

Desarrollo de un sistema electrónico para comunicación remota en dispositivos industriales Modbus RTU a través de MQTT

Development of an electronic system for remote communication in modbus RTU industrial devices via MQTT

MSc(c). Guillermo Acevedo ¹, PhD. Omar Pinzón ¹

¹ Universidad Pontificia Bolivariana seccional Bucaramanga, Facultad de ingeniería eléctrica y electrónica, Floridablanca, Santander, Colombia.

Correspondencia: omar.pinzon@upb.edu.co

Recibido: 15 octubre 2023. Aceptado: 17 diciembre 2023. Publicado: 01 marzo 2024.

Cómo citar: G. A. Acevedo Hernández y O. Pinzón Ardila, «Desarrollo de un sistema electrónico para comunicación remota en dispositivos industriales Modbus RTU a través de MQTT», RCTA, vol. 1, n.º 43, pp. 44–50, mar. 2024.
Recuperado de <https://ojs.unipamplona.edu.co/index.php/rcta/article/view/2800>

Derechos de autor 2024 Revista Colombiana de Tecnologías de Avanzada (RCTA).
Esta obra está bajo una licencia internacional [Creative Commons Atribución-NoComercial 4.0](https://creativecommons.org/licenses/by-nc/4.0/).



Resumen: Este documento muestra el proceso de desarrollo y las fases iniciales de prueba de un sistema electrónico diseñado para facilitar la comunicación a través de la web de un dispositivo industrial. El dispositivo industrial opera mediante comunicación Modbus RTU y el sistema emplea el protocolo de telemetría MQTT, el cual está integrado en un microcontrolador de 32 bits. El microcontrolador opera dentro de un marco de sistema operativo en tiempo real, asegurando una gestión eficiente de la comunicación. En el desarrollo del equipo se ha establecido la integración del sistema con un PLC y se ha logrado la validación mediante la utilización de una base de datos integral, scripts de Python y añadiendo capacidades de visualización proporcionadas por una plataforma basada en la web. El prototipo es capaz de leer y modificar de forma remota registros Modbus a través de una interfaz web que mediante una interfaz no sólo permite ajustes de parámetros, sino que también ofrece acceso a los registros históricos de mediciones, mejorando la funcionalidad general del dispositivo industrial.

Palabras clave: IIOT, MQTT, Modbus, Microcontroladores, RTOS.

Abstract: This document presents the development process and initial testing phases of an electronic system designed to facilitate web-based communication for an industrial device. The industrial device operates using Modbus RTU communication, and the system utilizes the MQTT telemetry protocol, which is integrated into a 32-bit microcontroller. The microcontroller operates within a real-time operating system framework, ensuring efficient communication management. The system integration with a PLC has been established during the equipment development, and validation has been achieved through the utilization of a comprehensive database, Python scripts, and the incorporation of visualization capabilities provided by a web-based platform. The prototype is capable of remotely reading and modifying Modbus registers through a web interface, which not only allows parameter adjustments but also provides access to historical measurement records, enhancing the overall functionality of the industrial device.

Keywords: IIOT, MQTT, Modbus, Microcontroller, RTOS.

1. INTRODUCCIÓN

Desde la Revolución Industrial, ha existido una constante búsqueda de la monitorización continua de equipos y maquinaria. En sus inicios, se requería la presencia constante de personal para supervisar el funcionamiento de cada equipo [1]. Con el paso de los años, se desarrollaron sensores y comunicaciones industriales primitivas, con el propósito fundamental de garantizar la seguridad de los trabajadores, el óptimo desempeño de los equipos y la implementación de mantenimiento preventivo [2]. Con el tiempo, surgieron diversos protocolos de comunicación industrial, entre ellos Modbus, que se convirtió en uno de los protocolos más utilizados a nivel mundial [3]. No obstante, Modbus presenta limitaciones, como la restricción de 255 clientes conectados por nodo y la comunicación que solo se inicia cuando el servidor solicita datos, careciendo de la posibilidad de un acceso de tipo asíncrono.

En la actualidad, en un mundo caracterizado por la presencia del internet y distancias extendidas, resulta imperativo que las comunicaciones industriales evolucionen, especialmente en industrias como el Petróleo y Gas, así como en el transporte de estos recursos, donde la información en tiempo real de los activos es crucial para la toma de decisiones [4]. Empresas innovadoras han instalado cables de fibra óptica junto a tuberías para lograr comunicación y detección de movimientos [5]. Sin embargo, el costo de implementar una infraestructura extensa de fibra óptica y la vulnerabilidad a daños, particularmente en regiones como Colombia, hacen necesaria la implementación de comunicaciones inalámbricas para evitar pérdida de información.

En la literatura se encuentra que diferentes autores han desarrollado avances en comunicaciones inalámbricas utilizando diversos protocolos a través de la web [6], [7], [8]. Sin embargo, algunas de estas soluciones resultan costosas debido al alto consumo de datos. Con lo anterior hay que adicionar que las direcciones IP fijas en comunicaciones inalámbricas generan gastos adicionales. Para abordar estos desafíos, surge la tecnología Publicador-Subscriber (Pub-Sub). A diferencia del modelo cliente-servidor ampliamente utilizado, Pub-Sub implica que los publicadores colocan mensajes en un intermediario (broker), que los suscriptores recuperan utilizando sus identificadores. Los mensajes permanecen en la nube, asegurando la disponibilidad de datos incluso si la conexión a Internet es intermitente [9].

Entre los protocolos Pub-Sub, destaca MQTT (MQ Telemetry Transport). MQTT es un protocolo de comunicación liviano con un sobre costo mínimo, capaz de conectar múltiples dispositivos simultáneamente, lo que lo convierte en un componente crucial para el desarrollo del Internet de las Cosas (IoT). El protocolo MQTT se ha utilizado ampliamente en IoT, permitiendo el monitoreo continuo de equipos y PLC a través de Modbus [10], [11], [12]. Sin embargo, la literatura señala la necesidad de una comunicación bidireccional para controlar Modbus nativo, ya que las aplicaciones existentes sólo permiten la lectura de variables y no la escritura. Por lo tanto, este trabajo propone una comunicación bidireccional para la configuración de registros Modbus a través de la web utilizando MQTT.

Desarrollos previos en la literatura a menudo se centran en dispositivos de un solo uso o máquinas individuales. Este trabajo tiene como objetivo crear un dispositivo versátil aplicable a diferentes máquinas industriales, PLC y equipos de monitoreo remoto que emplean el protocolo de comunicación Modbus [11], [10], [13], [14]. La centralización de varios dispositivos en una sola plataforma, personalizada según el conocimiento de los equipos conectados, sin necesidad de cableado adicional, es viable. Es posible crear una aplicación tipo SCADA a través de la web, aprovechando la capacidad de MQTT para integrar múltiples suscriptores con funcionalidades independientes simultáneamente [15].

La seguridad de la comunicación inalámbrica sigue siendo una preocupación importante para la industria, con temores de pérdida de datos debido a desconexiones. Para abordar esto, la solución propuesta incluye una memoria SD para garantizar los datos. La confirmación de calidad de servicio (QoS) de MQTT asegura la recepción de datos por parte del intermediario (broker). La memoria SD funciona como una FIFO, almacenando y reenviando datos durante la recuperación de la red.

2. DESARROLLO

El proceso inicial en el desarrollo del dispositivo se debe seleccionar el microcontrolador y el sistema operativo en tiempo real (RTOS) más apropiado para el desarrollo proyecto. Para tomar esta decisión, se consideraron diversos parámetros esenciales, como la capacidad de memoria para almacenar el programa, la velocidad del reloj del microcontrolador, la disponibilidad de una tarjeta de desarrollo con puerto Ethernet, el uso de software de

programación de código abierto y la integración con el hardware del depurador del programa.

En relación al RTOS, se realizó un análisis exhaustivo que cubre aspectos como el espacio de memoria requerido por el sistema, las capacidades de planificación de tareas, la compatibilidad con el dispositivo específico, la escalabilidad, el soporte ofrecido, la documentación disponible, el rendimiento obtenido y los costos asociados. Considerando estos criterios se determina optar por la placa de desarrollo MSP432E LaunchPad de Texas Instruments, junto con el RTOS proporcionado por la misma empresa, conocido como TI-RTOS.

Esta elección se basa en una evaluación exhaustiva que garantiza la alineación de las necesidades con las capacidades y características ofrecidas por la MSP432E LaunchPad y el RTOS TI-RTOS. Ambos elementos se complementan para brindar un entorno de desarrollo eficiente y confiable.

A continuación, se desarrolla la librería Modbus Master, que cumple el rol de intermediario entre el dispositivo y el equipo con el cual se establece la comunicación. En este proceso, es esencial considerar la configurabilidad de los parámetros Modbus, asegurando la capacidad de conexión sin inconvenientes a una amplia gama de dispositivos Modbus. Además, se busca permitir la conexión simultánea a múltiples dispositivos, cada uno con su propio ID distintivo.

Una vez completada la creación de la librería, se realizan pruebas exhaustivas de comunicación entre el microcontrolador y diversos dispositivos Modbus. Estas pruebas involucraron la modificación de parámetros, como la velocidad del puerto serial, la paridad, el ID Modbus, los métodos de comunicación RS232 y RS485, el número y tipo de registros, entre otros. Estos ajustes permitieron verificar de manera minuciosa la funcionalidad de la librería diseñada para esta tarea específica.

El siguiente paso consiste en la elaboración del código necesario para la comunicación a través de Ethernet. Para lograrlo, se lleva a cabo la configuración de los puertos del microcontrolador, utilizando las pertinentes librerías del RTOS. Esta configuración abarca la preparación del puerto y la implementación de los pasos requeridos para establecer una conexión TCP/IP entre el equipo y la infraestructura de Internet.

Como parte de la fase de desarrollo y pruebas, se realizan verificaciones de conectividad y funcionamiento utilizando pruebas de tipo *Ping*. Estas pruebas son esenciales para garantizar que el equipo se encuentre correctamente conectado a Internet y que la comunicación se esté llevando a cabo de manera efectiva.

Luego de esto, se procede a desarrollar el módulo de comunicaciones MQTT. Para este propósito, se implementan funciones centradas a la conexión, desconexión, suscripción y publicación de mensajes. Además, se considera la implementación de tareas asincrónicas del RTOS, las cuales se ajustan de acuerdo al modo de operación específico de MQTT.

Un aspecto clave es la implementación de múltiples formatos de texto para la transmisión de datos mediante MQTT. Esta flexibilidad permite al usuario final seleccionar el formato más adecuado para la plataforma de destino y el nivel de seguridad deseado en el sistema. Entre los formatos de texto incorporados destacan JSON, BSON, entre otros.

Entre las funcionalidades a destacar de este dispositivo se resalta su capacidad de almacenamiento de datos y de cola para envíos, útil en situaciones de comunicación intermitente. Con esta finalidad, se integra una memoria SD, donde los datos se almacenan en el formato de texto seleccionado. Cuando se restablece la conexión a Internet, los datos se transmiten en el mismo orden en que fueron almacenados en el equipo. La configuración de los puertos del microcontrolador se ajustó para el manejo de la SD, aprovechando las librerías disponibles para esta tarea.

Adicionalmente, se implementan diversos periféricos personalizables, como la lectura de niveles de batería, mediciones analógicas, entradas y salidas digitales, y puertos de comunicación adicionales según las necesidades particulares.

En la fase final de codificación del equipo se aborda la implementación de tareas en tiempo real y de interrupciones para los periféricos. Estas tareas se destinan a las distintas actividades que el equipo debe llevar a cabo de manera periódica y asincrónica. Los intervalos de ejecución de estas tareas pueden ser ajustados por el usuario final, permitiendo una adaptación óptima del dispositivo a la aplicación específica en la que será utilizado.

Después de completar esta etapa se procede a implementar el broker basado en Mosquitto en un

servidor. Este broker opera de manera continua y tiene asignada una dirección IP estática para que pueda ser localizado fácilmente por los dispositivos conectados. Es importante destacar que la única entidad con una IP estática es el broker, ya que una de las ventajas de MQTT es su capacidad para descubrir dispositivos conectados sin necesidad de una IP estática para determinar la dirección de envío del mensaje.

Posteriormente, se requiere la implementación de una base de datos MySQL. Esta base de datos recibe los datos provenientes de todos los dispositivos conectados a través del equipo. Cabe señalar que el equipo tiene la capacidad de conectarse a cualquier servidor MQTT configurado por el usuario. Sin embargo, para propósitos de las pruebas, se optó por implementar esta base de datos.

Como etapa final del desarrollo, se crea una plataforma web que permite la visualización amigable de los datos enviados por los diferentes dispositivos. Además, esta plataforma permite realizar modificaciones en los registros Modbus de los dispositivos conectados. Esta interfaz web brinda al usuario final una manera intuitiva y conveniente de interactuar con el dispositivo y acceder a los datos de manera eficiente.

En resumen, el desarrollo del dispositivo involucró una serie de etapas clave, desde la selección del microcontrolador y el RTOS, pasando por la creación de la librería Modbus Master, la implementación de la comunicación MQTT y la configuración del broker y la base de datos, hasta culminar en la creación de una plataforma web para la visualización y modificación de los datos. Cada una de estas etapas se aborda de manera rigurosa, garantizando un dispositivo versátil y eficiente que puede adaptarse a diversas aplicaciones industriales con facilidad.

3. RESULTADOS

En la fase de pruebas se validan todas las funcionalidades desarrolladas en el dispositivo. Estas pruebas se realizaron con el objetivo de asegurar que el equipo cumpla con los requerimientos de diseño.

En la etapa de validación, se prueba una conexión de un PLC M221 de la empresa Schneider Electric a través de la interfaz Modbus RS232, con un programa que utiliza registros enteros y flotantes, además de registros de lectura y escritura. Esta configuración permite evaluar en detalle la

interoperabilidad y la correcta comunicación entre el dispositivo desarrollado y el PLC. Al igual que se realizan pruebas de transferencia de datos, la lectura y escritura de registros Modbus, así como la respuesta del dispositivo ante comandos y solicitudes enviadas desde y hacia el PLC. El montaje de la prueba de laboratorio se puede observar en la Fig. 1. La configuración Modbus del PLC usada para esta prueba se muestra en la Fig. 2.

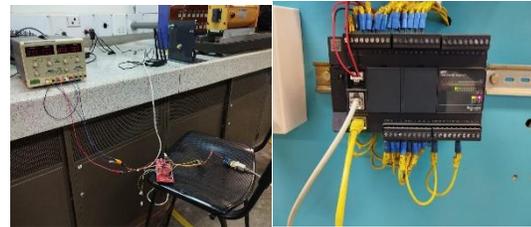


Fig. 1. Montaje de la prueba del equipo conectado a PLC M221.

Fuente: elaboración propia.



Fig. 2. Configuración serial y Modbus del PLC M221.

Fuente: elaboración propia.

Además de las pruebas de comunicación con el PLC, se evalúan las funcionalidades de almacenamiento de datos y cola para envíos en situaciones de comunicación intermitente. Se verifica la capacidad del dispositivo para guardar datos en la memoria SD en el formato de texto seleccionado y estos son retransmitidos en el mismo orden después de restablecer la conexión a Internet. Es importante destacar que estos datos fueron enviados a través de comunicación GPRS, utilizando un modem RUT240 conectado a la placa de desarrollo mediante Ethernet.

A continuación, se resumen algunas pruebas realizadas al equipo.

3.1 Pruebas de comunicación con broker MQTT

Se realizan pruebas de comunicación utilizando tanto una conexión por cable mediante Ethernet como una conexión inalámbrica a través de GPRS

mediante un módem. Ambos métodos establecen la conexión de manera satisfactoria. No obstante, se observa que la conexión Ethernet muestra una mayor velocidad en el envío y recepción de datos en comparación con la comunicación inalámbrica, lo cual era de esperarse. Aunque la conexión Ethernet es más rápida, es importante destacar que la comunicación inalámbrica es la más adecuada para esta aplicación, ya que permite a los dispositivos conectados operar sin depender de conexiones físicas por cable, lo que facilita la comunicación en entornos industriales.

Las pruebas se llevaron a cabo utilizando el puerto 1883 para el protocolo MQTT. Los datos fueron enviados con el tópico "PLC/001/W" y se realiza la suscripción al tópico "PLC/001/R", a través del cual se recibieron los datos utilizados para configurar los registros del PLC.

La Fig. 3 ilustra la conexión al broker MQTT, la suscripción a los tópicos y el envío de datos encriptados a través del tópico mencionado anteriormente. Esta actividad se visualiza mediante un monitor serial conectado al microcontrolador.

```

CONNACK:
Connection Success
Client subscribed on PLC/001/LED.
Client subscribed on PLC/001/W.
Publishes the following message
Topic: PLC/001/R

```

Fig. 3. Conexión al Broker, suscripción y publicación de mensajes.

Fuente: elaboración propia.

3.2 Pruebas de comunicación con script Python

Otras pruebas de lectura y escritura de datos del PLC se realizan utilizando una aplicación Python que se conecta al broker a través de Internet. Esta aplicación permite llevar un registro de los datos, realizar lecturas suscribiéndose al tópico donde el equipo envía la información y efectuar escrituras en los registros del PLC mediante el tópico al que el equipo está suscrito.

Una de las pruebas más significativas de este script consiste en encender y apagar un LED del PLC en intervalos de 10 minutos, lo que activa un contador para registrar cuántas veces se enciende el LED. Los comandos enviados para esta finalidad a través del script Python se presentan en la Fig. 4.

```

a10000
result: 0 mid: 1
-----
DATA PUBLISHED
client: <puho_mqtt.client.Client object at 0x000002F7099E7850>
userdata: None
result: 1
-----
Printing my First UUID of version 1
86c4e0a0-3859-11e6-b936-70c4e1909d7
a10001
result: 0 mid: 1
-----
DATA PUBLISHED
client: <puho_mqtt.client.Client object at 0x000002F7099E7850>
userdata: None
result: 1
-----
Printing my First UUID of version 1
607c2501-385a-11e6-a2f3-70c4e1909d7
a10000
result: 0 mid: 1
-----
DATA PUBLISHED
client: <puho_mqtt.client.Client object at 0x000002F7099E7850>
userdata: None
result: 1
-----

```

Fig. 4. Encendido y apagado de LED desde un PC conectado al broker.

Fuente: elaboración propia.

3.3 Pruebas de comunicación con plataforma online

La plataforma web desarrollada para la visualización de datos y la modificación de registros Modbus también se ha sometido a una serie de pruebas. Se verifica la interfaz de usuario, la precisión en la presentación de los datos y la eficiencia en la realización de modificaciones en tiempo real.

El dispositivo conectado se presenta en un mapa interactivo donde, al hacer clic en el icono correspondiente, se despliegan los datos actuales del equipo junto con su última información enviada. La ubicación del PLC en el mapa se muestra en la Fig. 5.



Fig. 5. Posición en mapa del PLC conectado.

Fuente: elaboración propia.

Al realizar un acercamiento detallado, es posible visualizar las gráficas que representan algunas de las variables o registros enviados por el PLC. En la Fig. 6 se muestra un ejemplo de uno de los canales analógicos, donde se ha conectado un sensor de distancia durante un período de un día.

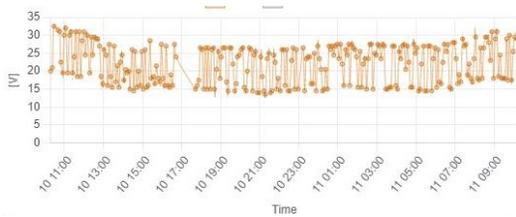


Fig. 6. Grafica de un día del sensor analógico conectado al PLC

Fuente: elaboración propia.

Para concluir las pruebas, se realiza el envío de datos a través de la plataforma web para modificar un registro específico. Dicho cambio permite controlar el encendido y apagado de un LED en el PLC, así como reiniciar los contadores del sistema.

Los resultados obtenidos en estas pruebas demuestran que el dispositivo ha superado exitosamente los criterios de rendimiento y funcionalidad establecidos. La interacción con el PLC de modelo M221 de Schneider Electric por medio de Modbus RS232 se realiza validando la interoperabilidad entre ambos sistemas. Las funcionalidades de almacenamiento de datos, cola para envíos y plataforma web también se desempeñaron según lo esperado, asegurando la integridad y disponibilidad de los datos en diversas condiciones.

4. CONCLUSIONES

En este trabajo se realiza la selección de componentes y herramientas en conformidad con las necesidades específicas de la aplicación. Como resultado, se logra implementar un dispositivo capaz de establecer comunicación bidireccional mediante el protocolo Modbus, facilitando la transferencia de datos hacia un broker MQTT. Este proceso se materializa a través de la interacción de scripts especializados y plataformas web que permiten la lectura y escritura de información.

El propósito subyacente de este enfoque radica en fomentar un avance significativo en la industria colombiana, direccionándola hacia las prerrogativas de la industria 4.0. Lo conseguido es un dispositivo de bajo costo que, crucialmente, no exige la alteración de la infraestructura preexistente. Esta innovadora solución no sólo responde a la demanda actual, sino que también allana el camino para una transición fluida y rentable hacia un entorno industrial más avanzado y tecnológicamente conectado.

En conjunto, el desarrollo y las pruebas realizadas confirman la robustez y funcionalidad del dispositivo en entornos de laboratorio. La combinación de comunicación Modbus, MQTT y la plataforma web proporciona una solución integral para la monitorización y control eficiente de dispositivos, allanando el camino para su implementación exitosa en diversas aplicaciones industriales que requieran una comunicación confiable y eficiente con dispositivos como el PLC M221 de Schneider Electric a través del protocolo Modbus RS232.

REFERENCIAS

- [1] E. Lalvay, Diseño y construcción de una interfaz hombre máquina para el monitoreo y supervisión del sobrellenado del tanque V4H en la empresa INTERQUIMEC., 2021.
- [2] W. Olarte, M. Botero y B. Cañon, «Importancia del mantenimiento industrial dentro de los procesos de producción,» *Scientia et Technica*, vol. 16, n° 44, pp. 354-356, 2010.
- [3] Modbus Inc. , «Modbus org.,» [En línea]. Available: <https://www.modbus.org>. [Último acceso: 17 Agosto 2023].
- [4] T. Wanasinghe, R. Gosine, L. James, G. Mann, O. De Silva y P. Warriar, «The internet of things in the oil and gas industry: a systematic review,» *IEEE Internet of Things Journal*, vol. 7, n° 9, pp. 8654-8673, 2020.
- [5] P. Salgado Díaz, Estudio y desarrollo de sensores de fibra óptica para detección de vibraciones en ductos ascendentes submarinos, Telecomunicacion, 2013.
- [6] Q. Bai, B. Jin, D. Wang, Y. Wang y X. Liu, «Compact Modbus TCP/IP protocol for data acquisition systems based on limited hardware resources,» *Journal of Instrumentation*, vol. 13, n° 4, p. T04004, 2018.
- [7] J. Song, . S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon y W. Pratt, «WirelessHART: Applying wireless technology in real-time industrial process control,» *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 377-386, 2008.
- [8] A. Jestratjew y A. Kwiciczen, «Performance of HTTP protocol in networked control systems,» *IEEE Transactions on Industrial Informatics*, vol. 9, n° 1, pp. 271-276, 2012.
- [9] N. Defossé , R. López, M. Gómez, P. Konstantinoff, S. Whaler, L. Castro y G. Harris, «Implementación de middleware publicador/subscriptor para aplicaciones web

- de monitoreo,» de *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*, 2017.
- [10] D. Torres, A. Soto, D. Coral, P. Parra, H. Muñoz, J. Rojas y N. Córdoba, «Sistema basado en internet de las cosas (IoT) para la monitorización en tiempo real de variables de temperatura y humedad en un equipo de refrigeración de área de farmacia de un hospital de cuarto nivel,» de *Encuentro Internacional de Educación en Ingeniería-EIEI-ACOFI 2019*, 2019, pp. 1-9.
- [11] C. Silva y F. Silva, «An IoT gateway for Modbus and MQTT integration,» de *2019 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, IEEE, 2019, pp. 1-3.
- [12] M. Tyrovolas y T. Hajnal, «Inter-communication between Programmable Logic Controllers using IoT technologies: A Modbus RTU/MQTT Approach,» *arXiv preprint arXiv:2102.05988*, 2021.
- [13] M. Ali, A. Miry y T. Salman, «IoT based water tank level control system using PLC,» de *2020 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, 2020, pp. 7-12.
- [14] S. Palacio, Diseño de un sistema de monitorización y control energético basado en IoT, Universitat Politècnica de Catalunya, 2021.
- [15] MQTT org, «MQTT org,» [En línea]. Available: <https://mqtt.org/> . [Último acceso: 14 agosto 2023].
- [16] Modbus Inc., «Application protocol V1,» [En línea]. Available: https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf . [Último acceso: 15 Agosto 2023].