimentales, incluyendo una estrategia de asignación de precios minoristas mediante comunicación publicación–suscripción basada en MQTT y complementada con OpenADR para aplicaciones de respuesta a la demanda. Los resultados demuestran que la plataforma permite una operación coordinada y confiable entre múltiples nodos, constituyendo una herramienta útil para la validación de estrategias de gestión energética en sistemas eléctricos distribuidos y para el estudio de nuevos esquemas de control y optimización en microrredes interconectadas.

---

[*] Trabajo de investigación de maestría en Ingeniería de Eléctrica

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: PhD. Juan Manuel Rey López. Co-director: PhD. César Antonio Duarte Gualdrón.

Abstract

Title: IoT Cyber-Physical Module for simulating the operation of distribution systems with multi-microgrids. [*]

Author: Jhon Hector Sandoval Manrique [**]

Keywords: IoT, Cyber-Physical System, OpenaADR, MQTT, Electrical Microgrids.

Description: The increasing integration of distributed energy resources and the rapid development of information and communication technologies have driven power systems toward more flexible, distributed, and intelligent architectures. In this context, cyber-physical systems and the Internet of Things (IoT) provide suitable tools for monitoring, control, and optimization of microgrids. This work proposes the development of an IoT-based cyber-physical module for distributed simulation of multiple microgrids using embedded devices and a hybrid communication scheme based on MQTT and OpenADR protocols. The proposed architecture enables interaction between a central node and multiple embedded boards representing microgrids, supporting real-time information exchange and distributed optimization processes. Several experimental strategies were implemented to validate the module, including a retail pricing strategy using an MQTT-based publish–subscribe communication scheme complemented with OpenADR for demand response applications. The results demonstrate that the platform enables coordinated and reliable operation among multiple nodes, constituting a useful tool for validating energy management strategies in distributed power systems and for studying new control and optimization schemes in interconnected microgrids.

---

[*] Master's thesis in Electrical Engineering

[**] Faculty of Physical-Mechanical Engineering. Department of Electrical, Electronics and Telecommunications Engineering. Director: PhD. Juan Manuel Rey López. Codirector: PhD. César Antonio Duarte.

Introducción

La transición energética actual plantea importantes desafíos tanto ambientales como operativos. Este proceso está impulsado principalmente por la necesidad de reducir la dependencia de combustibles fósiles mediante la adopción de fuentes de energía renovables y sostenibles, con el propósito de mitigar los impactos ambientales asociados a la generación convencional (Sahraei Manjili et al., 2012). Sin embargo, más allá del componente ambiental, esta transformación implica cambios estructurales en la forma en que los sistemas eléctricos son diseñados, operados y gestionados. Desde una perspectiva operativa, la gestión del sistema eléctrico se vuelve progresivamente más compleja debido a la transición de un flujo de potencia unidireccional hacia un esquema bidireccional, resultado de la integración de microrredes (MGs) y recursos energéticos distribuidos (DERs) en las redes de distribución. Esta evolución transforma a los consumidores tradicionales en prosumidores, es decir, usuarios capaces de consumir energía, generarla localmente e inyectar excedentes a la red (Grijalva and Tariq, 2011). La bidireccionalidad resultante introduce nuevas dinámicas operativas y plantea retos significativos relacionados con la coordinación, el control y la estabilidad del sistema (Xu et al., 2022).

Una MG se define como un sistema eléctrico de pequeña escala, con límites eléctricos claramente establecidos, capaz de operar de manera autónoma o en conjunto con la red principal (Ton and Smith, 2012). Entre sus principales componentes se encuentran la generación renovable, como los sistemas fotovoltaicos y las turbinas eólicas; la generación despachable, como los generadores diésel; los sistemas de almacenamiento y las cargas locales (Farrokhabadi et al., 2020; Rey et al.,

2022). Las MGs pueden operar en modo isla, suministrando energía de forma independiente, o en modo conectado a red, intercambiando energía con el sistema de distribución según las condiciones de generación y demanda (Tian et al., 2020; Rey et al., 2019).

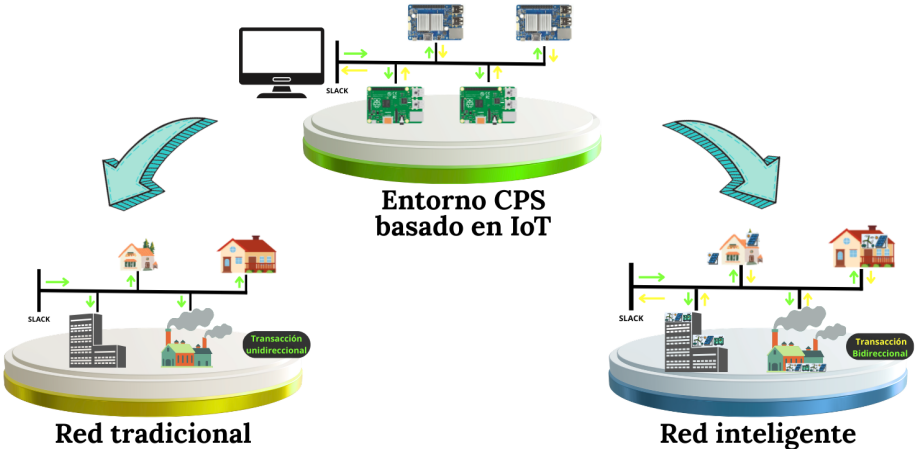
En estos escenarios de intercambio bidireccional de energía e información entre el operador del red (OR) y las MGs, la Infraestructura de Medición Avanzada (AMI) y el Internet de las Cosas (IoT) desempeñan un papel fundamental (Thirugnanam et al., 2021; Wang et al., 2022). El AMI permite la supervisión en tiempo real del consumo energético, habilita operaciones remotas y facilita la gestión eficiente ante eventos o anomalías (Huang et al., 2015). Por su parte, el IoT amplía estas capacidades al integrar dispositivos interconectados capaces de recolectar, procesar y transmitir información de manera continua, utilizando protocolos de comunicación ligeros y escalables (Magade and Sharma, 2023). Esta conectividad masiva permite una mayor visibilidad del estado del sistema, facilita la toma de decisiones distribuida y mejora la coordinación entre recursos energéticos distribuidos. En conjunto, AMI e IoT constituyen la base tecnológica que soporta la digitalización de las redes eléctricas modernas, habilitando esquemas avanzados de control, automatización y gestión energética en entornos con múltiples MGs (Stefan Sacala et al., 2021).

La creciente digitalización de los sistemas eléctricos exige entornos de simulación que modelen de manera realista la interacción entre múltiples MGs, DER y ORs. Las plataformas de simulación convencionales, basadas únicamente en modelos matemáticos ideales, no capturan completamente los efectos asociados a la comunicación, latencias, sincronización y procesamiento distribuido. Por ello, se hace necesario el desarrollo de herramientas experimentales que integren tanto la dimensión física como la computacional, permitiendo representar de forma más precisa

la dinámica real de los sistemas de distribución (Parrado Duque et al., 2018). En este contexto, el presente trabajo propone el desarrollo de un módulo ciber-físico basado en IoT para la simulación de sistemas de distribución con múltiples MGs. Los sistemas ciber-físicos (CPS) integran procesos físicos con capacidades de computación y comunicación, combinando hardware, software y redes para supervisar y controlar procesos en tiempo real. En la última década, estos sistemas han adquirido gran relevancia en el ámbito energético, aunque su implementación enfrenta desafíos técnicos asociados a seguridad, interoperabilidad y robustez (Meera and Arjun, 2024). El módulo desarrollado en esta investigación busca reducir la brecha existente entre los entornos puramente simulados y las implementaciones reales. Al incorporar dispositivos físicos que ejecutan procesos distribuidos y protocolos de comunicación utilizados en entornos reales, se logra una plataforma intermedia que facilita una transición más progresiva y controlada entre la simulación académica o de investigación y la aplicación práctica en sistemas eléctricos reales. Esta conceptualización y relevancia se ilustra en la Figura 1.

Figura 1

Relevancia de la implementación del módulo ciber-físico IoT.



1. Objetivos

1.1. Objetivo general

Desarrollar un módulo ciber-físico IoT para simular la operación de sistemas de distribución con múltiples microrredes.

1.2. Objetivos específicos

1. Diseñar la configuración de las capas del módulo ciber-físico IoT considerando las características de los sistemas de distribución con múltiples microrredes.

2. Implementar el diseño del módulo ciber-físico IoT para la simulación del sistema de distribución con múltiples microrredes.

3. Validar la operación del módulo ciber-físico por medio de una estrategia de asignación de precios minoristas.

1.3. Esquema del trabajo de investigación

Este trabajo de investigación está organizado de la siguiente manera:

- **Capítulo 1:** Presenta una contextualización de las necesidades por las cuales surge este trabajo de investigación. En este capítulo se presenta la introducción y los objetivos del trabajo realizado.

- **Capítulo 2:** Presenta el estado del arte de las plataformas IoT usadas en el contexto de la gestión de las redes eléctricas. En este capítulo se describen los avances recientes en el campo de los sistemas IoT implementados en los sistemas eléctricos con múltiples MGs.

- **Capítulo 3:** Presenta los requerimientos del módulo ciber-físico IoT. En este capítulo se detallan los requisitos de diseño.
- **Capítulo 4:** Presenta el diseño del módulo ciber-físico IoT. En este capítulo se describen el software y el hardware, así como su funcionamiento.
- **Capítulo 5:** Presenta la verificación del sistema. En este capítulo se describen los experimentos utilizados para la validación del correcto funcionamiento del módulo ciber-físico IoT.
- **Capítulo 6:** Presenta las conclusiones del proyecto y se identifican áreas clave para la continuación de investigaciones futuras.

Al final del documento, en el anexo 1, se presenta brevemente la guía para el uso del módulo ciber-físico IoT.

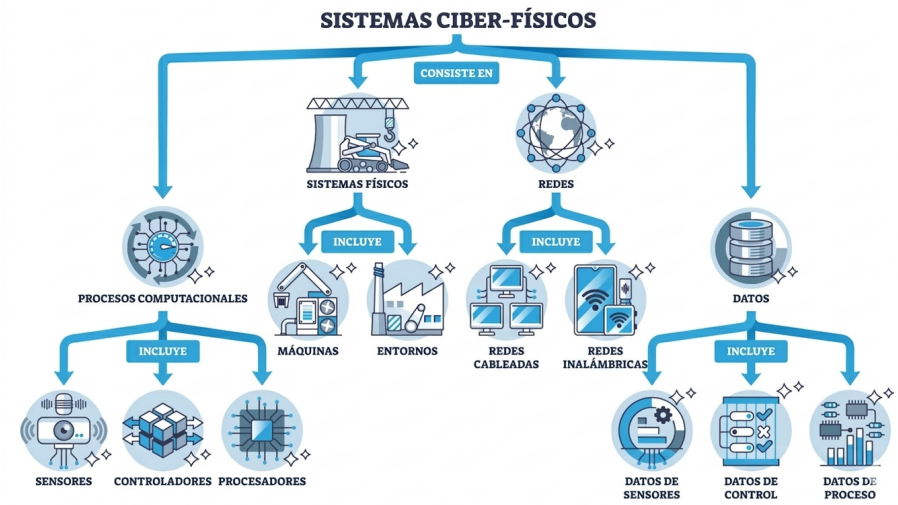
2. Estado del arte

2.1. Sistemas ciber-físicos

Los sistemas ciber-físicos (CPS, por sus siglas en inglés) han adquirido una relevancia significativa en múltiples áreas de la ingeniería (Meera and Arjun, 2024). Los CPS han sido ampliamente estudiados como una evolución natural de los sistemas embebidos y las infraestructuras inteligentes y pueden definirse como sistemas que integran procesos físicos con capacidades de computación, comunicación y control. Estos sistemas combinan hardware, software y redes de comunicación para supervisar, analizar y actuar sobre procesos físicos en tiempo real o de simulación (Kayamboo et al., 2022). Esta integración permite cerrar el ciclo de sensado–procesamiento–actuación, generando entornos altamente automatizados, adaptativos y resilientes. La Figura 2 presenta los componentes fundamentales de un CPS.

Figura 2

Diagrama general de los componentes de un CPS. Adaptado de (Farnós, 2025).



En el contexto de los sistemas eléctricos, la implementación de CPS requiere un análisis detallado de las redes de comunicación y del desempeño de la Infraestructura de Medición Avanzada (AMI), dado que estos elementos impactan directamente la supervisión, el control y la confiabilidad operativa del sistema (Torres-Olguin et al., 2023; Kayamboo et al., 2022; Cicceri et al., 2023).

La adecuada coordinación entre medición, comunicación y control permite mejorar la planificación del sistema, incrementar su seguridad y optimizar la toma de decisiones en escenarios dinámicos. Cuando múltiples CPS interactúan para coordinar procesos físicos complejos en tiempo real, se conforma lo que se conoce como un Sistema Ciber-físico Distribuido (DCPS). En aplicaciones orientadas a la simulación y operación de sistemas eléctricos, un DCPS suele estructurarse en al menos tres capas principales (Sahraei Manjili et al., 2012; Torres-Olguin et al., 2023):

- **Capa física:** Incluye todos los componentes tangibles del sistema eléctrico, tales como fuentes de generación, sistemas de almacenamiento de energía, líneas de transmisión y distribución, transformadores, dispositivos AMI y equipos IoT. Esta capa representa la infraestructura energética real o emulada, donde ocurren los procesos eléctricos.

- **Capa de red:** Permite la comunicación bidireccional entre la capa física y la capa de control, garantizando coordinación y operación confiable. Se fundamenta en infraestructuras de comunicación distribuidas, ya sea mediante enlaces cableados, inalámbricos o híbridos, apoyándose en tecnologías AMI avanzadas y soluciones basadas en IoT para el intercambio continuo de datos.

- **Capa de control:** Procesa la información proveniente de la capa física y genera señales

de control orientadas a cumplir objetivos específicos, tales como optimización económica, estabilidad o gestión de la demanda. Puede apoyarse en múltiples servidores o nodos de procesamiento para administrar el tráfico de comunicaciones y emitir comandos personalizados hacia los dispositivos físicos, en función del estado actual del sistema eléctrico.

2.2. Internet de las cosas (IoT)

El IoT se fundamenta en la integración de tecnologías de sensado, procesamiento embebido y comunicación en red, permitiendo la interacción entre dispositivos heterogéneos y sistemas distribuidos (Liao et al., 2018; Dorsemaine et al., 2015; Xu et al., 2014). Esta infraestructura posibilita la recolección continua de datos, el monitoreo remoto y la automatización de procesos en múltiples dominios, incluyendo la industria, la salud, el transporte y, de manera particular, los sistemas energéticos inteligentes.

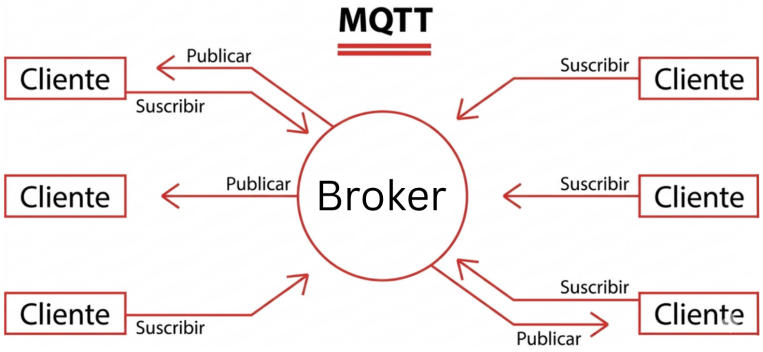
El IoT ha experimentado una adopción creciente en una amplia variedad de aplicaciones, consolidándose como una tecnología habilitadora en múltiples sectores con múltiples protocolos de comunicación. En la literatura se encuentran desarrollos que van desde sistemas de control remoto y seguimiento mediante GPS utilizando el protocolo MQTT (Aroon, 2016), hasta el diseño de inversores controlados remotamente basados en este mismo protocolo (Yang et al., 2021). Asimismo, MQTT ha sido implementado en entornos de ciudades inteligentes (Kaushik and Bagga, 2021) y en estaciones meteorológicas para la adquisición y transmisión de datos ambientales en tiempo real (Mohapatra and Subudhi, 2022; Amelia et al., 2020).

2.3. MQTT

El protocolo MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y eficiente, diseñado específicamente para la comunicación entre dispositivos en redes con recursos limitados y conexiones inestables. MQTT se destaca por su ligereza, escalabilidad, eficiencia en el uso de recursos y facilidad de implementación (Bender et al., 2021). Estas propiedades lo convierten en una solución adecuada para dispositivos con capacidades computacionales limitadas, como los empleados en arquitecturas IoT. Además, el protocolo se caracteriza por su bajo consumo energético y por una estructura organizada de distribución de información hacia los dispositivos receptores (Amjad et al., 2021). MQTT se basa en una arquitectura de publicación–suscripción (publish/subscribe), en la cual los dispositivos publicadores envían información a un *broker* central, mientras que los dispositivos suscriptores reciben los datos correspondientes a los tópicos de interés (Kaskatiiski and Boyanov, 2021). Esta arquitectura desacopla espacial y temporalmente a los dispositivos, facilitando comunicaciones múltiples, escalables y distribuidas. La Figura 3 presenta la arquitectura general del protocolo MQTT.

Figura 3

Diagrama de funcionamiento del protocolo MQTT. Adaptado de (PickData, 2019).



2.4. OpenADR

OpenADR (Open Automated Demand Response) es un protocolo de comunicación estandarizado diseñado para automatizar el intercambio de señales entre operadores del sistema eléctrico y usuarios finales o DERs (McParland, 2011; Gao et al., 2013; Yi et al., 2015). Su principal objetivo es facilitar la implementación de programas de respuesta a la demanda (Demand Response, DR), permitiendo que los consumidores ajusten su consumo o generación en función de señales externas, como precios dinámicos o eventos de reducción de carga.

OpenADR fue desarrollado bajo estándares internacionales para garantizar interoperabilidad, escalabilidad y seguridad, siendo ampliamente utilizado en redes eléctricas inteligentes (*Smart Grids*) (Rahman and Parizad, 2025). Este protocolo define dos entidades principales y su diagrama de comunicación general se presenta en la Figura 4:

- **VTN (Virtual Top Node)**: Representa al operador del sistema, agregador o entidad de mercado que envía señales de control, precios o eventos.
- **VEN (Virtual End Node)**: Corresponde al cliente o recurso energético que recibe las señales y ejecuta acciones automáticas en respuesta a ellas.

Esta estructura jerárquica permite una comunicación organizada y desacoplada entre el operador y múltiples recursos distribuidos. Sus características principales se presentan a continuación:

1. **Estandarización internacional:** Cumple con especificaciones reconocidas que permiten la interoperabilidad entre distintos fabricantes y plataformas.

2. **Automatización de la respuesta a la demanda:** Permite que los recursos energéticos reaccionen automáticamente ante eventos como: Reducción obligatoria de carga, señales de precios dinámicos y restricciones operativas del sistema (El Meslouhi et al., 2025).

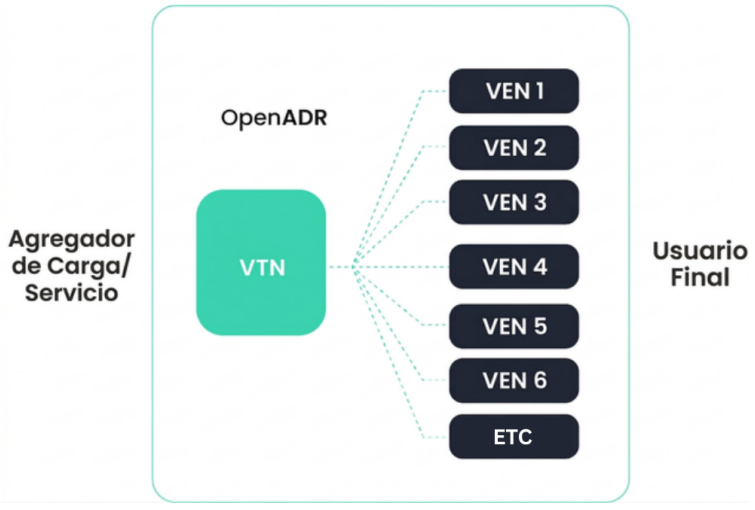
3. **Comunicación segura:** Implementa mecanismos de autenticación y cifrado (basados en TLS/HTTPS), garantizando la integridad y confidencialidad de los datos intercambiados (Pereira and Gomes, 2018).

4. **Escalabilidad:** Un único VTN puede gestionar múltiples VEN, lo que permite su aplicación en sistemas con gran cantidad de usuarios o recursos distribuidos.

5. **Interoperabilidad con sistemas energéticos inteligentes:** Está diseñado específicamente para integrarse en entornos de *smart grids*, mercados eléctricos y sistemas con generación distribuida (Kawoosa and Prashar, 2021).

Figura 4

Diagrama comunicación protocolo OpenADR. Adaptado de (Lumos Controls, 2023).



2.5. Aplicaciones y trabajos relacionados

La integración de los CPSs en infraestructuras energéticas ha permitido avanzar hacia sistemas eléctricos más inteligentes, capaces de monitorear, analizar y controlar procesos físicos en tiempo real mediante la interacción entre componentes computacionales, redes de comunicación y dispositivos físicos. En este contexto, los protocolos de comunicación juegan un papel fundamental para garantizar el intercambio eficiente de información entre los distintos elementos del sistema. Gracias a sus características, MQTT resulta particularmente adecuado para aplicaciones que requieren infraestructuras de comunicación escalables y confiables, como es el caso de los sistemas eléctricos inteligentes y las redes energéticas distribuidas (Ghani et al., 2024; Meyer et al., 2017; Budiawan et al., 2018).

Un aspecto relevante en la implementación de CPS en sistemas eléctricos es la seguridad y confiabilidad de la comunicación, dado que estos sistemas están expuestos a diversos riesgos asociados a la transmisión de datos en redes abiertas. En este sentido, la implementación de MQTT junto con mecanismos de seguridad como TLS/SSL permite establecer canales de comunicación cifrados, garantizando la confidencialidad e integridad de la información intercambiada entre los dispositivos del sistema (Khudhur and Croock, 2021; Faro et al., 2020). Asimismo, diferentes estudios han demostrado que la arquitectura basada en MQTT puede contribuir a mejorar la resiliencia del sistema frente a ataques cibernéticos, incluyendo ataques de denegación de servicio (DoS), lo cual es especialmente relevante en infraestructuras críticas como las redes eléctricas (Khudhur and Croock, 2021).

En el ámbito específico de los sistemas de potencia, el uso de MQTT dentro de arquitecturas CPS ha sido aplicado en diversas soluciones orientadas a la supervisión, control y automatización de redes inteligentes. Estas aplicaciones incluyen la monitorización en tiempo real de los DERs, la gestión de MGs y el intercambio de información entre dispositivos inteligentes dentro de la red eléctrica (Hasan et al., 2023; Cintuglu et al., 2020). Además, en sistemas energéticos locales inteligentes (Smart Local Energy Systems – SLES), el uso de MQTT permite coordinar tecnologías distribuidas y facilitar procesos de optimización energética multiobjetivo orientados a mejorar la eficiencia y sostenibilidad del sistema (Dong et al., 2022).

Diversos desarrollos experimentales han demostrado la viabilidad de esta integración. Por ejemplo, algunas implementaciones utilizan plataformas embebidas basadas en Raspberry Pi para desarrollar sistemas de monitoreo y control energético, en los cuales MQTT actúa como protocolo de comunicación entre sensores, actuadores y plataformas de visualización o análisis de datos. Estas arquitecturas suelen complementarse con interfaces hombre–máquina (HMI) y sistemas de almacenamiento de datos históricos, permitiendo el análisis y la gestión de la información generada por el sistema en tiempo real (Lin et al., 2024). Este tipo de implementaciones evidencia el potencial de los CPS basados en MQTT para el desarrollo de plataformas experimentales orientadas al estudio y validación de estrategias avanzadas de control y gestión energética.

A pesar de los avances reportados en la literatura sobre la aplicación de los CPSs en redes eléctricas inteligentes, aún existen limitaciones en la implementación práctica de arquitecturas distribuidas que integren de manera eficiente protocolos de comunicación ligeros como MQTT con plataformas de control y simulación de sistemas eléctricos. Diversos estudios han analizado el uso

de MQTT para la transmisión de datos en entornos IoT y su aplicación en la monitorización y control de MGs y DERs; sin embargo, la mayoría de estos trabajos se centran en aspectos específicos como la comunicación, la seguridad o la supervisión de dispositivos, sin abordar de forma integral la coordinación distribuida entre múltiples MGs dentro de un CPS. Asimismo, aunque protocolos estandarizados de respuesta a la demanda como OpenADR han sido ampliamente estudiados para la gestión de flexibilidad energética, su integración con arquitecturas CPS basadas en MQTT sigue siendo limitada en la literatura actual. En consecuencia, existe una necesidad de desarrollar y evaluar plataformas experimentales que combinen CPS, MQTT y modelos de sistemas eléctricos, permitiendo estudiar la operación coordinada de MGs, la resiliencia ante fallos de comunicación y la interoperabilidad con protocolos de gestión energética. Este trabajo busca contribuir en esta dirección mediante el desarrollo de un módulo ciber-físico IoT para la simulación y operación distribuida de sistemas de distribución con múltiples MGs.

Para llevar a cabo la contribución y el desarrollo de este proyecto se tomó como referencia los trabajos (Nguyen et al., 2019; Nelson et al., 2016). En la literatura reciente se han desarrollado plataformas experimentales ciber-físicas orientadas a la validación de estrategias de control y operación en MGs. En (Nguyen et al., 2019) se presenta la implementación de un flujo óptimo de potencia distribuido (OPF) en una MG ciber-física utilizando un entorno hardware-in-the-loop, donde un simulador digital en tiempo real OPAL-RT se integra con un conjunto de Raspberry Pi que ejecutan agentes distribuidos basados en el algoritmo ADMM para resolver el problema de optimización utilizando únicamente información local y de nodos adyacentes. Este enfoque demuestra la viabilidad de emplear dispositivos embebidos y arquitecturas distribuidas para el control

óptimo de MGs en tiempo real. De manera complementaria, en (Nelson et al., 2016) se presenta una plataforma de prueba ciber-física para MGs que combina simulación en tiempo real, hardware físico y simuladores de red, permitiendo analizar el impacto de las condiciones de comunicación como retardos de red y procesamiento de routers en el desempeño de los sistemas de control. Los resultados evidencian que las limitaciones en la infraestructura de comunicación pueden afectar significativamente la capacidad del controlador para mantener condiciones operativas deseadas en la red eléctrica. Estos trabajos constituyen referencias relevantes para el desarrollo de plataformas ciber-físicas aplicadas a sistemas eléctricos, y sirven como base conceptual para la implementación del módulo ciber-físico IoT propuesto en esta investigación, orientado a la simulación y operación distribuida de múltiples MGs mediante mecanismos de comunicación y coordinación entre dispositivos.

La tabla 1 resume las principales características de los trabajos analizados frente al módulo ciber-físico propuesto, considerando cuatro criterios clave: (i) bajo coste, (ii) comunicación basada en IIoT, (iii) arquitectura descentralizada y (iv) compatibilidad con simulaciones en tiempo real y desacopladas en el tiempo. La revisión muestra que las plataformas con alta fidelidad experimental, especialmente las basadas en OPAL-RT o hardware propietario, presentan altos costos y baja accesibilidad Nguyen et al. (2019); Nelson et al. (2016); Guo et al. (2013); Xie et al. (2018). Por otro lado, los entornos de co-simulación basados en software son escalables y económicos, pero limitados a operación fuera de línea y sin protocolos IIoT ligeros Ciraci et al. (2014); Palmintier et al. (2017). Asimismo, infraestructuras de gran escala Blank et al. (2016) ofrecen rigor metodológico, aunque con arquitecturas centralizadas y poco accesibles.

Ningún trabajo integra simultáneamente comunicación IIoT mediante MQTT, interoperabilidad OpenADR 2.0, arquitectura distribuida y compatibilidad con diferentes modos de simulación manteniendo bajo costo mediante miniordenadores y software libre. El módulo propuesto aborda esta limitación, ofreciendo una plataforma replicable, accesible y orientada a investigación y formación en redes inteligentes.

Tabla 1

Tabla comparativa de los bancos de pruebas de los CPS.

Referencia	**Bajo costo**	**Protocolo IIoT**	**Arquitectura distribuida**	**Tiempo real y desacoplado**
Ref. Nguyen et al. (2019)	✘	✘	✓	✓
Ref. Nelson et al. (2016)	✘	✓	✘	✓
Ref. Guo et al. (2013)	✘	✘	✘	✓
Ref. Ciraci et al. (2014)	✓	✘	✘	✘
Ref. Palmintier et al. (2017)	✓	✘	✘	✘
Ref. Blank et al. (2016)	✘	✘	✘	✓
Ref. Xie et al. (2018)	✘	✘	✘	✓
Este trabajo	✓	✓	✓	✓

3. Requerimientos del sistema

Con el fin de definir las características esperadas del módulo ciber-físico IoT, se establecieron una serie de requerimientos funcionales y no funcionales asociados a la operación, comunicación y validación del sistema. Estos requerimientos permiten verificar el correcto funcionamiento del módulo y garantizar la trazabilidad entre los objetivos planteados y las fases de validación desarrolladas durante la investigación. La Tabla 2 presenta los principales requerimientos definidos para el desarrollo del módulo.

3.1. Características del módulo ciber-físico IoT

El módulo ciber-físico está diseñado para realizar simulaciones que contengan la componente de comunicación sobre el intercambio de información bidireccional entre el OR y las MGs del sistema de distribución. Este está conformado por:

1. **Ordenador central.** Representa el cerebro y la capa de control del sistema, actúa como OR. En este componente se ejecutan los algoritmos de gestion de energía, se envían señales de control a las tarjetas que simulan a las MGs y se reciben las variables de operación correspondientes. Adicionalmente, se realiza el cálculo del flujo de potencia del sistema eléctrico.

2. **Tarjetas Raspberry Pi y Odroids.** Corresponden a miniordenadores que simulan las MGs en el sistema de distribución. Cada tarjeta recibe información del ordenador central y resuelve un problema de optimización interno para decidir, de acuerdo a sus recursos energéticos y su demanda local, si compra o vende energía al sistema de distribución.

3. **Sistema de comunicación IoT.** El intercambio de información entre el ordenador central y las tarjetas, se realiza mediante un sistema de comunicación híbrido basado en IoT, utilizando el protocolo OpenADR y MQTT. MQTT emplea un servidor central a través del cual se gestionan los mensajes intercambiados entre el ordenador central y las tarjetas, facilitando la comunicación eficiente y bidireccional.

El diagrama general del módulo se presenta en la Figura 5, donde sus principales capas son la capa de comunicaciones y la capa de potencia. La implementación del módulo se presenta en la Figura 6.

Figura 5

Diagrama general módulo ciber-físico IoT.

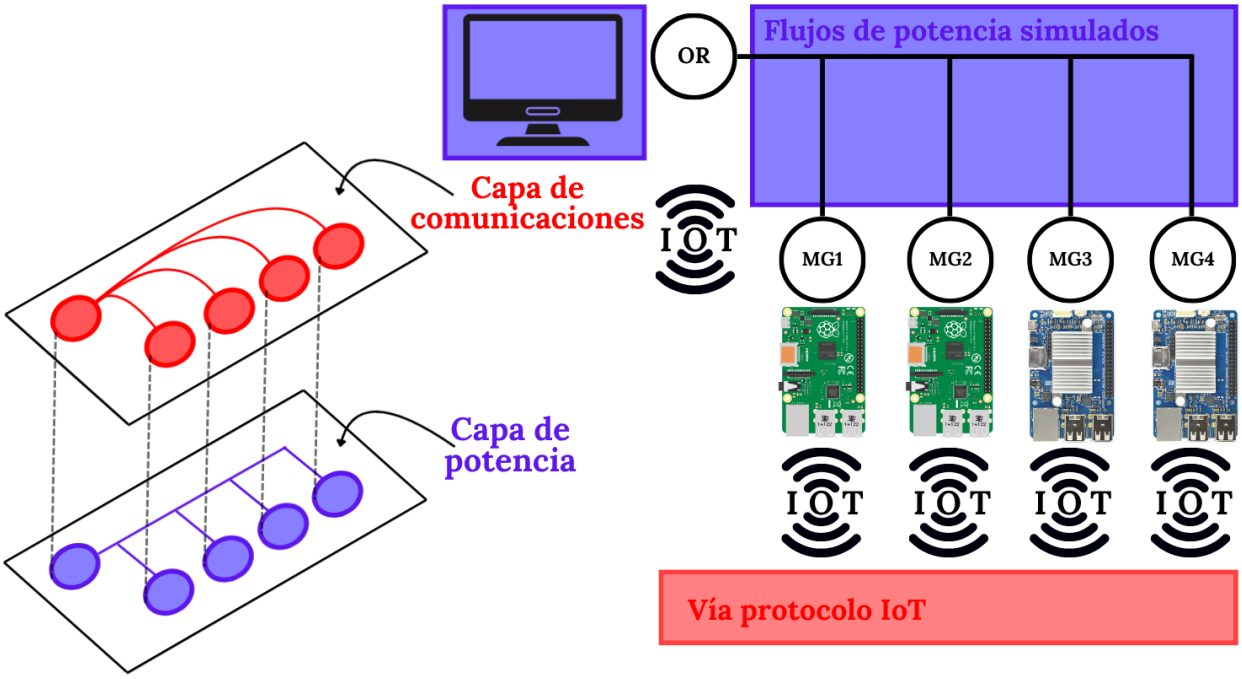
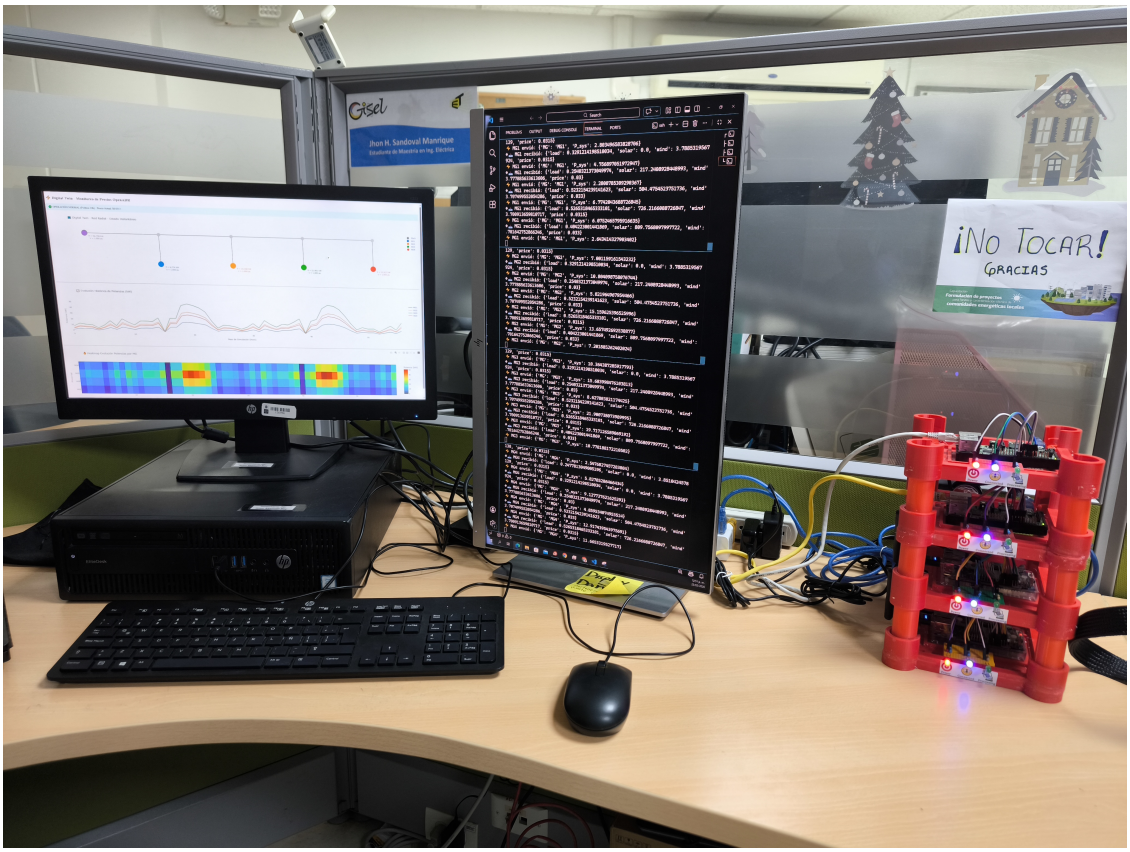


Figura 6

Implementación módulo ciber-físico IoT.



3.1.1. *Funcionamiento y selección de componentes*

El módulo ciber-físico se encuentra estructurado de acuerdo con las tres capas fundamentales de un sistema ciber-físico: capa física, capa de control y capa de red. La capa de control es la encargada de procesar la información del sistema y enviar señales de control a la capa física a través de la capa de red, con el fin de garantizar el cumplimiento de un objetivo operativo común.

Para la selección de los componentes del módulo se realizaron pruebas con distintos dispositivos y plataformas. En el caso del ordenador central, se determinó que este puede ser cualquier

computador con capacidad computacional suficiente para ejecutar cálculos de flujo de potencia y procesos de optimización, utilizando herramientas como PandaPower o AMPL, integradas mediante Python y sus respectivas librerías.

En cuanto a la capa física, se evaluaron diferentes dispositivos como Arduino, Raspberry Pi y Odroid, concluyendo que se requieren miniordenadores capaces de ejecutar Python y sus librerías actualizadas, tales como NumPy, SciPy, Pandas y Paho-MQTT, necesarias para la resolución de los problemas de optimización y la comunicación con el sistema central.

Finalmente, para la capa de red del módulo ciber-físico se seleccionó el protocolo MQTT, complementado con la integración del estándar OpenADR, debido a sus características de comunicación ligera, estandarizada, escalable y ampliamente utilizada en aplicaciones IoT y sistemas eléctricos inteligentes. El protocolo MQTT fue adoptado como mecanismo principal de comunicación interna del módulo, permitiendo organizar la información mediante temas estructurados, lo que garantiza que el ordenador central que simula al OR tenga acceso a la información de todas las tarjetas que simulan a las MGs, evitando la comunicación directa entre ellas y manteniendo una arquitectura jerárquica y controlada. Esta estructura facilita la sincronización, el intercambio bidireccional de datos y la implementación de estrategias de control distribuido. Por su parte, OpenADR se integra como un protocolo de nivel superior orientado a la comunicación estandarizada entre sistemas eléctricos y operadores de mercado.

3.1.2. *Uso científico y trabajo publicado*

El módulo ciber-físico IoT fue diseñado y desarrollado con el objetivo de contar con una herramienta de uso académico y científico en el ámbito de los sistemas eléctricos y las redes de distribución. Desde el punto de vista académico, el módulo permite la realización de pruebas y laboratorios orientados a la simulación de microrredes en redes eléctricas y sistemas eléctricos con generación distribuida, facilitando el análisis de estrategias de optimización descentralizada en un entorno controlado y realista.

En el ámbito científico, el módulo ha sido utilizado como plataforma para la simulación distribuida de la operación de un sistema eléctrico basada en IoT. Como resultado de este trabajo, se realizó la publicación titulada “Sistema ciber-físico distribuido basado en IoT para simular la operación de las redes eléctricas”, presentada en el evento SICEL 2025 en San Juan, Argentina.

3.1.3. *Colaboración y apoyo*

El proyecto desarrollado corresponde a un esfuerzo colaborativo que involucra a:

- **Profesores y grupo de investigación:** El proyecto fue dirigido por el profesor Juan Manuel Rey López y codirigido por el profesor César Antonio Duarte Gualdrón, y cuenta con el apoyo de los profesores del grupo de investigación de Sistemas de Energía Eléctrica (GISEL) de la UIS, incluyendo al profesor Iván David Serna Suárez.
- **Colaboración internacional:** Científicos como Nilson Henao en apoyo con el grupo de investigación LIREI de la Universidad de Quebec de Trois-Rivieres de Canadá.

Tabla 2

Requerimientos del módulo ciber-físico IoT

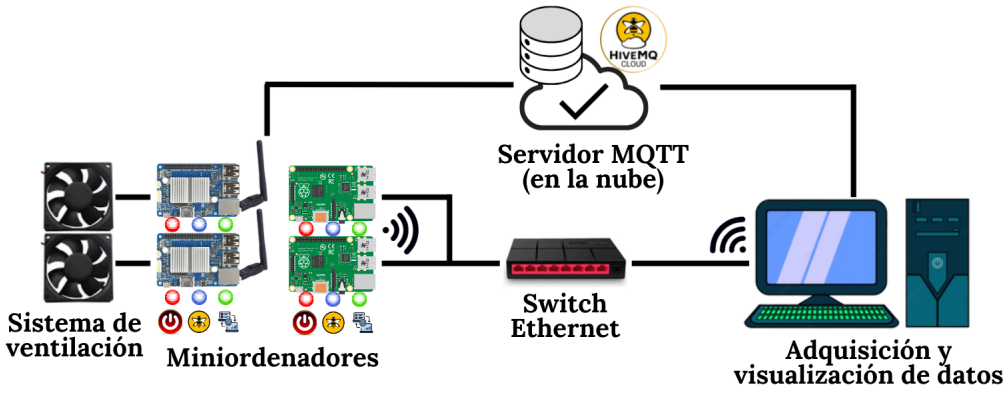
ID	Requerimiento	Tipo	Criterio de aceptación	Objetivo	Fase de validación
RF-01	Permitir comunicación bidireccional entre el ordenador central y las microrredes mediante MQTT.	Funcional	Intercambio exitoso de mensajes entre todos los dispositivos conectados.	Obj. 1 y 2	Fase 1
RF-02	Ejecutar procesos de optimización distribuida en cada tarjeta que representa una microrred.	Funcional	Obtención de resultados coherentes y sincronizados entre dispositivos.	Obj. 2 y 3	Fase 3 y 5
RF-03	Visualizar en tiempo real las variables del sistema eléctrico mediante una interfaz gráfica.	Funcional	Actualización continua y correcta de la información en el dashboard.	Obj. 2	Fase 2
RF-04	Integrar comunicación híbrida mediante MQTT y OpenADR.	Funcional	Recepción y procesamiento correcto de eventos OpenADR y traducción a MQTT.	Obj. 2 y 3	Fase 5
RNF-01	Garantizar tolerancia a fallos en la comunicación distribuida.	No funcional	Reconexión automática y continuidad de operación ante desconexiones.	Obj. 2	Fase 4
RNF-02	Garantizar confiabilidad en la entrega de mensajes.	No funcional	Uso correcto de QoS 2 evitando pérdida o duplicidad de mensajes.	Obj. 2	Fase 4
RNF-03	Permitir escalabilidad y modularidad del sistema.	No funcional	Posibilidad de agregar nuevas tarjetas o microrredes al módulo.	Obj. 1	Validación general
RNF-04	Implementar una arquitectura de bajo costo y fácil reproducibilidad.	No funcional	Uso de hardware embebido comercial y software de libre acceso.	Obj. 1	Diseño e implementación

4. Diseño del módulo ciber-físico IoT

A continuación se realiza una descripción detallada de cada uno de los componentes que conforman el sistema. La arquitectura del módulo ciber-físico IoT se muestra en la Figura 7. Los componentes del sistema se dividen en 3 componentes: hardware, software y comunicaciones.

Figura 7

Esquema general del módulo ciber-físico IoT.



4.1. Hardware

Para la implementación del hardware del sistema ciber-físico, se utilizó un ordenador central, equipado con un procesador Intel Core i7, el cual se empleó para ejecutar las tareas de control, comunicación y simulación del sistema eléctrico. En cuanto a las microrredes, estas fueron simuladas mediante cuatro miniordenadores, correspondientes a dos tarjetas Raspberry Pi 2 Model B y dos tarjetas Odroid. Las Raspberry Pi se conectaron al sistema a través de Ethernet, mientras que las tarjetas odroid se conectaron a internet mediante comunicación inalámbrica (Wi-Fi), utilizando una antena externa. Durante las pruebas experimentales, la latencia entre dispositivos resultó

prácticamente imperceptible, permitiendo un intercambio de información estable y eficiente para la operación del módulo.

Adicionalmente, se empleó un switch ethernet de ocho puertos para facilitar la conexión a internet del ordenador central y de las tarjetas Raspberry Pi, dejando puertos disponibles para una eventual escalabilidad del sistema mediante la incorporación de nuevas tarjetas. Finalmente, se realizó el montaje de un sistema de ventilación para las tarjetas, dado que, tras periodos prolongados de operación, se observó un incremento en la temperatura de los dispositivos, lo cual hacía necesario garantizar condiciones adecuadas de funcionamiento. La estructura del módulo fue diseñada en Tinkercad, tanto para la base de las tarjetas y sus conexiones entre bases, como el soporte del sistema de ventilación. El montaje del módulo ciber-físico propuesto está compuesto por los siguientes elementos y sus características se presentan en la Tabla 3:

- Ordenador central
- Cuatro tarjetas de cómputo (dos raspberry pi y dos odroid)
- Estructura física del módulo
- Un switch Ethernet
- Tres cables Ethernet
- Dos antenas Wi-Fi (para las odroid)
- Dos ventiladores para la ventilación

- Cuatro fuentes de alimentación independientes para las tarjetas
- Una fuente de alimentación para los ventiladores
- Cuatro mini protoboards
- Doce luces LED utilizadas como indicadores
- Doce resistencias de 470 ohmios
- Cables jumpers de conexión hembra-macho

Tabla 3

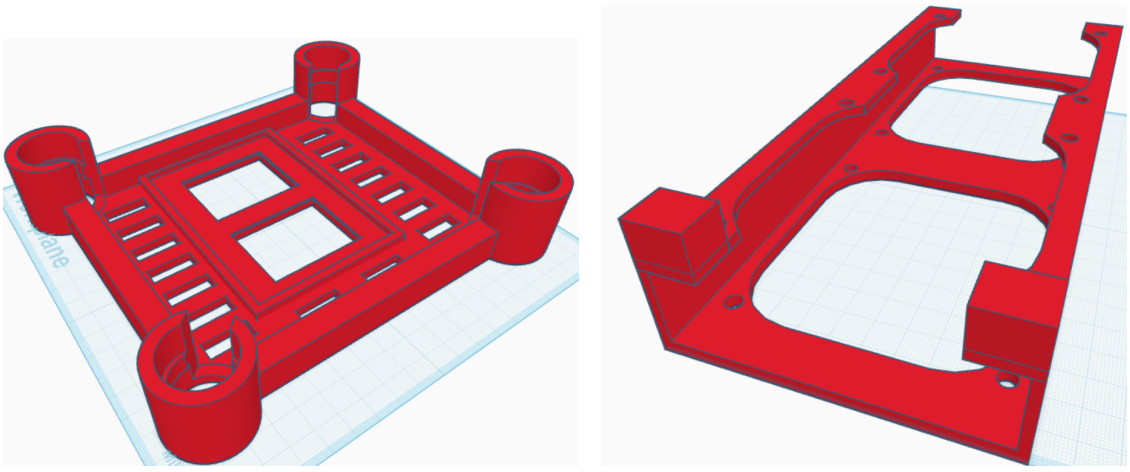
Descripción técnica de los dispositivos de cómputo y red utilizados.

Especificaciones de Hardware	Dispositivo
PC de Escritorio: Procesador Intel Core i7-6700 @ 3.40GHz, RAM 12.0 GB, sistema de 64 bits. Almacenamiento en disco local de 291 GB.	Ordenador Central
Raspberry Pi 2 Modelo B: SoC Broadcom BCM2836 (Quad-core Cortex-A7 @ 900 MHz), 1 GB RAM LPDDR2, Ethernet 10/100 Mbps. Almacenamiento mediante tarjeta MicroSD.	Dispositivo 1 y 2
Odroid-C1+: SoC Amlogic S805 (Quad-core Cortex-A5), 1GB RAM DDR3, Gigabit Ethernet. Almacenamiento: socket para módulo eMMC 4.5 y ranura MicroSD UHS-1.	Dispositivo 3 y 4
Switch Ethernet Mercusys MS108G: 8 puertos 10/100/1000Mbps con auto-negociación, soporte para Half/Full Duplex y MDI/MDIX. Tasas de transferencia de hasta 2000Mbps en Full Duplex.	Conectividad Red
Ventilador Techman VN-4051: Ventilador de alta fluidez (High Fluidity Box Fan), alimentación de 12V DC, corriente de 180mA, velocidad de 2800 RPM. Dimensiones: 92x92x25 mm.	Ventilación

Adicionalmente, se realizó el diseño tridimensional (3D) de la estructura física que soporta los componentes del módulo ciber-físico IoT. Este diseño fue concebido con el objetivo de garantizar una adecuada organización del cableado, correcta disposición de los dispositivos, ventilación apropiada y facilidad de mantenimiento y escalabilidad del sistema. Su diseño se muestra en la Figura 8.

Figura 8

Diseño 3D de la estructura física del módulo ciber-físico IoT.



(a) Base para las tarjetas RPi y Odroid (b) Soporte base para sistema de ventilación

4.2. Software

A continuación, se describen los principales componentes de software utilizados y requeridos para el desarrollo e implementación del módulo ciberfísico IoT.

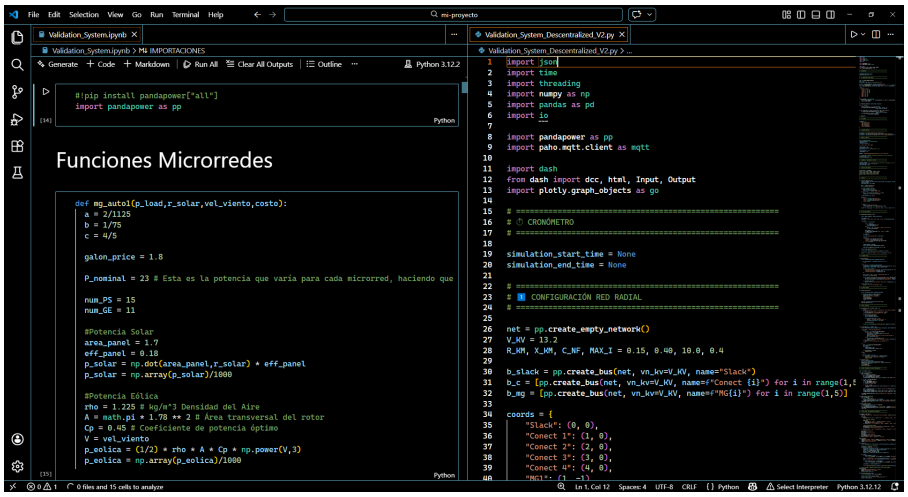
4.2.1. *Visual Studio Code y Python*

La implementación del módulo se llevó a cabo utilizando el entorno de desarrollo Visual Studio Code (VSCode) y Python, debido a sus características, versatilidad y ventajas específicas para este tipo de aplicaciones. VSCode, gracias a su interfaz intuitiva, sus múltiples opciones de configuración y su alto nivel de extensibilidad, permitió un desarrollo ágil y eficiente del proyecto. Por su parte, Python fue seleccionado por su amplia disponibilidad de librerías especializadas orientadas a la optimización, la comunicación y la visualización de datos, lo que facilitó la implementación.

Entre las principales librerías empleadas se encuentran **NumPy**, **Pandas**, **PandaPower**, **SciPy**, **Paho-MQTT**, **Dash** y **Plotly (graph_objects)**, las cuales constituyeron una base fundamental para el desarrollo del sistema, evitando la necesidad de implementar funcionalidades desde cero y optimizando significativamente el tiempo de desarrollo. Estas herramientas permitieron tanto la resolución de problemas de optimización, simulaciones del flujo de carga, como la implementación de la comunicación IoT y el desarrollo de interfaces gráficas para la visualización de las simulaciones, las cuales se describen en secciones posteriores. En la Figura 9 se presenta la interfaz de VSCode utilizada para el desarrollo de las simulaciones sobre el módulo en Python.

Figura 9

Interfaz de Visual Studio Code con Python.



4.2.2. *Interfaz con Dashboard*

Mediante el uso de Python y las librerías previamente mencionadas, se desarrolló una interfaz gráfica interactiva que permite visualizar en tiempo real el sistema eléctrico bajo estudio. Esta funcionalidad es posible gracias a que tanto las tarjetas que simulan las MGs como el script en Python encargado del *dashboard* se encuentran conectados al broker MQTT HiveMQ, permitiendo un intercambio continuo y sincronizado de información.

La interfaz muestra el estado operativo actual del sistema eléctrico, incluyendo variables como potencia en cada MG. Asimismo, permite consultar el historial de evolución de la potencia, representado mediante una gráfica temporal continua y mapa de calor (*heatmap*), lo que facilita la identificación de patrones y variaciones en el comportamiento del sistema. Adicionalmente, se muestra un cronómetro que resulta útil para visualizar el tiempo de simulación real. Por otra parte,

la interfaz ofrece la posibilidad de descargar los resultados en formato Excel, generando archivos con el registro histórico completo de la simulación. Esta funcionalidad facilita el análisis posterior, la comparación de escenarios y la validación cuantitativa de los resultados obtenidos. En la figura 10 se presenta la interfaz desarrollada en Dash, utilizada para la visualización en tiempo real de las actualizaciones del sistema eléctrico, la cual actúa como un gemelo digital del caso de estudio.

Figura 10

Interfaz gemelo digital caso de estudio.

4.3. Sistema de comunicaciones

El sistema de comunicaciones del módulo ciber-físico IoT se basa en una arquitectura robusta, eficiente y tolerante a fallos, diseñada para la transmisión, coordinación y monitoreo de información en sistemas eléctricos distribuidos con múltiples microrredes.

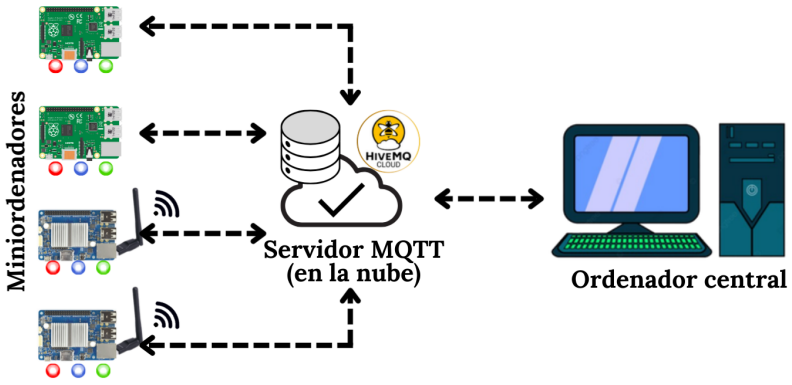
Comunicación interna mediante MQTT. La comunicación interna entre el ordenador central (OR) y las tarjetas que simulan las MGs se realiza mediante el protocolo MQTT, utilizando diferentes tópicos estructurados para organizar el intercambio bidireccional de información. Dado que el sistema opera bajo un enfoque distribuido, el ordenador central implementa un mecanismo de sincronización que garantiza que cada mensaje de control sea recibido una única vez por cada MG. Asimismo, el procesamiento global no se ejecuta hasta haber recibido la información de retorno de todas las tarjetas activas. Este mecanismo de coordinación se implementa mediante:

- Definición de timeouts para la espera de respuestas.
- Uso del nivel de *Quality of Service* (QoS) definido en MQTT según la criticidad del mensaje.
- Verificación de integridad y control de iteraciones en cada ciclo de simulación.

La gestión del intercambio de mensajes es soportada por el broker MQTT HiveMQ, el cual actúa como servidor central, permitiendo además el monitoreo en tiempo real del tráfico y las conexiones activas. La Figura 11 representa el diagrama de la comunicación del módulo.

Figura 11

Diagrama de comunicación MQTT.



Integración de OpenADR: Comunicación externa estandarizada. Adicionalmente, el módulo incorpora una capa de interoperabilidad basada en OpenADR (Open Automated Demand Response), permitiendo extender el sistema hacia un entorno compatible con estándares internacionales de comunicación en redes eléctricas inteligentes. En esta arquitectura híbrida:

- El ordenador central puede actuar como un *Virtual End Node* (VEN).
- Un sistema externo opera como *Virtual Top Node* (VTN), enviando señales tales como eventos de respuesta a la demanda, restricciones operativas o señales de precios dinámicos.
- Los eventos recibidos mediante OpenADR son procesados por el ordenador central y traducidos en comandos internos que se transmiten a las MGs a través de MQTT.

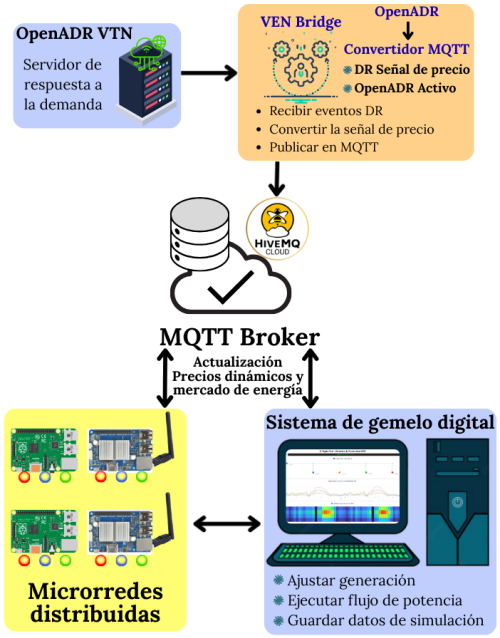
Es importante destacar que OpenADR no reemplaza MQTT, sino que lo complementa. Mientras MQTT gestiona la comunicación interna de baja latencia entre el operador y las MGs, OpenADR permite la integración del módulo con escenarios reales de mercado eléctrico y programas de gestión de demanda. De esta manera, la arquitectura propuesta combina la eficiencia operativa de MQTT con la interoperabilidad y estandarización de OpenADR, fortaleciendo el carácter académico, experimental y potencialmente implementable del módulo ciber-físico IoT.

Monitoreo y visualización en tiempo real. De forma paralela, toda la información intercambiada es transmitida hacia la interfaz de visualización desarrollada en Dash, permitiendo el monitoreo en tiempo real del estado del sistema y su evolución durante la simulación. De esta manera, el módulo no solo ejecuta procesos de control distribuido, sino que también actúa como

un gemelo digital interactivo del sistema eléctrico de estudio. La arquitectura de esta integración híbrida OpenADR-MQTT se ilustra en la Figura 12.

Figura 12

Arquitectura de control y comunicación híbrida OpenADR y MQTT.



4.3.1. *HiveMQ Cloud*

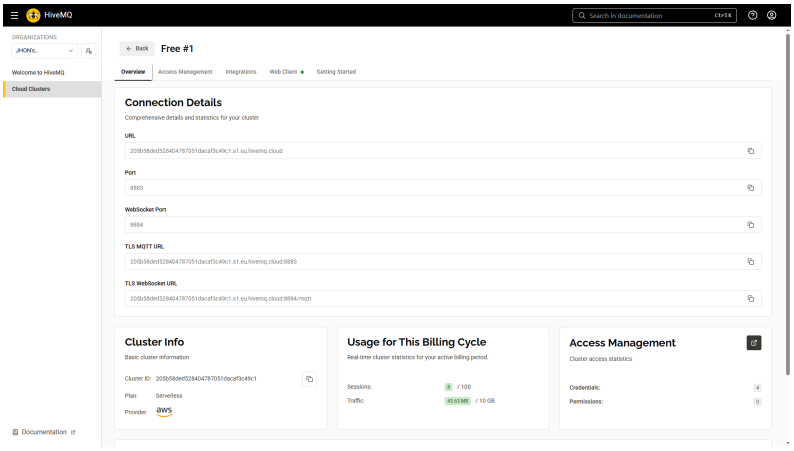
HiveMQ Cloud es un broker MQTT gestionado en la nube, basado en una arquitectura distribuida y serverless, que cumple estrictamente con los estándares del protocolo MQTT, incluyendo MQTT 5.0, así como las versiones 3.1.1 y 3.1. Esta compatibilidad permite el uso de funcionalidades avanzadas como los distintos niveles de calidad de servicio (Quality of Service, QoS), sesiones persistentes y mensajes retenidos, garantizando una comunicación flexible y confiable entre dispositivos (HiveMQ, 2024).

Uno de los aspectos más relevantes de HiveMQ para el desarrollo del módulo ciber-físico IoT es el

soporte completo de los tres niveles de QoS definidos por el protocolo MQTT: QoS 0 (como mucho una vez), QoS 1 (al menos una vez) y QoS 2 (exactamente una vez). Estos niveles permiten controlar el grado de fiabilidad en la entrega de los mensajes, lo cual resulta fundamental para la correcta sincronización entre el ordenador central y las tarjetas. En los *scripts* desarrollados en Python, la selección del nivel de QoS se utiliza para asegurar la recepción adecuada de las señales de control y de los datos de retorno, de acuerdo con la criticidad de la información transmitida. Desde el punto de vista de la arquitectura, HiveMQ opera gestionando de forma transparente la asignación de recursos y el manejo de múltiples conexiones simultáneas, lo que permite una escalabilidad automática sin necesidad de configuración adicional por parte del usuario. Además, el *broker* soporta mensajes retenidos y el almacenamiento temporal de mensajes para clientes desconectados, complementando el uso de QoS para mantener la coherencia del sistema. Adicionalmente, HiveMQ implementa mecanismos de seguridad robustos, utilizando cifrado obligatorio TLS/SSL para la transmisión de datos y autenticación mediante credenciales. La interfaz se ilustra en la Figura 13.

Figura 13

Interfaz HiveMQ (HiveMQ, 2024).

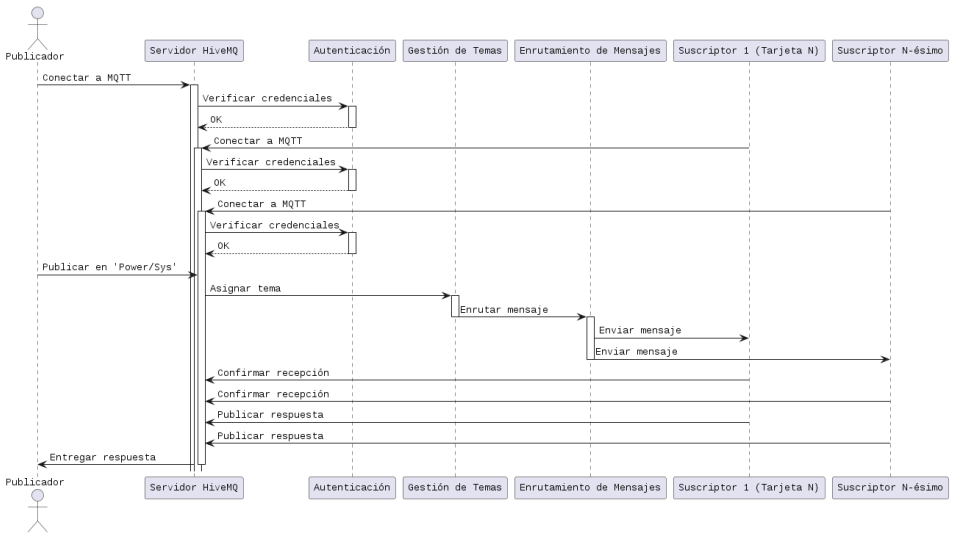


Finalmente, en la Figura 14 se presenta el diagrama de secuencia que describe el funcionamiento del módulo ciber-físico IoT y el flujo secuencial de comunicación entre sus componentes. En dicho diagrama se ilustra la interacción entre el ordenador central (publicador), el broker MQTT (servidor HiveMQ) y las tarjetas que representan las microrredes (suscriptor(es)), mostrando el proceso de: Conexión del Publicador con el servidor, Verificación de credenciales, Publicación y organización de *Topics* para cada suscriptor, Publicación y confirmación de mensajes por parte de los suscriptores y Publicación de resultados al publicador.

Este diagrama permite comprender de manera estructurada la lógica de operación del sistema, evidenciando la coordinación iterativa y el intercambio bidireccional de información que garantizan una comunicación adecuada y coherente dentro del módulo. De esta forma, se consolida la representación integral de la arquitectura implementada, tanto desde el punto de vista funcional como comunicacional.

Figura 14

Diagrama de secuencia de comunicación.



5. Verificación del módulo ciber-físico IoT

Una vez realizado el diseño del módulo ciber-físico IoT y el montaje de sus componentes, se llevó a cabo el proceso de validación del sistema, el cual se desarrolló de manera progresiva a través de pruebas experimentales enfocadas en sus partes fundamentales. Estas pruebas se organizaron de la siguiente manera.

1. **Intercambio de información a través MQTT:** Prueba experimental inicial de comunicación entre dos tarjeta Raspberry Pi y el ordenador central utilizando el protocolo MQTT, con el objetivo de verificar el correcto envío y recepción de mensajes entre los dispositivos.

2. **Visualización de la información en consola y HiveMQ e implementación de ADMM:** A partir del intercambio de información mediante Python y el broker HiveMQ, se verificó la correcta visualización de los datos y su adecuada organización en los tópicos MQTT, tanto en la consola de HiveMQ como en consola del ordenador central, además de la implementación de un caso base de optimización usando ADMM.

3. **Simulación descentralizada mediante ADMM de un sistema eléctrico:** En esta etapa se llevó a cabo la simulación de una optimización descentralizada de potencia en un sistema IEEE de 9 nodos modificado, el cual disponía de tres generadores, cada uno de ellos representado por una tarjeta del sistema ciber-físico. Como resultado de esta fase, se realizó la publicación de una conferencia internacional basada en el sistema desarrollado.

4. **Mejoras en la robustez, tolerancia a fallos y almacenamiento de datos:** Posteriormente

a la presentación del módulo en la conferencia, se implementaron mejoras orientadas a incrementar la robustez del sistema, su tolerancia a fallos y el almacenamiento de los datos de simulación generados en cada iteración o realización experimental.

5. **Validación mediante una estrategia de asignación de precios minoristas:** Finalmente, se realizó la validación del módulo mediante la simulación de una estrategia de asignación de precios minoristas, tanto de forma centralizada como descentralizada (utilizando el módulo propuesto). Los resultados obtenidos fueron comparados en términos de tiempo de simulación y error porcentual, permitiendo evaluar las diferencias entre ambos enfoques.

Las secciones siguientes detallan el proceso de validación de cada una de estas pruebas experimentales. Además, el material audiovisual de los experimentos está disponible a través de los enlaces proporcionados en cada sección.

5.1. Fase 1. Intercambio de información a través MQTT

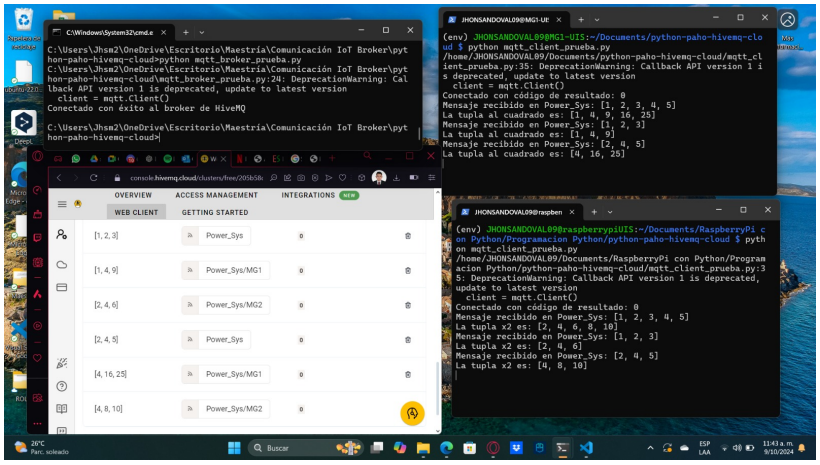
En esta fase se realizó la configuración de dos tarjetas Raspberry Pi, estableciendo la conexión a cada una de ellas mediante SSH desde el ordenador central, con el fin de permitir su control y monitoreo remoto. Posteriormente, a través de HiveMQ, se crearon las credenciales de conexión y el espacio de trabajo necesarios para enlazar las tarjetas con el *broker* MQTT.

Una vez configuradas y suscritas a los tópicos correspondientes, se llevó a cabo una prueba inicial de intercambio de información. En esta prueba, ambas tarjetas se encontraban suscritas al mismo tópico MQTT, sin embargo, cada una ejecutaba un proceso diferente sobre la información recibida. Específicamente, se realizó el envío de un vector de tres posiciones a través de la interfaz

de publicación de mensajes de HiveMQ. La primera tarjeta procesó el vector elevando cada uno de sus elementos al cuadrado, mientras que la segunda tarjeta realizó la duplicación de los valores del vector. Posteriormente, cada tarjeta publicó el resultado de su procesamiento de vuelta al *broker* MQTT. La prueba experimental correspondiente a esta fase se presenta en la Figura 15, donde se evidencia el correcto intercambio de información y la ejecución diferenciada de los procesos en cada dispositivo.

Figura 15

Interfaz de validación de intercambio de información entre dispositivos.



5.2. Fase 2. Visualización de la información en consola y HiveMQ e implementación de ADMM

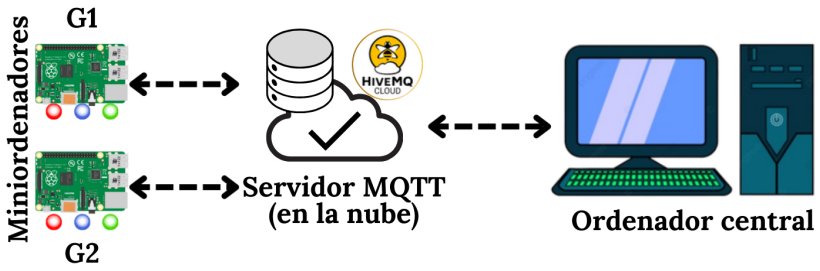
Para validar la estrategia de control y optimización del sistema, se implementó el algoritmo de multiplicadores de dirección alterna (ADMM). Esta implementación se diseñó bajo una arquitectura de computación distribuida, donde la carga computacional se reparte entre los diferentes nodos de hardware del proyecto: el ordenador central y los nodos de cómputo local (tarjetas).

Escenario de simulación. Para esta fase, el escenario de simulación se presenta en la Figura 16,

donde se muestra la interacción entre el ordenador central y las dos Raspberry Pi.

Figura 16

Escenario de validación ordenador central y 2 Raspberry Pi.



Formulación matemática del problema. El objetivo es resolver un problema de optimización para la asignación de potencia de dos generadores (G_1, G_2), sujeto a restricciones de red. El algoritmo ADMM descompone el problema en tres etapas fundamentales por iteración (k):

Paso proximal (Minimización local). Cada nodo generador (representado por una Raspberry Pi) resuelve de manera independiente y paralela su propia función de costo regularizada:

$$P_{g,k+1} = \text{prox}_{f,\rho}(\hat{P}_{g,k} - \mu_{g,k}) = \arg\min_{P_g} \left\{ f_g(P_g) + \frac{\rho}{2} \left\| P_g - \hat{P}_{g,k} + \mu_{g,k} \right\|^2 \right\} \quad (1)$$

Donde $f_g(P_g)$ es la función de costo del generador, ρ es el parámetro de penalización ($\rho = 40$), $\hat{P}_{g,k}$ es el valor de potencia proyectado en la iteración anterior y $\mu_{g,k}$ es el multiplicador de Lagrange escalado.

Las funciones de costo de cada generador se presentan a continuación:

$$G_1(P_1) = 9P_1^2 + 1200P_1 + 400000, 10 \leq P_1 \leq 80 \quad (2)$$

$$G_2(P_2) = 25P_2^2 + 1200P_2 + 100000, 10 \leq P_2 \leq 70 \tag{3}$$

Paso de proyección (Consenso global). El ordenador central actúa como el coordinador central. Recibe las soluciones locales $P_{g,k+1}$ y realiza una proyección sobre el conjunto de restricciones viables (SET):

$$\hat{P}_{k+1} = \Pi_{SET}(P_{k+1} + \boldsymbol{\mu}_k) = \arg\min_{\mathbf{y} \in SET} \|\mathbf{y} - (P_{k+1} + \boldsymbol{\mu}_k)\|^2 \tag{4}$$

Este paso garantiza que la suma de las potencias generadas cumpla con la demanda y las restricciones físicas del sistema de distribución. Donde,

$$SET -> P_1 + P_2 = 100[MW] \tag{5}$$

es la restricción en potencia del caso de estudio.

Actualización del multiplicador. Finalmente, se actualiza el residuo dual para ajustar los precios o penalizaciones en la siguiente iteración:

$$\mathbf{r}_{k+1} = \mathbf{P}_{k+1} - \hat{\mathbf{P}}_{k+1} \quad \text{(Residuo Primal)} \tag{6}$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \mathbf{r}_{k+1} \quad \text{(Actualización de Lagrange)} \tag{7}$$

$$\mathbf{s}_{k+1} = \rho(\hat{\mathbf{P}}_{k+1} - \hat{\mathbf{P}}_k) \quad \text{(Residuo Dual)} \tag{8}$$

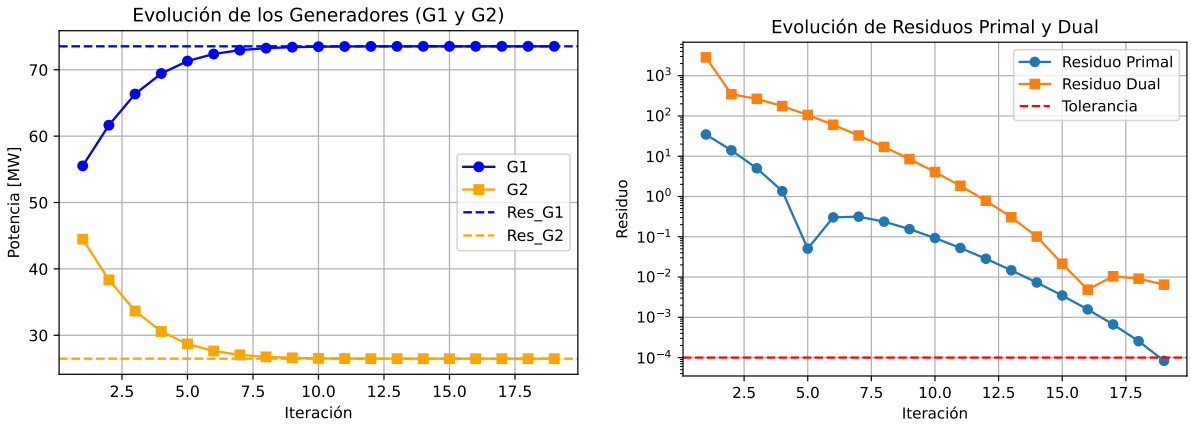
El proceso iterativo continúa hasta que los residuos son inferiores a la tolerancia $\varepsilon = 10^{-4}$, la cual

es el criterio de parada seleccionado para el caso de estudio.

Resultados de la validación de la fase 2. Los resultados de esta simulación son $P_1 = 73.529396[MW]$ y $P_2 = 26.470601[MW]$, convergiendo después de 19 iteraciones. Las gráficas que muestran la evolución de cada iteración de los resultados y la evolución de los criterios de parada (Residuo primal y dual) se presentan en la Figura 17.

Figura 17

Resultados de simulación fase 2.



(a) Resultados evolución generadores.

(b) Resultados evolución de residuos.

El enlace que presenta la simulación de esta fase es: `https://youtu.be/6WC5Tn6pLPw`.

5.3. Fase 3. Simulación descentralizada mediante ADMM de un sistema eléctrico

Para esta fase se llevó a cabo una publicación de una conferencia internacional, la cual se titula *"An IoT-Based Distributed Cyber-Physical System for Simulating Operation of Electrical Networks"* o "Sistema ciber-físico distribuido basado en IoT para simular la operación de las redes eléctricas" la cual fue presentada en el evento Simposio Internacional sobre Calidad de la Energía

Eléctrica (SICEL 2025), llevado a cabo en Argentina. Esta aporte científico buscó presentar el módulo ciber-físico IoT con una implementación de optimización usando ADMM. El resumen del trabajo se presenta a continuación:

Resumen. Desde una perspectiva operativa, el creciente despliegue de fuentes de energía renovables en los sistemas eléctricos y las redes de distribución plantea un reto debido al consiguiente aumento de la incertidumbre y a la granularidad temporal necesaria para un funcionamiento seguro y protegido. En este contexto, la Infraestructura de Medición Avanzada (AMI) y el Internet de las Cosas (IoT) desempeñan un papel crucial al permitir la medición en tiempo real y el intercambio de datos entre los componentes de la red. Aunque los entornos de simulación con condiciones ideales pueden proporcionar información valiosa sobre las operaciones del sistema eléctrico, a menudo no logran captar los efectos críticos de la comunicación y las limitaciones que, en cambio, se representan con mayor precisión en los entornos ciber-físicos. Para abordar esta carencia, este trabajo de investigación presenta el desarrollo de un Sistema Ciber-físico Distribuido (DCPS) basado en IoT diseñado para simular el funcionamiento de los sistemas de energía eléctrica. Para validar el rendimiento del DCPS, se comparó una implementación de simulación de flujo de potencia distribuida basada en el Método de Multiplicadores de Dirección Alterna (ADMM) con una simulación de flujo de potencia centralizada. Se obtuvieron resultados favorables, con un error inferior al 1 %, lo cual valida el funcionamiento adecuado del DCPS y respalda su viabilidad como una herramienta implementable y reproducible en entornos experimentales y prácticos.

Para la validación de esta fase, se presenta el montaje del sistema ciberfísico distribuido (DCPS). Como referencia, se consideran dos trabajos relacionados, cuya comparación se resume

en la Tabla 4, donde se destacan las características comunes y las diferencias entre la propuesta desarrollada y las reportadas en (Nguyen et al., 2019) y (Nelson et al., 2016). Las principales diferencias radican en la forma en que se utiliza el hardware, dado que en este trabajo se utilizaron tarjetas de placa única para implementar un sistema ciber-físico diferente. También se identificaron diferencias en el protocolo de comunicación seleccionado y en el sistema considerado como caso de estudio. Sin embargo, los dos primeros estudios coinciden en el lenguaje de programación utilizado y difieren de (Nelson et al., 2016), así como en el método de optimización empleado.

Tabla 4

Comparación entre la propuesta y otros trabajos de vanguardia.

Componentes	**Propuesta**	**Ref (Nguyen et al., 2019)**	**Ref (Nelson et al., 2016)**
Hardware	PC, RPi, Odroid	PC, RPi, OPAL-RT	PC, OPAL-RT
Protocolo	MQTT	gRPC	HTTP
Caso de estudio	IEEE 9 buses	6 buses	IEEE 13 buses
Método de Optimización	ADMM	ADMM	—
Lenguaje	Python	Python	MATLAB

Validación del sistema. Para validar el funcionamiento del DCPS, se llevó a cabo una simulación de flujo de potencia óptimo utilizando el método ADMM y la aproximación DC-OPF. El DC-OPF es una aproximación del ACOPF (flujo de potencia óptimo de corriente alterna) y se utiliza ampliamente en aplicaciones relacionadas con los mercados eléctricos. Se ha demostrado que este enfoque ofrece resultados fiables y destaca por su simplicidad, precisión y escalabilidad en el análisis y la resolución de problemas de flujo de potencia óptimo (Rui et al., 2022). El sistema

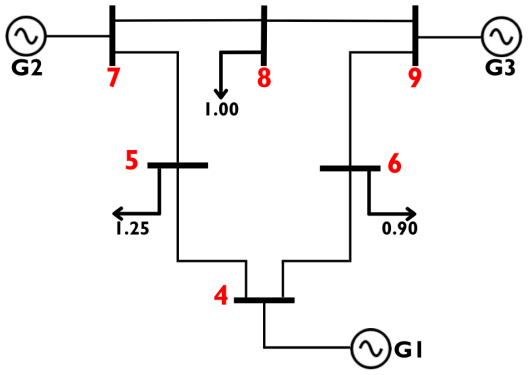
IEEE de 9 buses (Anderson and Fouad, 2003), con modificaciones específicas para este caso práctico, sirvió de base para la simulación. El sistema tiene tres generadores, seis buses y tres cargas, con sus respectivos valores por unidad (p.u.) indicados en la Figura 18. Se realizan dos tipos de simulaciones:

Simulación centralizada: El DC-OPF se simula de forma centralizada utilizando AMPL. En este escenario, todo el proceso de optimización se ejecuta en el ordenador central.

Simulación descentralizada: El DC-OPF se simula utilizando el DCPS. En este caso, cada tarjeta (Raspberry Pi u Odroid) simula el comportamiento de un generador individual, mientras que el ordenador central simula el flujo de potencia de todo el sistema basándose en la información recibida de cada tarjeta y ADMM.

Figura 18

Sistema IEEE de 9 buses modificado.



Formulación centralizada. Para este estudio de caso, se utilizó AMPL para resolver el DC-OPF, que puede formularse como el siguiente problema de optimización:

$$\begin{aligned} &\min \sum_{g\in\mathscr{G}} C(P_g) \\ \text{s.a.} \quad &\sum_{g\in\mathscr{G}_n} P_g - \sum_{d\in\mathscr{D}_n} p_d = \sum_{(n,m)\in\mathscr{L}_n} F_{(n,m)}, \quad \forall n \in \mathscr{B} \\ &0 < P_g < P_{max}, \quad \forall g \in \mathscr{G} \end{aligned} \tag{9}$$

donde,

$$C(P_g) = a_2 P_g^2 + a_1 P_g + a_0 \tag{10}$$

siendo a_2, a_1 y a_0 los coeficientes de costo operativo de cada generador. Y donde la función $F_{(n,m)}$ puede expresarse como

$$F_{(n,m)} = \frac{\delta_n - \delta_m}{x_{(n,m)}}, \quad \forall (n,m) \in \mathscr{L}. \tag{11}$$

donde $\delta_{n,m}$ corresponde al ángulo en grados para cada generador y $x_{(n,m)}$ corresponde a la reactancia de línea en p.u.

Formulación descentralizada. El método ADMM se utiliza para descentralizar el proceso de toma de decisiones asociado a los generadores. El ADMM es una formulación especializada del método de multiplicadores que permite desacoplar variables, lo que lo hace adecuado para la optimización descentralizada. La idea clave es reformular el problema de optimización utilizando una función indicadora. Para empezar, se define $\mathscr{R}$ como el conjunto de soluciones viables al problema de optimización, es decir:

$$\begin{aligned} &\min_x f(x) \\ &\text{s.a. } x \in \mathscr{R} \end{aligned} \tag{12}$$

A continuación, se define la variable auxiliar $\aleph$ como un duplicado de la variable primal x. Utilizando esta formulación, el problema de optimización se puede expresar como:

$$\begin{aligned} \min_{x,\aleph} \quad & f(x)+g(\aleph) \\ \text{s.a.} \quad & x-\aleph=0 \end{aligned} \tag{13}$$

donde $g(\aleph)$ es la función indicadora del conjunto factible $\mathscr{R}$, es decir:

$$g(\aleph)=\begin{cases} 0, & \aleph \in \mathscr{R} \\ \infty, & \aleph \notin \mathscr{R} \end{cases} \tag{14}$$

El problema DC-OPF se adapta a la estructura ADMM desacoplando las variables de decisión del problema, que son las inyecciones de potencia de cada generador. Así, en este enfoque, cada agente optimiza sus variables sujetas a restricciones locales, al tiempo que se coordina con el sistema global a través de variables duales y de consenso. Esta reformulación permite resolver el problema de forma descentralizada, aprovechando la estructura separable tanto de la función objetivo como de las restricciones.

Para realizar las iteraciones asociadas con ADMM, seguimos el enfoque propuesto por N. Parikh y S. Boyd en (Parikh and Boyd, 2014). Concretamente, cada iteración k consta de tres pasos: el paso *proximal*, el paso *projection* y el paso *update*, como se indica a continuación:

$$x^{k+1}=\text{prox}_{f,\rho}\left(\aleph^k-\mu^k\right),\ \aleph^k \in \mathscr{R} \tag{15}$$

$$\aleph^{k+1} = \text{proj}_{\mathscr{R}}\left(x^{k+1} + \mu^k\right) \tag{16}$$

$$\mu^{k+1} = \mu^k + r^k \tag{17}$$

donde $r^k = x^{k+1} - \aleph^{k+1}$ y μ es el multiplicador de Lagrange escalado del consenso de optimización, actualizado hasta alcanzar el punto óptimo del problema. El paso *proximal* se define como:

$$\text{prox}_{f,\rho}(x) = \arg\min_y \left\{ f(y) + \frac{\rho}{2}\|y-x\|_2^2 \right\} \tag{18}$$

Este operador devuelve un valor que se encuentra entre el óptimo de f y x. Concretamente, si ρ es muy pequeño, el valor resultante se acercará más al óptimo de f; por el contrario, si ρ es grande, el valor resultante se acercará más al valor de x. Aquí, f representa la función de costo operativo de cada generador ($f(y) = C(P_g)$).

La función que se debe optimizar es:

$$A_{P_g,\rho} = a_2 P_g^2 + a_1 P_g + a_0 + \frac{\rho}{2}\left\|P_g - x\right\|_2^2 \tag{19}$$

Derivando respecto a P_g :

$$\frac{dA_{P_g,\rho}}{dP_g} = 2a_2 P_g + a_1 + \rho(P_g - x). \tag{20}$$

Estableciendo la expresión igual a cero para obtener la solución en forma cerrada y resolviendo para P_g:

$$\text{prox}_{f,\rho}(x) = \frac{\rho x - a_1}{\rho + 2a_2}. \tag{21}$$

A continuación, el paso *proximal* se reformula en cada ordenador de placa única, como se muestra en la ecuación (22), con el fin de reducir el tiempo de cálculo y el uso de recursos. El coeficiente para todos los generadores es $a_2 = 1$ y $a_0 = 0$, el coeficiente $a1$ para cada generador se define como se muestra en la Tabla 5.

Tabla 5

Coeficientes para cada generador.

No. Generador	a_1
G1	4,06
G2	8,16
G3	11,44

Bajo estos supuestos, el paso *proximal* se expresa como:

$$\text{prox}_{f,\rho}(x) = \frac{\rho x - a1}{\rho + 2} \tag{22}$$

A continuación, el paso de *proyección* se define como:

$$\text{proj}_{\mathscr{R}}(x) = \arg\min_{y} \{\|y - x\| : y \in \mathscr{R}\} \tag{23}$$

En otras palabras, el operador produce el punto de la región factible $\mathscr{R}$ más cercano a x.

En el DCPS, cada tarjeta ejecuta el paso proximal, generando un valor que se optimizará. A continuación, este valor se somete a un paso de proyección, que se lleva a cabo utilizando el lenguaje de modelado AMPL e implica una serie de iteraciones. Las iteraciones continúan hasta

que el algoritmo converge en la solución óptima. Para el criterio de parada, se utiliza el residuo primario $primal_{res} = ||y - x||$ y el residuo dual $dual_{res} = ||\rho(y - x)||$.

Resultados de la validación de la fase 3. El objetivo principal de esta metodología es comparar los resultados obtenidos de la simulación centralizada con los de la simulación descentralizada. La comparación revela un alto grado de correspondencia, con un error muy inferior al 1 %, lo que valida el correcto funcionamiento del DCPS.

Tabla 6

Resultados de la simulación centralizada (solo AMPL) y descentralizada (utilizando el DCPS).

No. Generador	Centralizado	Descentralizado
G1	2,38573	2,38573171
G2	0,664268	0,66426827
G3	0,1	0,10000001

Los resultados de potencia para cada generador en la simulación centralizada (AMPL) y la simulación distribuida (DCPS) se presentan en la Tabla 6. Los resultados de la simulación centralizada tardan menos de 1 segundo. Sin embargo, la simulación distribuida convergió después de seis iteraciones con un $\rho = 12$ y un tiempo de simulación de $t = 6[s]$; esto se debe al proceso de intercambio de información entre la tarjetas y el ordenador central.

Estos resultados demuestran que la diferencia en los valores de potencia obtenidos de las dos simulaciones, es decir, los resultados de AMPL frente a los de DCPS, es despreciable. La Figura 19 muestra los valores obtenidos en cada iteración. Además, los cambios en los criterios de detención de ADMM se ilustran en la Figura 20, donde se alcanza el valor de tolerancia de $1 * 10^{-4}$

después de la sexta iteración. En la séptima iteración, se realiza el ajuste final de los valores del generador mediante los pasos de proyección y actualización.

Figura 19

Comparación de resultados. Simulación centralizada frente a descentralizada.

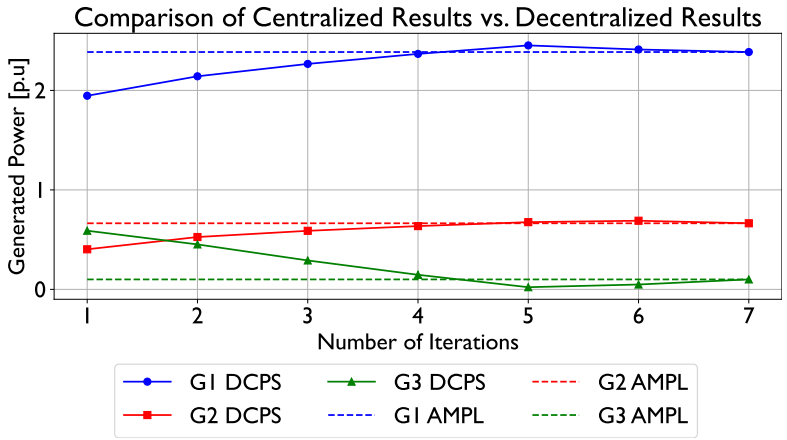
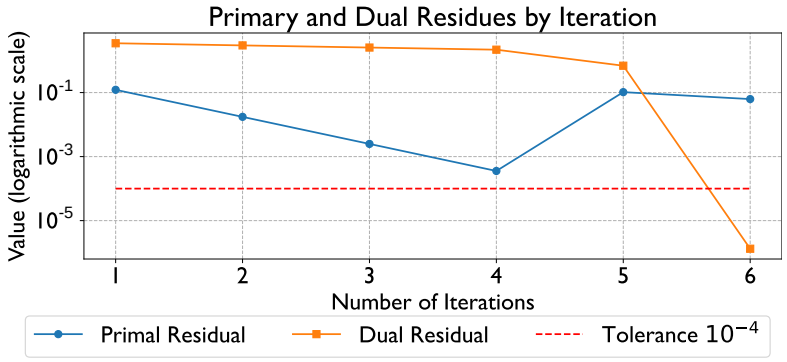


Figura 20

Evolución de los criterios de finalización de ADMM.



5.4. Fase 4: Mejoras en la robustez, tolerancia a fallos y almacenamiento de datos

El módulo ciber-físico IoT desarrollado corresponde a un sistema distribuido, entendido como un conjunto de dispositivos interconectados que comparten recursos, información y tareas

para cumplir un objetivo común. En este tipo de sistemas, la tolerancia a fallos es un aspecto fundamental, ya que garantiza la continuidad del servicio y reduce el impacto de errores tanto en la red como en los dispositivos o en el software de control.

5.4.1. *Identificación de fallas potenciales en el sistema*

Durante la evaluación experimental del módulo, se identificaron los principales tipos de fallas:

1. **Fallas de Red:** Pérdida de conexión con el broker MQTT, latencia elevada o desincronización de mensajes e interrupciones intermitentes en la conectividad.
2. **Fallas de Hardware:** Sobrecalentamiento o apagado inesperado de las tarjetas, corrupción de la tarjeta SD e interrupciones en el suministro eléctrico.
3. **Fallas de Software:** Errores en el esquema de publicación/suscripción, pérdida o duplicación de mensajes e inconsistencias en el formato de los datos (por ejemplo, estructuras JSON mal formadas).

Dado que el módulo opera bajo un esquema distribuido y requiere sincronización iterativa entre dispositivos, estas fallas pueden comprometer la coherencia global del proceso de simulación si no se gestionan adecuadamente.

5.4.2. *Estrategias de tolerancia a fallos implementadas*

Con el objetivo de incrementar la robustez del sistema, se implementaron las siguientes estrategias:

1. **Reconexión automática:** Se incorporaron mecanismos de reintento y reconexión automática en caso de pérdida de conexión con el broker MQTT, permitiendo que los dispositivos restablezcan la comunicación sin intervención manual. Los mecanismos de reconexión automática son ampliamente utilizados en sistemas basados en MQTT para garantizar la continuidad del servicio ante fallas de red, permitiendo a los clientes restablecer la conexión con el broker de manera transparente (OASIS, 2014; Light, 2017).

2. **Control de tiempos de espera (timeouts):** El ordenador central fue programado para esperar la respuesta de cada tarjeta durante un tiempo definido. En caso de que una tarjeta no responda dentro del intervalo establecido, el sistema puede:

   - Mantener el último valor válido recibido o si se recibe notificación de reconexión realizar un reintento de envío de la información.
   - Registrar el evento como anomalía.
   - Continuar la iteración evitando la detención completa del sistema.

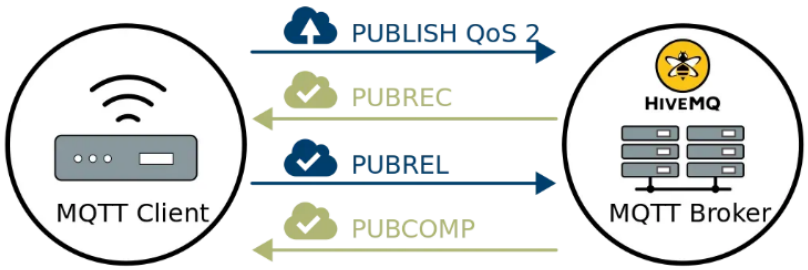
   Esta estrategia permite mantener la continuidad del proceso iterativo incluso ante fallas. El uso de mecanismos de control de tiempos de espera (timeouts) es una práctica fundamental en sistemas distribuidos para detectar fallas y evitar bloqueos del sistema, permitiendo la continuidad del procesamiento aun en presencia de nodos no disponibles (Tanenbaum and Van Steen, 2017; Coulouris et al., 2012).

3. **Uso de *Quality of Service* (QoS 2)**

Se configuró el sistema para operar con QoS 2, el nivel más alto de calidad de servicio en MQTT, el cual garantiza que cada mensaje sea entregado exactamente una vez. Este mecanismo funciona mediante un proceso de verificación en cuatro pasos entre emisor y receptor, reduciendo la probabilidad de pérdida o duplicación de mensajes. Su implementación resulta fundamental para la seguridad y estabilidad en un sistema descentralizado donde la coherencia de las iteraciones depende de la recepción correcta de cada mensaje como se muestra en la Figura 21. El uso de niveles de calidad de servicio en MQTT, particularmente QoS 2, permite garantizar la entrega exacta de los mensajes mediante un protocolo de confirmación en múltiples etapas, lo cual resulta crítico en sistemas distribuidos donde la integridad de la información es esencial (OASIS, 2014, 2019).

Figura 21

Diagrama de verificación QoS 2. Tomado de (HiveMQ, 2015).



Monitoreo y control. El uso del broker HiveMQ permitió supervisar en tiempo real: Conexiones activas, tráfico de mensajes y errores de autenticación o desconexión. Este monitoreo facilitó la identificación temprana de fallas durante las pruebas experimentales. El monitoreo en tiempo real de variables del sistema es una práctica clave en sistemas distribuidos modernos, per-

mitiendo la detección temprana de fallas y la mejora de la confiabilidad operativa (Burns et al., 2016; Turnbull, 2014).

Estrategia de almacenamiento de datos. Adicionalmente, se fortaleció el sistema mediante la implementación de un mecanismo de almacenamiento estructurado de resultados. Durante cada simulación: Los datos fueron registrados iteración por iteración, se generaron archivos en formato excel, almacenando las variables relevantes del sistema (potencias, tensiones y variables de optimización) y el almacenamiento se realizó por realización experimental, permitiendo trazabilidad completa y reproducibilidad de resultados. Este enfoque no solo mejora la robustez ante pérdidas de información, sino que también facilita el análisis comparativo posterior y la validación cuantitativa de los escenarios simulados.El almacenamiento estructurado de datos y la trazabilidad de las simulaciones son fundamentales para garantizar la reproducibilidad de los resultados y facilitar el análisis posterior en sistemas experimentales (Marz and Warren, 2015).

Conclusión de la implementación de la fase 4. Las mejoras implementadas en esta fase fortalecieron significativamente la resiliencia del módulo ciber-físico IoT, permitiendo: Continuidad operativa ante fallas parciales, integridad y coherencia en el intercambio de mensajes, registro estructurado y seguro de los resultados y mayor confiabilidad en los experimentos distribuidos. Estas características consolidan al módulo ciber-físico IoT como una plataforma experimental adecuada para aplicaciones académicas y de investigación en sistemas eléctricos distribuidos.

El enlace que presenta la simulación de esta fase es el siguiente: `https://youtu.be/3ab5gWGMsBU`

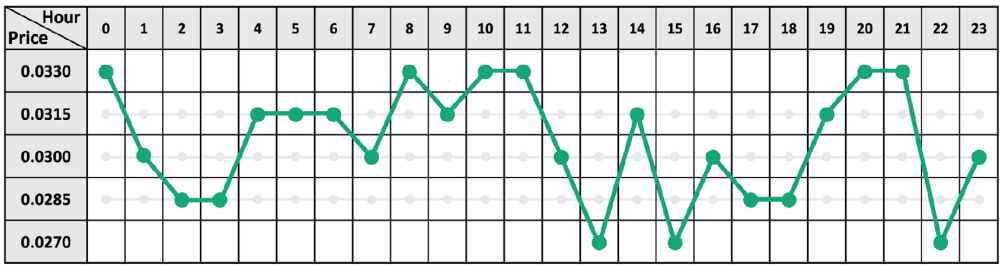
5.5. Fase 5: Validación mediante una estrategia de asignación de precios minoristas

Para la fase final de validación, una vez implementadas las mejoras de robustez y tolerancia a fallos del sistema, se evaluó el desempeño del módulo ciber-físico IoT mediante la aplicación de una **estrategia de asignación de precios minoristas** previamente desarrollada.

La estrategia utilizada se fundamenta en el enfoque presentado en la publicación titulada *"A Reinforcement Learning-Based Approach for Retail Electricity Pricing Strategy for Multi-Microgrid Distribution Systems"* (Rey et al., 2025), desarrollada por integrantes del grupo de investigación GISEL, así como en el trabajo de grado titulado *"Asignación de Precios Minoristas en Sistemas de Distribución con Múltiples Microrredes Usando Aprendizaje por Refuerzo"*. La política de precios obtenida en (Rey et al., 2025; Carrillo-Valera and Sandoval-Manrique, 2023), se presenta en la Figura 22.

Figura 22

Política de precios. Tomado de (Rey et al., 2025).



El objetivo principal de esta fase fue verificar que el módulo ciber-físico IoT reproduce resultados cuantitativos consistentes respecto al esquema centralizado tradicional, evaluando además su desempeño computacional y comportamiento iterativo bajo un enfoque distribuido.

5.5.1. *Descripción sistema eléctrico de estudio*

Para el caso de estudio, se trabajó con un conjunto de MGs que incluyen paneles solares, generadores eólicos, generadores diésel (despachable) y sus respectivas cargas como se muestra en la Figura 23. En total, se utilizaron 4 MGs con una estructura general similar. La configuración de cada MG se detalla en la Tabla 7.

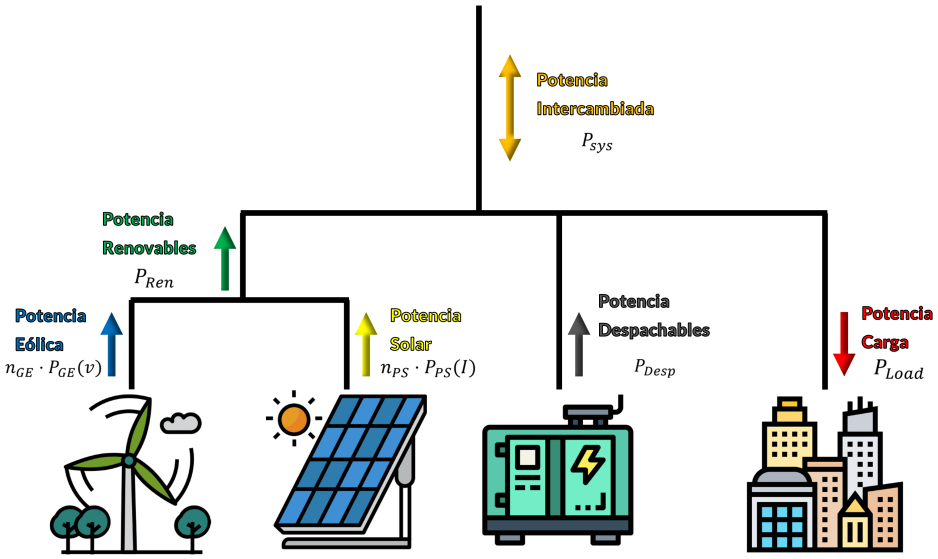
Tabla 7

Parámetros de las microrredes.

Microrred	Números paneles solares	Número generadores eólicos	Potencia nominal [kW]
1	15	11	23
2	28	19	45
3	39	26	63
4	24	17	39

Figura 23

Diagrama microrred usada en el caso de estudio. Tomado de (Carrillo-Valera and Sandoval-Manrique, 2023).



Para garantizar la reproducibilidad de las simulaciones, la variabilidad de los recursos energéticos y la demanda fue modelada a partir de los datos históricos del conjunto de datos utilizado. La radiación solar se representó mediante una distribución Beta, la velocidad del viento mediante una distribución Weibull y la demanda eléctrica mediante una distribución normal basada en la media y desviación estándar de los perfiles de carga. Adicionalmente, se utilizó una semilla fija en la generación de datos aleatorios, permitiendo reproducir consistentemente los escenarios y resultados obtenidos. A continuación se describen los modelos de potencia de cada uno de los elementos de las MGs.

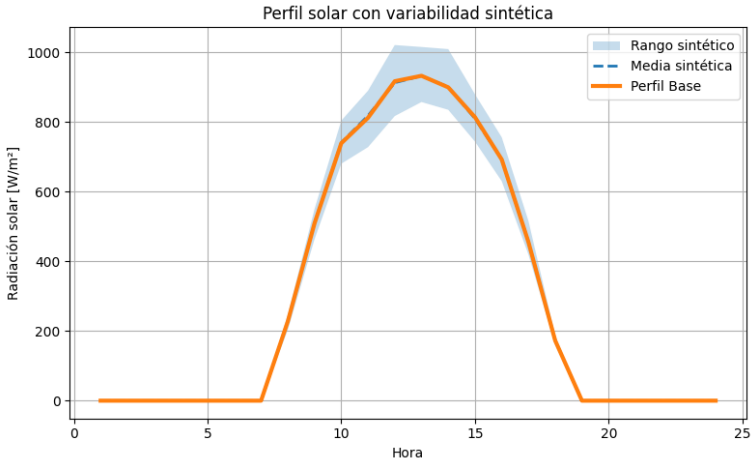
Potencia producida por un panel solar fotovoltaico. La potencia generada por un panel solar fotovoltaico depende de tres factores principales: el área del panel (A_{PS}), su eficiencia (η) y la irradiación solar (I), medida en $\left(W/m^2\right)$ y determinada por condiciones geográficas y climáticas (Vergara et al., 2014)(IDEAM, 2005). La combinación de estos factores permite estimar la potencia eléctrica generada mediante la expresión:

$$P_{PS}(I) = \frac{A_{PS} \cdot \eta \cdot I}{1000} \tag{24}$$

La ecuación (24) está divida en 1000 para así obtener la potencia en unidades de (kW), donde A_{PS} se consideró como 1.7 $\left(m^2\right)$ y la eficiencia (η) de 0.18. La figura que presenta el perfil usado en las simulaciones se presenten la Figura 24.

Figura 24

Perfil de radiación solar con variabilidad sintética.



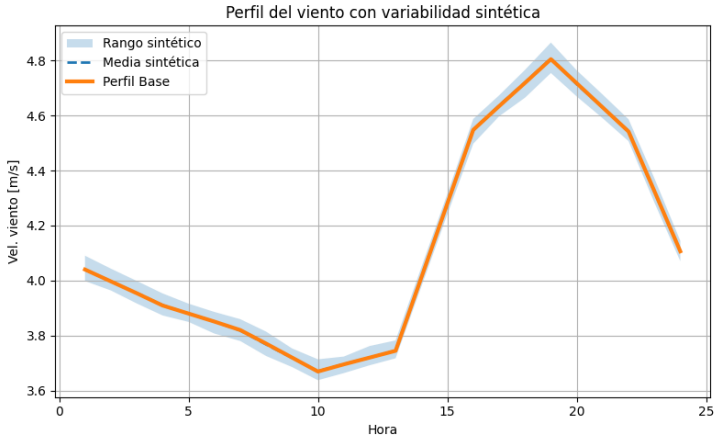
Potencia producida por un generador eólico. La potencia generada por un aerogenerador depende del área de barrido de las aspas (A_{GE}), la densidad del aire (ρ), la velocidad del viento (v) y el coeficiente de potencia (C_p), el cual representa la eficiencia de conversión de energía (Vergara et al., 2014). Estos factores permiten estimar la potencia eólica mediante la siguiente expresión (Ilinca et al., 2003):

$$P_{GE}(v) = \frac{\frac{1}{2} \cdot \rho \cdot A_{GE} \cdot C_p \cdot v^3}{1000} \tag{25}$$

La ecuación (25) se divide en 1000 para obtener el resultado en unidades de (kW), donde (ρ) tiene un valor de 1.225 $\left(\frac{kg}{m^3}\right)$, el área de las aspas está definida como πr^2 $\left(m^2\right)$ en que el radio toma el valor de 1.78 (m), la velocidad está en $\left(\frac{m}{s}\right)$ y el factor de C_p representa un coeficiente de potencia con valor de 0.45. La Figura 25 presenta el perfil usado en las simulaciones.

Figura 25

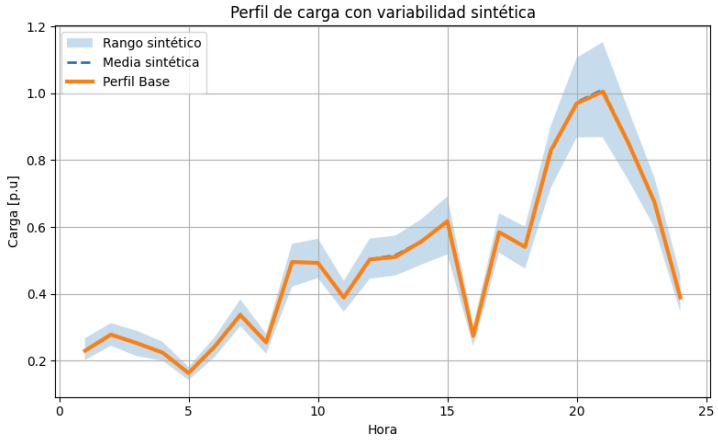
Perfil de velocidad de viento con variabilidad sintética.



Perfil de carga. Este perfil de carga proporciona el comportamiento de la demanda de la microrred, tomando valores de potencia hora a hora en por unidad (normalizada) para realizar el respectivo flujo de potencia. Estos perfiles de carga sintéticos se generarán partiendo de los valores de carga de un día semilla. Este perfil se presenta en la Figura 26.

Figura 26

Perfil de carga con variabilidad sintética.



Potencia producida por el generador diésel. La inclusión de un generador diésel en las MGs proporciona un respaldo confiable ante situaciones en las que las fuentes renovables no pueden cubrir completamente la demanda eléctrica. La operación del generador está determinada por la referencia de potencia que asigne el controlador central de la microrred. Es decir, por el valor de potencia que se defina en el despacho. El costo de operación de este generador se define mediante la siguiente ecuación cuadrática:

$$\$_{desp}(P_{desp}) = \$_{galon} \cdot \left(a \cdot P_{desp}^2 + b \cdot P_{desp} + c\right) \tag{26}$$

En esta ecuación, $\$_{galon}$ representa el costo por galón de diésel en dólares, que para este trabajo se asumirá como 1.8 $\left[\frac{USD}{g}\right]$, P_{desp} es la potencia despachada por el generador, y las constantes a, b y c describen el comportamiento específico del generador (Du and Li, 2020).

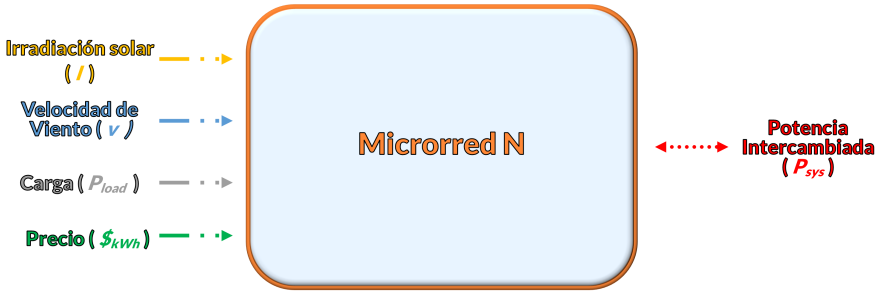
Para el cálculo de las constantes de la ecuación, se utilizó la ficha técnica (Clifford-Power, 2023) que contiene información sobre los puntos operativos de un generador diésel particular. Se tienen tres puntos de costos de operación para el generador. Utilizando el método de mínimos cuadrados, se realizaron los cálculos correspondientes para determinar las constantes de la ecuación de costos. Teniendo en cuenta que la potencia máxima que puede producir el generador es de 30 [kW], se obtuvieron los siguientes valores de constantes: $a = \frac{2}{1125}\left[\frac{g}{kW^2}\right]$, $b = \frac{1}{75}\left[\frac{g}{kW}\right]$ y $c = \frac{4}{5}[g]$, por lo que la ecuación (26) puede sustituirse con valores numéricos como

$$\$_{desp}(P_{desp}) = 1.8 \cdot \left(\frac{2}{1125} \cdot P_{desp}^2 + \frac{1}{75} \cdot P_{desp} + \frac{4}{5}\right) \tag{27}$$

Despacho de la microrred. En la Figura 27 se evidencia los respectivos parámetros de entrada y salida de la MG, con la cual se llevará a cabo la minimización de su costo de operación.

Figura 27

Arquitectura del modelo de optimización de la microrred. Tomado de (Carrillo-Valera and Sandoval-Manrique, 2023).



El objetivo del problema de despacho es minimizar el costo de operación de la MG, que está compuesto por dos términos: el costo de operación del generador $\$_{desp}$ (definido en la ecuación (27)) y el costo de compra/venta de energía al sistema de distribución $\$_{sys}$ definido como

$$\$_{sys}(P_{sys}) = \$_{kWh} \cdot P_{sys} \tag{28}$$

donde $\$_{kWh}$ es el precio de energía minorista asignado en $[kWh]$ y P_{sys} es la potencia intercambiada con el sistema de distribución.

Así, el problema de despacho, en donde deben definirse las variables P_{desp} y P_{sys}, se formula:

$$\begin{aligned} &\min_{P_{desp},P_{sys}} \{\$_{desp} + \$_{sys}\} \\ &\text{s.a.} \quad P_{ren} + P_{desp} + P_{sys} = P_{load} \end{aligned} \tag{29}$$

donde P_{ren} corresponde a la energía renovable producida, que se define a partir del número de paneles solares n_{PS}, la potencia producida por un panel $P_{PS}(I)$ (ecuación 24), el número de generadores eólicos n_{GE} y la potencia producida por un generador eólico $P_{GE}(v)$ (ecuación 25):

$$P_{ren} = n_{PS} \cdot P_{PS}(I) + n_{GE} \cdot P_{GE}(v). \tag{30}$$

Por su parte, P_{load} corresponde a la carga total de la MG. Nótese que la restricción es el balance de potencia de la MG, asumiendo pérdidas nulas en sus líneas de distribución, aproximación generalmente válida para microrredes de potencia baja y media (ver Figura 23). Este problema de optimización es resuelto para cada microrred usando la librería Pyomo de Python. Para ser resuelto, es necesario que se especifique el precio minorista de energía $\$_{kWh}$ por el OR.

Sistema de distribución. Como se mencionó anteriormente, el sistema de distribución está compuesto por 4 microrredes, conectadas en un sistema radial balanceado a un nodo de interconexión a la red principal (barra Slack, barra 0) como se puede ver en la Figura 28. Para el caso de estudio, se hace uso de la librería de Python llamada PandaPower, para realizar el flujo de potencia. En la Tabla 8 se registran los valores de impedancia y longitud de cada línea con cada una de las etiquetas que se evidencian en la Figura 28.

5.5.2. *Simulación centralizada*

Para la validación se realizó una creación de datos tanto de radiación solar, velocidad de viento y carga de 100 días correspondiente a 2400 realizaciones horarias usando la política de

precios presentada en la Figura 22.

Figura 28

Topología de sistema de distribución de estudio.

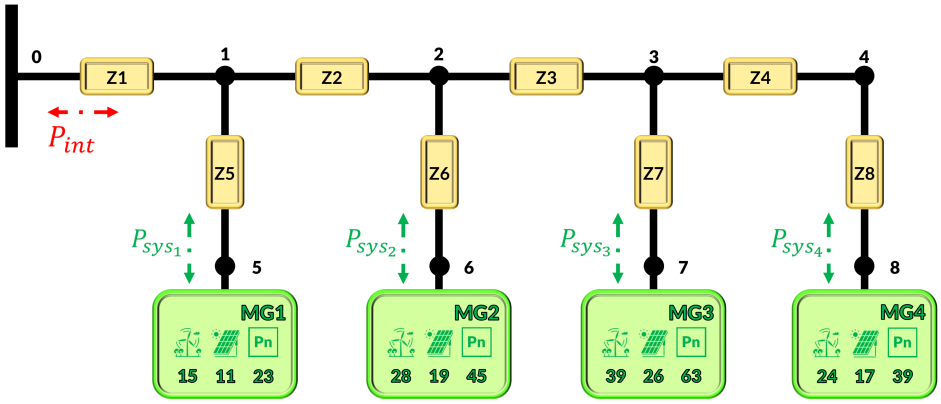


Tabla 8

Parámetros de las líneas del sistema de distribución.

Línea	Resistencia $\left[\frac{\Omega}{km}\right]$	Reactancia $\left[\frac{\Omega}{km}\right]$	Distancia [km]
Z_1	1.60816	16.0816	1.0
Z_2	0.33856	3.3856	1.5
Z_3	1.57113	15.7113	1.3
Z_4	1.57113	15.7113	1.2
Z_5	1.48649	14.8649	1.0
Z_6	0.57132	5.7132	1.0
Z_7	1.57113	15.7113	1.0
Z_8	1.57113	15.7113	1.0

Nota. Valores obtenidos utilizando la herramienta de Panda-Power y etiquetas de línea de la Figura 28.

Para el flujo de potencia se le asignaron a las potencias intercambiadas por cada microrred (P_{sys_1}, P_{sys_2}, P_{sys_3} y P_{sys_4}) signos de acuerdo con su comportamiento. Se asumirá signo positivo si una microrred consume/compra energía y signo negativo si una microrred entrega/vende energía.

Solucionando los flujos de potencia del sistema de distribución considerando las pérdidas causadas por las impedancias de las líneas, se calcula $P_{int}(t)$ que corresponde a la potencia intercambiada por el sistema de distribución en el nodo de interconexión a la red principal (nodo 0). El flujo de potencia en cada realización de la simulación fue resuelta por PandaPower y Pyomo con la ayuda del solver Gurobi. El flujo de potencia del sistema está descrito por la ecuación (31).

$$P_{int} = P_{sys_1} + P_{sys_2} + P_{sys_3} + P_{sys_4} + P_{losses} \tag{31}$$

Los resultados obtenidos de esta simulación correspondiente a los 100 días, se presenta en la Figura 29, con un tiempo de simulación total de **547 segundos** (9.1 minutos).

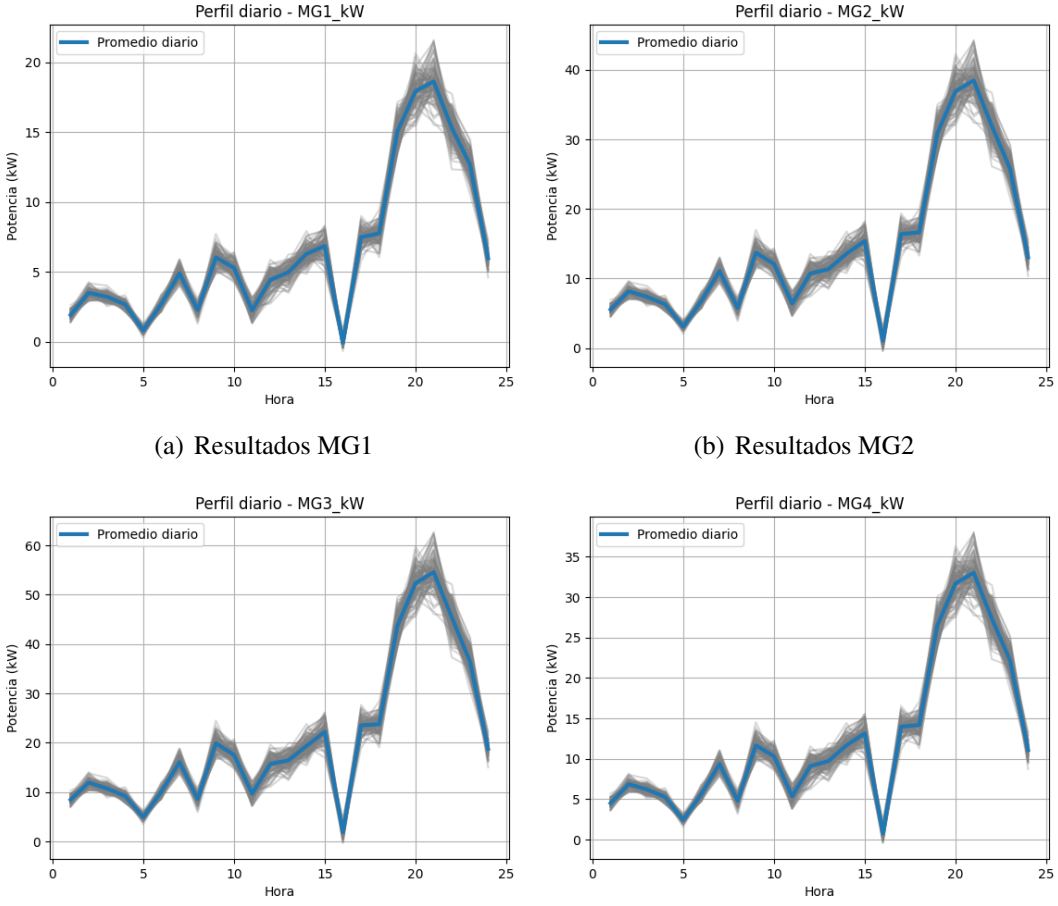
Se evidencia que la potencia generada en cada MG sigue la misma tendencia del perfil de carga (véase Figura 26) ajustándose proporcionalmente a la potencia nominal instalada en cada caso. Este comportamiento se debe a que todas las MGs comparten la misma función de costo para el generador diésel; sin embargo, difieren en su capacidad nominal y en la configuración de recursos renovables, particularmente en el número de paneles solares y generadores eólicos instalados. En consecuencia, aunque la forma de la curva de potencia mantiene una tendencia similar entre MGs, las magnitudes varían de acuerdo con la capacidad instalada y la disponibilidad de generación renovable en cada una de ellas.

5.5.3. *Simulación descentralizada*

Para la simulación descentralizada se resolvió el mismo problema formulado en el esquema centralizado. La simulación se ejecutó durante 100 días, evaluando el sistema hora a hora y manteniendo las mismas condiciones de estudio, incluyendo la política de precios y los perfiles de generación renovable de cada MG.

Figura 29

Resultados en potencia de cada MG del problema centralizado.



(a) Resultados MG1

(b) Resultados MG2

(c) Resultados MG3

(d) Resultados MG4

La principal diferencia respecto al esquema centralizado radica en la implementación computacional. En el enfoque descentralizado, el problema es resuelto de manera distribuida en las tarjetas que simulan las MGs, las cuales poseen recursos de procesamiento limitados en comparación con un computador convencional. Por esta razón, fue necesario optimizar el uso de los recursos computacionales disponibles en cada dispositivo, con el objetivo de reducir los tiempos de procesamiento y garantizar la estabilidad del sistema iterativo. En consecuencia, se realizó un ajuste en la formulación e implementación del problema de optimización ejecutado en las tarjetas, priorizando eficiencia computacional sin alterar la naturaleza matemática del problema original. Este ajuste se fundamenta en el siguiente análisis.

El problema a resolver en cada MG está definido como:

$$\begin{aligned} &\min_{P_{desp},P_{sys}} \{\$_{desp} + \$_{sys}\} \\ \text{s.a.} \quad &P_{sys} = P_{load} - P_{desp} - P_{ren} \end{aligned} \tag{32}$$

Donde $\$_{desp}$ esta definido como:

$$\$_{desp}(P_{desp}) = \$_{galon} \cdot \left(a \cdot P_{desp}^2 + b \cdot P_{desp} + c\right) \tag{33}$$

mientras $\$_{sys}$ definida como:

$$\$_{sys}(P_{sys}) = \$_{kWh} \cdot P_{sys} \tag{34}$$

Sustituyendo ambas expresiones en la función objetivo y definiendo el costo total $(C_\$)$, se tiene:

$$\mathrm{C}_{\$}(P_{desp}) = \$_{galon} \cdot \left(a \cdot P_{desp}^2 + b \cdot P_{desp} + c\right) + \$_{kWh} \cdot (P_{load} - P_{desp} - P_{ren}) \tag{35}$$

De esta manera, el problema se reduce a una función de una sola variable P_{desp}, lo cual permite obtener una solución analítica directa. Derivando el costo total respecto a P_{desp}:

$$\frac{\mathbf{d}C_{\$}}{\mathbf{d}P_{desp}} = \$_{galon} \cdot \left(2 \cdot a \cdot P_{desp} + b\right) + \$_{kWh} \cdot (-1)$$

Igualando la derivada a cero y despejando P_{desp} para encontrar el punto óptimo se tiene:

$$\$_{galon} \cdot \left(2 \cdot a \cdot P_{desp} + b\right) - \$_{kWh} = 0$$

$$P_{desp} = \frac{\$_{kWh} - \$_{galon} \cdot b}{2 \cdot a \cdot \$_{galon}} \tag{36}$$

Este valor corresponde a la potencia óptima despachada por el generador diésel en cada iteración, y es calculado localmente por cada MG durante cada realización de la simulación.

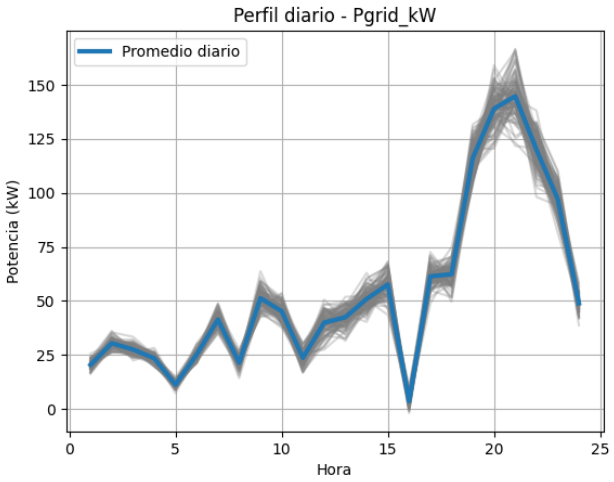
El tiempo total de simulación para el esquema descentralizado fue de **1 hora, 9 minutos y 1 segundo**, equivalente a **4141.6 segundos**. Este valor refleja el costo computacional asociado a la ejecución iterativa del algoritmo en un entorno distribuido, donde cada MG resuelve localmente su problema de optimización y se intercambia información a través de IoT.

La Figura 30 presenta la potencia intercambiada con el sistema principal durante los 100

días simulados. En dicha figura, las líneas grises representan el comportamiento horario individual de cada día, mientras que la línea azul corresponde al promedio de los 100 días analizados. Se observa que el perfil promedio mantiene la tendencia general del sistema, suavizando las variaciones diarias y permitiendo identificar el comportamiento típico de intercambio de potencia bajo la estrategia de precios implementada.

Figura 30

Resultados potencia del sistema.

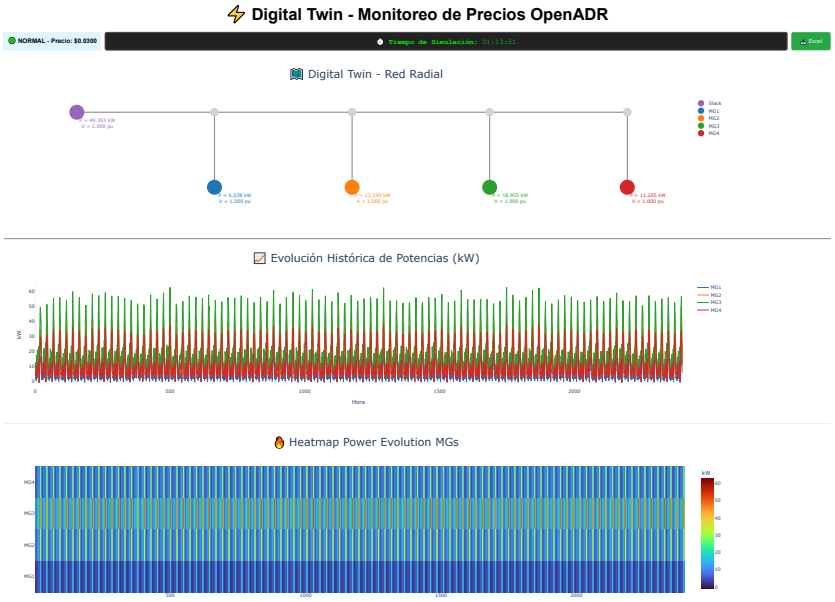


Por otra parte, en la Figura 31 se presenta la *dashboard* correspondiente al gemelo digital de la simulación descentralizada. En ella se visualiza la totalidad de las **2400 realizaciones** (100 días evaluados hora a hora), así como el tiempo total de ejecución del proceso. La interfaz permite observar de manera integrada la evolución temporal de las variables principales del sistema, evidenciando la actualización en tiempo real durante el desarrollo de la simulación. Esta representación confirma no solo la correcta ejecución del algoritmo descentralizado, sino también la funcionalidad del módulo como herramienta de monitoreo y análisis interactivo. De esta manera,

la *dashboard* actúa como un entorno de supervisión que replica el comportamiento del sistema eléctrico estudiado, consolidando el concepto de gemelo digital implementado en el módulo ciber-físico IoT.

Figura 31

Resultados del gemelo digital de la simulación descentralizada.



El enlace que evidencia una parte de la simulación es el siguiente: `https://youtu.be/AXyC2ay-YGs`

5.5.4. *Análisis de resultados fase 5 de validación*

Los resultados obtenidos en los esquemas centralizado y descentralizado fueron altamente consistentes, presentando diferencias mínimas entre ambos enfoques. La Figura 32 muestra el error porcentual asociado a los resultados obtenidos, evidenciando que la discrepancia entre las soluciones es prácticamente nula.

En términos de tiempo de simulación, la Tabla 9 presenta los tiempos correspondientes a cada caso, evidenciando un incremento aproximado de **7.57 veces** en el tiempo total de ejecución.
Tabla 9

Tabla comparativa de tiempos de simulación.

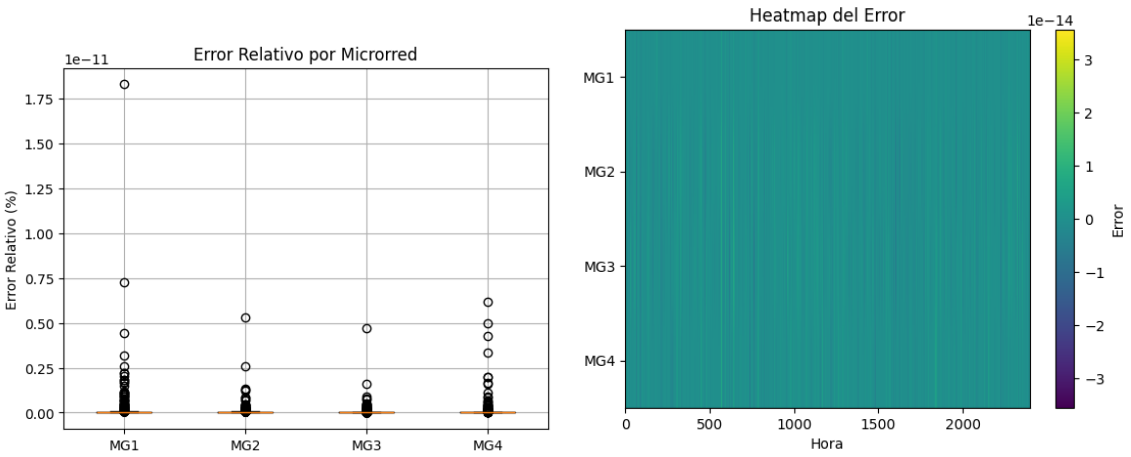
Tipo de Simulación	Tiempo [s]
Centralizada	547.02
Distribuida	4141.68

No obstante, esta diferencia es coherente con la naturaleza del sistema distribuido implementado. En el esquema descentralizado, el tiempo adicional corresponde al proceso de: Comunicación híbrida mediante OpenADR y MQTT, sincronización entre dispositivos, procesamiento local en cada tarjeta y gestión de iteraciones y validaciones de recepción. Por lo tanto, aunque el enfoque descentralizado implica un mayor costo computacional, este permite trasladar la simulación a un entorno más realista, donde los tiempos de comunicación y procesamiento forman parte inherente de la operación de sistemas eléctricos distribuidos donde la planeación y control sea un factor importante. En consecuencia, la validación confirma que el módulo ciber-físico IoT reproduce con alta precisión los resultados del modelo centralizado, demostrando su viabilidad como plataforma experimental para la simulación de estrategias de operación y mercado en sistemas de distribución con múltiples microrredes.

A partir de las Figuras 32 y 33, se evidencia que el error se encuentra en el orden de magnitud de 10^{-11}. Este valor corresponde a un error numérico prácticamente despreciable, atribuible únicamente a diferencias en precisión computacional y manejo de decimales.

Figura 32

Resultados del error comparativo entre la simulación centralizada y descentralizada.

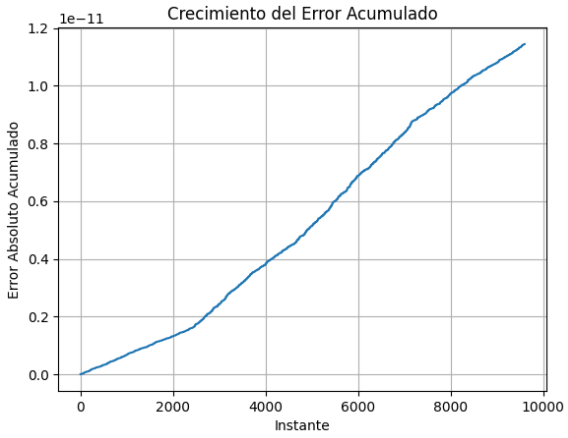


(a) Resultados error relativo.

(b) Mapa de calor(*Heatmap*) del error.

Figura 33

Resultados error absoluto acumulado de todas las MGs.



Los resultados de las simulaciones son numéricamente equivalentes, presentando solo pequeñas variaciones decimales atribuibles a la implementación computacional. No obstante, estas variaciones no son significativas, lo que valida la consistencia matemática y de simulación del módulo.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

Se diseñó la configuración por capas del módulo ciber-físico IoT considerando las características propias de los sistemas de distribución con múltiples microrredes. La arquitectura propuesta integra de manera estructurada: una capa física (dispositivos embebidos), una capa de comunicación (MQTT y OpenADR), una capa de procesamiento y control (ordenador central) y adicionalmente, una capa de visualización (gemelo digital). Como resultado, se concluye que el módulo constituye una herramienta versátil, flexible y escalable, capaz de adaptarse a distintos escenarios de simulación. Se demostró que cualquier miniordenador con capacidad para ejecutar Python y sus librerías puede representar una microrred dentro del sistema, lo que amplía significativamente las posibilidades de implementación. Asimismo, el uso de protocolos ligeros como MQTT permite reducir los recursos computacionales en las tarjetas, manteniendo eficiencia en la comunicación distribuida.

Se logró implementar satisfactoriamente el diseño del módulo ciber-físico IoT para la simulación de un sistema de distribución con múltiples microrredes. En esta implementación el ordenador central simuló el rol del operador de red, dos Raspberry Pi y dos Odroid representaron las microrredes y se integró una arquitectura de comunicación híbrida basada en OpenADR y MQTT. La combinación del estándar OpenADR (orientado a redes eléctricas inteligentes) con el protocolo ligero y escalable MQTT permitió crear un entorno de simulación más cercano a condiciones reales de operación. Esta integración demuestra que es posible combinar estándares industriales

con tecnologías IoT modernas para desarrollar plataformas experimentales robustas y funcionales. Además, la incorporación de mecanismos de tolerancia a fallos, control mediante QoS 2 en MQTT y estrategias de sincronización fortaleció la confiabilidad del sistema distribuido.

Se logró validar la operación del módulo ciber-físico IoT mediante la implementación de una estrategia existente de asignación de precios minoristas en sistemas de distribución con múltiples microrredes. La comparación entre la simulación centralizada y la distribuida evidenció: Resultados prácticamente idénticos, con errores en el orden de 10^{-11}, correcta ejecución del proceso de optimización distribuida, consistencia matemática entre la solución cerrada implementada en las tarjetas y el modelo centralizado. Aunque el tiempo de simulación distribuida fue mayor debido al proceso de comunicación y procesamiento distribuido, esta diferencia representa un comportamiento más realista, al considerar latencias, intercambio de mensajes y ejecución en dispositivos físicos. Adicionalmente, la implementación del algoritmo ADMM y otras estrategias de optimización demostró la versatilidad del módulo como herramienta experimental para el estudio de sistemas eléctricos distribuidos, superando las limitaciones de entornos puramente simulados.

Así entonces, el módulo ciber-físico IoT desarrollado constituye una plataforma experimental robusta, flexible y validada, que permite estudiar la operación de sistemas de distribución con múltiples microrredes en un entorno híbrido físico-digital. La integración de comunicación distribuida, estándares industriales, tolerancia a fallos y visualización en tiempo real posiciona esta herramienta como un aporte significativo para la investigación en redes eléctricas inteligentes y sistemas energéticos distribuidos.

Aunque el módulo ciber-físico IoT desarrollado demostró un funcionamiento adecuado pa-

ra la simulación distribuida de múltiples microrredes, existen algunas limitaciones asociadas al alcance de esta investigación. En primer lugar, la plataforma fue concebida como un entorno experimental y de validación distribuida, por lo que no representa una implementación directa sobre infraestructura eléctrica real. Adicionalmente, la integración del protocolo OpenADR se enfocó principalmente en la comunicación y traducción de eventos de control hacia MQTT, sin profundizar en una validación completa bajo escenarios reales de respuesta a la demanda o integración con operadores eléctricos externos.

6.2. Recomendaciones y trabajo futuro

El éxito de la implementación de un módulo ciber-físico IoT proporcionará tanto al grupo GISEL como a la UIS una herramienta con alto potencial de uso en los ámbitos académico y científico, orientada al análisis, validación y enseñanza de la operación de sistemas eléctricos con microrredes. A partir de los resultados obtenidos y de la validación satisfactoria del módulo ciber-físico IoT, se identifican diversas líneas de trabajo futuro que pueden ampliar su alcance, fortalecer su aplicabilidad y consolidarlo como una herramienta de investigación y desarrollo tecnológico.

Escalabilidad del módulo: Una línea de trabajo futuro consiste en ampliar el número de tarjetas que representan microrredes dentro del sistema. Gracias a su arquitectura modular basada en MQTT, el módulo puede escalarse fácilmente incorporando nuevos dispositivos sin modificar su estructura principal. Esto permitiría analizar sistemas eléctricos de mayor tamaño, estudiar efectos de latencia y sincronización, e implementar estrategias avanzadas de coordinación distribuida. Además, el módulo podría replicarse como herramienta experimental en otros laboratorios o grupos de investigación.

Implementación de nuevas estrategias de optimización: El módulo demostró ser funcional para implementar estrategias como ADMM y asignación de precios minoristas basada en aprendizaje por refuerzo. Sin embargo, su arquitectura permite integrar otros enfoques como optimización multiobjetivo, métodos estocásticos, programación robusta, algoritmos distribuidos y técnicas de inteligencia artificial. Esto permitiría estudiar problemas asociados a la gestión activa de la demanda, integración de energías renovables y mercados locales de energía. Además, se propone profundizar en el análisis de mercados transaccionales en tiempo real, evaluando el impacto de la comunicación, la latencia y la coordinación distribuida entre múltiples microrredes utilizando el módulo como plataforma experimental.

Aplicación a otros escenarios eléctricos: El módulo puede extenderse para simular: Redes de distribución de mayor escala, sistemas híbridos con almacenamiento energético, comunidades energéticas locales, microredes interconectadas y operación en modo isla y reconexión a red principal. Esto permitiría evaluar escenarios más cercanos a aplicaciones reales de redes inteligentes y transición energética.

Evolución hacia una herramienta transferible: Gracias a su diseño e integración de comunicación híbrida (OpenADR–MQTT), el módulo presenta potencial para evolucionar hacia una plataforma reproducible para simulación distribuida, gestión energética comunitaria y formación en redes inteligentes. Asimismo, su arquitectura podría ser transferida a comunidades o regiones para la gestión local de recursos energéticos, facilitando la coordinación de precios y la optimización del consumo, en línea con la transición hacia sistemas eléctricos inteligentes.

Referencias Bibliográficas

Amelia, A., Roslina, Fahmi, N., Pranoto, H., Sundawa, B. V., Hutauruk, I. S., and Arief, A. (2020). MQTT protocol implementation for monitoring of environmental based on IoT. In *2020 International Conference on Applied Science and Technology (iCAST)*, pages 700–703.

Amjad, A., Azam, F., Anwar, M. W., and Butt, W. H. (2021). A systematic review on the data interoperability of application layer protocols in industrial IoT. *IEEE Access*, 9:96528–96545.

Anderson, P. M. and Fouad, A. A. (2003). *Power System Control and Stability*. IEEE Press; Wiley-Interscience, Piscataway, NJ, 2nd edition.

Aroon, N. (2016). Study of using MQTT cloud platform for remotely control robot and GPS tracking. In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6.

Bender, M., Kirdan, E., Pahl, M.-O., and Carle, G. (2021). Open-source MQTT evaluation. In *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pages 1–4.

Blank, M., Lehnhoff, S., Heussen, K., Morales Bondy, D., Moyo, C., and Strasser, T. (2016). Towards a foundation for holistic power system validation and testing. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4.

Budiawan, I., Pranoto, H. R., Hidayat, E. M. I., and Arief, S. R. (2018). Design and Implementation of Cyber-Physical System-Based Automation on Plant Chemical Process: Study Case Mini Batch Distillation Column. In *2018 6th International Conference on Information and Communication Technology (ICoICT)*, pages 360–365.

Burns, B., Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (2016). *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media.

Carrillo-Valera, C. E. and Sandoval-Manrique, J. H. (2023). Asignación de Precios Minoristas en Sistemas de Distribución con Múltiples Microrredes Usando Aprendizaje por Refuerzo.

Cicceri, G., Tricomi, G., D'Agati, L., Longo, F., Merlino, G., and Puliafito, A. (2023). A Deep Learning-Driven Self-Conscious Distributed Cyber-Physical System for Renewable Energy Communities. *Sensors*, 23(9). https://doi.org/10.3390/s23094549.

Cintuglu, M., Kondabathini, A., and Ishchenko, D. (2020). Real-Time Implementation of Secure Distributed State Estimation for Networked Microgrids. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6.

Ciraci, S., Daily, J., Fuller, J., Fisher, A., Marinovici, L., and Agarwal, K. (2014). Fncs: a framework for power system and communication networks co-simulation. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, DEVS '14, San Diego, CA, USA. Society for Computer Simulation International.

Clifford-Power (2023). *Diesel Generator Fuel Consumption Chart*. Clifford Power.

Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G. (2012). *Distributed Systems: Concepts and Design*. Addison-Wesley, 5 edition.

Dong, S., Cao, J., Flynn, D., and Fan, Z. (2022). Cybersecurity in smart local energy systems: requirements, challenges, and standards. *Energy Informatics*, 5(1):9.

Dorsemaine, B., Gaulier, J.-P., Wary, J.-P., Kheir, N., and Urien, P. (2015). Internet of Things: A Definition Taxonomy. In *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 72–77.

Du, Y. and Li, F. (2020). Intelligent Multi-Microgrid Energy Management based on Deep Neural Network and model-free reinforcement learning. *IEEE Transactions on Smart Grid*, 11(2):1066–1076.

El Meslouhi, S., El Fadil, H., Lassioui, A., Hasni, A., and Fhail, A. (2025). Internet of Energy in Microgrids and Smart Grids: State-of-the-Art. In *2025 5th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 1–7.

Farnós, J. D. (2025). Trabajando con la física del aprendizaje y la computación neural: su rol en la estabilidad y el rendimiento de los sistemas de IA (Universidad e Ingeniería Técnica y Cognitiva).

Faro, A., Giordano, D., and Venticinque, M. (2020). Deploying Wifi, RF and BLE sensors for pervasive monitoring and control. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, pages 605–610. IEEE.

Farrokhabadi, M., Cañizares, C. A., Simpson-Porco, J. W., Nasr, E., Fan, L., Mendoza-Araya, P. A., Tonkoski, R., Tamrakar, U., Hatziargyriou, N., Lagos, D., Wies, R. W., Paolone, M., Liserre, M., Meegahapola, L., Kabalan, M., Hajimiragha, A. H., Peralta, D., Elizondo, M. A., Schneider, K. P., Tuffner, F. K., and Reilly, J. (2020). Microgrid Stability Definitions, Analysis, and Examples. *IEEE Trans. on Power Systems*, 35(1):13–29. https://doi.org/10.1109/TPWRS.2019.2925703.

Gao, C., Liang, T., Li, H., and Zhai, H. (2013). Development and application of open automated demand response. *Power System Technology*, 37(3):692–698.

Ghani, S. A. C., Kettner, M., Steinmann, J., Salim, M. N. M., and Sazali, N. (2024). Using MQTT protocol for controlling manual transmission car on a chassis dynamometer. *AIP Conference Proceedings*, 2998(1):030008. https://doi.org/10.1063/5.0189030.

Grijalva, S. and Tariq, M. U. (2011). Prosumer-based smart grid architecture enables a flat, sustainable electricity industry. In *ISGT 2011*, pages 1–6.

Guo, F., Herrera, L., Alsolami, M., Li, H., Xu, P., Lu, X., Lang, A., Wang, J., and Long, Z. (2013). Design and development of a reconfigurable hybrid microgrid testbed. In *2013 IEEE Energy Conversion Congress and Exposition*, pages 1350–1356.

Hasan, M. K., Habib, A. A., Shukur, Z., Ibrahim, F., Islam, S., and Razzaque, M. A. (2023). Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. *Journal of network and computer applications*, 209:103540.

HiveMQ (2015). MQTT Essentials Part 6: MQTT Quality of Service Levels. https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/.

HiveMQ (2024). HiveMQ: The Industrial AI Platform for MQTT and IoT. https://www.hivemq.com.

Huang, S.-C., Lu, C.-N., and Lo, Y.-L. (2015). Evaluation of AMI and SCADA data synergy for distribution feeder modeling. *IEEE Transactions on Smart Grid*, 6(4):1639–1647. https://doi.org/10.1109/TSG.2015.2408111.

IDEAM (2005). Atlas de radiación solar de colombia.

Ilinca, A., McCarthy, E., Chaumel, J.-L., and Rétiveau, J.-L. (2003). Wind potential assessment of quebec province. *Renewable Energy*, 28(12):1881–1897.

Kaskatiiski, N. and Boyanov, L. (2021). Efficiency of data exchange of IoT communication protocols. In *2021 International Conference Automatics and Informatics (ICAI)*, pages 358–361.

Kaushik, N. and Bagga, T. (2021). Smart cities using IoT. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–6.

Kawoosa, A. I. and Prashar, D. (2021). A review of cyber securities in smart grid technology. In *2021 2nd International conference on computation, automation and knowledge management (ICCAKM)*, pages 151–156. IEEE.

Kayamboo, S., Ray, B., Das, N., and Tom, M. (2022). IoT-Based cyber-physical distribution system planning. In *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6. https://doi.org/10.1109/IEMTRONICS55184.2022.9795702.

Khudhur, D. D. and Croock, M. S. (2021). Developed security and privacy algorithms for cyber physical system. *International Journal of Electrical and Computer Engineering*, 11(6):5379.

Liao, Y., de Freitas Rocha Loures, E., and Deschamps, F. (2018). Industrial Internet of Things: A Systematic Literature Review and Insights. *IEEE Internet of Things Journal*, 5(6):4515–4525.

Light, R. (2017). Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software*, 2(13):265.

Lin, M.-H., Wu, S.-H., Huang, B.-W., Chen, P.-H., Huang, C.-H., Chen, C.-Y., and Yang, C.-F. (2024). Node-RED Web-based Monitor and Control of Power System Using Modbus and Message Queuing Telemetry Transport Communication in Raspberry Pi Embedded Platform. *Sensors & Materials*, 36.

Lumos Controls (2023). How OpenADR take the demand response to the next level? https://www.lumoscontrols.com/blog/how-openadr-take-the-demand-response-to-the-next-level/.

Magade, K. and Sharma, A. (2023). The Significant Role of IoT and Cyber-Physical Systems in Context-Awareness and Ambient Intelligence. In Sharma, A., Magade, K., and Reddy, S. R. N.,

editors, *The Next Generation Innovation in IoT and Cloud Computing with Applications*, pages 23–38. CRC Press, Boca Raton, 1 edition.

Marz, N. and Warren, J. (2015). *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Manning Publications.

McParland, C. (2011). OpenADR open source toolkit: Developing open source software for the Smart Grid. In *2011 IEEE power and energy society general meeting*, pages 1–7. IEEE.

Meera, V. M. and Arjun, K. P. (2024). *Challenges in Ensuring Security for Smart Energy Management Systems Based on CPS*, chapter 7, pages 217–254. John Wiley Sons, Ltd.

Meyer, O., Ollesch, J., Gries, S., Wessling, F., and Gruhn, V. (2017). Property-based routing in clustered message brokers for CPS: Doctoral Symposium. In *Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems*, DEBS '17, page 366–369, New York, NY, USA. Association for Computing Machinery.

Mohapatra, D. and Subudhi, B. (2022). Development of a cost-effective IoT-based weather monitoring system. *IEEE Consumer Electronics Magazine*, 11(5):81–86.

Nelson, A., Chakraborty, S., Wang, D., Singh, P., Cui, Q., Yang, L., and Suryanarayanan, S. (2016). Cyber-physical test platform for microgrids: Combining hardware, hardware-in-the-loop, and network-simulator-in-the-loop. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5. https://doi.org/10.1109/PESGM.2016.7741176.

Nguyen, T.-L., Tran, Q.-T., Caire, R., Luu, N.-A., and Besanger, Y. (2019). Controller hardware-in-the-loop implementation for agent-based distributed optimal power flow using ADMM on cyber-physical microgrids. In *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pages 712–717. https://doi.org/10.1109/GTDAsia.2019.8715852.

OASIS (2014). MQTT Version 3.1.1. `https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/`.

OASIS (2019). MQTT Version 5.0. `https://docs.oasis-open.org/mqtt/mqtt/v5.0/`.

Palmintier, B., Krishnamurthy, D., Top, P., Smith, S., Daily, J., and Fuller, J. (2017). Design of the helics high-performance transmission-distribution-communication-market co-simulation framework. In *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6.

Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239. https://doi.org/10.1561/2400000003.

Parrado Duque, A., Ordóñez Plata, G., and Osma Pinto, G. A. (2018). Proposal for Advanced Metering Infrastructure in Distribution Systems with Generation Distributed in Low Voltage Electrical Networks. In *2018 IEEE ANDESCON*, pages 1–6. https://doi.org/10.1109/ANDESCON.2018.8564587.

Pereira, F. and Gomes, L. (2018). A JSON/HTTP communication protocol to support the development of distributed cyber-physical systems. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pages 23–30.

PickData (2019). MQTT vs CoAP, the battle to become the best IoT protocol. https://www.pickdata.net/news/mqtt-vs-coap-best-iot-protocol.

Rahman, S. and Parizad, A. (2025). Energy Efficiency in Smart Buildings Through IoT Sensor Integration: Implementation of BEMOSS TM Platform. *Smart Cyber-Physical Power Systems: Fundamental Concepts, Challenges, and Solutions*, 1:247–291.

Rey, J. M., Carrillo-Valera, C., Sandoval-Manrique, J., Luna, A., and Serna Suárez, I. D. (2025). A Reinforcement Learning-Based Approach for Retail Electricity Pricing Strategy for Multi-Microgrid Distribution Systems. *Ingeniería y Desarrollo*, 44(1):109–129.

Rey, J. M., Vera, G. A., Acevedo-Rueda, P., Solano, J., Mantilla, M. A., Llanos, J., and Sáez, D. (2022). A review of microgrids in Latin America: Laboratories and test systems. *IEEE Latin America Transactions*, 20(6):1000–1011. https://doi.org/10.1109/TLA.2022.9757743.

Rey, J. M., Vergara, P. P., Solano, J., and Ordóñez, G. (2019). Design and optimal sizing of microgrids. *Microgrids Design and Implementation*, pages 337–367. https://doi.org/10.1007/978-3-319-98687-6$_1$3.

Rui, X., Liu, M., Sahraei-Ardakani, M., and Nudell, T. R. (2022). ADMM-Based Distributed DC Optimal Power Flow with Power Flow Control. In *2022 North American Power Symposium (NAPS)*, pages 1–6. https://doi.org/10.1109/NAPS56150.2022.10012190.

Sahraei Manjili, Y., Rajaee, A., Jamshidi, M., and Kelley, B. T. (2012). Intelligent decision making for energy management in microgrids with air pollution reduction policy. In

2012 7th International Conference on System of Systems Engineering (SoSE), pages 13–18. https://doi.org/10.1109/SYSoSE.2012.6384132.

Stefan Sacala, I., Pop, E., Alexandru Moisescu, M., Dumitrache, I., Iuliana Caramihai, S., and Culita, J. (2021). Enhancing CPS Architectures with SOA for Industry 4.0 Enterprise Systems. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 71–76.

Tanenbaum, A. S. and Van Steen, M. (2017). *Distributed Systems: Principles and Paradigms*. CreateSpace Independent Publishing Platform, 2 edition.

Thirugnanam, K., Moursi, M. S. E., Khadkikar, V., Zeineldin, H. H., and Al Hosani, M. (2021). Energy Management of Grid Interconnected Multi-Microgrids Based on P2P Energy Exchange: A Data Driven Approach. *IEEE Transactions on Power Systems*, 36(2):1546–1562. https://doi.org/10.1109/TPWRS.2020.3025113.

Tian, L., Cheng, L., Wan, Y., Yuan, K., Liu, R., and Wu, K. (2020). From Distributed Energy Resources to Virtual Power Plants: A Cyber-Physical System Solution for Integrating Demand-side in Smart Grid. In *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, pages 3463–3467. https://doi.org/10.1109/EI250167.2020.9346925.

Ton, D. T. and Smith, M. A. (2012). The U.S. Department of Energy's Microgrid Initiative. *The Electricity Journal*, 25(8):84–94. https://doi.org/10.1016/j.tej.2012.09.013.

Torres-Olguin, R. E., Sanchez-Acevedo, S., Mo, O., and Garcés-Ruiz, A. (2023). Cyber-Physical Power System Testing Platform for Topology Identification in

Power Distribution Grids. In *2023 IEEE Belgrade PowerTech*, pages 1–6. https://doi.org/10.1109/PowerTech55446.2023.10202733.

Turnbull, J. (2014). *The Art of Monitoring*. James Turnbull.

Vergara, P. P., Rey, J. M., Osma, G. A., and Ordoñez, G. (2014). Evaluación del potencial solar y eólico del campus central de la Universidad Industrial de Santander y la ciudad de Bucaramanga, Colombia. *UIS Ingenierías*, 13(2):49–57.

Wang, B., Guo, Q., and Yu, Y. (2022). Mechanism design for data sharing: An electricity retail perspective. *Applied Energy*, 314:118871. https://doi.org/10.1016/j.apenergy.2022.118871.

Xie, J., Bedoya, J. C., Liu, C.-C., Hahn, A., Kaur, K. J., and Singh, R. (2018). New educational modules using a cyber-distribution system testbed. *IEEE Transactions on Power Systems*, 33(5):5759–5769.

Xu, L. D., He, W., and Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243.

Xu, T., Chen, T., and Gao, C. (2022). Intelligent Energy Management Strategy with Internal Pricing Sensitivity for Residential Customers Based on Reinforcement Learning and Internet-of-Things. In *2022 IEEE 6th Conference on Energy Internet and Energy System Integration (EI2)*, pages 1760–1765. https://doi.org/10.1109/EI256261.2022.10116749.

Yang, K., Zhang, B., Zhang, J., and Zhu, J. (2021). Design of Remote Control Inverter Based on

MQTT Communication Protocol. In *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1374–1378.

Yi, L., Fanbo, M., Nan, W., Miao, Y., Junnan, W., and Wangzhang, C. (2015). OpenADR protocol of the wireless transmission. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 1059–1062. IEEE.

Apéndice A. Guía de uso módulo ciber-físico IoT

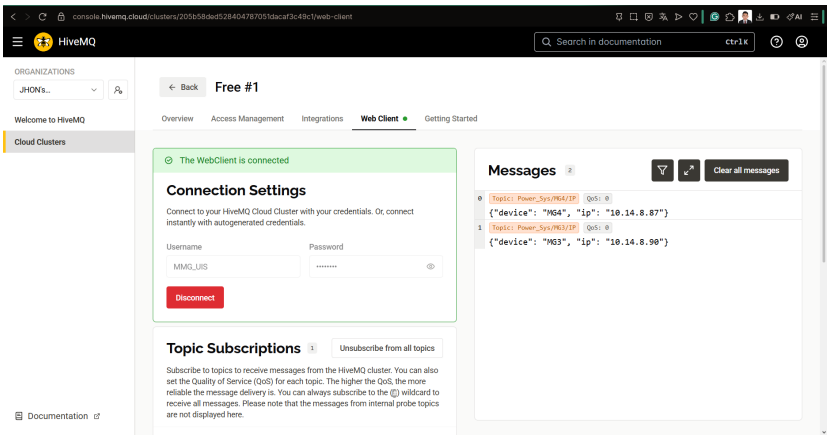
Acceso remoto a las tarjetas

Al iniciar o energizar las tarjetas que conforman el módulo, es importante tener en cuenta que únicamente las Raspberry Pi cuentan con dirección IP fija, debido a que se encuentran conectadas a la red mediante Ethernet. En contraste, las Odroid, al estar conectadas mediante Wi-Fi, reciben una dirección IP dinámica, la cual puede cambiar en cada reinicio del sistema.

Con el fin de facilitar su localización en la red, se implementó un script en Python que se ejecuta automáticamente al encender cada dispositivo. Este script envía la dirección IP asignada al broker HiveMQ, publicándola en un tópico previamente definido, como se muestra en la Figura 34. Por esta razón, se recomienda que, antes de encender el módulo completo, el usuario acceda primero a la plataforma HiveMQ utilizando las credenciales correspondientes, con el fin de monitorear la publicación de las direcciones IP. Una vez identificadas, se puede establecer la conexión remota a cada tarjeta mediante el protocolo SSH, permitiendo así su administración y supervisión.

Figura 34

Interfaz HiveMQ con IPs de las Odroids.



El protocolo SSH (Secure Shell) es un protocolo de red seguro que permite controlar dispositivos de forma remota desde otro ordenador, accediendo a su línea de comandos (terminal) mediante una conexión cifrada. En el contexto del módulo ciberfísico IoT, SSH se utiliza para administrar las tarjetas Raspberry Pi y Odroid desde el ordenador central. Para establecer la conexión remota es necesario:

- Tener habilitado el servicio SSH en cada tarjeta.
- Conocer la dirección IP asignada al dispositivo.
- Utilizar un cliente SSH, como la terminal del sistema operativo (Windows, macOS o Linux) o herramientas como PuTTY.

Una vez identificada la dirección IP de cada tarjeta (según el procedimiento descrito), la conexión se realiza desde la terminal del ordenador central mediante el siguiente comando general:

```
ssh nombre_usuario@IP_tarjeta
```

donde:

- `nombre_usuario` corresponde al usuario configurado en la tarjeta.
- `IP_tarjeta` es la dirección IP asignada al dispositivo.

A continuación, se muestra el procedimiento de conexión para las cuatro tarjetas que conforman el módulo.

Conexión a las Raspberry Pi. En el caso de las Raspberry Pi, al estar conectadas mediante Ethernet, cada una cuenta con una dirección IP fija y un nombre de host configurado dentro de la red local. Esto permite establecer la conexión remota utilizando directamente el nombre del dispositivo, sin necesidad de especificar manualmente la dirección IP.

La conexión se realiza desde la terminal del ordenador central mediante el siguiente formato:

```
ssh usuario@nombre_host
```

Para este módulo, las conexiones se realizaron utilizando los siguientes comandos:

```
ssh JHONSANDOVAL09@MG1-UIS y ssh JHONSANDOVAL09@raspberrypiUIS
```

Una vez ejecutado el comando correspondiente, el sistema solicita la contraseña configurada para el usuario. Tras ingresar las credenciales correctas, se obtiene acceso a la terminal de la tarjeta, permitiendo su administración y ejecución de scripts.

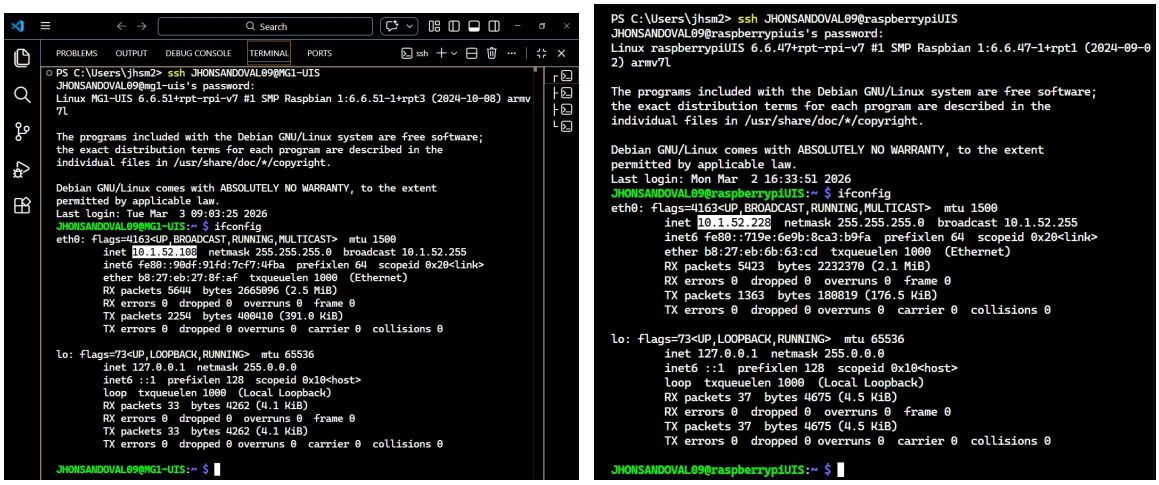
En la Figura 35 se muestran las direcciones IP asignadas a cada Raspberry Pi (resaltadas), aunque la conexión se realizó mediante el nombre de host, lo que simplifica el acceso dentro de la red local.

Conexión a las odroid. En el caso de las Odroid, al estar conectadas a la red mediante Wi-Fi, estas reciben una dirección IP dinámica, la cual puede cambiar cada vez que el dispositivo se reinicia. Por esta razón, es necesario consultar previamente la dirección IP publicada en el servidor HiveMQ, según el procedimiento descrito en la sección anterior.

Una vez identificada la dirección IP correspondiente, la conexión remota se realiza desde la terminal del ordenador central mediante el siguiente formato:

Figura 35

Acceso remoto a las Raspberry Pi.



(a) Acceso remoto RPi 1. (b) Acceso remoto RPi 2.

```
ssh usuario@IP_dispositivo
```

Para el módulo implementado, las conexiones se realizaron utilizando los siguientes comandos:

```
ssh odroid@10.14.8.87 y ssh odroid@10.14.8.90
```

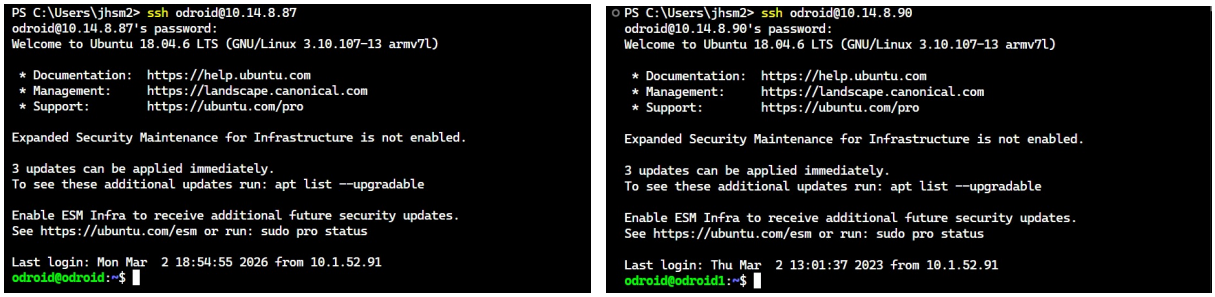
Después de ejecutar el comando, el sistema solicita la contraseña configurada para el usuario del dispositivo. Tras ingresar las credenciales correctas, se obtiene acceso a la terminal de la Odroid, permitiendo la ejecución de scripts y la supervisión del sistema.

Es importante resaltar que, debido a la naturaleza dinámica de la conexión Wi-Fi, este procedimiento debe repetirse cada vez que las Odroid sean reiniciadas o cambien de red. En la Figura 36 se evidencia su conexión remota.

Nota: Para hacer la conexión a cada una de las tarjetas, se usa un terminal diferente. Se recomienda abrir el terminal en VSCode, como se muestra en la Figura 37.

Figura 36

Acceso remoto a las Odroids.



(a) Acceso remoto odroid 1. (b) Acceso remoto odroid 2.

Figura 37

Terminal con conexión a todas las tarjetas.

Acceso al entorno virtual para la ejecución de los scripts de simulación

Cada una de las tarjetas que conforman el módulo ciberfísico IoT cuenta con un entorno virtual de Python previamente configurado. Este entorno contiene todas las librerías necesarias para la ejecución de los scripts de simulación, tales como herramientas de optimización, comunicación MQTT y manejo de datos. La activación del entorno virtual es un paso fundamental, ya que garantiza:

- El uso de las versiones correctas de las librerías.
- El aislamiento del sistema base del dispositivo.
- La correcta ejecución de los scripts de simulación.

Procedimiento de activación

Primero, se debe acceder a la carpeta donde se encuentra ubicado el entorno virtual correspondiente en cada tarjeta, como se muestra en la Figura 38.

Una vez dentro del directorio del proyecto, se ejecuta el siguiente comando en la terminal:

```
source env/bin/activate
```

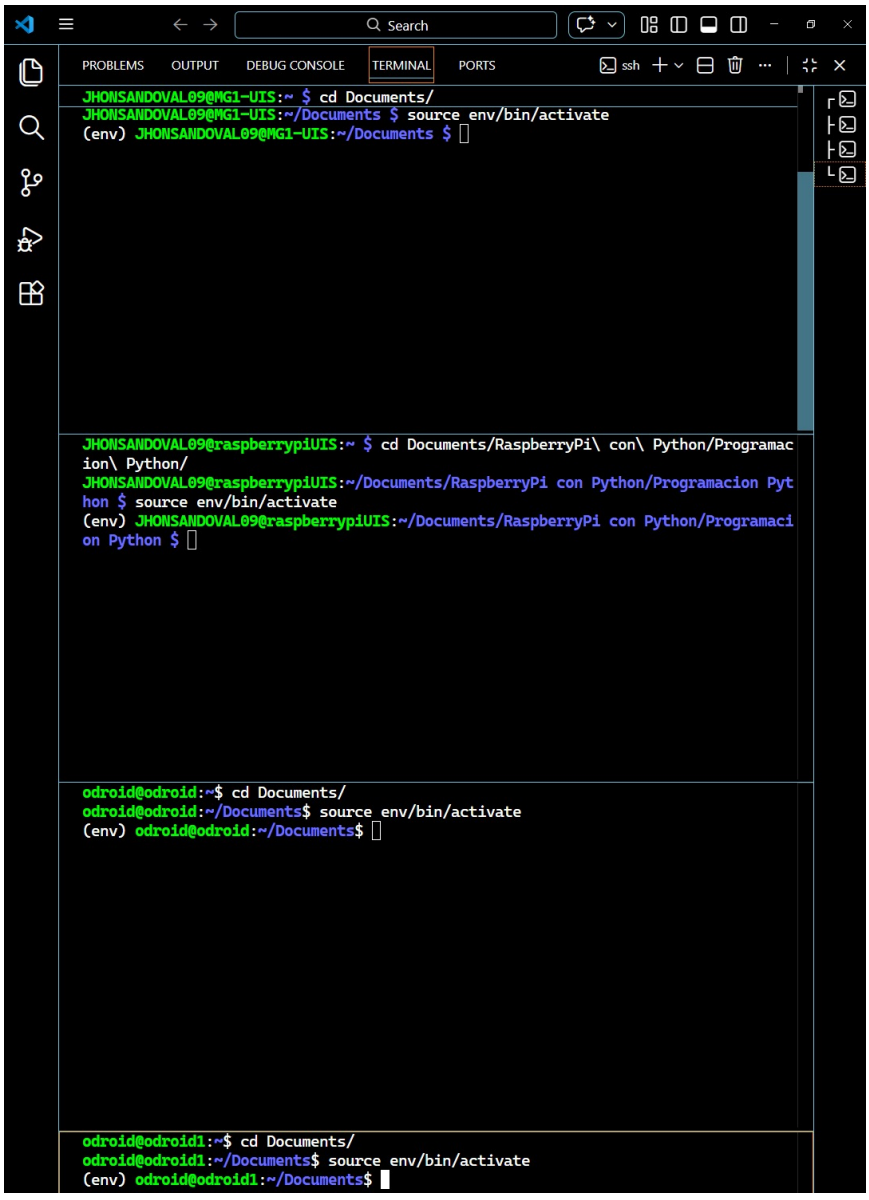
Figura 38

Terminal con ingreso de ruta de ejecución de todas las tarjetas.

Tras ejecutar este comando, el nombre del entorno virtual aparecerá al inicio de la línea de comandos, indicando que el entorno ha sido activado correctamente, como se muestra en la Figura 39.

Figura 39

Terminal con entorno virtual activado en todas las tarjetas.



A partir de este momento, es posible ejecutar los scripts de simulación en Python con todas las dependencias necesarias cargadas.

Ejecución del script de simulación en las tarjetas

Una vez activado el entorno virtual en cada dispositivo, se procede a ejecutar el script correspondiente a la simulación descentralizada.

En las raspberry pi. En el caso de las raspberry pi, el script se ejecuta mediante el siguiente comando:

```
python Validation_Code.py
```

En las odroid. Para las tarjetas odroid, el script se ejecuta con privilegios de administrador, utilizando el siguiente comando:

```
sudo python3 Validation_Code.py
```

Después de ejecutar el comando, el sistema solicitará la contraseña correspondiente al usuario configurado en la tarjeta. Una vez ingresadas las credenciales correctamente, el script iniciará su ejecución.

Estado de espera y verificación de funcionamiento. Tras la ejecución del script:

- Cada tarjeta queda en estado de espera, suscrita a los tópicos MQTT definidos.
- El dispositivo permanece escuchando mensajes provenientes del ordenador central.
- Se activa un indicador luminoso (LED azul), el cual comienza a parpadear, señalando que el dispositivo se encuentra correctamente conectado y en espera de recibir instrucciones.

Este comportamiento puede observarse en la Figura 40 (estado de escucha MQTT) y en la Figura 41 (indicador luminoso activo).

Figura 40

Terminal con las tarjetas escuchando mensajes MQTT.

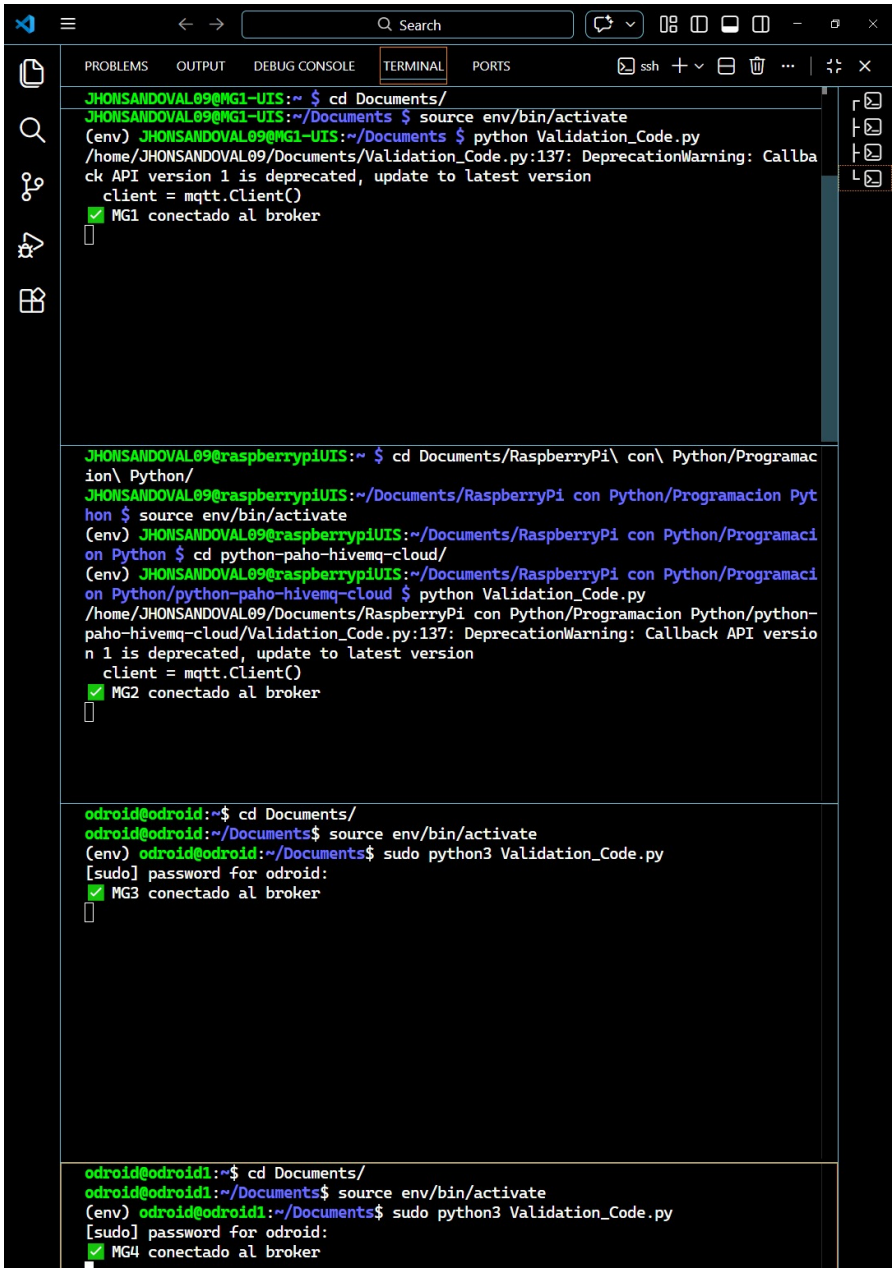
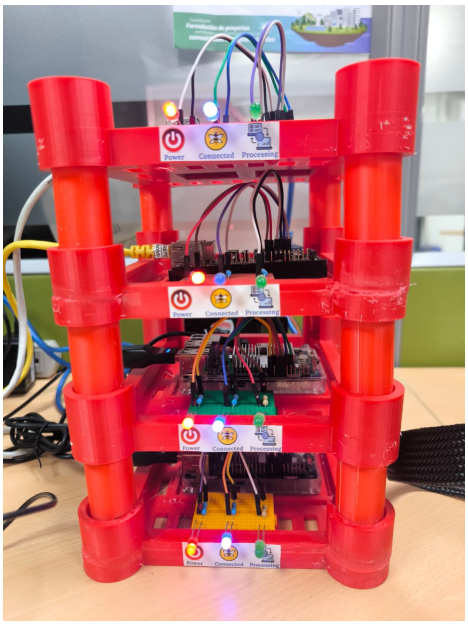


Figura 41

Estructura módulo con indicador azul parpadeando.



Inicio de simulación ordenador central

Activación del servidor VTN (OpenADR)

Una vez que todas las tarjetas (Raspberry Pi y Odroid) se encuentran encendidas, conectadas por SSH y ejecutando el script de simulación en modo espera, el siguiente paso para iniciar la operación del módulo ciberfísico IoT es activar el servidor VTN del protocolo OpenADR.

El VTN (Virtual Top Node) actúa como entidad central dentro de la arquitectura OpenADR, permitiendo la gestión de eventos y la coordinación con los dispositivos registrados como VEN (Virtual End Node).

Ubicación del servidor VTN. El servidor OpenADR se ejecuta desde el Ordenador Central, donde se encuentra configurado su entorno virtual en la siguiente ruta:

C:\Users\jhsm2\vtn_service

Para activarlo, se debe:

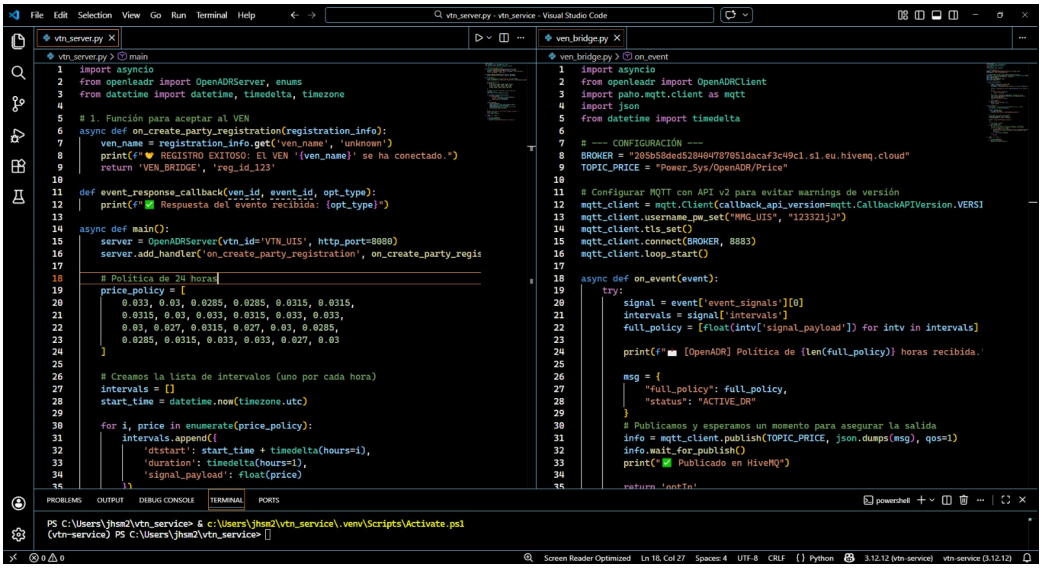
1. Abrir una nueva ventana de Visual Studio Code (VSCode).
2. Seleccionar la opción **“Open Folder”**
3. Abrir el directorio: **C:\Users\jhsm2\vtn_service**

En este directorio se encuentran los scripts principales del servidor y este entorno se muestra en la Figura 42:

- `vtn_server.py` → Script principal del servidor VTN.
- `ven_bridge.py` → Script que permite la integración híbrida entre OpenADR y MQTT, funcionando como puente de comunicación entre ambos protocolos.

Figura 42

VSCode con el entorno de inicio de OpenADR.



Orden correcto de ejecución de la simulación

Para garantizar el funcionamiento adecuado de la arquitectura híbrida OpenADR–MQTT, es fundamental respetar estrictamente el orden de ejecución de los scripts en el Ordenador Central. Cada uno de ellos cumple una función específica dentro del sistema.

Paso 1: Activar el servidor OpenADR (VTN). El primer paso consiste en ejecutar el script `vtn_server.py`, el cual inicia el servidor VTN (Virtual Top Node) del protocolo OpenADR. Este servidor será el encargado de gestionar la señal de precios y los eventos energéticos que posteriormente serán enviados al sistema. Desde el directorio: **C:\Users\jhsm2\vtn_service**

Se ejecuta el siguiente comando en la terminal de VSCode:

```
uv run .\vtn_server.py
```

Una vez ejecutado correctamente, el servidor OpenADR quedará activo y en espera de conexiones de los nodos VEN. Este paso debe ejecutarse primero, ya que los demás scripts dependen de que el VTN esté en funcionamiento. En este paso debe mostrarse la terminal como se muestra en la Figura 43.

Figura 43

Terminal mostrando activación de servidor VTN.

Paso 2: Ejecutar el sistema principal descentralizado. El segundo paso consiste en ejecutar el script principal del sistema:

`Validation_System_Descentralized.py`

Este archivo se encuentra en el siguiente directorio: **C:\Users\jhsm2\mi-proyecto**

Importante: Este script debe ejecutarse en una ventana diferente de VSCode, ya que utiliza un entorno virtual distinto al del servidor VTN. El comando de ejecución es:

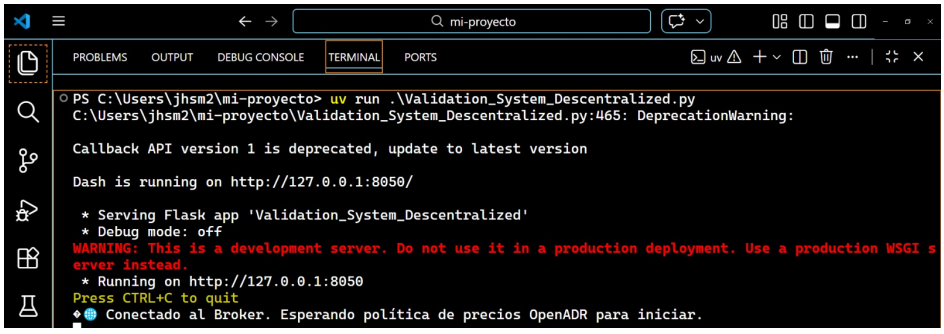
```
uv run .\Validation_System_Descentralized.py
```

Al ejecutarlo: El sistema inicializa la simulación, se prepara la comunicación con las tarjetas vía MQTT, se establece la lógica de coordinación descentralizada e inicia la *Dashboard* con el gemelo digital y el monitor en tiempo real.

El estado de ejecución puede observarse en la terminal, como se muestra en la Figura 44.

Figura 44

Mensaje en terminal del inicio del script principal.



Paso 3: Ejecutar el puente de comunicación OpenADR–MQTT. Finalmente, se debe ejecutar el script `ven_bridge.py`, el cual cumple la función de traducir la política de precios generada en OpenADR hacia mensajes MQTT entendibles por las tarjetas del módulo.

Desde el directorio: **C:\Users\jhsm2\vtn_service**

Se ejecuta:

```
uv run .\ven_bridge.py
```

Este script: Se conecta al servidor VTN, recibe la señal de precio, la convierte en mensajes MQTT, publica la información hacia las microrredes y da inicio formal a la simulación.

El funcionamiento de este paso se muestra en la Figura 45.

Figura 45

Mensaje en terminal del inicio del VEN.

Resumen del orden obligatorio de ejecución. `vtn_server.py` → Activa OpenADR (VTN). `Validation_System_Descentralized.py` → Inicializa el sistema descentralizado. `ven_bridge.py` → Traduce la política de precios y arranca la simulación.

Inicio de la simulación y verificación de funcionamiento

Una vez ejecutados correctamente los tres scripts en el orden establecido la simulación de validación inicia automáticamente.

Despliegue del gemelo digital. Al comenzar la simulación:

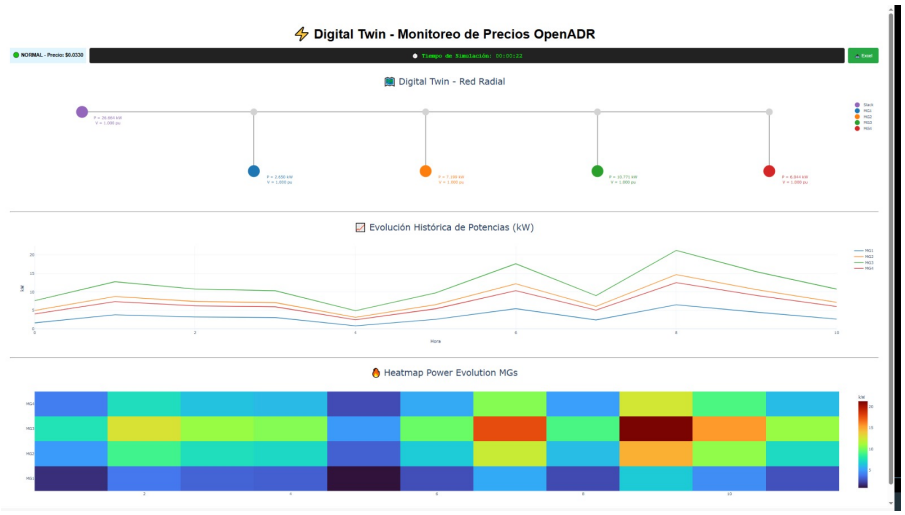
- Se abre automáticamente la Dashboard del gemelo digital desarrollada en Dash.

- En la interfaz se visualiza el estado del sistema eléctrico en tiempo real.
- Se muestran las curvas de potencia de cada microrred.
- Se actualizan los resultados hora a hora durante las 2400 realizaciones (100 días).

Esta interfaz permite monitorear: Potencia intercambiada, evolución de cada microrred, tiempo total de simulación e historial completo del proceso como se muestra en la Figura 46.

Figura 46

Dashboard con gemelo digital y monitoreo en tiempo real.



Actividad en las tarjetas. De manera simultánea:

- En cada terminal SSH de las tarjetas comienzan a visualizarse los mensajes (`print`) correspondientes al proceso interno de simulación.
- Se evidencia la recepción de señales vía MQTT.
- Se muestra el cálculo de la optimización local.

- Se confirma el envío de resultados al ordenador central.

Indicadores luminosos de operación. Durante la ejecución:

- Los indicadores verdes de cada tarjeta comienzan a parpadear.
- Este parpadeo indica que: Se está ejecutando el proceso interno de optimización, Se están enviando y recibiendo mensajes MQTT y El sistema está operando correctamente. El parpadeo continuo confirma que la comunicación y el procesamiento descentralizado se están realizando de manera adecuada.

Criterio de verificación correcta de la simulación. Se considera que la simulación está funcionando correctamente cuando:

- El servidor VTN permanece activo.
- La Dashboard muestra actualizaciones en tiempo real.
- Las terminales de las tarjetas imprimen resultados continuamente.
- Los LEDs verdes parpadean de forma constante.

Nota: Al finalizar el uso del módulo, se recomienda apagar correctamente las tarjetas para prevenir posibles daños, utilizando el siguiente comando:

```
sudo shutdown -h now
```

Finalmente, en el siguiente enlace se evidencia el correcto funcionamiento de la simulación: `https://youtu.be/AXyC2ay-YGs`.