Redes de Telecomunicaciones
Trabajo 1: Control y monitoreo de temperatura de productos con sistema de acceso a bodega

Franklin Mauricio Gómez López, Juan Diego Tenesaca

Universidad de Cuenca, Facultad de Ingeniería,
Av. 12 de Abril y Agustín Cueva, EC010112, Cuenca, ECU
franklin.gomez@ucuenca.edu.ec, juan.tenesaca@ucuenca.edu.ec

Resumen En el presente documento se discutió la importancia del control y la seguridad en el acceso, almacenamiento y gestión de productos en distintos tipos de establecimientos comerciales. Se mencionó la preocupación de los propietarios y administradores de almacenes y negocios en cuanto a la prevención de robos y daños a los productos almacenados, y la crítica cuestión del control de la temperatura en la industria alimentaria para garantizar la calidad y seguridad de los productos.
Se abordó también la automatización del control de nivel de stock de un producto, y cómo esta solución permite una mejor gestión de los recursos y una reducción de costos en personal. Además, se destacó la importancia de contar con una visualización remota en tiempo real del nivel de los productos para tomar decisiones informadas y oportunas sobre la reordenación de los mismos.
Se presentaron dos soluciones para el monitoreo y control remoto de variables en sistemas de automatización: usando elementos como son sensores de temperatura (DHT11), ultrasonido (HC-SR04), lector de tarjetas RFID (PN532), reles para controlar sistemas de alto voltaje, modulo ESP32 para unir todo estos sensores y publicar sus datos en las plataformas de Node-Red como Blynk, que permite a los usuarios crear aplicaciones móviles personalizadas para monitorear y controlar dispositivos conectados a Internet de manre remota y localmente.

Keywords: IoT · Node-Red · ESP32 · monitoreo · DHT11 · automático · Blynk

1. Introducción

El control de acceso a bodegas que almacenan productos es un tema crucial para los propietarios y administradores de almacenes y tiendas. Es esencial mantener la integridad y la calidad de los productos almacenados, y esto solo puede lograrse con un control adecuado del acceso a las áreas de almacenamiento. La seguridad es un factor importante en este proceso, ya que se deben prevenir robos, daños y otros incidentes.

En la industria de alimentos y bebidas, el control de temperatura es crucial para garantizar la calidad y la seguridad de los productos. Los alimentos perecederos, como la carne, el pescado, los productos lácteos y los productos frescos, deben mantenerse a una temperatura adecuada para evitar el crecimiento de bacterias y otros microorganismos que pueden causar enfermedades transmitidas por los alimentos.

El control automatizado del nivel de stock de un producto permite a las empresas tener un control más preciso y eficiente de su inventario. Esto se traduce en una mejor gestión de los recursos y una reducción de costos en personal. Además, al contar con una visualización remota en tiempo real del nivel de los productos, se pueden tomar decisiones informadas y oportunas sobre la reordenación de los productos para asegurar que los niveles estén siempre en línea con la demanda.

Además, el uso de la plataforma de Nod-red permite a los usuarios visualizar y controlar los dispositivos conectados dentro de la red local. La plataforma de Nod-red ofrece una amplia variedad de opciones de visualización y control, como el control rotativo de la plataforma de los productos y el monitoreo de la temperatura y humedad de la bodega, lo que proporciona una solución integral para el control y monitoreo remoto de las variables críticas de la industria de alimentos y bebidas.

La plataforma Blynk ofrece una solución para el control remoto y la visualización de variables de los sistemas de automatización. Los usuarios pueden crear aplicaciones móviles personalizadas para monitorear y controlar dispositivos conectados a Internet, lo que facilita la supervisión y el control en tiempo real de los sistemas de seguridad, temperatura y nivel de stock de los productos.

Este informe está dividido en varias secciones. En la primera sección se encuentra la Introducción, que proporciona una visión general del tema a tratar. La segunda sección, es la del Marco Teórico, en donde se exponen los conceptos y se describe los programas, sensores y módulos empleados en el informe.
En la tercera sección, se encuentra el Desarrollo, sección en la que se presenta la metodología, el circuito y los resultados obtenidos en la implementación de esta práctica. Las conclusiones obtenidas se presentan en la siguiente sección. Finalmente, el informe se cierra con la sección de referencias, donde se incluyen todas las fuentes consultadas.

2. Marco Teórico

2.1. Node-Red

Node-RED es un entorno de programación visual basado en Node.js que permite crear flujos de trabajo automatizados, integrando diferentes servicios y dispositivos IoT en una misma plataforma de manera sencilla y rápida. Con una interfaz gráfica intuitiva, el usuario puede arrastrar y soltar diferentes nodos y configurarlos para realizar diferentes tareas. Node-RED es altamente personalizable y puede ser utilizado para crear soluciones IoT de diferentes complejidades. Además, cuenta con una comunidad activa que desarrolla y comparte diferentes nodos y flujos para su uso en proyectos específicos. [1]

2.2. Protocolo MQTT

MQTT (Message Queue Telemetry Transport) es un protocolo de comunicación ligero y de bajo ancho de banda diseñado para dispositivos con recursos limitados, como sensores y actuadores, para enviar y recibir mensajes en tiempo real en redes de alta latencia y ancho de banda limitado. MQTT utiliza un modelo de publicación/suscripción

en el que los dispositivos se suscriben a un tema (topic) y reciben mensajes publicados por otros dispositivos en ese tema. Fue desarrollado por Andy Stanford-Clark de IBM y Arlen Nipper de Arcom (ahora Cirrus Link) en 1999 [2].

2.3. ESP32

El ESP32 es un microcontrolador de bajo costo y alto rendimiento que se utiliza en aplicaciones de Internet de las cosas (IoT). Tiene un procesador de doble núcleo, conectividad Wi-Fi y Bluetooth, así como una variedad de periféricos como puertos GPIO, ADC y DAC. El ESP32 se programa en lenguaje C++ y se puede programar a través de una variedad de entornos de desarrollo integrado (IDE). Es popular en la industria IoT debido a su bajo costo, su conectividad inalámbrica y su capacidad para manejar múltiples tareas. [3]

2.4. PN532

PN532 es un circuito integrado que funciona como lector/escritor de tarjetas RFID y NFC. Permite la comunicación inalámbrica entre dispositivos y tarjetas mediante campos electromagnéticos. Este chip cuenta con diversas funcionalidades y se puede utilizar en una amplia variedad de aplicaciones, como el control de acceso, el pago sin contacto y la identificación de objetos.

2.5. Sensor HC-SR04

El HC-SR04 es un sensor de distancia que utiliza ondas ultrasónicas para medir la distancia entre el sensor y un objeto. El sensor emite una señal de ultrasonido a través del transmisor (triger) y luego recibe la señal reflejada a través del receptor (echo). Al medir el tiempo que tarda la señal en viajar desde el sensor al objeto y regresar al sensor, se puede calcular la distancia. La precisión del sensor puede ser afectada por factores como la temperatura y la dirección del objeto en relación al sensor. Sin embargo, es una herramienta útil para aplicaciones de medición de distancia en proyectos de electrónica y robótica. [4]

2.6. Sensor DHT11

La DHT11 es un sensor de temperatura y humedad que utiliza un termistor para medir la temperatura y un sensor de humedad capacitivo para medir la humedad relativa del aire. Este sensor es económico y fácil de usar, lo que lo hace popular para proyectos de electrónica y automatización del hogar. La DHT11 proporciona datos digitales de temperatura y humedad a través de un solo pin, lo que simplifica su conexión a microcontroladores. [5]

2.7. Blynk

Blynk es una plataforma de Internet de las cosas (IoT) que proporciona una aplicación móvil y una biblioteca de software para permitir a los usuarios controlar dispositivos conectados a Internet de forma remota. La plataforma también incluye una herramienta de creación de aplicaciones que permite a los usuarios crear interfaces de usuario personalizadas para sus proyectos de IoT sin necesidad de conocimientos de programación avanzados. La plataforma es compatible con una amplia gama de placas de desarrollo, microcontroladores y protocolos de comunicación. [6]

3. Desarrollo

En términos simplificados, esta práctica consta de tres partes:

Primero, el control y la energización de los sensores se realiza mediante un ESP32. Segundo, la información recopilada por los sensores es enviada al servidor MQTT Mosquitto a través de mensajes del protocolo MQTT.
Finalmente, los datos de las mediciones y parámetros de los sensores son visualizados a través de un entorno gráfico. Para este fin, se utilizó Node-RED como entorno local y la plataforma Blynk como entorno público.

La figura 6 es un diagrama que muestra claramente la conexión de los sensores y cómo se utilizan en el sistema. Este diagrama es una herramienta útil para visualizar cómo se relacionan los diferentes componentes del sistema y cómo la información fluye entre ellos.
Así mismo, en la figura 3 se presenta un diagrama que ilustra el proceso de publicación y suscripción MQTT que se lleva a cabo en esta práctica con los diferentes sensores.

Figura 1: Esquema del sistema de refrigeración, control de productos y control de acceso a una bodega IoT

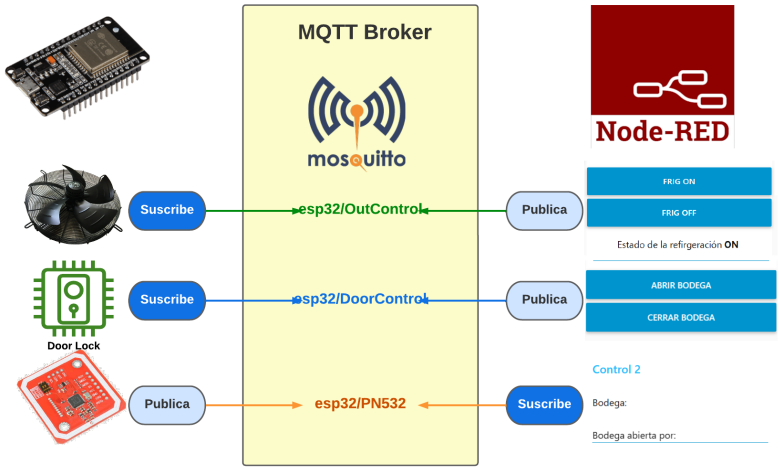

Figura 2: Proceso de publicación y suscripción MQTT entre el microcontrolador y el entorno de visualización

3.1. Sistema de refrigeración de productos

Metodología: Para garantizar que los productos se mantengan a la temperatura adecuada durante su almacenamiento y transporte, es esencial contar con sistemas de refrigeración confiables y efectivos. En este sentido, el uso de sensores de humedad y temperatura, como el DHT11, es fundamental para monitorear la temperatura y la humedad del ambiente en el que se encuentran los productos.

El ESP32, por su parte, es un microcontrolador de bajo costo y alta capacidad de procesamiento, que permite el monitoreo y control remoto de dispositivos electrónicos. Al utilizar el DHT11 y el ESP32 en conjunto, es posible controlar el sistema de refrigeración en tiempo real, de forma automatizada y precisa.

Cuando el sensor DHT11 detecta que la temperatura en la zona de almacenamiento ha aumentado por encima de un umbral establecido, el ESP32 activa un relé, que a su vez acciona el sistema de refrigeración. Esto permite mantener la temperatura en el rango adecuado y evitar la proliferación de microorganismos en los productos almacenados.

En resumen, la utilización de tecnologías como el DHT11 y el ESP32 en los sistemas de refrigeración garantiza la calidad y seguridad de los productos alimentarios, a la vez que reduce el riesgo de desperdicio de alimentos debido a una inadecuada conservación de los mismos.

Circuito implementado

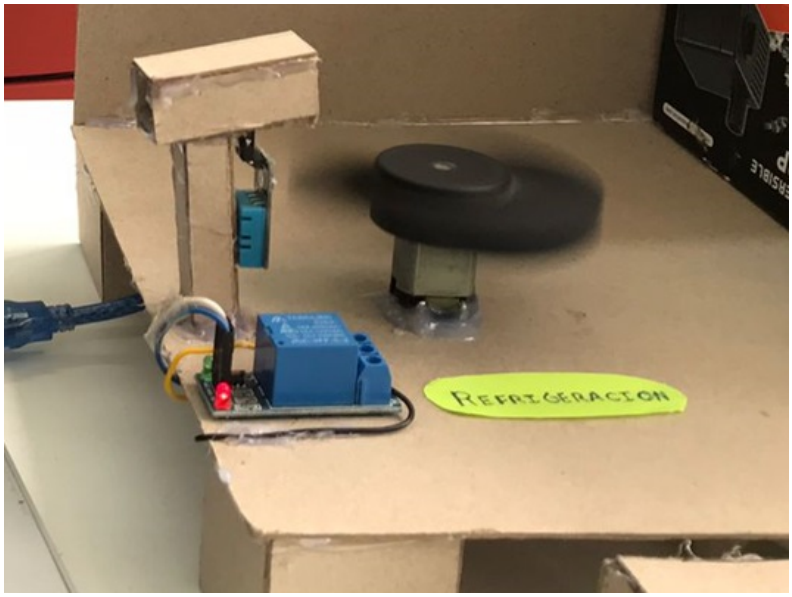

Figura 3: Circuito implemento para el control de refrigeración

3.2. Control de acceso a la bodega

Metodología: La tecnología de lectura de tarjetas como la PN532 se ha convertido en una solución común para el control de acceso en una amplia variedad de entornos. Esta tecnología permite a los usuarios autorizados acceder a áreas restringidas, como la bodega de productos, mediante la presentación de una tarjeta con un identificador único.

Al utilizar el módulo ESP32 en conjunto con el lector de tarjetas PN532, es posible crear un sistema de control de acceso seguro y confiable. El ESP32, al ser un microcontrolador de bajo costo y alta capacidad de procesamiento, es capaz de gestionar el proceso de autenticación y control de acceso en tiempo real.

Además, al utilizar la plataforma de NOD-RED y Blink, se puede tener una gestión de datos en tiempo real, lo que permite monitorear el acceso a la bodega de forma remota y en tiempo real. De esta manera, se puede visualizar quién ha ingresado a la bodega y cuándo, lo que proporciona un registro preciso y detallado de los movimientos en la zona de almacenamiento.

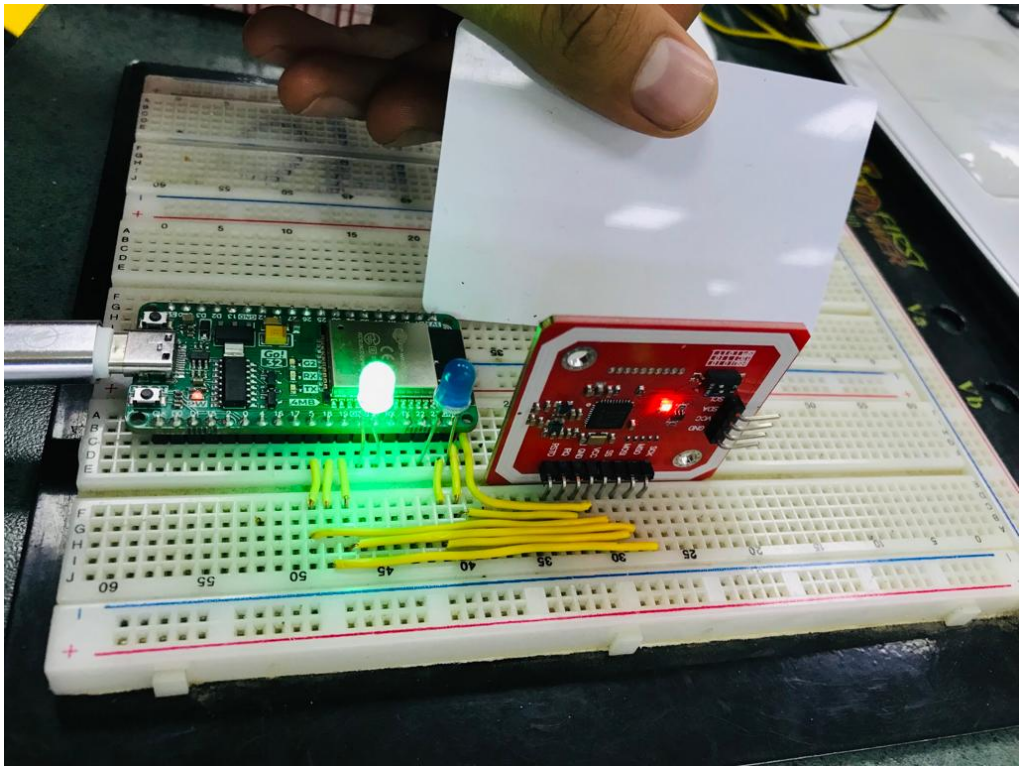

Figura 4: Esquema del sistema para control de acceso

Circuito Implementado Control del nivel de stock de un producto

Metodología El monitoreo y control de los niveles de stock de los productos es una tarea crítica para los negocios, especialmente para aquellos que manejan grandes cantidades de inventario. Sin embargo, tener personal dedicado a la supervisión de los niveles de stock puede ser costoso y poco práctico.

El uso de tecnologías como el sensor de ultrasonido HC-SR04 y el módulo ESP32 se han vuelto cada vez más populares en el control de stock automatizado. Este sistema de control consta de un sensor de ultrasonido que mide la distancia entre el sensor y el producto, y luego el módulo ESP32 procesa esa información para calcular la profundidad del producto en el contenedor.

A partir de esa información, se puede programar el sistema para notificar a los administradores del inventario cuando el nivel de un producto esté bajo, medio o lleno. Además, al utilizar plataformas de monitoreo y visualización remota como NOD-RED y Blink, se puede tener una visión completa y en tiempo real del estado de los niveles de stock.

Circuito implementado :

Figura 5: Esquema del sistema para control de stock del producto

3.3. Control de un mostrador rotativo

Metodología La exhibición adecuada de productos en un mostrador es esencial para atraer la atención de los clientes y garantizar una experiencia de compra satisfactoria. Sin embargo, a menudo puede ser difícil para los clientes alcanzar o acceder a ciertos productos que están lejos o fuera de su alcance.

Para abordar esta problemática, se ha desarrollado un circuito de control de un mostrador rotativo de un producto, que consta de un servo que controla la plataforma de manera presencial y remota. Con este sistema, se puede establecer un horario de rotación o rotar en cualquier momento, lo que permite a los clientes tener acceso fácil y cómodo a todos los productos en exhibición.

Además, el circuito de control también puede ayudar a mantener una temperatura uniforme en los productos, lo que es especialmente importante en la exhibición de alimentos o bebidas. Al utilizar el módulo ESP32, se pueden enviar notificaciones a plataformas de monitoreo y visualización remota como NOD-RED y Blink, lo que permite a los administradores del negocio supervisar el funcionamiento del mostrador en tiempo real y tomar decisiones informadas sobre la rotación de los productos y la temperatura en el mostrador.

Circuito implementado

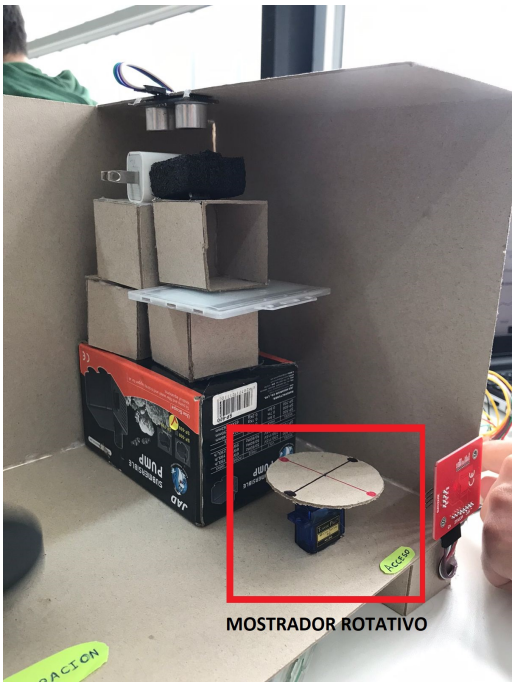

Figura 6: Esquema del sistema para control rotativo del producto

3.4. Servidor MQTT

Metodología: Como servidor MQTT se usó el servicio de Mosquitto *broker*, en Windows tanto su instalación como uso es muy sencillo, basta con descargar el instalador desde el sitio oficial `https://mosquitto.org/download/` y seguir todos los pasos.

Se puede ejecutar el servidor como si fuera un servicio de Windows, sin embargo de esta forma las peticiones anónimas están desactivadas, lo que impide que el ESP32 lo cual generó muchos inconvenientes en la realización de esta práctica, ya que no era posible la conexión del ESP32 con el *broker*, para solucionar este inconveniente, se modifico el archivo de configuración de mosquitto, agregando las líneas que se muestran en la figura 7.
Para poder modificar el archivo de configuración, se debe ejecutar como administrador el editor de texto empleado, para poder guardar las modificaciones.

Figura 7: Edición del archivo de configuración del *broker*

Con las modificaciones realizadas ya es posible conectar cualquier dispositivo externo con el servidor MQTT.

Resultados Una vez que se ha configurado correctamente el servidor mosquitto y que se tiene implementada la librería para que el ESP32 pueda emplear el protocolo MQTT, solo basta con asignar los nodos de Node-Red con los *topics* correspondientes y se

podrá leer los datos desde la interfaz de usaurio, como se muestra en la figura 8.

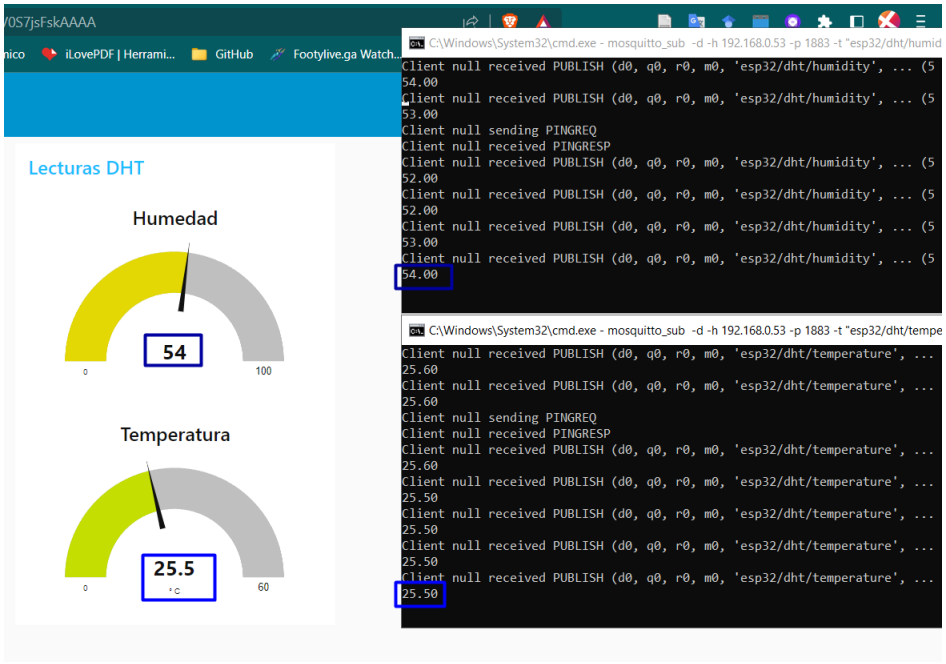

Figura 8: Conexión entre Node-Red y el servidor Mosquitto

3.5. MQTT con ESP32

Metodologia: Para usar el protocolo MQTT con ESP32 se empleó la librería *Async MQTT Client Library* de Marvin Roger, esta librería no se encuentra disponible en el administrador de bibliotecas de Arduino y puede ser descargada desde el repositorio de esta práctica.

Esta librería crea un objeto *AsyncMqttClient*, el cual permite manejar todos los parámetros necesarios para establecer conexión con un servidor MQTT, como la configuración del servidor y la suscripción y publicación en los *topics*, tal como se muestra a continuación.

```
1 AsyncMqttClient mqttClient;
2
```

```
// funcion para establecer conexion MQTT
void connectToMqtt() {
  Serial.println("Connecting to MQTT...");
  mqttClient.connect();
}

mqttClient.onConnect(onMqttConnect);
mqttClient.onDisconnect(onMqttDisconnect);
mqttClient.onPublish(onMqttPublish);
mqttClient.onSubscribe(onMqttSubscribe);
mqttClient.onUnsubscribe(onMqttUnsubscribe);
mqttClient.onMessage(onMqttMessage);
mqttClient.setServer(MQTT_HOST, MQTT_PORT);
```

A continuación se muestran algunos de los *topics* MQTT empleados en esta práctica, los mismos deben ser declarados dentro del código para poder publicar y recibir mensajes en los mismos.

```
//__________ MQTT Topics _____
#define MQTT_PUB_UID  "esp32/PN532"
#define MQTT_SUB_OUT_TEMP "esp32/OutputControl"
#define MQTT_SUB_DOOR "esp32/DoorControl"
```

Para publicar un mensaje en un *topic* se emplea la función *.publis()*, en donde se indica el nombre del *topic*, la calidad del servicio y el mensaje que se va a enviar, también es importante establecer un ID para cada paquete que contendrá un mensaje.
A continuación se muestran los mensajes que se publican cuando se acerca una tarjeta RFID al lector. Al reconocer una tarjeta válida, se publica un mensaje en el *topic esp32/DoorControl* con el mensaje *OPEN* y al mismo tiempo, se envía la identidad a la que pertenece el UID de la tarjeta leída por el *topic esp32/PN532*, estos mensajes luego son leídos por los nodos suscriptores de Node-RED y pueden presentar la información en el *Dashboard*.

```
 // Publica un mensaje MQTT en el topic  "esp32/DoorControl"
uint16_t packetIdPub3 = mqttClient.publish(MQTT_SUB_DOOR, 0,
    true, String("OPEN").c_str());
Serial.printf("Publishing on topic %s at QoS 1, packetId: %i"
    , MQTT_SUB_DOOR, packetIdPub3);
Serial.println("");

  // Publica un mensaje MQTT en el topic  "esp32/PN532"
uint16_t packetIdPub4 = mqttClient.publish(MQTT_PUB_UID, 0,
    true, String("Franklin").c_str());
Serial.printf("Publishing on topic %s at QoS 1, packetId: %i"
    , MQTT_PUB_UID, packetIdPub4);
Serial.println("");
```

Para suscribir el ESP32 a un *topic* se emplea la función *suscribe(topic, QoS)*, en la cual se especifica el nombre del tema y la calidad del servicio, con ello el microcontrolador será capaz de recibir todos los mensajes enviados por el nodo publicador de Node-RED y, según el mensaje recibido el ESP32 activará o desactivará el pin 21, tal como se muestra a continuación.

```
    //suscripcion para control de la puerta
    uint16_t packetIdSub2 = mqttClient.subscribe(MQTT_SUB_DOOR,
         2);
    Serial.print("Subscribing  at QoS 2, packetId: ");
    Serial.println(packetIdSub2);

     if (messageTemp == "OPEN"){
    digitalWrite(DoorPin, HIGH);
    Serial.println("Puerta abierta");
    }
    if (messageTemp == "CLOSE"){
    digitalWrite(DoorPin, LOW);
    Serial.println("Puerta cerrada");
    }
```

El código completo puede encontrarse en el repositorio creado para esta práctica.

3.6. Entorno de control y monitoreo usando Node-Red

Metodologia: Se ha desarrollado un entorno de visualización y control de los circuitos de control anteriormente mencionados. Esta solución permite a los administradores y encargados del negocio tener acceso remoto y en tiempo real a los datos y variables que se están monitoreando en los circuitos.

Entre las áreas de visualización y control que se han desarrollado en el entorno de NodeRED, se encuentran la visualización del nivel de stock de los productos, el control de acceso a la bodega, el monitoreo de la temperatura y humedad, el control de la refrigeración y el control rotativo de la plataforma de los productos.

El uso de esta plataforma permite a los encargados del negocio supervisar y controlar de manera eficiente todos los aspectos relevantes para el éxito de su empresa. La información y datos que se reciben de los circuitos de control permiten tomar decisiones informadas, prevenir problemas y optimizar el rendimiento del negocio.

Implementación y Resultados Debido a que Node-Red se basa en la programación mediante nodos, solo basca con arrastrar al flujo de trabajo los nodos necesarios, en este caso se empleó nodos de publicación y suscripción MQTT, los cuales se configuran para que se conecten al servidor Mosquitto en donde el ESP32 envía todos los datos de

los sensores. Además se se agregó nodos de escritura y lectura para conectarse con la plataforma Blynk y, finalmente, se agregó nodos de la aplicación de mensajería Telegram, los cuales permiten enviar notificaciones y mensajes a un *bot* de Telegram.
En la figura 9 se muestra el flujo de trabajo para conectar los diferentes sensores, módulos y dispositivos.
Mientras que en la figura 10 se muestra la interfaz de usuario creada mediante el flujo de Node-Red.

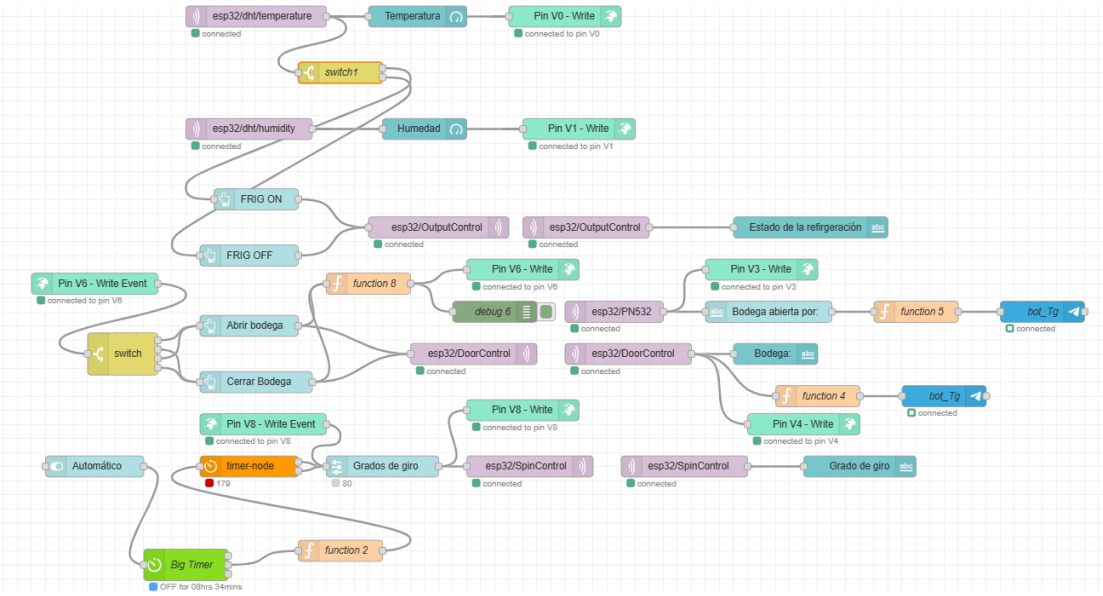

Figura 9: Flujo de trabaja en Node-Red

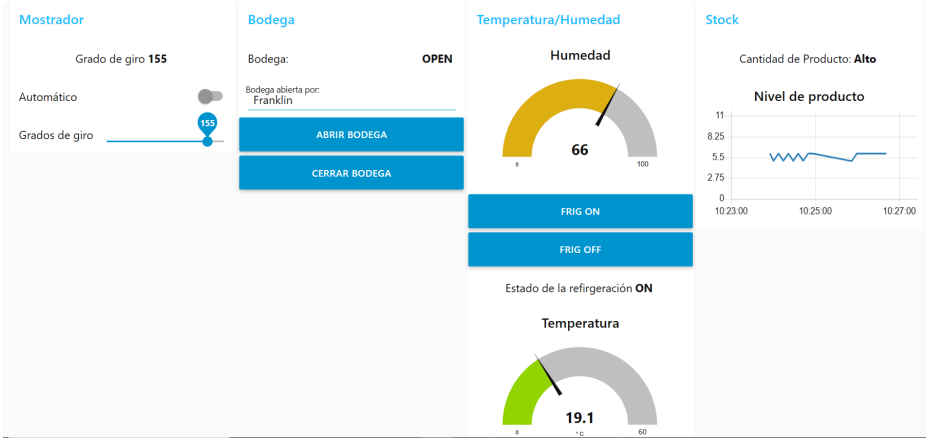

Figura 10: Interfaz de usuario en Node-Red

3.7. Entorno de control y monitoreo usando BLYNK

En este caso, se ha desarrollado un entorno de visualización y control en la plataforma Blynk para los circuitos de control mencionados anteriormente de manera remota . Esto permite a los encargados del negocio tener acceso remoto a la información y datos de los circuitos en tiempo real.

El uso de la plataforma Blynk permite a los encargados del negocio monitorear y controlar de manera eficiente los aspectos clave del negocio, lo que les permite tomar decisiones informadas y evitar problemas potenciales. Además, la plataforma Blynk es fácil de usar y personalizable, lo que significa que se puede adaptar a las necesidades específicas de cada negocio. Este entorno tiene la capacidad de monitorear y controlar áreas críticas como el acceso, la temperatura, la humedad y la rotación de los productos,

Implementación y Resultados La plataforma Blynk es capaz de conectar varios dispositivos IoT que envían diferentes flujos de datos o *datastreams*. Para simular estos flujos de datos, en esta práctica se utilizó nodos de Node-Red, y para conectar ambas plataformas es necesario declarar estos *datastreams* en Blynk. En la Figura 12, se puede observar cómo se declaran los *datastreams* en Blynk y se especifica el tipo de dato que se transmitirá en cada flujo.

Una vez declarados los *datastreams* y enlazados con los dispositivos, Blynk permite crear una interfaz de usuario personalizada utilizando diferentes widgets preconstruidos, como botones, interruptores, medidores y gráficos. En la Figura 11, se muestra un ejemplo de la interfaz de usuario creada en Blynk, que es similar a la interfaz desarrollada en Node-Red (ver Figura 10), y que permite controlar y monitorear los diferentes dispositivos conectados.

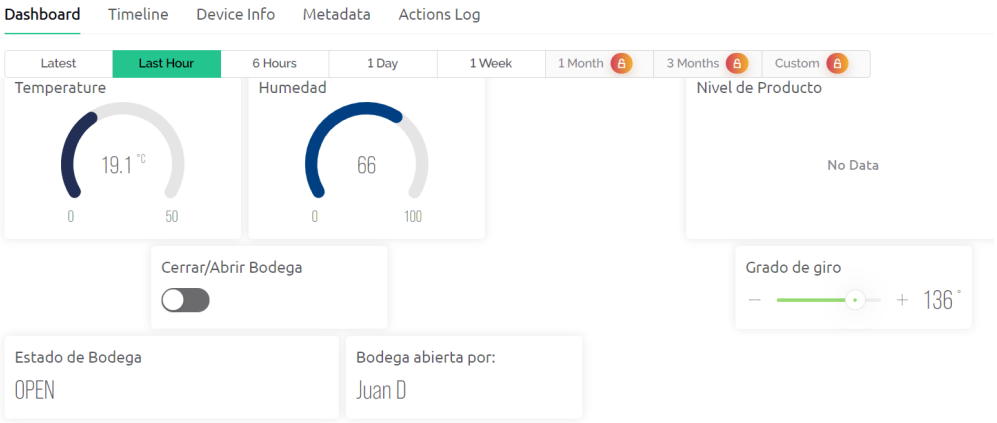

Figura 11: Entorno de control $BLYNK$

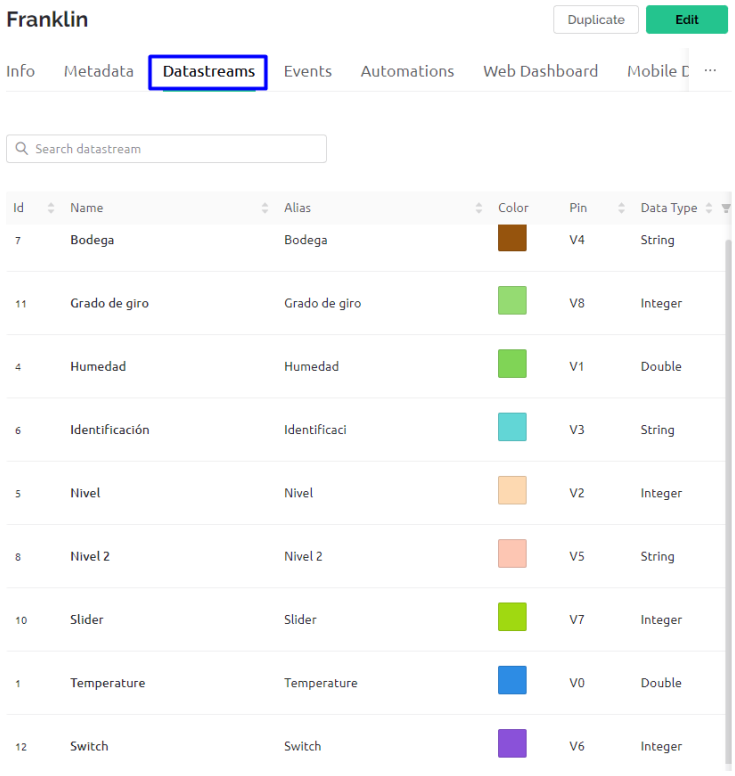

Figura 12: Declaración de los *datastreams* del dispositivo Blynk

4. Conclusiones

- Una de las ventajas de emplear un servidor MQTT en una implementación de IoT es que permite una gestión más eficiente de los datos recolectados por los dispositivos. En lugar de que cada dispositivo tenga que preocuparse por el almacenamiento y procesamiento de los datos, estos simplemente se envían al servidor MQTT. Luego, se puede utilizar un cliente MQTT para recoger, procesar y almacenar los datos, o incluso para controlar los dispositivos conectados, como el caso de Node-Red en esta práctica. Esta arquitectura escalable permite que los sistemas de IoT crezcan sin preocuparse por la complejidad de la recolección y procesamiento de los datos. De esta manera, el servidor MQTT actúa como una capa de abstracción que facilita la comunicación entre los dispositivos y el sistema de gestión de datos de IoT.

- Además de las ventajas de emplear un servidor MQTT en una implementación de IoT, en esta práctica también se evidenció la utilidad de utilizar Node-Red. Al ser una plataforma de código libre, Node-Red cuenta con una gran cantidad de nodos que pueden ser utilizados para diferentes soluciones IoT. De esta forma, después de recibir los datos desde el servidor MQTT, Node-Red puede procesar esta información y conectarlos a otros nodos, como los de Telegram o los de Blynk. Esto permite una integración más fácil de diferentes soluciones y herramientas de IoT, lo que resulta en una arquitectura más flexible y escalable. En conjunto, la combinación de un servidor MQTT y Node-Red facilita el proceso de recolección, procesamiento y visualización de los datos de IoT, y brinda la capacidad de conectar y controlar dispositivos de manera eficiente.

Referencias

1. N. Nicholls, D. O'Leary, and A. Rose, "Node-red: A visual tool for wiring the internet of things," https://nodered.org/, 2016, accessed: March 29, 2023.
2. S. Bose, R. Islam, and A. K. Paul, "Internet of things security: Issues, challenges, and solution," in *Internet of Things and Big Data Analytics toward Next-Generation Intelligence*. Cham, Switzerland: Springer International Publishing, 2019, pp. 193–220. [Online]. Available: https://doi.org/10.1007/978-3-319-89483-5_8
3. E. Systems, "ESP32 Technical Reference Manual," https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf, 2021, accessed on: March 29, 2023.
4. ElecFreaks, "Hc-sr04 ultrasonic sensor," https://www.elecfreaks.com/store/hcsr04-ultrasonic-sensor-distance-measuring-module.html, 2014, [Online; accessed 29-Mar-2023].
5. L. Aosong Electronics Co., "Dht11: Datasheet," https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf, 2007.
6. Blynk, "Blynk," https://blynk.io/, 2021.