

Sistema de Monitoreo y Control con tecnología BT e IoT, utilizando la placa de desarrollo ESP32

Fernando J. Carranza, Francisco M. Cortez Nachar, Roberto M. Murdocca, Sergio F. Hernandez Velazquez
Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: {fernandocarranza2018, francorteznachar, mmurdocc, sergio.sfhv}@gmail.com

II. DESARROLLO

Resumen - Este trabajo contribuye al conocimiento existente en el área de la IoT al presentar una arquitectura flexible y escalable para la construcción de sistemas de monitoreo y control remoto. Se buscó implementar un sistema de visualización de datos analógicos y digitales, monitoreo de variables físicas específicas y actuadores las cuales deberían poder interactuar con cualquier dispositivo portátil con conexión a Internet utilizando una interfaz gráfica. Esta última, diseñada bajo los principios de usabilidad, presenta una estructura intuitiva y elementos visuales claros.

Palabras clave: IoT, Wi-Fi, Node-Red, MQTT, Bluetooth.

I. INTRODUCCIÓN

La creciente demanda de soluciones IoT ha generado un gran interés en el desarrollo de sistemas de monitoreo y control remoto flexibles y escalables. Este trabajo contribuye a este campo al proponer una arquitectura modular y abierta que puede ser adoptada y extendida por otros investigadores. La interfaz gráfica desarrollada, basada en principios de usabilidad, establece un punto de referencia para el diseño de interfaces en sistemas IoT, facilitando la interacción entre el usuario y el sistema. Los resultados obtenidos en este proyecto abren nuevas líneas de investigación dentro del Departamento de Electrónica, como la integración de algoritmos de aprendizaje automático para la toma de decisiones autónomas y la optimización de recursos energéticos.

El sistema utiliza un microcontrolador ESP32 [1], acompañado de diversos componentes electrónicos, como un sensor de temperatura y humedad, una fotoresistencia, pulsadores, un potenciómetro, un módulo de relés, y elementos para el acondicionamiento de señales.

Para la comunicación mediante Bluetooth, se diseñó una interfaz gráfica en la aplicación móvil Bluetooth Electronics [2], que permite al usuario monitorear datos y controlar dispositivos de forma local. En cuanto a la implementación para IoT, se emplea el protocolo MQTT, utilizando una interfaz gráfica desarrollada en Node-RED [3] para la visualización y gestión de los datos proporcionados por el microcontrolador.

El sistema completo se presenta en la Fig. 1, mediante un diagrama de bloques distribuido de la siguiente manera:

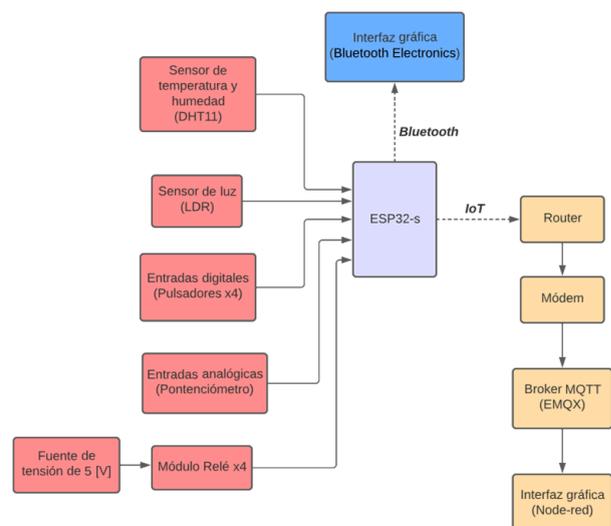


Fig. 1. Diagrama en bloques del sistema

Se tuvieron en cuenta las siguientes restricciones de diseño para asegurar el correcto funcionamiento y eficiencia del sistema:

- **Conectividad dual:** El sistema debe ser capaz de manejar simultáneamente conexiones Bluetooth y Wi-Fi, sin interferencias.
- **Tamaño compacto:** Se requiere un diseño que permita integrar todos los componentes en un espacio reducido, para facilitar su implementación en entornos pequeños.
- **Compatibilidad con múltiples dispositivos:** El sistema debe ser compatible con una variedad de dispositivos (smartphones, tablets, y computadoras) para asegurar una amplia accesibilidad y usabilidad.

III. DISEÑO DEL SISTEMA

A. MICROCONTROLADOR

El sistema integra un microcontrolador ESP32 (Fig.2) fabricado por Expressif Systems [1][4], el cual proporciona capacidades de comunicación Bluetooth y Wi-Fi, lo que permite la implementación simultánea de tecnologías de comunicación local y remota.

A continuación se detallan las características principales del microcontrolador:

- Integra un procesador Tensilica Xtensa 32 bits LX6, capaz de operar a velocidades de hasta 240 MHz.
- Memoria Flash de 4MB.
- Memoria RAM de 520kB.
- Wi-Fi compatible con los estándares 802.11b/g/n/e/i con velocidades de hasta 150 Mbit/s en la banda de 2.4 GHz.
- Soporta Bluetooth V4.2 en sus versiones VR/EDR (Bluetooth clásico) y Bluetooth Low Energy (BLE).
- 34 pines programables
 - 18 canales ADC de 12 bits.
 - 2 canales DAC de 8 bits.

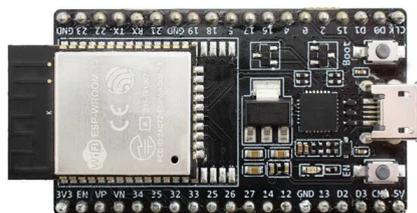


Fig. 2. ESP32.

- 16 canales PWM.
- 3 interfaces UART.
- 3 interfaces SPI.
- 2 interfaces I2C.

- Diferentes modos para gestionar el consumo de energía.

El ESP32 actúa como núcleo de procesamiento, coordinando la adquisición de datos de los sensores y el control de los actuadores.

B. SENSORES

Se detallan a continuación los diversos sensores implementados en el sistema de monitoreo:

- **Sensor de temperatura y humedad:** El sensor DHT11 [5] mide temperatura y humedad relativa. Este sensor utiliza un termistor y un sensor capacitivo de humedad para proporcionar datos digitales mediante una señal de un solo cable. Sus principales características son:
 - Tensión de operación: 3.3V-5.5V.
 - Corriente de operación: 0.3 mA (medición) y 60 μ A (en reposo).
 - Consumo de potencia máximo de 2.5 mW.
 - Rango de medición de temperatura: 0°C a 50°C \pm 2°C.
 - Rango de medición de humedad: 20% a 90% de humedad relativa (HR) con una precisión de \pm 5% HR.
 - Frecuencia de muestreo: 1 Hz (una vez por segundo).
 - Interfaz de salida: Serial, comunicación unidireccional.

La Fig.3 ilustra la configuración eléctrica empleada para interconectar y operar el sensor.

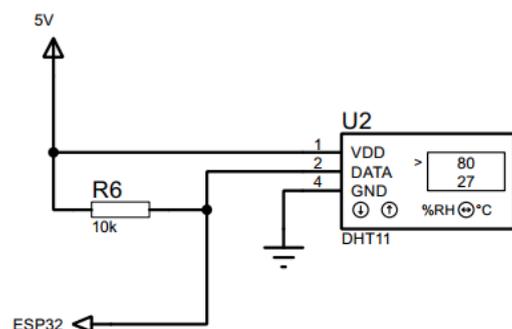


Fig.3. Circuito eléctrico DHT11.

- Fotorresistencia (LDR - Light Dependent Resistor)[6]: Es una resistencia eléctrica la cual varía su valor en función de la cantidad de luz que incide sobre su superficie. Su valor resistivo varía entre $1\text{ M}\Omega$ en la oscuridad y 100Ω con luz brillante. El consumo máximo es de 50 mW a 1W . En la Fig.4 se detalla el conexionado eléctrico, donde a mayor intensidad de luz, menor será la lectura de tensión en el pin del ESP32. Es decir, el ESP32 "verá" una tensión más alta cuando la fotorresistencia esté en la oscuridad y una tensión más baja cuando esté expuesta a la luz.

C. ENTRADAS ANALÓGICAS Y DIGITALES

En esta sección se detallan las funcionalidades de entrada y salida digital utilizadas para el sistema.

- Potenciómetro: varía de 0Ω a $10\text{ k}\Omega$ y se conecta a una entrada analógica del microcontrolador En la Fig.5, se detalla el conexionado eléctrico.
- Pulsadores: son del tipo de conexión

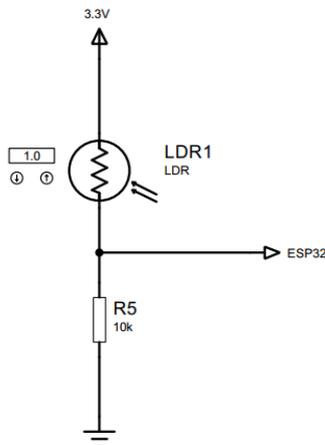


Fig.4. Circuito eléctrico del LDR.

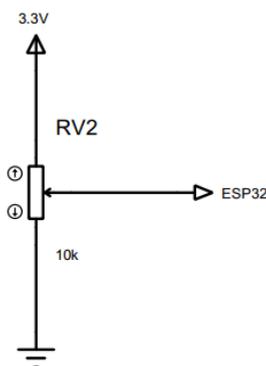


Fig. 5. Circuito eléctrico del potenciómetro.

normalmente abiertos (NA) y se encuentran configurados en la configuración pull-down, lo que garantiza que los pines digitales se mantengan en un estado bajo hasta que se active el pulsador. En la Fig.6, se detalla el conexionado eléctrico.

D. ACTUADORES

El control remoto del sistema se llevará a cabo mediante un 4 Relay Module [7] (Fig.7). Este módulo permite el control de hasta cuatro cargas eléctricas de manera independiente a través de señales TTL (Transistor-Transistor Logic) de 3.3 V a 5 V . Sus principales características son:

- 4 relés de 1 polo y 2 posiciones.
- Tensión de la bobina de 5 V .
- El circuito de control debe proveer una corriente de 15 mA a 20 mA .

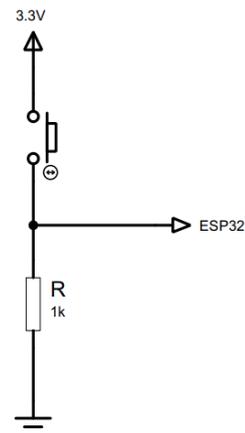


Fig. 6. Circuito eléctrico Pulsadores.

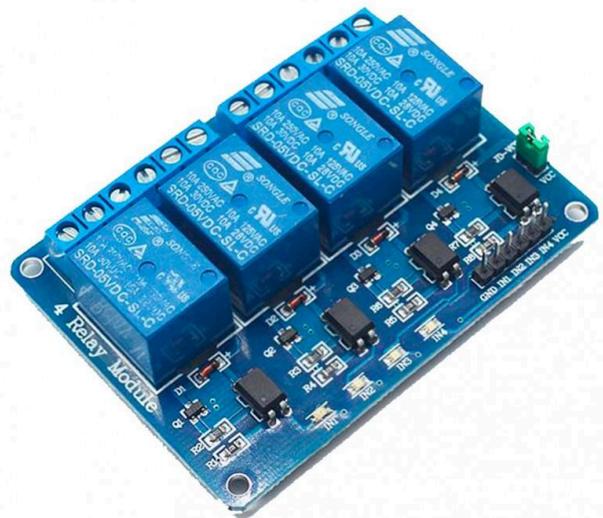


Fig. 7. 4 Relay Module.

La Fig.8 muestra el conexionado eléctrico.

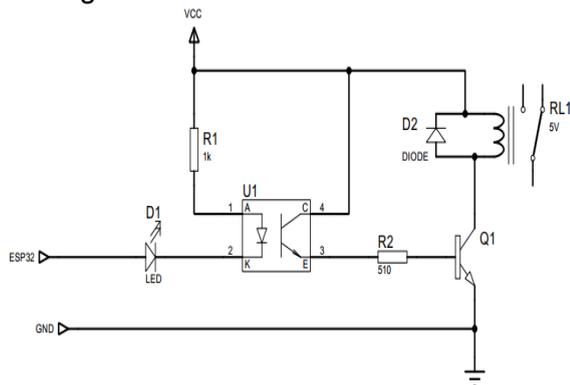


Fig.8. Circuito eléctrico de relé.

IV. INTERFAZ DE MONITOREO Y CONTROL

A. MQTT

El MQTT (Message Queuing Telemetry Transport) [8], es un protocolo de mensajería altamente eficiente y versátil, que ha demostrado ser una herramienta fundamental para el desarrollo de aplicaciones IoT y otras soluciones basadas en la comunicación máquina a máquina. Su arquitectura simple y su capacidad para manejar grandes volúmenes de datos lo convierten en una opción atractiva para investigadores y desarrolladores en diversos campos.

Emplea una arquitectura de publicación-suscripción, lo que permite que los dispositivos (clientes) envíen mensajes (publicaciones) a temas específicos. Otros dispositivos, al suscribirse a estos temas, reciben de manera eficiente los mensajes relevantes. Esta flexibilidad facilita la creación de sistemas escalables y desacoplados, donde los productores y consumidores de datos no necesitan conocerse entre sí.

B. NODE RED

Node-RED [9] es una herramienta de programación visual basada en flujo, desarrollada por IBM y ahora parte de la Fundación OpenJS. Utiliza un modelo de programación de nodos, donde cada nodo realiza una tarea específica y los datos fluyen entre ellos, facilitando la comprensión y manipulación de procesos complejos sin necesidad de conocer detalles del código.

El entorno de Node-RED se ejecuta sobre Node.js y permite a los usuarios crear

aplicaciones a través de un editor Web, conectando nodos en un espacio de trabajo.

C. CONECTIVIDAD

El sistema implementa un esquema de conectividad dual que permite la interacción local y remota mediante Bluetooth y Wi-Fi, optimizando el monitoreo y control de los dispositivos en tiempo real, como se muestra en la Fig.9.

El ESP32 puede conectarse a redes Wi-Fi 802.11b/g/n/e/i, facilitando así la comunicación remota a través de Internet. En este modo, los datos recopilados por los sensores del sistema se envían a la nube mediante el protocolo MQTT, con la ayuda del broker, que actúa como servidor para la transmisión de mensajes entre el ESP32 y el sistema de control remoto.

Para una gestión local y autónoma del sistema de monitoreo y control, se emplea Bluetooth. Esta tecnología inalámbrica permite la comunicación directa entre el ESP32 y un dispositivo móvil, sin requerir una conexión a Internet, brindando movilidad y flexibilidad al usuario.

D. PROGRAMACIÓN

La programación del sistema se dividió en dos partes principales: el desarrollo de la interfaz local mediante Bluetooth Electronics y la implementación remota utilizando Node-RED a través de MQTT. Ambas interfaces permiten al usuario monitorear los datos de los sensores y controlar los actuadores en tiempo real, proporcionando diferentes niveles de interacción, ya sea local o remota.

Para la comunicación local, se diseñó una interfaz gráfica en la aplicación móvil Bluetooth Electronics [1], que permite al usuario visualizar datos obtenidos de los sensores de temperatura, humedad, y luminosidad, además de interactuar con las salidas digitales y entradas analógicas del sistema.

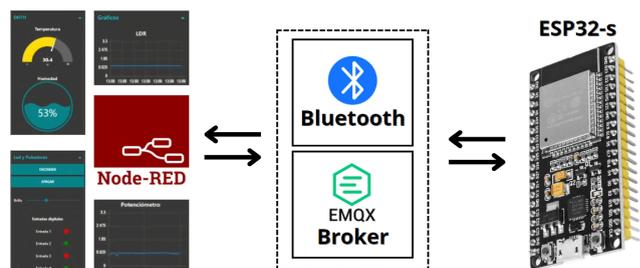


Fig.9. Esquema de conectividad.

La programación del ESP32 se realizó utilizando Arduino IDE [10], integrando librerías específicas para la comunicación Bluetooth como BluetoothSerial, esp_bt_main y esp_bt_device.

Para la implementación remota del sistema y la comunicación con el broker EMQX [11], se utilizaron las librerías WiFi.h y PubSubClient.h en el Arduino IDE, permitiendo que el ESP32 se conecte a la red Wi-Fi y publique los datos de los sensores en el servidor MQTT. Estas librerías facilitan la comunicación eficiente del ESP32 con el broker, permitiendo la transmisión en tiempo real de los datos de los sensores a la plataforma Node-RED.

En la programación en Node-RED (Fig.10) se incluyeron los nodos de suscripción y publicación en el broker MQTT, así como la configuración de nodos para la interfaz gráfica que permiten monitorear los datos y controlar las salidas del sistema.

E. INTERFAZ LOCAL

Se implementa a través de la aplicación móvil Bluetooth Electronics [1], esto permitirá la conexión directa con el ESP32.

La interfaz gráfica (Fig.11), se explica a continuación:

- Control de los relés (R1, R2, R3, R4): permiten al usuario activar (botón verde) o desactivar (botón rojo) los cuatro relés del sistema. El indicador de color azul muestra cuando un relé está activo.
- Indicadores de temperatura y humedad: se muestra la temperatura (en rojo) y la humedad (en azul). Ambas se

representan tanto numéricamente como en formato gráfico (termómetro vertical para la temperatura y gráfico de nivel para la humedad).

- Control deslizante del brillo del LED: situado justo debajo del indicador de temperatura, permite al usuario ajustar la intensidad de luz del LED.
- Indicadores LDR y POTENC.: se observa una gráfica que muestra la variación de la tensión registrada por el LDR y el potenciómetro en tiempo real. Además, se facilitan las lecturas numéricas actuales de ambos sensores.
- Indicadores P1, P2, P3 y P4: muestran el estado de las entradas digitales, representados con color naranja cuando la entrada está activa.

F. INTERFAZ REMOTA

Desarrollada en Node-RED, proporciona una visualización intuitiva del panel de monitoreo y control del sistema (Fig.12).

A continuación se detalla la estructura de la mismas:

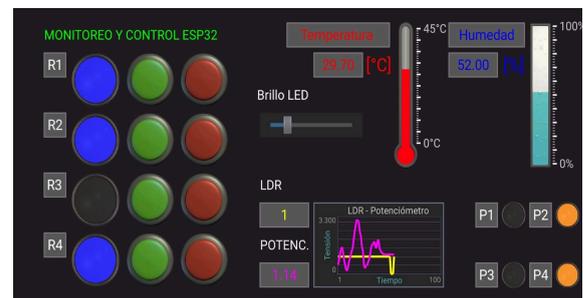


Fig.11. Interfaz local.

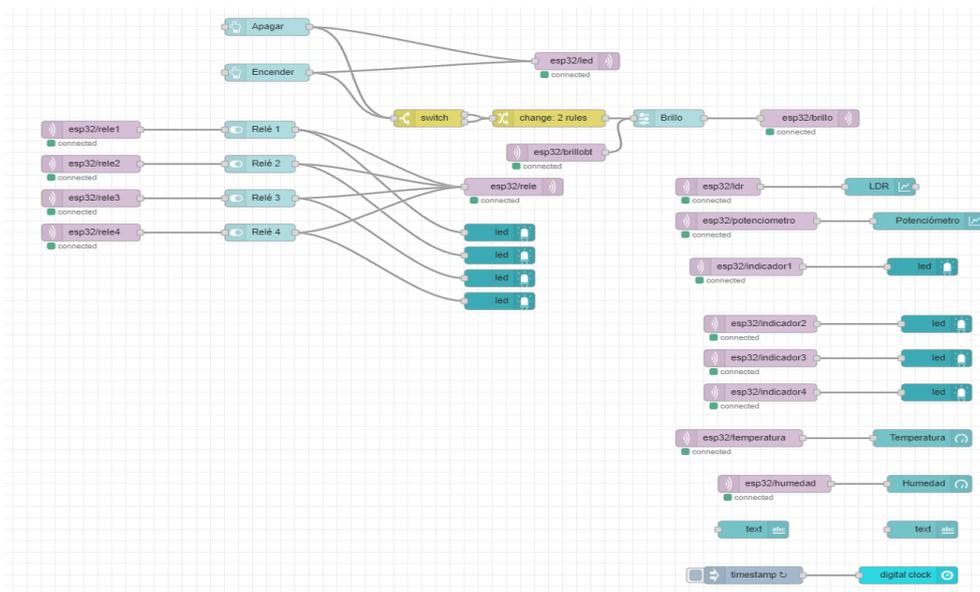


Fig.10 Programación de Node-RED.

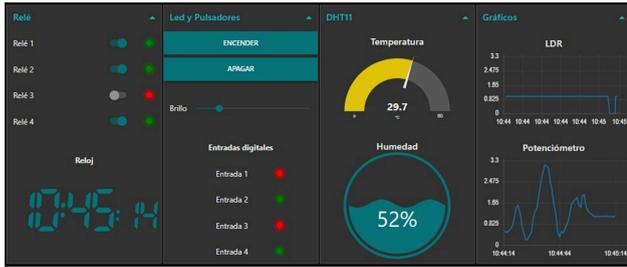


Fig.12. Interfaz remota.

- Panel de relés: Control mediante interruptores, cada uno con su indicador on/off.
- Panel de LED y Pulsadores: Incluye un botón para encender y otro para apagar un LED, junto con un control deslizante que ajusta el nivel de brillo. También se muestran los estados de cuatro entradas digitales, representadas por indicadores luminosos.
- Panel de DHT11: Presenta un medidor analógico para la temperatura en grados Celsius y un indicador circular para la humedad relativa.
- Panel de Gráficos: Muestra los datos en tiempo real de la intensidad de luz captada por el sensor LDR y la variación de la resistencia del potenciómetro.

V. RESULTADOS

El prototipo físico de diseño puede observarse en la Fig.13. Para el mismo se lograron los siguientes apartados:

Se logró implementar y operar de manera efectiva el sistema utilizando Wi-Fi y Bluetooth de manera simultánea. Durante las pruebas, identificamos una oportunidad para optimizar el rendimiento del sistema al observar una ligera latencia en la transmisión de datos entre las diferentes plataformas. Este hallazgo nos permitirá investigar a fondo factores como la calidad de la señal, la distancia a los puntos de acceso, la congestión de red y las

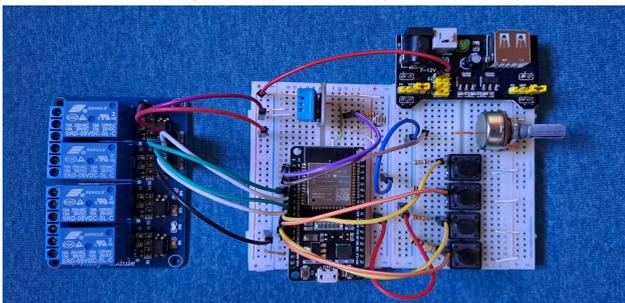


Fig. 13. Prototipo del sistema.

características del hardware utilizado para implementar soluciones que reduzcan este inconveniente y mejoren la experiencia del usuario.

Estos resultados destacan la funcionalidad básica y la interconexión efectiva lograda, mostrando el potencial del sistema en entornos donde se requiere comunicación dual, a pesar de algunas limitaciones observadas durante la fase de pruebas.

En la Fig.14, se proporciona un código QR que redirige a un video en YouTube. Este video permite observar en detalle el funcionamiento del prototipo en tiempo real, mostrando cómo el sistema interactúa a través de sus diferentes componentes y tecnologías de comunicación, como WiFi y Bluetooth.



Fig.14. QR Video funcionamiento.

REFERENCIAS

- [1] Espressif Systems (Shanghai) Co., ESP32 Series Datasheet, Version 4.5, February, 2024.
- [2] Keelsoft Electronics. Bluetooth Electronics. Obtenido de <https://www.keuwl.com/electronics.html> (Accedido 03/2023)
- [3] Node-RED. Programación de bajo código para aplicaciones basadas en eventos. Obtenido de <https://nodered.org/> (Accedido 03/2023)
- [4] Espressif Systems (Shanghai) Co., ESP32 Technical Reference Manual, Version 5.1, April, 2024.
- [5] Sensor DHT11. Product documentation. Obtenido de <https://mkelectronica.com/producto/sensor-temperatura-humedad/> (Accedido 03/2023)
- [6] MecatrónicaLATAM. Resistencia LDR. Obtenido de <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/ldr/> (Accedido 03/2023)
- [7] Profe Tolocka. Modulo de Relés. Obtenido de <https://www.profetolocka.com.ar/2015/05/09/modulo-de-4-rel-es-para-arduino/> (Accedido 03/2023)
- [8] Amazon Web Service, ¿Que es MQTT?, 2023, Disponible en <https://aws.amazon.com/es/what-is/mqtt/> (Accedido 03/2024)
- [9] Node-RED. Sobre Node-RED. Obtenido de <https://nodered.org/about/> (Accedido 03/2023)
- [10] Arduino IDE. Software. Obtenido de <https://www.arduino.cc/en/software> (Accedido 03/2024)
- [11] .EMQX. Disponible en <https://www.emqx.com/en> (Accedido 03/2023)