



**Desarrollo de un sistema IoT para el control y monitoreo remoto; y detección y notificación de bombillas apagadas del sistema actual de iluminación de crisantemos en la empresa Flores El Capiro S.A**

Luis Eduardo Rodríguez Ascuntar

Informe de práctica para optar al título de Ingeniero Electrónico

Asesor interno

David Stephen Fernández Mc Cann  
Profesor de la Facultad de Ingeniería

Asesor externo

Jhon Fredy García Hincapié  
Director de innovación tecnológica y nuevos  
negocios

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería Electrónica  
Medellín, Antioquia, Colombia

2023

<b>Cita</b>	(Rodríguez Ascuntar, 2023)
<b>Referencia</b>	Rodríguez Ascuntar, L. E. (2023). <i>Desarrollo de un sistema IoT para el control y monitoreo remoto; y detección y notificación de bombillas apagadas del sistema actual de iluminación de crisantemos en la empresa Flores El Capiro S.A.</i> [Semestre de industria]. Universidad de Antioquia, Medellín, Colombia.
<b>Estilo APA 7 (2020)</b>	



Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

**Rector:** John Jairo Arboleda Céspedes.

**Decano/Director:** Jesús Francisco Vargas Bonilla.

**Jefe departamento:** Augusto Enrique Salazar Jiménez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## Tabla de contenido

Resumen.....	7
Abstract.....	8
1. Introducción.....	9
2. Objetivos .....	10
2.1    Objetivo general.....	10
2.2    Objetivos específicos .....	10
3. Marco teórico .....	11
3.1    Resistor.....	11
3.2    Fotorresistencia .....	11
3.3    Transistor.....	11
3.4    Relevador .....	11
3.5    ESP32-WROOM-32D .....	12
3.6    Protocolo MQTT.....	12
3.7    Mosquitto .....	12
3.8    Node-red.....	12
3.9    Arduino IDE .....	13
4. Metodología .....	13
4.1    Diagnóstico del SI actual en el invernadero. ....	13
4.2    Diseño de circuito detector de luminancia en invernadero. ....	13
4.3    Instalación de software. ....	15
4.3.1    Instalación del servidor Mosquitto. ....	15
4.3.2    Instalación de Herramienta programación Node-Red.....	16
4.3.3    Instalación de base de datos MongoDB. ....	16
4.4    Conexión entre elementos principales. ....	16
4.4.1    Conexión entre Esp32 y Circuito detector de luminancia y SI.....	16

4.4.2	Conexión entre Esp32 y servidor Mosquito. ....	17
4.4.3	Conexión entre Servidor Mosquito y Node-Red.....	18
4.5	Diseño de aplicación web .....	19
5.	Resultados y análisis.....	19
6.	Conclusiones.....	23
7.	Referencias .....	24

## Lista de figuras

<b>Figura 1.</b> Circuito divisor de tensión. a) Esquemático con resistores, b) Esquemático con fotoresistores, c) Montaje real. ....	14
<b>Figura 2.</b> Fotoresistor o LDR. a) LDR sin cubierta, b) LDR Con cubierta. ....	15
<b>Figura 3.</b> Montaje de LDR en el bombillo de una guirnalda.....	15
<b>Figura 4.</b> Esquema de la conexión entre Esp32 y circuito detector de luminancia. ....	17
<b>Figura 5.</b> Esquema de la conexión entre Esp32 y Contactor.....	17
<b>Figura 6.</b> Esquema del circuito de conexión entre Esp32 y Servidor Mosquito. ....	18
<b>Figura 7.</b> Esquema de conexión entre Servido Mosquito y Node-red.....	19
<b>Figura 8.</b> PCB de la conexión de esp32, circuito detector de luminancia y sistema de iluminación. ....	20
<b>Figura 9.</b> Montaje de fotoresistor en bombillo de cada guirnalda. ....	20
<b>Figura 10.</b> Interfaz gráfica de usuario.. ....	21
<b>Figura 11.</b> Comportamiento de Sistema de iluminación. ....	21
<b>Figura 12.</b> Recepción de alerta de bombillo apagado en Smartphone. ....	22

## **Siglas, acrónimos y abreviaturas**

<b>UdeA</b>	Universidad de Antioquia
<b>SI</b>	Sistema de Iluminación
<b>IoT</b>	Internet of Things
<b>MQTT</b>	Message Queuing Telemetry Transport
<b>IDE</b>	Integrated Development Environment
<b>PCB</b>	printed circuit board

## Resumen

El presente documento trata del desarrollo de un sistema basado en IoT, el cual tiene como objetivo realizar el control y monitoreo remoto; adicionalmente, la detección y notificación de bombillas apagadas del sistema actual de iluminación de crisantemos en la empresa Flores El Capiro S.A. Inicialmente se desarrolla un circuito eléctrico en capacidad de detectar cambios de luminancia y con conexión a la placa de desarrollo ESP32; luego se establece la arquitectura de conexión entre la placa ESP32 con el sistema de iluminación, circuito detector de luminancia y el servidor Mosquitto; posteriormente se diseña e implementa una interfaz gráfica en el software Node-red para la interacción remota del personal de la empresa y el sistema de iluminación; más adelante, se obtiene resultados favorables y concretos tales como la interacción remota entre personal de la empresa y el sistema de iluminación; además, la notificación temprana de un bombillo apagado a través de mensajes de texto hacia un Smartphone; finalmente se concluye que el sistema desarrollado cumple a cabalidad con los objetivos planteados y presenta un alto grado de escalabilidad.

*Palabras clave:* servidor, placa de desarrollo, invernadero, iot, software, sistema de iluminación.

### **Abstract**

This document is about the development of a system based on IoT, whose objective is to carry out remote control and monitoring; additionally, the detection and notification of turned off bulbs of the current lighting system for chrysanthemums in the company Flores El Capiro S.A. Initially, an electrical circuit capable of detecting changes in luminance and with connection to the ESP32 development board is developed; then the connection architecture between the ESP32 board with the lighting system, luminance detector circuit and the Mosquitto server is established; afterwards, a graphical interface is designed and implemented in the Node-red software for the remote interaction of company personnel and the lighting system; subsequently, favorable and concrete results are obtained, such as the remote interaction between the company's personnel and the lighting system; furthermore, the early notification of an out light bulb through text messages to a Smartphone; finally it is concluded that the developed system fully complies with the proposed objectives and presents a high degree of scalability.

*Keywords:* server, development board, greenhouse, iot, software, lighting system



## 1. Introducción

La plantación de crisantemos es uno de los cultivos de flor de corte más importante en Colombia y el que más se exporta después de las rosas y claveles. No obstante, para tener una producción exitosa y de calidad, esta planta requiere de mucho cuidado durante su propagación y floración, por ello las empresas que se dedican a la labor de producción de esta especie vegetal en invernaderos deben tener un control eficiente en las condiciones químicas y físicas del suelo, así mismo de las plagas e iluminación [1]. La iluminación juega un papel fundamental en el desarrollo de las plantas y el control de ésta permite manipular directamente el fotoperiodo (número de horas de luz por día) de las plantas. En el caso del crisantemo para su floración requiere de un fotoperiodo inferior a 14,5h el cual es considerado como crítico; por otra parte, para mantener la planta en estado vegetativo se exige un fotoperiodo mayor al crítico [2]; de ahí que, las empresas dedicadas a la floricultura utilicen sistemas eléctricos para el suministro de luz artificial a través de bombillos fluorescentes o led.

Durante el proceso de propagación o estado vegetativo es importante para una buena producción de esquejes que el crisantemo reciba luz artificial en el transcurso de un tiempo no menor a 3 horas por noche; por consiguiente, una falla en el sistema de iluminación(SI) como un apagón, ya sea de un solo bombillo, una sola guirnalda de luces o el sistema completo es perjudicial para dicho proceso si no se corrige de manera temprana; de no ser así, genera una gran cantidad de pérdidas de esquejes por planta. Con base en lo anterior, el presente proyecto trata sobre la actualización realizada al SI ya existente en un invernadero de la empresa Flores el Capiro S.A, dicha actualización consistió en integrar al sistema actual un nuevo sistema basado en IoT (Internet of things), con el fin de tener el control y noción de manera remota el estado en que se encuentra el SI: apagado, encendido y encendido con problema de iluminación (bombillo apagado en alguna de las guirnaldas de luces); en el caso de que el SI se encuentre en este último estado, automáticamente el nuevo sistema envía una alerta mediante un mensaje de texto al Smartphone del operario encargado. Es importante mencionar que el alcance de este proyecto solo abarca la instalación del nuevo sistema en la mitad del invernadero.

Para llevar a cabo la actualización en el SI se diseñó un circuito eléctrico sensible a los cambios de luz para la detección de bombillas apagadas; además, se utilizó la tarjeta de desarrollo ESP32 como elemento central de la captura, ejecución de eventos y

transmisión/recepción de datos a través de internet hacia/desde una computadora en la oficina central. Al mismo tiempo se desarrolló una interfaz gráfica y se la implementó en este equipo, para que el personal de la empresa interactúe de manera remota con el SI.

Finalmente, el nuevo sistema IOT funciono adecuadamente; las ordenes de encendido y apagado enviadas desde la interfaz gráfica hacia el SI se ejecutaron adecuadamente, de igual manera en la interfaz se evidencia el estado actual del SI. En cuanto a la detección y notificación de bombillos apagados el sistema IOT fue capaz de enviar la notificación al Smartphone del operario cuando se presentó dicho problema.

## **2. Objetivos**

### **2.1 Objetivo general**

Desarrollar un sistema IoT, para el control y monitoreo remoto; y detección y notificación de bombillas apagadas del sistema actual de iluminación de crisantemos en la empresa Flores El Capiro S.A, utilizando para ello la placa de desarrollo ESP32 y el software Node-red y Mosquitto.

### **2.2 Objetivos específicos**

- Desarrollar un sistema eléctrico sensible a la luz, con capacidad de detectar cambios de luminancia en los bombillos del sistema de iluminación, por medio del uso de elementos eléctricos como fotoresistores, resistores, entre otros; igualmente, que tenga una conexión física y lógica con la placa de desarrollo ESP32.
- Diseñar la arquitectura de conexión de la placa ESP32 con el sistema de iluminación y el servidor Mosquitto, para obtener el control de la iluminación en el invernadero y recibir o transmitir datos del estado del sistema usando el protocolo MQTT, a través de dispositivos con conexión a internet.
- Desarrollar una interfaz gráfica mediante el uso de la herramienta de desarrollo Node-red, para la interacción remota del personal de la empresa con el sistema de iluminación.
- Validar el sistema de iluminación mejorado, a través de pruebas como el envío de ordenes directamente de la interfaz gráfica y el apagado manual de algunos bombillos, con el fin

verificar su correcto funcionamiento en la capacidad de control y monitoreo remoto; y notificación de bombillas apagadas.

### **3. Marco teórico**

A continuación, se muestra de manera resumida una definición de todos los elementos que sirvieron de apoyo para llevar a cabo la actualización del SI actual en el invernadero.

#### **3.1 Resistor**

Es un elemento electrónico el cual tiene como propósito principal limitar la corriente eléctrica que circula a través de sus terminales. Un resistor funciona según la ley de ohm, la cual afirma que el voltaje entre sus terminales es directamente proporcional a la corriente que fluye por medio de ella; sus unidades de medida son los ohmios [3].

#### **3.2 Fotorresistencia**

También llamado LDR por sus siglas en inglés (Light-Dependent-Resistor), es un componente electrónico que varía su resistencia de acuerdo a la cantidad de luz incidente sobre su cuerpo, el cual es una célula fotorreceptora. La resistencia de este elemento es inversamente proporcional a la cantidad de luz; entre mayor cantidad de luz incida por su cuerpo menor será su resistencia [4].

#### **3.3 Transistor**

Es un dispositivo electrónico formado por materiales semiconductores que regula el flujo de corriente o de tensión sobre un circuito, actuando como interruptor o amplificador. El transistor consta de tres pines o terminales las cuales reciben como nombre colector, emisor y base. Cuando una corriente pequeña circula por la base del transistor se obtendrá una corriente mucho más grande circulando por el colector y emisor [5].

#### **3.4 Relevador**

También llamado relé. Es un aparato electromagnético que funciona como interruptor que permite o no el paso de la corriente eléctrica hacia un circuito de mucha potencia; dicho interruptor es controlado por un circuito eléctrico que generalmente es de baja potencia. Cuando se mantiene una corriente por la bobina interna del relé, esta crea un campo magnético y a su vez funciona

como electroimán, el cual atrae o acciona un conjunto de uno o varios contactos metálicos que posibilitan el flujo de una gran cantidad de corriente entre las terminales [6].

### **3.5 ESP32-WROOM-32D**

Es una placa de desarrollo con un consumo energía muy bajo. Tiene el microcontrolador de 32 bits Xtensa Dual-Core 32-bit LX6, comunicación inalámbrica Wifi y Bluetooth; trae consigo dos núcleos que se pueden controlar individualmente y una frecuencia de reloj de CPU que se puede ajustar de 80 MHz a 240 MHz, 448kb de ROM, 520 kb de SRAM; además, integra un amplio conjunto de periféricos, que van desde sensores táctiles capacitivos, sensores Hall, interfaz de tarjeta SD, Ethernet, SPI de alta velocidad, UART, PS y PC [7].

### **3.6 Protocolo MQTT**

Es un protocolo de comunicación Máquina a Máquina (M2M) que actualmente es bastante utilizado en aplicaciones IoT por su sencillez y ligereza. El funcionamiento de MQTT es un servicio de mensajería que utiliza la metodología de publicador/suscriptor (pub-sub). Para la comunicación en este tipo de red es necesario disponer de un servidor MQTT o bróker, el cual se encarga de administrar los mensajes que son enviados por los clientes hacia otros clientes. Un cliente puede publicar un mensaje en un determinado tópico, entonces otro cliente puede suscribirse a éste, y el bróker le hará llegar los mensajes suscritos [8].

### **3.7 Mosquitto**

Es un bróker o intermediario gratuito, ideal para administrar mensajes MQTT, que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Este servidor se encarga de recibir los mensajes enviados por un publicador y entregarlos a un suscriptor. Este software es liviano, por ende, adecuado para su uso en muchos dispositivos, desde computadoras de placa única de bajo consumo como raspberry pi hasta computadoras personales o equipos empresariales. Presenta una gran portabilidad y está disponible para una amplia gama de plataformas. [9].

### **3.8 Node-red**

Es una herramienta de programación visual de carácter libre, se utiliza para conectar dispositivos de hardware, APIs y servicios de internet. Proporciona es un editor de flujo basado en el navegador donde se puede añadir o eliminar nodos y conectarlos entre sí con el propósito de hacer que se comuniquen entre ellos. Algunos de los nodos más usados para aplicaciones IoT son MQTT Output (nodo para publicar mensajes en un tema), MQTT In (nodo para suscribirse

a los mensajes del tema especificado) y el módulo dashboard que proporciona un conjunto de nodos para crear rápidamente un tablero de datos en vivo [10].

### **3.9 Arduino IDE**

Es una aplicación gratuita y multiplataforma, es decir se puede instalar en diferentes sistemas operativos como Windows, GNU/Linux y Mac OS, la cual se puede utilizar para escribir y cargar programas en placas de desarrollo tales como Arduino Uno, Esp32, entre otras. Esta aplicación permite escribir el código con el lenguaje de programación C/C++, de igual manera admite cargar librerías que sean necesarias para el buen funcionamiento de los códigos a cargar en alguna placa de desarrollo compatible [11].

## **4. Metodología**

A continuación, se describe cada uno de los pasos realizados y técnicas empleadas para el desarrollo del proyecto.

### **4.1 Diagnóstico del SI actual en el invernadero.**

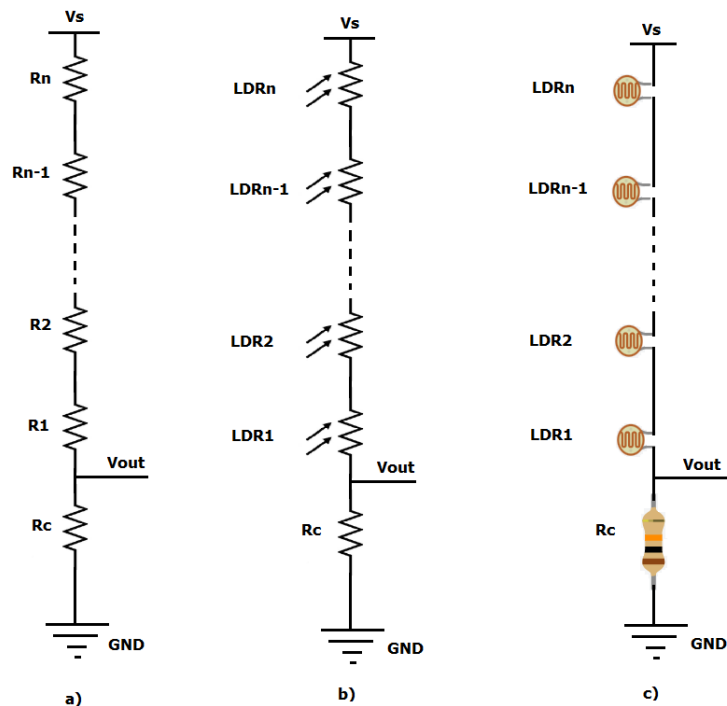
Actualmente los invernaderos que dispone la empresa Flores El Capiro S.A cuentan con un SI eléctrico para crisantemos que funciona a 220V, el cual está compuesto por una red de guirnalda conectadas en paralelo y estas a la vez tienen bombillos con el mismo tipo de conexión mencionado anteriormente, por tanto, posee la virtud de que si llegase a fallar una guirnalda o bombillo esto no afectara al resto de la red. En cuanto a la funcionalidad, el estado encendido o apagado del SI en las noches, es controlado por medio de un dispositivo electrónico llamado temporizador, el cual permite suministrar la energía eléctrica y, por ende, controla de manera indirecta la cantidad de luz suministrada a las plantas en determinados intervalos de tiempo. Cabe mencionar que el suplemento de luz artificial en la noche es indispensable para el cultivo de crisantemos, por ello es importante que un operario realice una inspección tanto en el día como en la noche verificando que los bombillos funcionen correctamente en todos los invernaderos presentes en los 20 bloques que dispone la finca de la empresa.

### **4.2 Diseño de circuito detector de luminancia en invernadero.**

Teniendo en cuenta el funcionamiento de un fotoresistor o LDR; es decir su variación de resistencia con respecto a la cantidad de luz incidente sobre su superficie y también que la

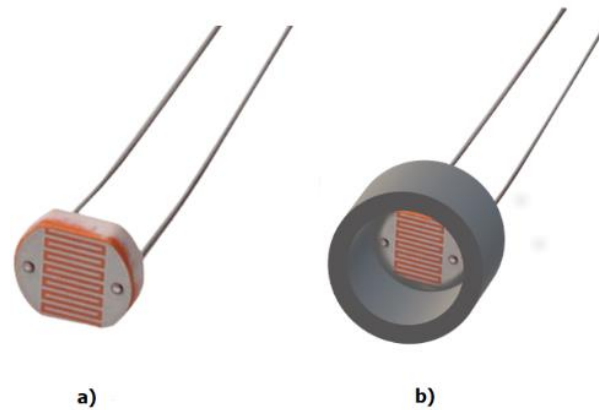
conexión de resistencias en serie resulta una resistencia equivalente a la suma de estas como se evidencia en (1), se diseñó un circuito divisor de tensión (**Figura 1**) y se realizó su montaje en cada guirnalda del invernadero (**Figura 3**).

$$R_{total} = R_1 + R_2 + \dots + R_{n-1} + R_n \quad (1)$$



**Figura 1.** Circuito divisor de tensión. a) Esquemático con resistores, b) Esquemático con fotoresistores, c) Montaje real.

Dado que la superficie receptora de luz de la LDR presenta un ángulo sólido grande, por lo cual, no solo recibe la luz proveniente de su respectivo bombillo sino también de los bombillos aledaños, se optó por reducir este ángulo sólido agregando un objeto cilíndrico a la superficie receptora de la LDR (**Figura 2**). Es importante resaltar que cada LDR se fija al platillo del bombillo y se ubica la superficie sensible a la luz en dirección hacia dicho bombillo (**Figura 3**).



**Figura 2.** Fotorresistor o LDR. a) LDR sin cubierta, b) LDR Con cubierta.



**Figura 3.** Montaje de LDR en el bombillo de una guirnalda.

### 4.3 Instalación de software.

Es importante mencionar que la instalación del software utilizado para este proyecto se realizó en una computadora con procesador Intel i5, RAM 8Gb y sistema operativo Fedora 35.

#### 4.3.1 Instalación del servidor Mosquitto.

Puesto que, en el sistema desarrollado se utilizó el protocolo de comunicación MQTT fue necesario la instalación del servidor Mosquitto, el cual es ideal para la gestión de mensajes en este tipo de comunicación (subscriber – publicador); este servidor se encarga de recibir los mensajes enviados por un publicador y entregarlos a un subscriber [12].

Para el funcionamiento óptimo del servidor en este proyecto, se requiere realizar algunos ajustes sobre su archivo de configuración, tal como configurar la visibilidad del servidor para establecer comunicación desde otros dispositivos y la autenticación [13].

### **4.3.2 Instalación de Herramienta programación Node-Red.**

En la mayoría de sistema IoT es indispensable disponer de una interfaz gráfica, con la cual las personas pueden interactuar de manera remota con los elementos a controlar o monitorear y este proyecto no es la excepción; por consiguiente, se instaló la herramienta de programación visual Node-Red [14]. Esta herramienta permitió desarrollar la interfaz gráfica con diferentes elementos como botones, graficas, entre otros, que permiten manipular y monitorear el estado del SI y de sus guirnaldas en el invernadero.

Para el funcionamiento óptimo del servidor en este proyecto, se requiere realizar algunos ajustes sobre su archivo de configuración, tal como configurar la visibilidad del servidor para establecer comunicación desde otros dispositivos y la autenticación [15].

### **4.3.3 Instalación de base de datos MongoDB.**

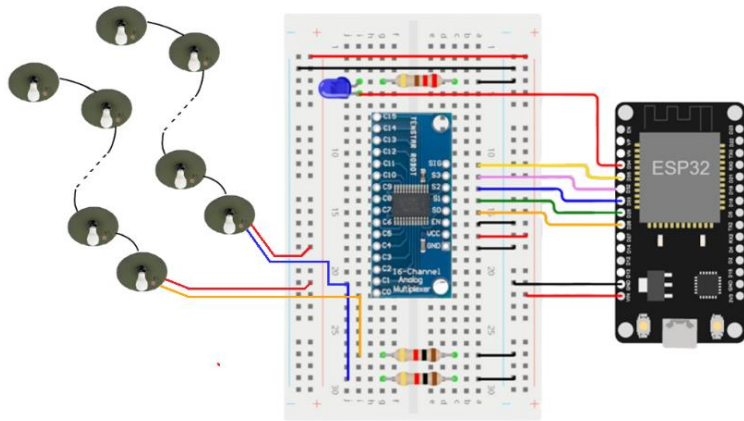
Cuando se requiere realizar algún estudio del comportamiento de un sistema y luego detectar que factores pueden mejorarlo o afectarlo ya sea a corto o largo plazo, es útil registrar o almacenar datos de dicho comportamiento. En el caso de este proyecto, para llevar a cabo el proceso de almacenar los datos se instaló la base de datos MongoDB, posteriormente se estableció una conexión con la herramienta de desarrollo node-red con el objetivo de almacenar los tiempos de temporización del SI y el momento de apagado de un bombillo en los invernaderos [16].

## **4.4 Conexión entre elementos principales.**

### **4.4.1 Conexión entre Esp32 y Circuito detector de luminancia y SI.**

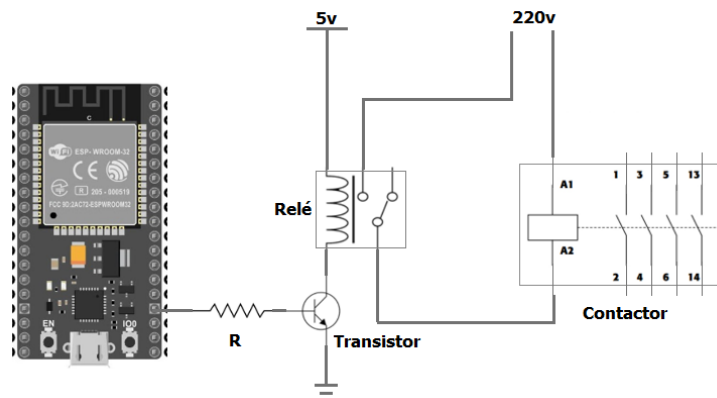
En cada invernadero que tiene la empresa, la cantidad de guirnaldas que maneja el SI es grande y aunque este proyecto solo abarca la mitad del invernadero, es decir un SI con 16 guirnaldas es necesario el uso de un multiplexor de 16 canales a 1, esto con el objetivo de ocupar la menor cantidad de pines analógicos de la Esp32 (ya que son limitados) y además medir la mayor cantidad de guirnaldas. En consecuencia, el circuito detector de luminancia ubicado en cada guirnalda se conecta a un único canal de entrada del multiplexor y el canal de salida se conecta a un pin analógico de la Esp32; además, de los pines de canal, el multiplexor trae los pines de control, con los cuales permite elegir el canal que deseamos leer; y -a diferencia del pin de canal estos se conectan a pines digitales de la Esp32 (**Figura 4**).





**Figura 4.** Esquema de la conexión entre Esp32 y circuito detector de luminancia.

Por otro lado, el suministro de energía eléctrica al SI es controlado por un Contactor, este dispositivo funciona como interruptor y cuenta con dos terminales de entrada (A1 y A2) para activarlo o desactivarlo. En este proyecto, el proceso de activación o desactivación se controló desde la Esp32 y dado que este proceso funciona con un voltaje de 110 o 220 voltios, se necesitó del uso de un relé<sup>1</sup> para la conexión entre la Esp32 y el Contactor (**Figura 5**).



**Figura 5.** Esquema de la conexión entre Esp32 y Contactor.

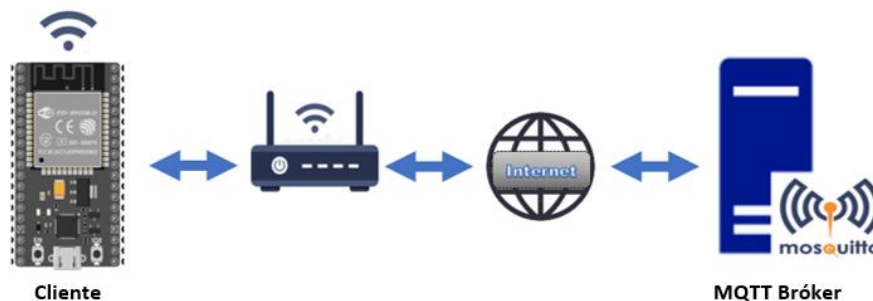
#### 4.4.2 Conexión entre Esp32 y servidor Mosquitto.

El protocolo de mensajería MQTT utiliza una metodología de publicador-subscriptor es decir, los elementos finales que componen esta red llamados clientes pueden publicar o

<sup>1</sup> Este dispositivo permite controlar circuitos eléctricos de gran potencia con circuitos de baja potencia.

subscribirse a un tópic<sup>2</sup> para enviar o recibir mensajes respectivamente. Para establecer la comunicación, los clientes se conectan con un servidor central denominado bróker, en este caso el servidor Mosquitto; este se encarga de recibir los mensajes de todos los dispositivos que publican y distribuirlos a los dispositivos que se subscriben [17].

A través de un router habilitado por la empresa, la Esp32 se conecto via wifi y tuvo salida a internet, esto con el fin establecer la comunicación con el servidor Mosquitto previamente instalado. En este caso, la Esp32 funcionó como cliente publicador y suscriptor, por tanto se implementó tres topicos; el primero para publicar el estado del SI; el segundo para subscribirse a los mensajes u ordenes de cambio de estado sobre el SI y el tercero para publicar el estado de las guirnaldas (**Figura 6**). En la Esp32, el protocolo MQTT se implento por medio de las funciones presentes en la librería “PubSubClient.h”; librería previametente instalada en arduino IDE.

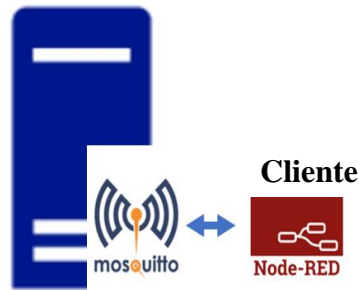


**Figura 6.** Esquema del circuito de conexión entre Esp32 y Servidor Mosquitto.

#### 4.4.3 Conexión entre Servidor Mosquito y Node-Red

Como se mencionó anteriormente, Node-Red es una herramienta que permite desarrollar un montón de aplicaciones entre ellas algunas con interfaz gráfica. Para el caso de controlar y/o monitorear dispositivos estas aplicaciones necesitan de alguna manera la recepción o envío de datos para desplegarla sobre su interfaz o enviar alguna orden desde ésta. Para el presente proyecto este proceso de adquisición y transmisión de datos se realizó a través de la conexión de Node-Red (cliente) con el servidor Mosquitto implementado el protocolo MQTT, y es lógico que también se utilizó los mismos tópicos mencionados en el numeral anterior para publicar y subscribirse (**Figura 7**). En Node-Red, el protocolo MQTT se implementó por intermedio de los nodos “mqtt in” y “mqtt out” para subscribirse y publicar respectivamente [18].

<sup>2</sup> Es una cadena de texto UTF-8, de una longitud máxima de 65536 caracteres, con la cual el Broker MQTT distingue a que clientes suscritos debe entregar el mensaje enviado por un cliente publicador.



**Figura 7.** Esquema de conexión entre Servido Mosquito y Node-red.

#### 4.5 Diseño de aplicación web

Cuando se requiere controlar o monitorear dispositivos a través de alguna aplicación, es recomendable que la interfaz gráfica sea amigable con el usuario; en otras palabras, que esta aplicación sea fácil de manipular y la información presentada en ésta sea entendible por el usuario. Así, para completar este sistema IoT se diseñó una aplicación con la herramienta de desarrollo Node-red para que el personal de la empresa interactúe de manera remota con el SI en el invernadero.

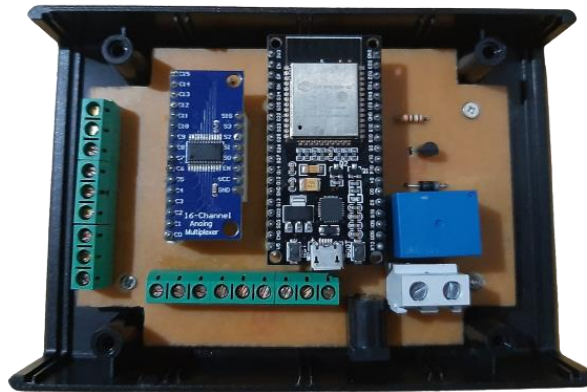
La interfaz gráfica de la aplicación cuenta con un menú de botones que permiten elegir los diferentes invernaderos (Bloques) que dispone la empresa. Cada botón despliega un grupo de elementos de control y de monitoreo. Los elementos de control le permiten al usuario enviar la orden de encendido o apagado al SI; en cambio los de monitoreo permiten visualizar el estado del SI y de las guirnaldas.

En el caso de que una guirnalda presente una falla en la iluminación, es decir cuando algún bombillo de ésta se daña o deja de emitir luz, la aplicación envía un mensaje de texto por Telegram al Smartphone del operario encargado del SI en el invernadero. Este mensaje contiene información como el número del bloque, sección y de la guirnalda que presenta dicha falla.

### 5. Resultados y análisis

Luego de realizar pruebas parciales para verificar el correcto funcionamiento del sistema compuesto por la placa de desarrollo Esp32, circuito detector de luminancia y SI, se realizó la placa de circuito impreso (PCB) de dicho sistema (**Figura 8**). Esta PCB tiene como elemento central de computo la Esp32, un multiplexor para elegir los diferentes canales de entrada; cuenta con 18 puertos para la conexión del circuito detector de luminancia: 16 puertos de entrada y dos

para alimentación energética; también dispone de 2 puertos para la conexión con el SI; por último, dispone de un puerto de entrada para su alimentación energética. Desde el punto de vista energético, la PCB al trabajar con voltajes y corrientes bajos consume muy poca potencia, y desde el lado espacial es ideal para ubicarla en lugares con espacios reducido, ya que tiene pequeños y pocos elementos.



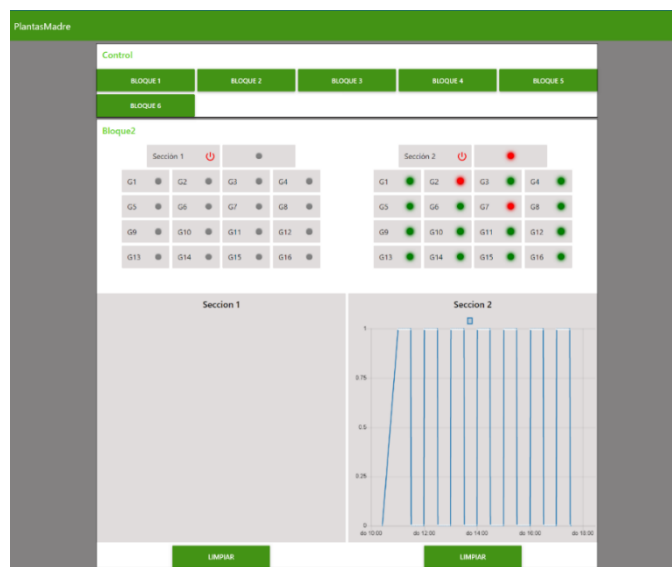
**Figura 8.** PCB de la conexión de esp32, circuito detector de luminancia y sistema de iluminación.

Para la detección de bombillos apagados, se llevó a cabo el montaje del circuito detector de luminosidad en cada guirnalda del SI, ubicando un fotoresistor encapsulado con un elemento cilíndrico en los platillos presentes en cada bombillo de las guirnaldas (**Figura 9**). Dado que se generaban lecturas erróneas al momento detectar un bombillo apagado, esto a razón de que la luz proveniente de los bombillos aledaños influía en la lectura, hubo la necesidad de encapsular el fotoresistor para evitar la luz ajena al propio bombillo, dando como resultado lecturas más certeras y precisas.

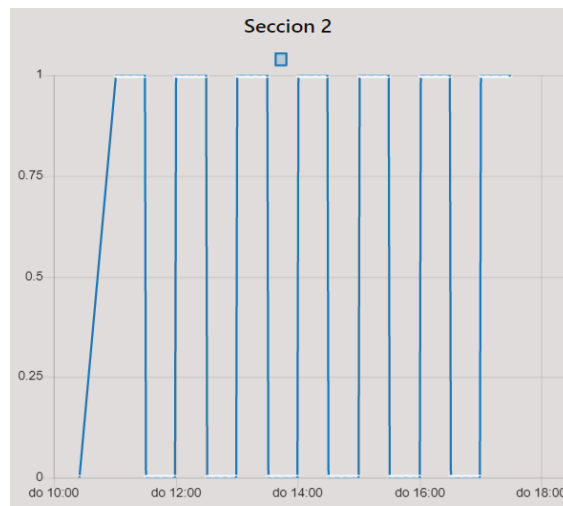


**Figura 9.** Montaje de fotoresistor en bombillo de cada guirnalda.

Por otra parte, la aplicación desarrollada contiene una interfaz gráfica fácil de entender y manipular; presenta diferentes botones y gráficas con los cuales se puede visualizar y controlar el estado del SI. Los botones en la parte de control corresponden a cada bloque en la finca y dado que, en estos bloques el SI está dividido en dos secciones y cada una de estas contiene 16 guirnaldas, la interfaz dispone de dos grupos de elementos para visualizar el estado del SI y de las guirnaldas correspondiente a cada sección. Cada grupo trae un botón para enviar la orden de apagado o encendido del SI, 16 leds para visualizar el estado de las guirnaldas (**Figura 10**) y una gráfica para visualizar el comportamiento del SI en el intervalo de un día (**Figura 11**).

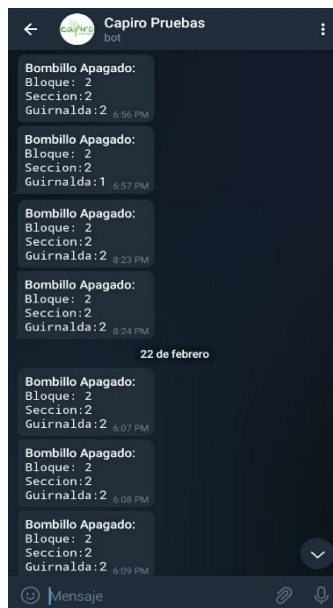


**Figura 10.** Interfaz gráfica de usuario.



**Figura 11.** Comportamiento de Sistema de iluminación.

Finalmente, cuando se realizó un apagado intencional de algunos bombillos del SI, la aplicación envió un mensaje de texto por medio de Telegram hacia el Smartphone previamente configurado para recibir mensajes desde esta aplicación. Estos mensajes contienen la información del número del bloque, sección y guirnalda en el cual ocurrió el evento de un bombillo apagado, los cuales son datos suficientes para que el operario encargado de revisar el SI identifique el problema y realice el cambio del bombillo apagado (**Figura 12**).



**Figura 12.** Recepción de alerta de bombillo apagado en Smartphone.

## 6. Conclusiones

Con la realización de este trabajo se obtuvo un sistema IoT, para el control y monitoreo remoto, y detección y notificación de bombillas apagadas, el cual está compuesto por un circuito detector de luz, una placa de desarrollo Esp32 y una aplicación web. Con el desarrollo de este sistema se logró detectar con buena exactitud y notificar correctamente los bombillos apagados en el sistema de iluminación; también se logró establecer una conexión lógica y estable entre los componentes dispuestos para el proyecto, por ende, se realizó el control remoto eficiente y una temporización apropiada sobre el sistema de iluminación. Además, con el desarrollo de la aplicación se logró monitorear adecuadamente y almacenar de manera coherente los datos del comportamiento del SI y el estado de las guirnaldas. Finalmente, dado que el protocolo MQTT utilizado para la transmisión de información requiere un ancho de banda reducido y es fácil de implementar, este sistema presenta un alto grado de escalabilidad en cuanto a la cantidad de invernaderos a actualizar.

## 7. Referencias

- [1]. Agr1col@dmin. (2021, noviembre 10). *Los secretos de la productora de crisantemos más grande del mundo*. Redagícola Colombia. <https://www.redagricola.com/co/los-secretos-de-la-productora-de-crisantemos-mas-grande-del-mundo/>
- [2]. Chica, F., & Correa, G. (2005). *EVALUACIÓN DE DOS TRATAMIENTOS FOTOPERIÓDICOS EN CRISANTEMO (Dendranthema grandiflorum (Ramat.) Kitam.), BAJO CONDICIONES DEL INTERTRÓPICO ANDINO ALTO*. [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0304-28472005000200005](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0304-28472005000200005)
- [3]. J.L, B. (2020, abril 10). → *Qué es un Resistor? ▷ Definición, Tipos y Usos*. Electrónica Online. <https://electronicaonline.net/componentes-electronicos/resistor/>
- [4]. *Fotoresistor, LDR o Fotoresistencia*. (2021, abril 23). Mecatrónica LATAM. <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/ldr/>
- [5]. *Conceptos básicos de transistores PNP y NPN con 2N3904, 2N3906, 2N2222, 2N2907*. (s. f.). Digi-Key Electronics. Recuperado 26 de febrero de 2023, de <https://www.digikey.com/es/articles/transistor-basics>
- [6]. *El Relé: Para qué es, para qué sirve y qué tipos existen | Blog SEAS*. (2019, agosto 22). <https://www.seas.es/blog/automatizacion/el-rele-para-que-es-para-que-sirve-y-que-tipos-existen/>
- [7]. *ESP-WROOM-02D pdf, ESP-WROOM-02D Description, ESP-WROOM-02D Datasheet, ESP-WROOM-02D view: ALLDATASHEET :::* (s. f.). Recuperado 26 de febrero de 2023, de <https://pdf1.alldatasheet.com/datasheet-pdf/view/1148033/ESPRESSIF/ESP-WROOM-02D.html>
- [8]. ¿Qué es MQTT? Su importancia como protocolo IoT. (2019, abril 17). Luis Llamas. <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- [9]. *Eclipse Mosquitto*. (2018, enero 8). Eclipse Mosquitto. <https://mosquitto.org/>
- [10]. *Node-RED*. (s. f.). Recuperado 26 de febrero de 2023, de <https://nodered.org/>
- [11]. Peña, C. (2020). *Arduino IDE: Domina la programación y controla la placa*. RedUsers.
- [12]. *How To Install mosquitto on Fedora 34*. (2021, agosto 1). Installati.One. <https://installati.one/fedora/34/mosquitto/>
- [13]. *Práctica 4: Instalar, Configurar y Securizar Mosquitto y Node-RED en Raspberry Pi | Aprendiendo Arduino*. (s. f.). Recuperado 26 de febrero de 2023, de <https://aprendiendoarduino.wordpress.com/2021/02/28/practica-4-instalar-configurar-y-securizar-mosquitto-y-node-red-en-raspberry-pi/>
- [14]. *Running Node-RED locally: Node-RED*. (s. f.). Recuperado 26 de febrero de 2023, de <https://nodered.org/docs/getting-started/local>



- [15]. *Securing Node-RED: Node-RED*. (s. f.). Recuperado 26 de febrero de 2023, de <https://nodered.org/docs/user-guide/runtime/securing-node-red>
- [16]. Instalador, E. (2022, junio 1). ▷ Cómo instalar MongoDB en Fedora 36 ✓ [2022] Paso a paso. *Cómo instalar...* <https://comoinstalar.me/como-instalar-mongodb-en-fedora/>
- [17]. Protocolos de comunicación para IoT. (2019, febrero 21). *Luis Llamas*. <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- [18]. Steve. (2018, abril 15). *Configuring the Node-Red MQTT Publish and Subscribe Nodes*. <http://stevesnoderedguide.com/configuring-the-mqtt-publish-node>