



# TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

---

Sistema de automatización con Arduino  
para el control de las cortinas de nave  
industrial para la empresa Distribuidora de  
Tecnología Especializada CJ SAPI DE C.V.

Informe Técnico de Residencias  
Profesionales

Para obtener el Título de:  
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Especialidad  
Desarrollo de Software

Presenta(n):  
Gómez Mendicuti Mario  
201920425

Coacalco de Berriozábal, Noviembre 2024

## AGRADECIMIENTOS

Gracias infinitas a mis padres, por su amor incondicional y su apoyo moral. Su fe en mí, incluso en los momentos más difíciles, ha sido el pilar de este logro. Sin ustedes, todo esto no habría sido posible. Su amor y sacrificio han sido la luz que guio mi camino a través de este viaje académico.

A mi asesor interno, M. en C. Luis Alberto Ovando Brito, quiero expresar mi profundo agradecimiento por su orientación experta y apoyo continuo a lo largo de este proceso de investigación. Sus valiosas sugerencias y comentarios han sido fundamentales para el desarrollo y éxito de este trabajo.

Quisiera expresar mi más profundo agradecimiento a mi asesor externo, el Ing. Fernando Rubio Rivera al brindarme su experiencia, comprensión y paciencia contribuyeron a mi experiencia en el complejo y gratificante camino del proyecto.

Al Ing. Carlos Mencos García y al Ing. Jairo Cesar Ayala Rivero y a todo el equipo de Distribuidora de Tecnología Especializada CJ SAPI DE C.V. quiero expresar mi sincero agradecimiento por brindarme la oportunidad de realizar mis prácticas profesionales en esta prestigiosa organización.

A mi querida pareja, agradezco desde lo más profundo de mi corazón tu amor incondicional por tu apoyo constante y por celebrar cada pequeño logro durante este viaje académico. Tu aliento y paciencia han sido mi mayor fuente de fortaleza.

## RESUMEN

Dentro de un parque industrial, es crucial manejar de forma eficiente las cortinas industriales. Sin un sistema centralizado para abrir y cerrar estas estructuras, se enfrenta un gran obstáculo. La falta de este sistema no solo complica la asignación de las cortinas a los operadores designados, sino que también aumenta los riesgos de seguridad. Implementar un sistema de control completo no solo mejorará la organización logística, sino que también hará que el manejo de las cortinas sea más seguro, lo que a su vez mejorará la eficiencia y reducirá los riesgos.

El proyecto de residencias "Sistema de Automatización con Arduino y ESP32 para el Control de Cortinas en una Nave Industrial" se centra en la aplicación de la tecnología del Internet de las Cosas (IoT) y microcontroladores para mejorar la eficiencia y la gestión de recursos en el entorno industrial. El objetivo principal es desarrollar un sistema innovador que permita la automatización del proceso de control de cortinas en una nave industrial para la empresa ya mencionada.

El sistema se basa en el uso del microcontrolador ESP32 y el lenguaje de programación Arduino para proporcionar una solución versátil y de bajo costo. Este proyecto busca ofrecer a la empresa una solución que les permita gestionar las cortinas de forma remota, programar horarios de apertura y cierre, y recopilar datos en tiempo real sobre el estado de las cortinas. Esto tiene como objetivo mejorar la eficiencia energética, la seguridad y la comodidad en la nave industrial.

El uso de tecnología IoT permite la conectividad y el control a través de dispositivos móviles o una plataforma web, brindando a los operadores y gerentes un control total sobre las cortinas, independientemente de su ubicación. Además, se espera que la implementación de este sistema proporcione ahorros significativos en costos de energía y una mayor productividad.

## ÍNDICE DE CONTENIDOS

|                                                     |    |
|-----------------------------------------------------|----|
| AGRADECIMIENTOS .....                               | 2  |
| RESUMEN.....                                        | 3  |
| ÍNDICE .....                                        | 4  |
| Índice de Imágenes.....                             | 7  |
| Índice de Tablas .....                              | 9  |
| GENERALIDADES DEL PROYECTO .....                    | 10 |
| 1.1 INTRODUCCIÓN.....                               | 11 |
| 1.2 DESCRIPCIÓN DE LA EMPRESA.....                  | 12 |
| 1.3 PROBLEMA A RESOLVER .....                       | 15 |
| 1.4 OBJETIVOS .....                                 | 16 |
| 1.5 JUSTIFICACIÓN.....                              | 17 |
| MARCO TEÓRICO.....                                  | 18 |
| 2.1 Internet.....                                   | 19 |
| 2.1.1 La web .....                                  | 20 |
| 2.2 IOT (Internet Of Things) .....                  | 21 |
| 2.2.1 Sectores de aplicaciones de IOT .....         | 23 |
| 2.3 Microcontrolador.....                           | 25 |
| 2.3.1 Uso de los microcontroladores .....           | 26 |
| 2.3.2 Programación de Microcontroladores .....      | 26 |
| 2.3.3 Microcontroladores vs Microprocesadores ..... | 27 |
| Similitudes y comparaciones.....                    | 29 |
| 2.4 ESP32 .....                                     | 31 |
| 2.4.1 Características .....                         | 31 |
| 2.4.2 Especificaciones técnicas .....               | 32 |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 2.4.3 Lenguaje de programación .....                                          | 33 |
| 2.4.4 Espressif Systems.....                                                  | 34 |
| 2.5 NodeMCU.....                                                              | 35 |
| 2.5.1 NODEMCU-32 30-PIN ESP32 WIFI .....                                      | 36 |
| 2.5.2 Especificaciones técnicas .....                                         | 36 |
| 2.5.3 KINCONY KC868 - A4.....                                                 | 38 |
| 2.6 Arduino .....                                                             | 39 |
| 2.6.1 ¿Para qué sirve? .....                                                  | 39 |
| 2.6.3 ¿Con que se programa Arduino?.....                                      | 40 |
| 2.7 Placa de circuito impreso (PCB) .....                                     | 41 |
| 2.7.1 Interior PCB.....                                                       | 42 |
| 2.8 Componentes de hardware necesarios .....                                  | 43 |
| 2.8.1 Tabla comparativa ESP8266 vs ESP32.....                                 | 43 |
| 2.8.2 ESP8266 .....                                                           | 44 |
| 2.8.3 ESP32 .....                                                             | 45 |
| 2.9 Lenguaje de programación más adecuado .....                               | 47 |
| 2.9.1 C y C++ (Espressif IoT Development Framework - ESP-IDF) .....           | 47 |
| 2.9.2 Arduino (C++) .....                                                     | 47 |
| 2.9.3 Micropython .....                                                       | 48 |
| 2.9.4 JavaScript (Espruino) .....                                             | 48 |
| 2.9.5 Blockly .....                                                           | 49 |
| 2.10 Módulo PCF8574 .....                                                     | 50 |
| DESARROLLO.....                                                               | 51 |
| 3.1 Levantamiento de requerimientos al cliente .....                          | 52 |
| 3.2 Crear un diseño esquemático detallado del sistema de automatización ..... | 54 |
| 3.3 Configuración del entorno de trabajo .....                                | 55 |

|                                                                  |     |
|------------------------------------------------------------------|-----|
| 3.4 Programación en Simulador Wokwi .....                        | 57  |
| 3.5 Programación placa KC868 con Blynk .....                     | 65  |
| 3.6 Programación placa KC868 con WhatsApp .....                  | 72  |
| 3.7 Programación placa KC868 con Telegram .....                  | 78  |
| 3.8 Creación de la placa.....                                    | 81  |
| 3.9 Programación con placa KC868 E16S .....                      | 82  |
| RESULTADOS .....                                                 | 88  |
| 4.1 Resultados de la simulación.....                             | 89  |
| 4.2 Resultados de la programación en la placa KC868 E16S.....    | 92  |
| 4.3 Diseño de ubicación de la placa KC868 E16S en cortinas ..... | 94  |
| 4.4 Cronograma esperado para implementación.....                 | 96  |
| 4.5 Propuestas .....                                             | 97  |
| CONCLUSIONES .....                                               | 98  |
| COMPETENCIAS DESARROLLADAS.....                                  | 99  |
| FUENTES DE INFORMACIÓN .....                                     | 101 |
| ANEXOS.....                                                      | 104 |
| Anexo 1. Autorización de código.....                             | 104 |
| Anexo 2. Glosario .....                                          | 105 |
| Anexo 3. Código para simulador .....                             | 106 |
| Anexo 4. Código completo conexión Blynk con KC868 A4 .....       | 110 |
| Anexo 5. Código completo conexión WhatsApp con KC868 A4 .....    | 111 |
| Anexo 6. Código completo conexión Telegram con KC868 A4.....     | 112 |
| Anexo 7. Programa completo para placa KC868 E16S .....           | 114 |
| Anexo 8. Conexión eléctrica para placa KC868 E16S.....           | 120 |
| Anexo 9. Esquema eléctrico KC868 E16S .....                      | 121 |
| Anexo 10. Esquema PCF8574 .....                                  | 122 |

## Índice de Imágenes

|                                                                    |                                                                                                                                                                                                                              |    |
|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| IMAGEN 1 ORGANIGRAMA INSTITUCIONAL ASPI                            | FUENTE: ASPI                                                                                                                                                                                                                 | 13 |
| IMAGEN 2 CERTIFICACIONES DE LA EMPRESA                             | FUENTE: ASPI                                                                                                                                                                                                                 | 14 |
| IMAGEN 3 ALIANZAS COMERCIALES DE LA EMPRESA                        | FUENTE: ASPI                                                                                                                                                                                                                 | 14 |
| IMAGEN 4 LÍNEA DEL TIEMPO HISTORIA DEL INTERNET PARTE 1            | FUENTE: PROPIA                                                                                                                                                                                                               | 19 |
| IMAGEN 5 LÍNEA DEL TIEMPO HISTORIA DEL INTERNET PARTE 2            | FUENTE: PROPIA                                                                                                                                                                                                               | 20 |
| IMAGEN 6 LÍNEA DEL TIEMPO ANTECEDENTES DEL INTERNET DE LAS COSAS   | FUENTE: PROPIA                                                                                                                                                                                                               | 21 |
| IMAGEN 7 EJEMPLO DE MICROCONTROLADOR                               | FUENTE: <a href="https://www.geekfactory.mx/tutoriales/programacion-multitarea-en-microcontroladores/">HTTPS://WWW.GEEKFACTORY.MX/TUTORIALES/PROGRAMACION-MULTITAREA-EN-MICROCONTROLADORES/</a>                              | 27 |
| IMAGEN 8 EJEMPLO DE MICROPROCESADOR                                | FUENTE: <a href="https://resources.altium.com/es/p/all-about-cmos-ecl-and-ttl-propagation-delay-high-speed-pcbs">HTTPS://RESOURCES.ALTUM.COM/ES/P/ALL-ABOUT-CMOS-ECL-AND-TTL-PROPAGATION-DELAY-HIGH-SPEED-PCBS</a>           | 28 |
| IMAGEN 9 MICROCONTROLADOR ESP32                                    | FUENTE: <a href="https://www.digikey.com/es/product-highlight/s/schtoeta/esp32-wroom-32-wi-fi-bluetooth-module">HTTPS://WWW.DIGIKEY.COM/ES/PRODUCT-HIGHLIGHT/S/SCHTOETA/ESP32-WROOM-32-WI-FI-BLUEETOOTH-MODULE</a>           | 31 |
| IMAGEN 10 ESQUEMA NODEMCU                                          | FUENTE: <a href="https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/">HTTPS://PROGRAMARFACIL.COM/PODCAST/NODEMCU-TUTORIAL-PASO-A-PASO/</a>                                                                      | 35 |
| IMAGEN 11 ESP-WROOM-32 ESP32 ESP-32S - PLACA DE DESARROLLO         | FUENTE: <a href="https://www.amazon.com.mx/AiTrip-Esp32-3-piezas-esp-32s/dp/B08D5ZD528?th=1">HTTPS://WWW.AMAZON.COM.MX/AiTrip-Esp32-3-PIEZAS-ESP-32S/DP/B08D5ZD528?TH=1</a>                                                  | 36 |
| IMAGEN 12 KINCONY KC868 A4                                         | FUENTE: <a href="https://www.kincony.com/kc868-a4-hardware-design-details.html">HTTPS://WWW.KINCONY.COM/KC868-A4-HARDWARE-DESIGN-DETAILS.HTML</a>                                                                            | 38 |
| IMAGEN 13 EJEMPLO PCB                                              | FUENTE: <a href="https://www.freeipk.es/vector-gratis/concepto-isometrico-placa-circuito_4027496.htm">HTTPS://WWW.FREEPIK.ES/VECTOR-GRATIS/CONCEPTO-ISOMETRICO-PLACA-CIRCUITO_4027496.HTM</a>                                | 41 |
| IMAGEN 14 ESQUEMA INTERIOR DE UN PCB                               | FUENTE: <a href="https://www.mokotechnology.com/es/how-to-manufacture-multilayer-pcb/">HTTPS://WWW.MOKOTECHNOLOGY.COM/ES/HOW-TO-MANUFACTURE-MULTILAYER-PCB/</a>                                                              | 42 |
| IMAGEN 15 PLACA DE DESARROLLO NODEMCU (ESP8266)                    | FUENTE: <a href="https://www.steren.com.mx/placa-de-desarrollo-nodemcu-esp8266.html">HTTPS://WWW.STEREN.COM.MX/PLACA-DE-DESARROLLO-NODEMCU-ESP8266.HTML</a>                                                                  | 44 |
| IMAGEN 16 PLACA DE DESARROLLO ESP32 PROCESADOR DE MICROCONTROLADOR | FUENTE: <a href="https://www.amazon.com.mx/desarrollo-procesador-microcontrolador-bluetooth-integrado/dp/B07RY9MVCV">HTTPS://WWW.AMAZON.COM.MX/DESARROLLO-PROCESADOR-MICROCONTROLADOR-BLUEETOOTH-INTEGRADO/DP/B07RY9MVCV</a> | 45 |
| IMAGEN 17 PCF8574                                                  | FUENTE: <a href="https://naylampmechatronics.com/interfaz-de-usuario/359-modulo-expansor-io-pcf8574.html">HTTPS://NAYLAMPMECHATRONICS.COM/INTERFAZ-DE-USUARIO/359-MODULO-EXPANSOR-IO-PCF8574.HTML</a>                        | 50 |
| IMAGEN 18 DIAGRAMA DE FLUJO DEL PROCESO                            | FUENTE: PROPIA                                                                                                                                                                                                               | 54 |
| IMAGEN 19 PANTALLA DE DESCARGA DE ARDUINO IDE                      | FUENTE: PROPIA                                                                                                                                                                                                               | 55 |
| IMAGEN 20 INSTALACIÓN DE ARDUINO IDE                               | FUENTE: PROPIA                                                                                                                                                                                                               | 55 |
| IMAGEN 21 INSTALACIÓN DE PAQUETES COMPLEMENTARIOS                  | FUENTE: PROPIA                                                                                                                                                                                                               | 56 |
| IMAGEN 22 PAQUETE ESP32 INSTALADO                                  | FUENTE: PROPIA                                                                                                                                                                                                               | 56 |
| IMAGEN 23 PANTALLA INICIO DE SESIÓN WOKWI                          | FUENTE: PROPIA                                                                                                                                                                                                               | 57 |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| IMAGEN 24 TEMPLATES EJEMPLO DE ESP32 FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 57 |
| IMAGEN 25 PANTALLA PRINCIPAL WOKWI FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 58 |
| IMAGEN 26 CIRCUITO EN SIMULADOR WOKWI FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 58 |
| IMAGEN 27 SIMULACIÓN ACTIVA ESTADO 1 FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 63 |
| IMAGEN 28 SIMULACIÓN ACTIVA ESTADO 2 FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 63 |
| IMAGEN 29 SIMULACIÓN ACTIVA ESTADO 3 FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 64 |
| IMAGEN 30 INSTALACIÓN DE PAQUETE BLYNK FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 65 |
| IMAGEN 31 SKECTH EJEMPLO CONEXIÓN BLYNK FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 66 |
| IMAGEN 32 PANTALLA PRINCIPAL DATASTREAM FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 67 |
| IMAGEN 33 PANTALLA PRINCIPAL WEB DASHBOARD FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 67 |
| IMAGEN 34 CONFIGURACIÓN DEL WIDGET FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 68 |
| IMAGEN 35 PANTALLA PRINCIPAL NUEVO SERVICIO FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 69 |
| IMAGEN 36 INFORMACIÓN TOKEN DE BLYNK FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 69 |
| IMAGEN 37 CONEXIÓN EXITOSA A BLYNK FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 71 |
| IMAGEN 38 PANTALLA PROYECTO CREADO EN THINGESP FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 72 |
| IMAGEN 39 PANTALLA PRINCIPAL THINGESP FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 72 |
| IMAGEN 40 ENCUESTA TWILIO FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 73 |
| IMAGEN 41 PRIMER MENSAJE WHATSAPP FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 73 |
| IMAGEN 42 MENSAJE DE PRUEBA EN WHATSAPP FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 74 |
| IMAGEN 43 INFORMACIÓN CONEXIÓN CON TWILIO FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 74 |
| IMAGEN 44 PANTALLA SANDBOX CONFIGURATION TWILIO FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 75 |
| IMAGEN 45 PANTALLA PROYECTO MODIFICADO FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 75 |
| IMAGEN 46 PANTALLA PRINCIPAL THINGESP EN LÍNEA FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 77 |
| IMAGEN 47 PRUEBA CONEXIÓN CON WHATSAPP FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 77 |
| IMAGEN 48 CONEXIÓN CON TELEGRAM FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 79 |
| IMAGEN 49 RESPUESTA DE TELEGRAM FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 80 |
| IMAGEN 50 PCB DISEÑADA PARA EL PROYECTO FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 81 |
| IMAGEN 51 PLACA KC868 E16S FUENTE: <a href="https://es.aliexpress.com/item/1005004794816007.html?GPS-ID=PLATFORMRECOMMENDH5FORSPIDER&amp;PVID=2D8EC439-7269-428E-AAF9-C0A7FEF06905&amp;_T=GPS-ID:PLATFORMRECOMMENDH5FORSPIDER,PVID:2D8EC439-7269-428E-AAF9-C0A7FEF06905,TPP_BUCKETS:66">HTTPS://ES.ALIEXPRESS.COM/ITEM/1005004794816007.HTML?GPS-ID=PLATFORMRECOMMENDH5FORSPIDER&amp;PVID=2D8EC439-7269-428E-AAF9-C0A7FEF06905&amp;_T=GPS-ID:PLATFORMRECOMMENDH5FORSPIDER,PVID:2D8EC439-7269-428E-AAF9-C0A7FEF06905,TPP_BUCKETS:66</a> | 82 |
| IMAGEN 52 ORGANIZACIÓN PLACA KC868 E16S PARA EL PROYECTO FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 82 |
| IMAGEN 53 CIRCUITO PARA PRIMER CORTINA Y ALARMA GENERAL FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 87 |
| IMAGEN 54 SIMULACIÓN CORTINAS CERRADAS FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 89 |
| IMAGEN 55 SIMULACIÓN PERMISO DE APERTURA CERRADAS FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 89 |
| IMAGEN 56 SIMULACIÓN CORTINA ABIERTA CERRADAS FUENTE: PROPIA .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 90 |
| IMAGEN 57 SIMULACIÓN SEMÁFORO AMARILLO CORTINA 2 CERRADAS FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 90 |
| IMAGEN 58 SIMULACIÓN SEMÁFORO VERDE CORTINA 2 CERRADAS FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 91 |
| IMAGEN 59 SIMULACIÓN DE CORTINA ABAJO LED ROJO ENCENDIDO FUENTE: PROPIA.....                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 92 |

|                                                                |                                                                                                                                                                                                             |     |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| IMAGEN 60 SIMULACIÓN DE CORTINA ARRIBA LED VERDE ENCENDIDO     | FUENTE: PROPIA.....                                                                                                                                                                                         | 92  |
| IMAGEN 61 CORTINA 1 ABIERTA SIN PERMISO LED AMARILLO ENCENDIDO | FUENTE: PROPIA .....                                                                                                                                                                                        | 93  |
| IMAGEN 62 CORTINA 4 ABIERTA, ALERTA QUE SE APROXIMA SU CIERRE  | FUENTE: PROPIA .....                                                                                                                                                                                        | 93  |
| IMAGEN 63 DISEÑO DE UBICACIÓN PARA DOS CORTINAS                | FUENTE: PROPIA.....                                                                                                                                                                                         | 94  |
| IMAGEN 64 DISEÑO DE UBICACIÓN PARA CINCO CORTINAS              | FUENTE: PROPIA .....                                                                                                                                                                                        | 94  |
| IMAGEN 65 DISEÑO DE LA COMUNICACIÓN CON 2 PLACAS               | FUENTE: PROPIA .....                                                                                                                                                                                        | 95  |
| IMAGEN 66 CIRCUITO COMPLETO PARA 15 CORTINAS CON ALARMA        | FUENTE: PROPIA.....                                                                                                                                                                                         | 120 |
| IMAGEN 67 ESQUEMA PLACA KC868 E16S                             | FUENTE: <a href="https://www.kincony.com/forum/showthread.php?tid=2809&amp;highlight=E16S">HTTPS://WWW.KINCONY.COM/FORUM/SHOWTHREAD.PHP?TID=2809&amp;HIGHLIGHT=E16S</a> .....                               | 121 |
| IMAGEN 68 ESQUEMA PCF8574                                      | FUENTE: <a href="https://www.luisllamas.es/mas-pines-digitales-con-arduino-y-el-expansor-es-i2c-pcf8574/">HTTPS://WWW.LUISLLAMAS.ES/MAS-PINES-DIGITALES-CON-ARDUINO-Y-EL-EXPANSOR-ES-I2C-PCF8574/</a> ..... | 122 |

## Índice de Tablas

|                                                           |    |
|-----------------------------------------------------------|----|
| TABLA 1 SECTORES DE LA APLICACIÓN DE IOT .....            | 23 |
| TABLA 2 COMPONENTES CLAVES DE UN MICROCONTROLADOR .....   | 25 |
| TABLA 3 MICROCONTROLADOR VS MICROPROCESADOR .....         | 29 |
| TABLA 4 CARACTERÍSTICAS DEL MICROCONTROLADOR ESP32.....   | 31 |
| TABLA 6 ESP8266 VS ESP32.....                             | 43 |
| TABLA 7 PRIMER REQUERIMIENTO.....                         | 52 |
| TABLA 8 SEGUNDO REQUERIMIENTO .....                       | 53 |
| TABLA 9 CRONOGRAMA ESTIMADO PARA ENTREGA DE PROYECTO..... | 96 |
| TABLA 10 COMPETENCIAS DESARROLLADAS .....                 | 99 |

# **CAPÍTULO I**

## **GENERALIDADES DEL PROYECTO**

## 1.1 INTRODUCCIÓN

En el presente trabajo, se documenta el proyecto planificado y desarrollado durante la práctica de residencia profesional. Se inicia con una breve descripción de la empresa donde se realiza la residencia profesional, proporcionando información general como su razón social, a qué giro pertenece, cuál es su misión, visión y una pequeña historia de la misma.

Dentro del marco teórico, se desglosan temas relevantes para el desarrollo del proyecto seleccionado, con el fin de tener una visión más clara de las herramientas utilizadas.

Se exponen de manera detallada todas las actividades declaradas dentro del cronograma para la creación y ejecución del proyecto, presentando productos como diagramas de flujo, diagramas de circuitos eléctricos y simulaciones de circuitos eléctricos. Después de la ejecución del proyecto, se describe cuál es el resultado obtenido comparándolo con un estado anterior a la ejecución, obteniendo una respuesta negativa o positiva a la propuesta brindada.

Finalmente, se brinda una conclusión de manera general sobre la experiencia personal y profesional adquirida durante la ejecución de la residencia profesional.

## 1.2 DESCRIPCIÓN DE LA EMPRESA

### Nombre, razón social

Distribuidora de Tecnología Especializada CJ SAPI DE C.V.

### Ubicación

Calle del Laurel #1 Lt 27 colonia Los Reyes Iztacala Tlalnepantla de Baz, Estado de México  
C.P. 54090

### Sector productivo de la empresa

Sector terciario

### Misión, Visión, Filosofía y/o Valores

- **Misión:** ASPI SYSTEMS es una empresa que brinda servicios de calidad y confianza a nuestros clientes en materia de Mantenimiento Industrial y Residencial en el área de Seguridad.
- **Visión:** Ser reconocida como una empresa líder en Materia de Mantenimiento Industrial y Residencial en el área de Seguridad a Nivel Nacional, cimentada en un excelente sistema de gestión de calidad, administración de recursos económicos, técnicos y humanos, trascendiendo por la calidad de nuestro servicio.
- **Filosofía y/o Valores:** Para ASPI Systems nuestra principal prioridad es cumplir los compromisos establecidos con nuestros clientes para otorgarles una solución Integral a sus necesidades, basándonos en la gestión de calidad e innovación en los procesos de implementación y solución de problemas

## Organigrama

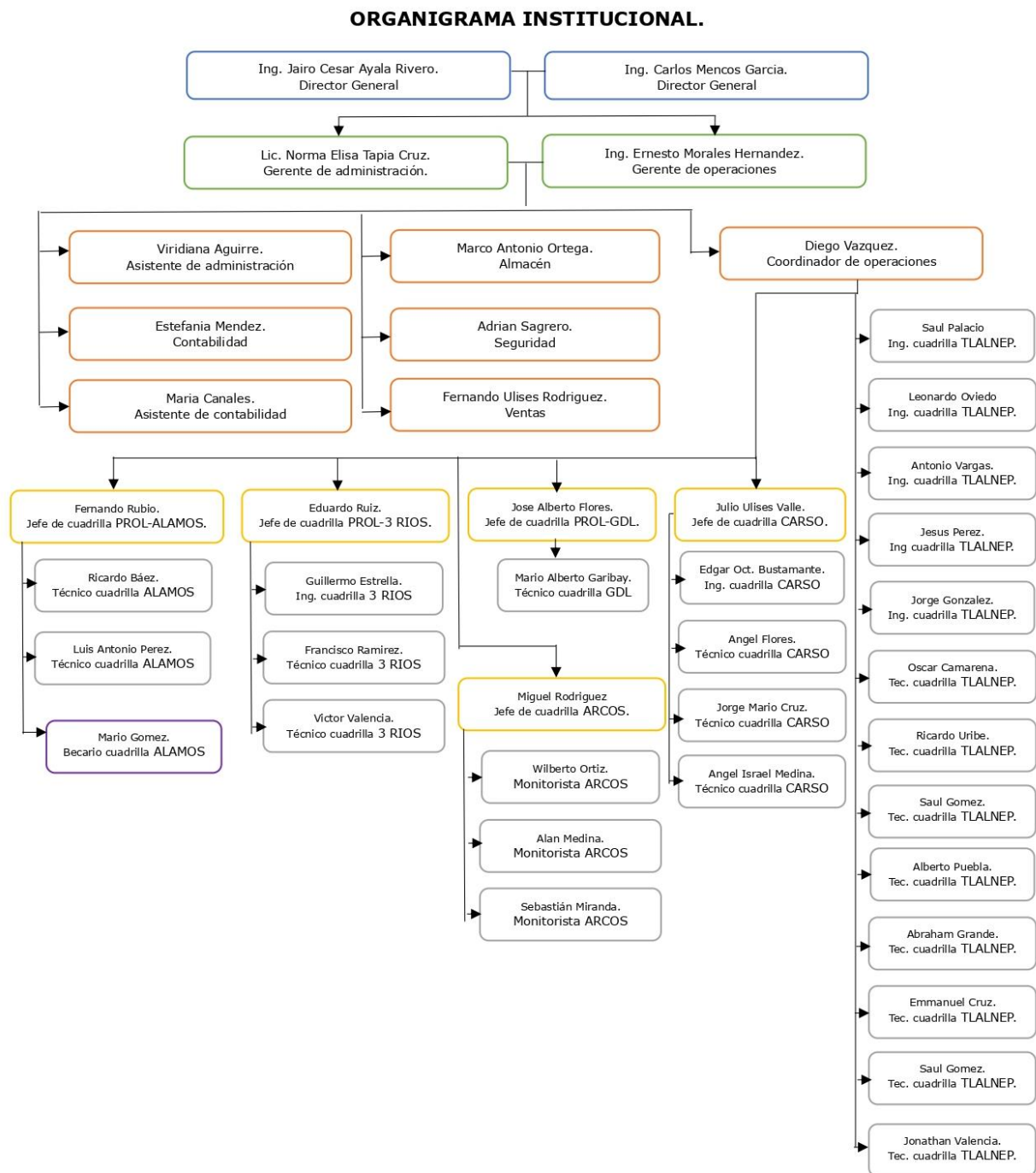


Imagen 1 Organigrama Institucional ASPI Fuente: ASPI

### Breve historia de la empresa.

Desde sus inicios nuestra empresa se ha caracterizado por mantener un trato cordial y honesto, este principio aplica en cada una de las partes involucradas en la totalidad del servicio, desde la venta al cliente pasando por la adquisición o creación del producto hasta la puesta en marcha y entrega final, asimismo, la responsabilidad de llevar hasta sus manos un servicio seguro y de calidad.

ASPI SYSTEMS cuenta con diversas certificaciones en la mayoría de las marcas Premium dentro de dichos sistemas de seguridad, como son:



Imagen 2 Certificaciones de la empresa Fuente: ASPI

Además de tener alianzas comerciales con las más importantes compañías proveedoras de equipo de seguridad a nivel mundial, para garantizar que cada proyecto cuente con el soporte de los mejores productos y de la mejor infraestructura técnica para llevar a buen fin un servicio, dichas empresas son:

|                                          |                                                                                                                                    |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>CONTROL DE ACCESO:</b>                | WINPAK – NORTHEM, HONEYWELL – NEXSENTRY, MANAGER, KERI SYSTEMS, ROSSLARE, SECURA KEY, CENTAUR, NEDAP, PCSC, TYCO - CCURE 800/8000. |
| <b>INTRUSIÓN Y SEGURIDAD PERIMETRAL:</b> | HONEYWELL, ROKONET, PYRONIX, PIMA, ADEMCO, PARADOX.                                                                                |
| <b>DETECCIÓN DE INCENDIO:</b>            | NOTIFIRE, FIRE LITE, EDWARDS SYSTEMS , TECHNOLOGY (EST).                                                                           |
| <b>CCTV:</b>                             | PELCO, SAMSUNG, HIKVISION, BOSCH, MARCH, DEDICATED MICROS, VIVOTEK.                                                                |
| <b>AUTOMATIZACIÓN:</b>                   | HONEYWELL - EXCEL 5000, ALERTON – IBEX, SCHNEIDER ELECTRIC, DELTA CONTROLS.                                                        |

Imagen 3 Alianzas comerciales de la empresa Fuente: ASPI

### **1.3 PROBLEMA A RESOLVER**

Dentro del parque industrial, a los clientes les falta una forma automatizada que les permita conocer si las cortinas industriales están disponibles u ocupadas, lo que ocasiona retrasos en el trabajo tanto para los operarios de unidades de carga pesada como para los encargados de las cortinas. Por lo tanto, se propone la creación de un sistema de automatización para el control de las naves industriales, con el objetivo de mejorar la eficiencia en el tiempo de trabajo de los clientes del parque industrial.

La ausencia de un sistema con las características mencionadas continuará generando retrasos en el proceso, lo que resultará en pérdidas de tiempo, dinero y, en algunos casos, de gasolina para los operarios de las unidades de carga. Además, también implicará una pérdida de tiempo para los encargados de las cortinas y para los propios clientes del parque industrial.

## 1.4 OBJETIVOS

### Objetivo general

Desarrollar e implementar un sistema de automatización basado en Arduino que permita el control eficiente de las cortinas en la nave industrial de Distribuidora de Tecnología Especializada CJ SAPI DE C.V.

### Objetivos específicos

1. Planificar el sistema de automatización.

#### **Actividad:**

- 1.1. Realizar el levantamiento de requerimientos al cliente.

2. Diseñar el software para el Sistema de automatización.

#### **Actividad:**

- 2.1. Crear un diseño esquemático detallado del sistema.

3. Desarrollar el software para el Sistema de automatización.

#### **Actividades:**

- 3.1. Configurar el entorno de trabajo.

- 3.2. Realizar la programación en simuladores en línea.

- 3.3. Realizar la programación en el entorno de trabajo configurado.

4. Construir un prototipo físicamente de las conexiones de las cortinas de nave industrial para el sistema de automatización.

#### **Actividad:**

- 4.1. Crear la PCB con los requisitos previamente establecidos.

5. Implementar el Sistema de automatización para el control de las cortinas.

#### **Actividad:**

- 5.1. Establecer la programación en la PCB creada.

6. Documentar el Sistema de automatización.

#### **Actividad:**

- 6.1. Realizar el informe técnico de residencias profesionales.

## **1.5 JUSTIFICACIÓN**

Ante la problemática previamente expuesta, se identifica la necesidad de una solución que mejore la eficiencia operativa y reduzca los retrasos que afectan tanto a los operarios de unidades de carga pesada como a los encargados de las cortinas industriales. Dichos retrasos conllevan pérdidas de tiempo y recursos, tanto para el personal como para los clientes, impactando negativamente en la calidad del servicio.

Al implementar el sistema propuesto, se logrará mitigar la problemática, generando beneficios significativos en términos de ahorro de tiempo, recursos financieros y combustible. Esta iniciativa redundará en un nivel superior de satisfacción para los clientes, consolidando al parque industrial como un referente destacado en el mercado.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

## 2.1 Internet

Internet es una red mundial que interconecta una amplia cantidad de datos, personas, computadoras y software que trabajan de manera cooperativa a nivel global. Funciona como un complejo entramado virtual que une computadoras de todo el mundo mediante variados medios, adoptando múltiples formas, como:

- Cables de red local.
- Cables telefónicos convencionales y/o digitales.
- Cables de fibra óptica.

La transmisión también puede ser vía satélite o a través de servicios como la telefonía celular.

Internet, sin duda, es una de las principales innovaciones del siglo XX. Desde su surgimiento, ha impulsado avances tecnológicos que continúan hasta hoy, transformando la forma en que vivimos y nos comunicamos. Para conocer un poco más sobre los acontecimientos que impulsaron estos avances, se puede incluir en una línea del tiempo eventos respaldados por los autores Timetoast (1969) y Ghostapi (2023), como se muestra a continuación.

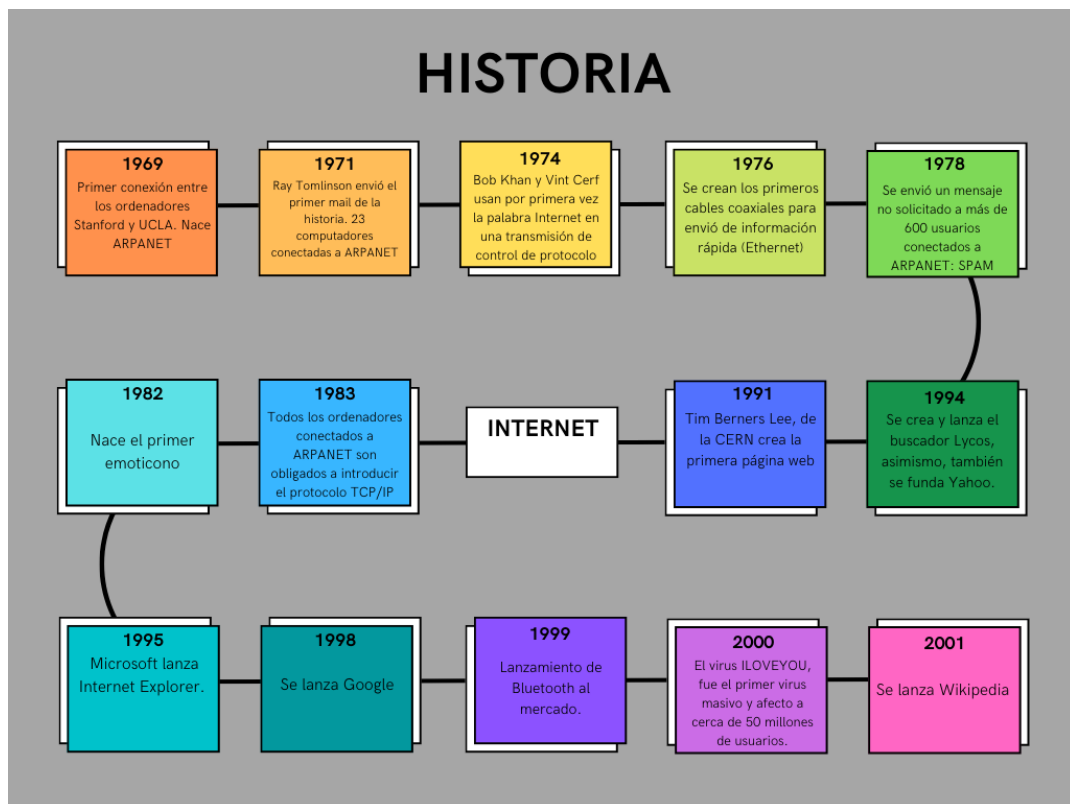


Imagen 4 Línea del tiempo historia del internet Parte 1 Fuente: Propia

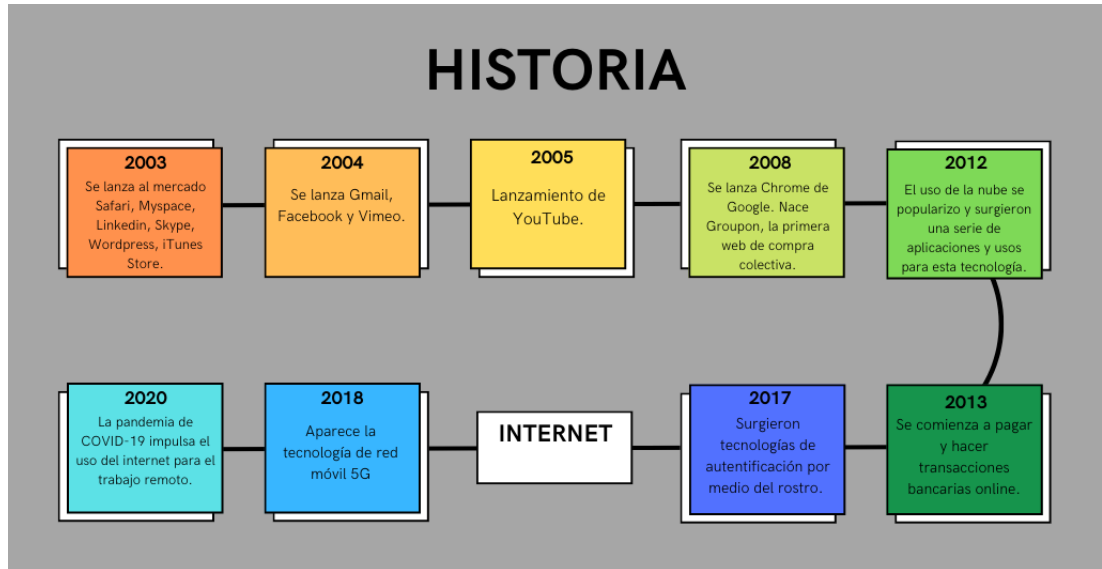


Imagen 5 Línea del tiempo historia del internet Parte 2 Fuente: Propia

El Internet ha tenido un impacto revolucionario en la sociedad y la economía en las últimas décadas. Su relevancia se manifiesta de diversas maneras, al permitir la interconexión global y el acceso a una amplia gama de información, noticias, recursos educativos y creativos. Además, ha impulsado una innovación tecnológica significativa que ha mejorado la comunicación, el comercio y la educación, transformando la vida cotidiana de las personas. La creación de Internet ha tenido un impacto masivo en todos los aspectos de la vida moderna, impulsando la innovación, la conectividad global y transformando la forma en que vivimos y hacemos negocios en la era digital.

### 2.1.1 La web

La World Wide Web (WWW o Web) constituye una amplia colección de documentos interconectados que se entrelazan para crear una red de información accesible a través de una interfaz gráfica sumamente amigable. Junto con el correo electrónico, se destaca como el servicio más popular entre los usuarios de Internet. La Web materializa la visión que durante muchos años anheló la gente: una base de datos universal con información disponible para todos, de fácil acceso y estructurada de forma que asegura una pronta recuperación de la información. Su descripción oficial la define como "una telaraña de información hipermedial que ofrece acceso a un extenso universo de documentos".

## 2.2 IOT (Internet Of Things)

El internet de las cosas o Internet of things (IOT siglas en inglés) son dispositivos que están conectados entre sí mediante el internet, estos dispositivos pueden ser electrodomésticos, maquinas industriales, sensores, casas completas y una amplia variedad de dispositivos más.

Antes de adentrarnos más en la Internet de las Cosas (IOT), es importante destacar algunos eventos que sirvieron como antecedentes para el inicio de esta tecnología.

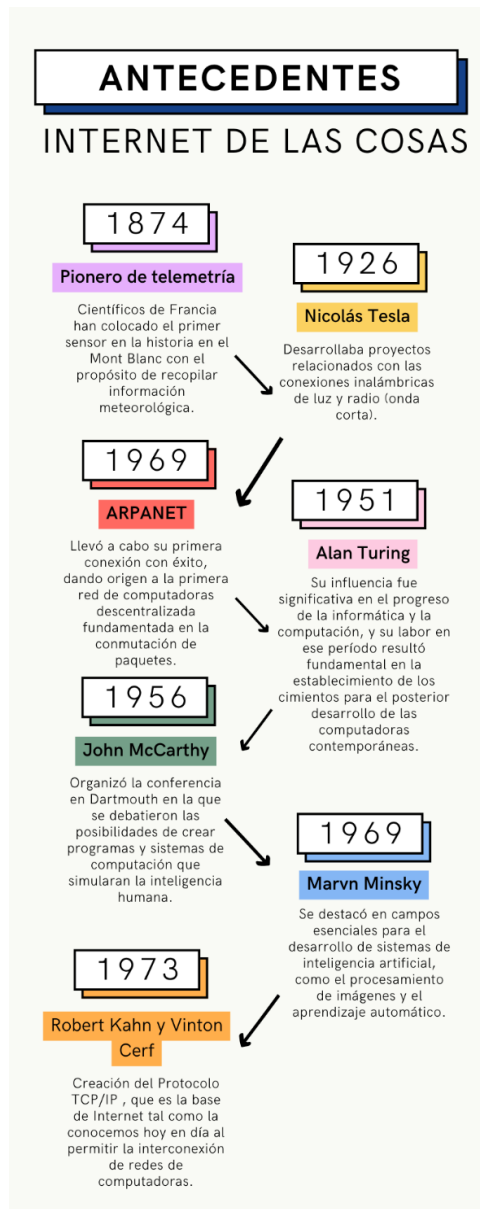


Imagen 6 Línea del tiempo Antecedentes del internet de las cosas Fuente: Propia

Con estos hitos establecidos, la primera conexión a Internet tuvo lugar en 1982 en la Universidad de Carnegie Mellon, en Pensilvania. Se trató de una máquina expendedora de refrescos que informaba sobre la cantidad de producto que tenía y la temperatura en la que se encontraba.

Un avance significativo que impulsó esta tendencia fue el desarrollo del protocolo HTTP, que sentó las bases para la World Wide Web (WWW). Este protocolo fue concebido por el científico británico Sir Timothy John Berners-Lee en 1990. En ese mismo año, en la conferencia COMDEX, se presentó una demostración notable de encender una tostadora mediante una computadora conectada a Internet, realizada por el ingeniero de software Jon Romkey.

El internet de las cosas (IOT) tuvo sus inicios en 1999 cuando el profesor Kevin Ashton, del Instituto Tecnológico de Massachusetts (MIT), presentó una conferencia sobre el tema. Sin embargo, no fue hasta 2009 que el concepto se popularizó más ampliamente gracias al artículo titulado " **That ‘Internet of Things’ thing** " en el cual se afirmaba que el IOT tenía un potencial transformador aún mayor que el propio internet para cambiar el mundo. (BBVA, 2021)

### 2.2.1 Sectores de aplicaciones de IOT

Referenciando a lo que comenta los autores Gallardo et. al. (2023), la Internet de las Cosas encuentra aplicaciones en diversos sectores. Se utiliza en soluciones de toma de decisiones automatizadas, mejora de los procesos de fabricación y cadena de suministro, y en la mejora de la seguridad de operarios y trabajadores.

*Tabla 1 Sectores de la aplicación de IOT*

| Sectores de Aplicación  |                                                                                                                                                                     |                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sector                  | ¿Para qué se utiliza?                                                                                                                                               | Ejemplos                                                                                                                                                                                    |
| Agricultura y Ganadería | Optimizar la gestión de recursos, automatizar tareas, mejorar la calidad y cantidad de la producción, y garantizar la salud de plantas y animales.                  | <ul style="list-style-type: none"> <li>➤ Monitorización de plantaciones con drones.</li> <li>➤ Automatización de riego en invernaderos.</li> </ul>                                          |
| Energía                 | Optimizar la gestión de recursos, monitorear y controlar infraestructuras eléctricas y reducir el consumo energético a través de la automatización y la eficiencia. | <ul style="list-style-type: none"> <li>➤ Supervisión y mantenimiento a distancia de plantas de energía.</li> <li>➤ Automatización y mantenimiento predictivo en aerogeneradores.</li> </ul> |
| Finanzas                | Mejorar la seguridad, automatizar transacciones, y recopilar datos en tiempo real para tomar decisiones de inversión más informadas.                                | <ul style="list-style-type: none"> <li>➤ Monitoreo remoto de plantas de energía.</li> <li>➤ Automatización y mantenimiento predictivo en aerogeneradores.</li> </ul>                        |
| Salud                   | Monitorear pacientes, administrar tratamientos, recopilar datos médicos en tiempo real y mejorar la atención médica.                                                | <ul style="list-style-type: none"> <li>➤ Monitoreo de salud con wearables conectados.</li> <li>➤ Atención hospitalaria personalizada.</li> </ul>                                            |
| Retail                  | Mejorar la gestión de inventario, ofrecer experiencias de compra personalizadas y optimizar                                                                         | <ul style="list-style-type: none"> <li>➤ Personalización de la experiencia del usuario a través de dispositivos móviles.</li> </ul>                                                         |

|                        |                                                                                                                                                                                                                         |                                                                                                                                                                                             |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | la eficiencia de la cadena de suministro.                                                                                                                                                                               | ➤ Optimización de la cadena de suministro.                                                                                                                                                  |
| Transporte y logística | Rastrear activos, mejorar la gestión de flotas, aumentar la eficiencia de la cadena de suministro y proporcionar seguimiento en tiempo real de la ubicación de productos y vehículos.                                   | <ul style="list-style-type: none"> <li>➤ Gestión de flotas y trazabilidad en tiempo real.</li> <li>➤ Gestión de almacenes y monitorización de mercancías delicadas o especiales.</li> </ul> |
| Ciudades Inteligentes  | Mejorar la gestión de servicios públicos, optimizar el tráfico, ahorrar energía y proporcionar una mejor calidad de vida a los ciudadanos a través de la recopilación de datos y la automatización de sistemas urbanos. | <ul style="list-style-type: none"> <li>➤ Estacionamiento e iluminación inteligentes.</li> <li>➤ Mejora de la seguridad ciudadana.</li> </ul>                                                |

## 2.3 Microcontrolador

Los microcontroladores son componentes esenciales en la electrónica moderna, desempeñando un papel fundamental en una amplia variedad de aplicaciones. Estos diminutos circuitos integrados incluyen una unidad central de procesamiento (CPU), memoria, puertos de entrada/salida (E/S) y otros elementos, y se emplean para controlar y automatizar sistemas en diversas industrias, que van desde la electrónica de consumo hasta la industria automotriz y la robótica.

Los microcontroladores son dispositivos electrónicos programables diseñados para llevar a cabo tareas específicas o controlar sistemas en tiempo real. A diferencia de las computadoras personales, que ejecutan una variedad de aplicaciones, los microcontroladores están diseñados para tareas específicas y constituyen la "inteligencia" detrás de muchos dispositivos electrónicos que usamos en nuestra vida cotidiana. (Pérez & Areny, 2007)

*Tabla 2 Componentes claves de un microcontrolador*

| Componentes claves                       |                                                                                                             |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| CPU                                      | Es la unidad central de procesamiento ejecuta instrucciones y realiza cálculos.                             |
| Memoria                                  | Esta almacena el programa (código) y datos temporales.                                                      |
| Puertos de E/S                           | Son conexiones para interactuar con el mundo exterior, como sensores, actuadores y dispositivos de usuario. |
| Temporizadores y contadores              | Se utilizan para controlar el tiempo y realizar mediciones de eventos.                                      |
| Convertidores Analógicos-digitales (ADC) | Sirven para interactuar con señales analógicas y digitales.                                                 |
| Periféricos específicos del fabricante   | Características adicionales como comunicación serial, PWM (modulación por ancho de pulso), etc.             |

### 2.3.1 Uso de los microcontroladores

Los microcontroladores desempeñan un papel esencial en una amplia gama de aplicaciones, abarcando desde dispositivos de consumo y electrónica integrada hasta aplicaciones industriales y educativas. Su capacidad para controlar sistemas en tiempo real y su versatilidad los convierten en una herramienta inestimable para la automatización y el control en el mundo contemporáneo. La constante evolución tecnológica continuará ampliando su alcance y sus aplicaciones.

Se utilizan en campos como:

- Electrónica de consumo
- Automoción
- Industria
- electrónica embebida
- Educación y experimentación

### 2.3.2 Programación de Microcontroladores

Programar microcontroladores involucra la creación de código, usualmente se utilizan lenguajes como C o C++, que establece cómo el microcontrolador interactúa con su entorno. Los programas pueden controlar una variedad de elementos, incluyendo luces, motores, sensores y más. Los microcontroladores se pueden programar directamente en el dispositivo o a través de un entorno de desarrollo en una computadora.

Los microcontroladores tienen una gran importancia, ya que posibilitan la automatización, el control preciso y la eficiencia en una amplia variedad de aplicaciones. Además, contribuyen a la seguridad, la interactividad, la conectividad y la conservación de energía. Su versatilidad y capacidad para adaptarse a diversas tareas los convierten en componentes fundamentales en el ámbito de la electrónica moderna.

### 2.3.3 Microcontroladores vs Microprocesadores

Según AWS (s. f.) Comparación entre un microcontrolador y un microprocesador. los microcontroladores y los microprocesadores son dos tipos de unidades de procesamiento utilizados en sistemas electrónicos, aunque presentan diferencias significativas en cuanto a diseño, aplicaciones y capacidades.



Imagen 7 Ejemplo de microcontrolador Fuente: <https://www.geekfactory.mx/tutoriales/programacion-multitarea-en-microcontroladores/>

Los microcontroladores son chips altamente integrados que incluyen una unidad central de procesamiento (CPU), memoria, periféricos de entrada/salida y, en muchas ocasiones, memoria de programa incorporada en un solo paquete. Están diseñados para llevar a cabo tareas específicas y controlar sistemas en tiempo real.

Cuenta con características como:

- **Integración:** Los microcontroladores incluyen CPU, memoria RAM, memoria ROM, puertos de E/S y otros componentes en un solo chip, lo que los hace altamente integrados y compactos.
- **Tareas específicas:** Los microcontroladores se utilizan para realizar tareas específicas.
- **Aplicaciones embebidas:** Se encuentran comúnmente en dispositivos electrónicos embebidos, como electrodomésticos, sistemas de automoción, dispositivos médicos, juguetes electrónicos y más.
- **Baja potencia y eficiencia:** Están diseñados para funcionar con bajo consumo de energía, aptos para dispositivos con batería o en sistemas que deben estar en funcionamiento las 24 horas del día.
- **Programación específica:** Cada microcontrolador tiene su propio conjunto de instrucciones y arquitectura, lo que requiere programación específica para el dispositivo en cuestión.

Por otro lado, los microprocesadores son la "unidad central" de una computadora y se emplean para realizar cálculos y ejecutar programas y a diferencia de los microcontroladores, los

microprocesadores no están destinados a ejecutar tareas específicas ni a controlar hardware en tiempo real, las características más importantes de los microprocesadores son los siguientes:



*Imagen 8 Ejemplo de microprocesador Fuente: <https://resources.altium.com/es/p/all-about-cmos-ecl-and-ttl-propagation-delay-high-speed-pcbs>*

- **CPU pura:** Se centran en la ejecución de instrucciones de software y no contienen periféricos de entrada/salida ni memoria de programa incorporada.
- **Tareas generales:** Se utilizan en sistemas de propósito general, como computadoras personales, servidores y estaciones de trabajo, para ejecutar una amplia variedad de aplicaciones y tareas.
- **Alta potencia de procesamiento:** Están diseñados para ofrecer un alto rendimiento en cálculos matemáticos y de propósito general.
- **Requieren hardware adicional:** Para interactuar con el entorno, los microprocesadores dependen de placas base, controladores, memoria y otros componentes externos.
- **Variedad de sistemas operativos:** Los microprocesadores son comunes en sistemas que ejecutan sistemas operativos completos, como Windows, macOS y Linux.

## Similitudes y comparaciones

Tabla 3 Microcontrolador vs Microprocesador

| Similitudes                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Microcontrolador                                                                                                                                                                                                                                                                                                                                                             | Microprocesador                                                                                                                                                                           |
| Son componentes semiconductores incorporados en un circuito integrado. Un circuito integrado es una diminuta placa cuadrada o rectangular que contiene miles o incluso millones de componentes electrónicos.                                                                                                                                                                 |                                                                                                                                                                                           |
| Incluyen una Unidad Central de Procesamiento (CPU). La CPU es la parte central del chip de la computadora que ejecuta las instrucciones proporcionadas por las aplicaciones o el firmware. La CPU también cuenta con una unidad aritmética lógica (ALU) especializada que realiza cálculos matemáticos y evalúa problemas lógicos según las instrucciones de la computadora. |                                                                                                                                                                                           |
| Los registros son unidades de memoria que la CPU utiliza para el procesamiento. Se componen de registros internos, aunque es común que los microprocesadores dispongan de más registros en comparación con los microcontroladores.                                                                                                                                           |                                                                                                                                                                                           |
| Comparación                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                           |
| Microcontrolador                                                                                                                                                                                                                                                                                                                                                             | Microprocesador                                                                                                                                                                           |
| Se utiliza la arquitectura Harvard                                                                                                                                                                                                                                                                                                                                           | Se utiliza la arquitectura Von Neumann                                                                                                                                                    |
| Contiene menos componentes eléctricos                                                                                                                                                                                                                                                                                                                                        | Contiene más componentes eléctricos                                                                                                                                                       |
| Están contruidos con memorias ROM y RAM internas, además de utilizar un bus interno para interactuar con los módulos de memoria integrados.                                                                                                                                                                                                                                  | No tienen módulos de memoria interna para almacenar los datos de las aplicaciones, se deben conectar a los almacenamientos de memoria externos, como la ROM y la RAM, con un bus externo. |
| Se conecta a los periféricos del chip mediante un bus de control interno. Permitiendo controlar los dispositivos con menos piezas adicionales.                                                                                                                                                                                                                               | No tiene periféricos integrados en su circuito integrado, los periféricos se deben conectar de manera externa para ampliar los casos de uso.                                              |
| Tienen una potencia de procesamiento menor impidiendo cálculos de punto                                                                                                                                                                                                                                                                                                      | Tienen una potencia de procesamiento mayor, que sirve para realizar tareas matemáticas y computacionales complejas.                                                                       |

|                                                                                                                                             |                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| flotante, pero se implementa una lógica específica.                                                                                         |                                                                                                        |
| Su velocidad se oscila entre Kilohercios (KHz) y cientos de megahercios (MHz).                                                              | Funciona con un rango de gigahercios (GHz).                                                            |
| Están diseñados para que trabajar de manera eficiente con mínima potencia, además de contar con un sistema de ahorro de energía (No todos). | Funcionan a una velocidad mayor por lo que gastan más energía; requiere una fuente de energía externa. |
| No requiere de S.O.                                                                                                                         | Requieren de un S.O. que proporcione las funcionalidades adecuadas.                                    |
| Requieren de un procesador especial para la conectividad de datos de alta velocidad.                                                        | Manejan tecnologías de comunicación diversas.                                                          |

## 2.4 ESP32

El ESP32 es un microcontrolador económico y de bajo consumo de energía que pertenece a la familia de dispositivos ESP (Espressif System Platform) desarrollada por la empresa china Espressif Systems. Es ampliamente utilizado en aplicaciones de Internet de las Cosas (IoT) y proyectos de desarrollo de hardware y software debido a su versatilidad y capacidades.

ESP32 es un dispositivo que pequeño y portátil que permite estar monitorizando y actuando mientras estás conectado a internet. (ESP32 ¿Qué es y para qué sirve? - programador novato, 2022)



Imagen 9 Microcontrolador ESP32 Fuente: <https://www.digikey.com/es/product-highlight/s/schtoeta/esp32-wroom-32-wi-fi-bluetooth-module>

### 2.4.1 Características

En relación con las características de la placa ESP32, Morales (2022) señala que:

Tabla 4 Características del microcontrolador ESP32

|                           |                                                                                                                                                                                                                                                                                                                                  |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Diseño robusto            | Opera en entornos hostiles, con un rango de temperatura de funcionamiento que va desde -40 °C hasta +125 °C. Gracias a circuitos de calibración avanzados, tiene la capacidad de protegerse contra imperfecciones en los circuitos externos que podrían dañarlo                                                                  |
| Bajo Consumo de energía   | Tiene un bajo consumo de energía. Además, incorpora características como la activación de reloj de grano fino, múltiples modos de gestión de energía y escalado dinámico de potencia.                                                                                                                                            |
| Alto nivel de integración | Está altamente integrado con interruptores de antena incorporados, balun de RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros y módulos de gestión de energía. Aporta funcionalidad y versatilidad significativas a las aplicaciones con requerimientos mínimos de placa de circuito impreso (PCB). |
| Wi-Fi y Bluetooth híbrido | Puede operar como un sistema independiente completo o como un dispositivo esclavo de una MCU anfitriona, lo que disminuye la                                                                                                                                                                                                     |

|                                      |                                                                                                                                                                                                                              |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                      | carga en la pila de comunicación del procesador de la aplicación principal. Además de poder comunicarse con otros sistemas para brindar funcionalidades de Wi-Fi y Bluetooth a través de sus interfaces SPI/SDIO o I2C/UART. |
| Integración con Arduino              | Muchas de sus funciones se asemejan con Arduino, e incluso puede trabajar con el entorno de desarrollo Arduino IDE.                                                                                                          |
| Diferentes lenguajes de programación | Puede ser programado en distintos lenguajes de programación, como: <ul style="list-style-type: none"> <li>➤ (C) Arduino IDE</li> <li>➤ (Python) MicroPython</li> <li>➤ LUA</li> </ul>                                        |

#### 2.4.2 Especificaciones técnicas

- Procesador Tensilica Xtensa 32bits LX6 hasta 240MHz.
- Wi-Fi: 802.11b/g/n/e/i (802.11n @ 2.4 Ghz hasta 150 Mbit/s).
- Bluetooth: v4.2 BR/EDR y bluetooth Low Energy (BLE).
- Rom:448 KiB.
- SRAM: 520 KiB.
- RTC slow SRAM: 8 KiB.
- RTC fast SRAM: 8 KiB.
- eFuse: 1 Kbit.
- Flash embebida: 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips); 2 MiB (ESP32-D2WD chip); 4 MiB (ESP32-PICO-D4 SIP module).
- Periféricos compatibles: ADC, DAC, I2C, UART, Interfaz CAN 2.0, SPI, I2S, RMII y PWM entre otros.
- Seguridad tipo IEEE 802.11, WFA, WPA/WPA2 y WAPI.
- Encriptación de memoria Flash.
- Criptografía soportada por acelerador de hardware: AES, SHA-2, RSA, ECC, RNG.
- Voltaje de trabajo 3.3VDC.
- Energía y datos via conector microUSB 5VDC.

### **2.4.3 Lenguaje de programación**

#### **2.4.3.1 (C) Arduino IDE**

El lenguaje de programación de Arduino es una versión simplificada de C/C++ diseñada para facilitar la programación de microcontroladores Arduino, es decir que sea más accesible para una amplia gama de usuarios, desde principiantes hasta programadores experimentados.

Es una excelente opción para desarrollar proyectos de hardware y electrónica debido a su facilidad de uso y a la disponibilidad de una gran comunidad y recursos en línea que pueden ayudarte a aprender y resolver problemas. (Fundación Aquae, 2020)

#### **2.4.3. 2 (Python) MicroPython**

Python es un lenguaje de programación potente y de sencillo aprendizaje. Ofrece estructuras de datos de alto nivel eficientes y un sistema de programación orientado a objetos simple pero efectivo. Su elegante sintaxis, tipado dinámico y naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y para múltiples plataformas.

Fue creado por Guido van Rossum y lanzado por primera vez en 1991, se utiliza en una amplia variedad de aplicaciones, incluyendo desarrollo web, análisis de datos, inteligencia artificial, aprendizaje automático, scripting, automatización, desarrollo de videojuegos y mucho más. (Gutiérrez, 2022)

#### **2.4.3.3 LUA**

Lua, que significa "luna" en portugués, es un lenguaje de programación extensible que se ha diseñado para soportar la programación procedimental general, además de ofrecer utilidades para la descripción de datos. También proporciona un sólido soporte para la programación orientada a objetos y la programación funcional. El propósito de Lua es servir como un lenguaje de scripting poderoso y liviano que pueda integrarse en cualquier programa que lo requiera. Lua se implementa como una biblioteca escrita en C limpio, lo que significa que está en conformidad con el subconjunto común de ANSI C y C++. (*Manual de referencia de LUA 5.1*, s. f.)

### 2.4.4 Espressif Systems

Espressif Systems es una empresa multinacional que se estableció en 2008 y tiene presencia en China, República Checa, India, Singapur y Brasil. Cuenta con un apasionado equipo de ingenieros y científicos de diversas partes del mundo, enfocados en el desarrollo de soluciones de bajo consumo para la Inteligencia Artificial en el Internet de las Cosas (AIoT), así como en tecnología de vanguardia en Wi-Fi y Bluetooth. Espressif es conocida por crear las populares series de chips, módulos y placas de desarrollo ESP8266, ESP32 y ESP32-S, que ofrecen soluciones seguras, robustas y eficientes en términos de energía. Además, mediante el uso de tecnología y soluciones de código abierto, Espressif tiene como objetivo permitir a los desarrolladores de todo el mundo utilizar sus soluciones y construir sus propios dispositivos inteligentes conectados. En julio de 2019, Espressif llevó a cabo su Oferta Pública Inicial en la Junta de Innovación Sci-Tech (STAR) de la Bolsa de Valores de Shanghai (SSE). (*About Espressif* | *Espressif Systems*, s. f.)

Espressif tiene un firme compromiso con la investigación y desarrollo de tecnología sostenible. Su enfoque está en la creación de soluciones AIoT que contribuyen a la reducción del consumo de energía y el desperdicio de material. Para lograrlo, diseñan chipsets de bajo consumo altamente integrados que minimizan la necesidad de componentes adicionales, como resistencias, condensadores, inductores, interruptores, baluns y chips de gestión de energía, lo que a su vez reduce el desperdicio en las placas de circuito impreso. (*Espressif*, s. f.)

## 2.5 NodeMCU

NodeMCU es una placa de desarrollo completamente abierta, tanto en términos de software como de hardware. Al igual que en Arduino, en NodeMCU se han implementado todas las herramientas necesarias para facilitar la programación de un microcontrolador o MCU (del inglés Microcontroller Unit).

Es importante destacar que un microcontrolador y una placa de desarrollo son dos cosas distintas. NodeMCU no es un microcontrolador en sí mismo. En realidad, se trata de placas o kits de desarrollo que incorporan un chip conocido como SoC (System on a Chip), que incluye un microcontrolador o MCU en su interior.

Se puede esquematizar de la siguiente manera:

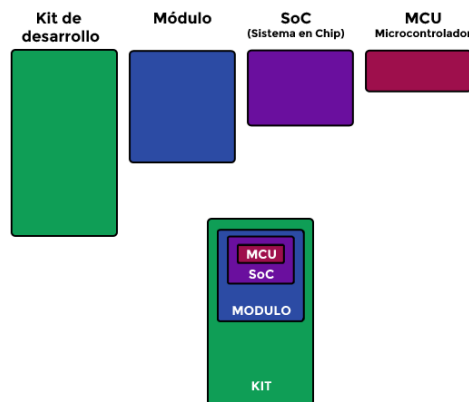
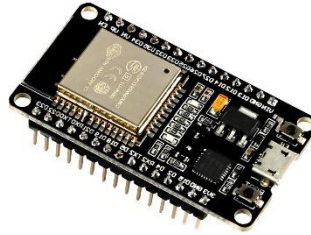


Imagen 10 Esquema NodeMCU Fuente: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

### 2.5.1 NODEMCU-32 30-PIN ESP32 WIFI



*Imagen 11 ESP-WROOM-32 ESP32 ESP-32S - Placa de desarrollo Fuente: <https://www.amazon.com.mx/AiTrip-Esp32-3-piezas-ESP-32S/dp/B08D5ZD528?th=1>*

Es una potente herramienta de desarrollo para la creación rápida de prototipos en proyectos de IoT (Internet de las cosas). Esta placa integra el módulo SoM ESP-WROOM-32, que se basa en el SoC ESP32, además del convertidor USB-serial CP2102 necesario para la programación del ESP32 a través de USB, reguladores de voltaje y LED indicadores. La plataforma ESP32 representa una evolución del ESP8266, mejorando sus capacidades de comunicación y capacidad de procesamiento. En cuanto a conectividad, permite utilizar varios protocolos de comunicación inalámbrica, incluyendo WiFi, Bluetooth y BLE. (NodeMCU-32 30-pin ESP32 WiFi, s. f.)

### 2.5.2 Especificaciones técnicas

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- Placa: ESP32 DEVKIT V1 (Espressif)
- SoM: ESP-WROOM-32 (Espressif)
- SoC: ESP32 (ESP32-D0WDQ6)
- CPU: Dual-Core Tensilica Xtensa LX6 (32 bit)
- Frecuencia de Reloj: hasta 240Mhz
- Desempeño: Hasta 600 DMIPS
- Procesador secundario: Permite hacer operaciones básicas en modo de ultra bajo consumo
- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- Memoria:
- 448 KByte ROM
- 520 KByte SRAM
- 16 KByte SRAM in RTC

- QSPI Flash/SRAM, 4 MBytes
- Pines: 30
- Pines Digitales GPIO: 24 (Algunos pines solo como entrada)
- Pines PWM: 16
- Pines Analógicos ADC: 18 (3.3V, 12bit: 4095, tipo SAR, ganancia programable)
- Conversor Digital a Analógico DAC: 2 (8bit)
- UART: 2
- Chip USB-Serial: CP2102
- Antena en PCB
- Seguridad: Estándares IEEE 802.11 incluyendo WFA, WPA/WPA2 and WAPI 1024-bit OTP, up to 768-bit for customers
- Aceleración criptográfica por hardware: AES, HASH (SHA-2), RSA, ECC, RNG
- Dimensiones: 55\*28 mm

### 2.5.3 KINCONY KC868 - A4

Es una placa de relés de control y automatización diseñada para aplicaciones de domótica y automatización en el hogar o en entornos industriales. Esta placa se utiliza para controlar dispositivos y sistemas eléctricos mediante relés, lo que la hace adecuada para una amplia variedad de aplicaciones (Admin & Admin, 2022)

Cuenta con características como:

- **Relés de conmutación:** Incluye 4 relés de conmutación independientes. Cada uno de estos relés se puede utilizar para controlar dispositivos eléctricos, como luces, persianas, electrodomésticos, sistemas de riego, etc.
- **Control remoto:** La placa se puede controlar de forma remota a través de una red local o, en algunos casos, a través de Internet. Esto permite controlar dispositivos y sistemas desde un teléfono inteligente, una tableta o una computadora.
- **Compatibilidad con sensores:** Puede estar diseñada para admitir sensores, lo que permite que los dispositivos se activen o desactiven en respuesta a condiciones específicas, como cambios de luz, temperatura, movimiento, etc.
- **Escalabilidad:** Algunas versiones de la KC868-A4 pueden ser escalables, lo que significa que es posible agregar más relés o dispositivos de control según sea necesario.
- **Seguridad:** Suelen estar diseñadas con medidas de seguridad para proteger contra sobrecargas y cortocircuitos, lo que garantiza un funcionamiento seguro.

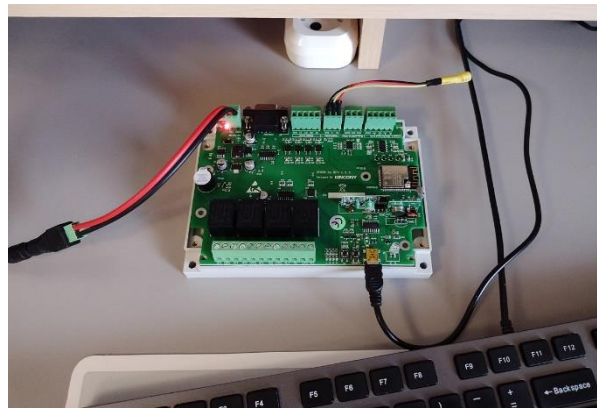


Imagen 12 KINCONY KC868 A4 Fuente: <https://www.kincony.com/kc868-a4-hardware-design-details.html>

## 2.6 Arduino

Arduino es una plataforma de electrónica de código abierto que se rige por los principios de contar con software y hardware fáciles de usar. En esencia, esta herramienta permite la creación de diversos tipos de microcontroladores en una sola placa, los cuales pueden adaptarse para una amplia variedad de usos según las necesidades del creador. En otras palabras, proporciona una forma sencilla de desarrollar proyectos interactivos para cualquier persona, por lo que es una de las placas más populares ya que permite tener versatilidad al trabajar con ella.

Tuvo sus comienzos en el año 2005 como un proyecto dirigido a estudiantes en el Instituto IVREA, en Ivrea, Italia. La idea surgió de la necesidad de tener un equipo que fuera más económico para estos, ya que utilizaban el microcontrolador BASIC Stamp, el cual tenía un costo de 100 dólares estadounidenses, lo que resultaba costoso para ellos.

El nombre del proyecto proviene del Bar di Re Arduino (Bar del Rey Arduino), donde Massimo Banzi (uno de los fundadores de Arduino) solía pasar tiempo. El rey Arduino gobernó Italia entre los años 1002 y 1014, de ahí el origen del nombre. (Fundación Aquae, 2020)

### 2.6.1 ¿Para qué sirve?

Arduino se puede emplear para desarrollar dispositivos autónomos o para conectarlos a otros dispositivos y comunicarse con otros programas. De esta manera, es posible lograr interacciones tanto con el hardware como con el software.

Su uso se puede clasificar en 2 grandes grupos, el primero, puede funcionar como un dispositivo independiente, descargando un programa a este desde una computadora, interactuando con la realidad y tomando decisiones según lo programado; Arduino monitorea constantemente la humedad del suelo a través de un sensor. Basado en esta lectura y un programa preconfigurado, este decide si es necesario el riego cuando la humedad está por debajo de un umbral. En caso afirmativo, activa una bomba de agua para regar el jardín, controlando la cantidad y duración del riego según el programa. Una vez que la humedad deseada se alcanza, el Arduino detiene la bomba. Además, puede registrar datos de riego y generar alertas según lo programado, actuando de forma autónoma al tomar decisiones.

El segundo, funciona con una interfaz entre una computadora u otro dispositivo y puede ejecutar alguna tarea especificada, para traducirla a una acción en físico y viceversa; Una persona dentro de su casa tiene una aplicación en su computadora desde la cual puede controlar las luces de

una habitación. A través de la interfaz de la aplicación, el usuario puede ajustar el brillo, el color y el encendido/apagado de las luces.

### 2.6.3 ¿Con que se programa Arduino?

Arduino es una plataforma de hardware de código abierto que ha ganado una gran popularidad en el ámbito de la electrónica y la informática. A menudo, surge la pregunta sobre el lenguaje de programación utilizado en Arduino, ya que no es tan evidente como en algunos lenguajes de programación tradicionales. En este artículo, exploraremos el lenguaje de programación que se emplea en el entorno de desarrollo integrado (IDE) de Arduino y cómo este lenguaje facilita la programación de microcontroladores y la creación de proyectos electrónicos.

El lenguaje de programación principal utilizado en Arduino se basa en C y C++. Esto significa que, si tienes experiencia en programación en C o C++, te sentirás más cómodo programando en Arduino. Sin embargo, el Arduino IDE proporciona una capa de abstracción que simplifica muchos aspectos de la programación, lo que lo hace más accesible incluso para aquellos que no tienen experiencia previa en programación.

Los programas de Arduino siguen una estructura básica que consta de dos funciones principales: `setup()` y `loop()`. La función `setup()` se ejecuta una sola vez al inicio del programa y se utiliza para configurar variables, pines y realizar configuraciones iniciales. Por otro lado, la función `loop()` se ejecuta de manera continua en un bucle y es donde se coloca el código principal del programa. Esta estructura simplifica la programación de aplicaciones que requieren realizar tareas repetitivas de manera eficiente.

## 2.7 Placa de circuito impreso (PCB)

El uso de las PCB fueron un paso de gigante en la evolución de los dispositivos electrónicos, ya que proporcionó un método innovador para la conexión de elementos sin necesidad de utilizar cables eléctricos.

PCB son las siglas de Placa de Circuito Impreso en inglés (Printed Circuit Board) estas son un soporte físico en donde se instalan componentes electrónicos y eléctricos y se interconectan entre ellos, estos pueden ser tanto chips, como diodos, resistencias, etc. La forma más común de encontrarlas actualmente es dentro de nuestras computadoras, son las placas que conecta los componentes electrónicos utilizando rutas conductoras grabadas en la placa. (Peterson, 2023)

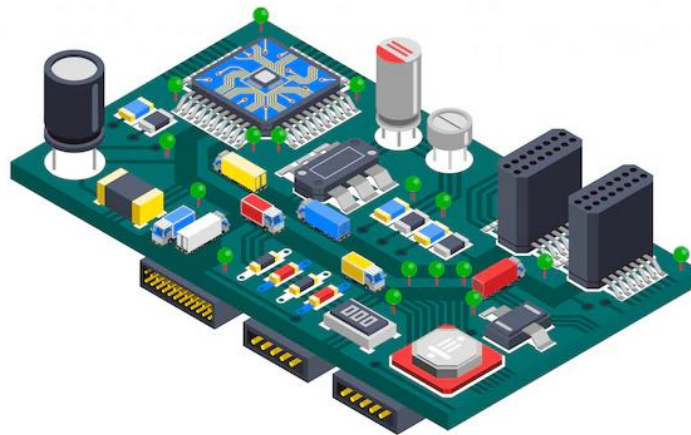


Imagen 13 Ejemplo PCB Fuente: [https://www.freepik.es/vector-gratis/concepto-isometrico-placa-circuito\\_4027496.htm](https://www.freepik.es/vector-gratis/concepto-isometrico-placa-circuito_4027496.htm)

Para conectar cada elemento en una PCB, se emplean pistas conductoras de cobre extremadamente finas que forman carriles conductores, simulando una conexión con un cable o jumper. En circuitos simples, estas pistas pueden observarse en una o ambas caras visibles de la PCB, mientras que en circuitos más complejos se pueden ver pistas eléctricas y componentes apilados en múltiples capas.

Las PCB se componen principalmente de una combinación de fibra de vidrio reforzada con materiales cerámicos, resinas, plástico y otros elementos no conductores. Actualmente, se utilizan también componentes como celuloide y pistas de pintura conductora para fabricar PCB flexibles.

La primera placa de circuito integrado fue construida a mano en 1936 por el ingeniero Paul Eisler para ser utilizada en una radio. A partir de entonces, los procesos se automatizaron para la fabricación a gran escala, primero en radios y luego en una variedad de componentes electrónico.

### 2.7.1 Interior PCB

Las PCB están conformadas por múltiples capas conductoras, cada una separada por un material aislante conocido como sustrato. Para conectar pistas de diferentes capas, se utilizan orificios llamados vías, que pueden atravesar completamente la PCB o llegar a una profundidad determinada.

El sustrato puede tener diversas composiciones, pero siempre está hecho de materiales no conductores para asegurar que cada pista eléctrica transporte su propia señal y voltaje. El material más comúnmente utilizado es el Pértinax, que es esencialmente papel recubierto de resina. Para equipos de alto rendimiento, se emplea un compuesto llamado FR-4, que consiste en fibra de vidrio cubierta de resina resistente al fuego.

Los componentes electrónicos suelen ubicarse en la zona externa de las PCB, instalados en ambas caras para aprovechar al máximo la extensión de estas. Antes de crear las pistas eléctricas, las diferentes capas de la PCB solo constan del sustrato y láminas muy delgadas de cobre u otro material conductor. La creación de las pistas eléctricas se realiza mediante una máquina similar a una impresora, a través de un proceso largo y complejo. (Castillo, 2023)

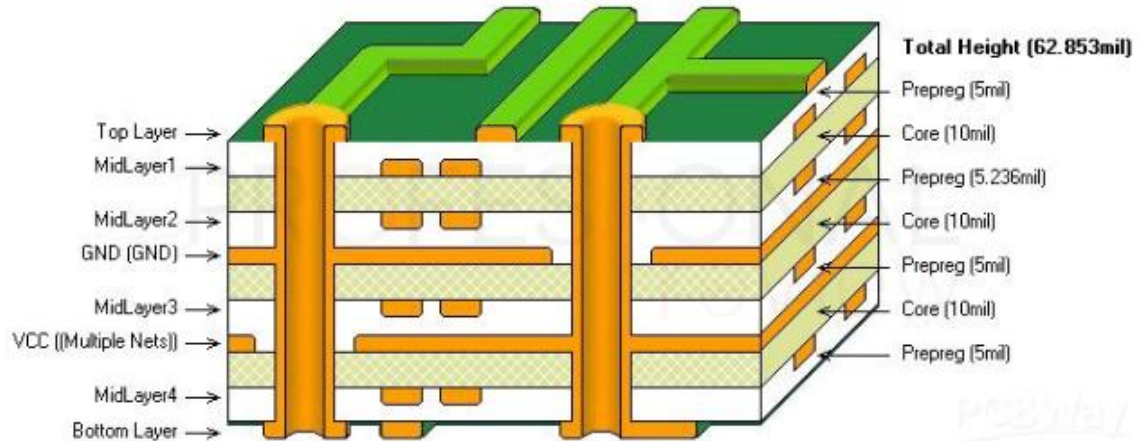


Imagen 14 Esquema interior de un PCB Fuente: <https://www.mokotechnology.com/es/how-to-manufacture-multilayer-pcb/>

## 2.8 Componentes de hardware necesarios

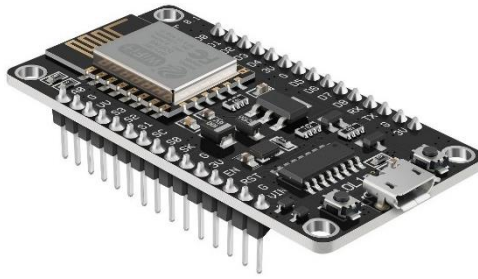
Para realizar el proyecto se tiene dos opciones de microcontrolador el ESP32 y el ESP8266 ambos son desarrollados por la empresa Espressif Systems, en China. Estos son utilizados en el mundo de la electrónica y la programación de sistemas embebidos, generalmente se utilizan para proyectos de Internet de las cosas (IOT) gracias a su integración de conectividad a Wifi, a continuación, se compartirá la información recabada de cada uno y cuál fue la elección para el proyecto.

### 2.8.1 Tabla comparativa ESP8266 vs ESP32

Tabla 5 ESP8266 vs ESP32

| Características                        | ESP8266                        | ESP32                                     |
|----------------------------------------|--------------------------------|-------------------------------------------|
| Microprocesador                        | Xtensa Single-core 32-bit L106 | Xtensa Dual-Core 32-bit LX6 con 600 DMIPS |
| Wi-Fi (802.11 b/g/n)                   | HT2                            | HT40                                      |
| Bluetooth                              | No posee                       | Bluetooth 4.2 y BLE                       |
| Frecuencia de operación (Valor típico) | 80 MHz                         | 160 MHz                                   |
| SRAM                                   | No posee                       | 448 KB                                    |
| Flash                                  | No posee                       | 520 KB                                    |
| Numero de GPIO                         | 17                             | 34                                        |
| PWM (Hardware)                         | No posee                       | No posee                                  |
| PWM (Software)                         | 8 canales                      | 16 canales                                |
| SPI                                    | 2                              | 4                                         |
| I2C                                    | 1                              | 2                                         |
| I2S                                    | 2                              | 2                                         |
| UART                                   | 2                              | 2                                         |
| ADC                                    | 10-bits de resolución          | 12-bits de resolución                     |
| CAN                                    | No                             | Si                                        |
| Interfaz MAC Ethernet                  | No                             | Si                                        |
| Sensor de tacto                        | No                             | Si                                        |
| Sensor de temperatura                  | No                             | Sí (solo las versiones antiguas)          |
| Sensor de efecto hall                  | No                             | Si                                        |
| Temperatura de trabajo                 | -40°C to 125°C                 | -40°C to 125°C                            |

### 2.8.2 ESP8266



*Imagen 15 Placa de desarrollo NodeMcu (ESP8266) Fuente: <https://www.steren.com.mx/placa-de-desarrollo-nodemcu-esp8266.html>*

#### **Características Principales:**

**Conectividad:** WiFi integrado (802.11 b/g/n).

**Procesador:** Tensilica L106 32-bit RISC CPU.

**Frecuencia de Reloj:** 80 MHz.

**Memoria Flash:** Varía entre 512 KB y 16 MB.

**Entradas/Salidas (GPIO):** Generalmente 17 pines GPIO.

**Interfaces:** UART, I2C, SPI.

#### **Programación:**

Generalmente se programa en lenguaje C++.

#### **Entornos de desarrollo comunes:**

Arduino IDE, PlatformIO.

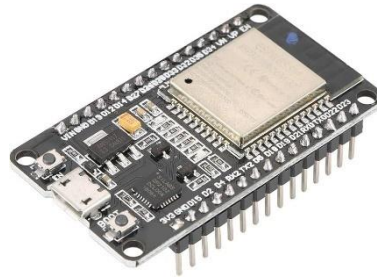
#### **Aplicaciones Típicas:**

Proyectos IoT sencillos.

Dispositivos de monitoreo.

Sensores y actuadores conectados a la red.

### 2.8.3 ESP32



*Imagen 16 Placa de Desarrollo ESP32 Procesador de microcontrolador Fuente: <https://www.amazon.com.mx/desarrollo-Procesador-microcontrolador-Bluetooth-integrado/dp/B07RY9MVCV>*

#### **Características Principales:**

**Conectividad:** WiFi (802.11 b/g/n), Bluetooth (BLE y Classic) integrados.

**Procesador:** Dual-core Xtensa LX6 a 32 bits.

**Frecuencia de Reloj:** Hasta 240 MHz.

**Memoria Flash:** Varía entre 4 MB y 16 MB.

**Entradas/Salidas (GPIO):** Generalmente 36 pines GPIO.

**Interfaces:** UART, I2C, SPI, I2S, CAN, Ethernet, entre otros.

#### **Programación:**

Puede programarse en lenguajes como C, C++, Micropython, y Arduino.

#### **Entornos de desarrollo comunes:**

Soporta entornos de desarrollo como Arduino IDE, PlatformIO y el Espressif IDF (IoT Development Framework).

#### **Aplicaciones Típicas:**

Proyectos IoT avanzados.

Desarrollo de productos comerciales.

Sistemas embebidos que requieren conectividad Bluetooth y WiFi.

Desarrollo de Software:

Para poder trabajar con estos microcontroladores se puede trabajar dentro del entorno de desarrollo de Arduino (Arduino IDE).

Para el proyecto se va a utilizar el microcontrolador ESP32 ya que es una versión más avanzada que el ESP8266, lo que sirve para poder exigirle más en términos de capacidad de procesamiento. Además de que cuenta con un procesador Dual-Core lo que permite ejecutar tareas en paralelo, lo que ayuda al rendimiento y la efectiva respuesta del sistema. También se eligió este microcontrolador por la conectividad Bluetooth con la que cuenta, ya que el ESP8266 solo cuenta con conectividad Wifi, la conectividad Bluetooth ayuda a ampliar las posibilidades de comunicación entre microcontroladores, por el caso de que la conectividad Wifi falle, se obtiene esta como respaldo.

Otro punto para considerar en la elección es el número de entradas y salidas, ya que el ESP32 cuenta con un total de 34 GPIOs y el ESP8266 tiene 17 GPIOs, lo que permite tener una mayor conexión de dispositivos.

Como conclusión, se elige el microcontrolador ESP32 para el proyecto por su versatilidad al trabajar con proyectos más exigentes siendo una versión más avanzada y permitiendo conectar más dispositivos por el mayor número de entradas y salidas en comparación al ESP8266.

## 2.9 Lenguaje de programación más adecuado

Anteriormente se eligió el microcontrolador ESP32, este puede programarse en lenguajes como C, C++, MicroPython o Arduino y puede trabajarse en entornos como Arduino IDE o PlatformIO, de igual manera Espressif (el fabricante del ESP32) brinda el ESP-IDF que es el SDK oficial, este nos ofrece un soporte oficial siempre a todas las funcionalidades, modelos nuevos, actualizaciones de seguridad, etc.

### 2.9.1 C y C++ (Espressif IoT Development Framework - ESP-IDF)

**Descripción:** El ESP-IDF es el entorno de desarrollo oficial proporcionado por Espressif para programar el ESP32 en lenguajes como C y C++. Está basado en FreeRTOS y ofrece una amplia variedad de bibliotecas y herramientas para aprovechar al máximo las características del ESP32.

**Uso Típico:** Proyectos que requieren un control preciso de hardware y acceso directo a las funciones específicas del ESP32.

**Ventajas:**

- Acceso completo al hardware.
- Alto rendimiento y eficiencia.

**Desventajas:**

- Mayor curva de aprendizaje en comparación con entornos más simplificados.

### 2.9.2 Arduino (C++)

**Descripción:** El ESP32 es compatible con el entorno de desarrollo de Arduino, que utiliza el lenguaje C++. Permite a los desarrolladores programar el ESP32 de una manera más accesible, aprovechando las ventajas del ecosistema de Arduino.

**Uso Típico:** Proyectos de prototipos rápidos.

Desarrollo rápido y fácil con acceso a una amplia variedad de bibliotecas.

**Ventajas:**

- Amplia comunidad y soporte.
- Fácil integración con hardware compatible con Arduino.

**Desventajas:**

- Puede tener un rendimiento ligeramente inferior en comparación con el uso directo de ESP-IDF.

### 2.9.3 Micropython

**Descripción:** Micropython es una implementación de Python 3 para microcontroladores y sistemas embebidos. Existen versiones adaptadas para el ESP32, lo que permite programar el microcontrolador utilizando Python.

**Uso Típico:** Desarrollo rápido y prototipado en Python.

Proyectos donde la simplicidad y la legibilidad del código son prioritarias.

**Ventajas:**

- Sintaxis simple y fácil de aprender.
- Facilita el desarrollo rápido de prototipos.

**Desventajas:**

- Puede tener un rendimiento inferior en comparación con C/C++.

### 2.9.4 JavaScript (Espruino)

**Descripción:** Espruino es un proyecto que permite ejecutar JavaScript directamente en microcontroladores, incluido el ESP32. Facilita la programación en JavaScript sin necesidad de intérpretes externos.

**Uso Típico:** Desarrollo en entornos donde JavaScript es el lenguaje preferido.

Proyectos que involucran la interacción con la web.

**Ventajas:**

- Facilita la programación en JavaScript directamente en el microcontrolador.

**Desventajas:**

- La comunidad y el soporte pueden ser más limitados que para C/C++ y Arduino.

### 2.9.5 Blockly

**Descripción:** Blockly es un entorno de programación visual basado en bloques. Algunas plataformas ofrecen soporte para programar el ESP32 utilizando este enfoque, donde los bloques representan diferentes funciones y estructuras de control.

**Uso Típico:** Introducción a la programación para principiantes.

Proyectos educativos y didácticos.

**Ventajas:**

- Enfoque visual y simplificado para aprender a programar.

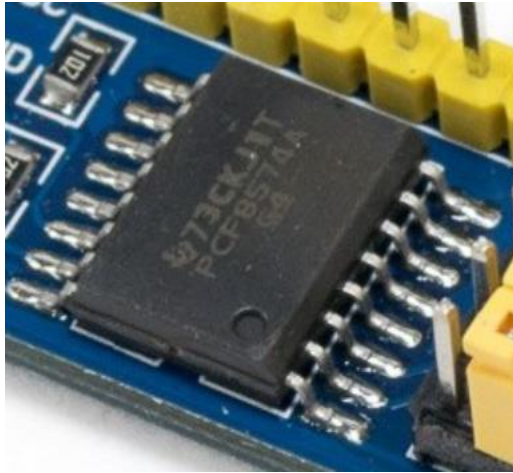
**Desventajas:**

- Menos flexibilidad y control en comparación con la programación textual.

Después de recopilar la información necesaria, se eligió trabajar con Arduino (C++) porque dentro del equipo de desarrollo, se tiene más conocimiento relacionado a este lenguaje de programación, aparte de que el entorno de desarrollo es amigable y cuenta con una comunidad de soporte activa el cual facilita recursos en línea, tutoriales, foros además de contar con una variedad de bibliotecas que ayudan a simplificar la programación de funciones ya predefinidas.

## 2.10 Módulo PCF8574

Con información de la empresa NAYLAMP MECHATRONICS el módulo PCF8574 permite expandir las entradas/salidas digitales de Arduino utilizando solo 2 líneas del bus I2C (SDA y SCL) [NAYLAMP MECHATRONICS,2023].



*Imagen 17 PCF8574 Fuente: <https://naylampmechatronics.com/interfaz-de-usuario/359-modulo-expansor-io-pcf8574.html>*

El módulo PCF8574 posibilita la conversión de datos en formato paralelo (8 E/S) a I2C y viceversa, siendo una opción ideal para la gestión de diversos dispositivos como displays LCD alfanuméricos, pulsadores/switches, teclados matriciales, LEDs, relés, entre otros. Este módulo ofrece la ventaja de compartir el bus I2C con otros dispositivos, permitiendo así ahorrar pines, por ejemplo, para conectar RTC, memorias, sensores, y más. Se destaca la capacidad de manejar hasta 8 módulos expansores PCF8574 en un mismo bus I2C, posibilitando el control de hasta 64 E/S con tan solo 2 pines. Además, el cambio de la dirección I2C es sencillo mediante la manipulación de jumpers en el módulo.

La dirección I2C del módulo (I2C address) dependerá del chip en la placa y la configuración de los jumpers. En el caso del chip PCF8574, la dirección I2C estará en el rango 0x20-0x27, mientras que para el chip PCF8574A, será en el rango de 0x38-0x3F. Es crucial identificar correctamente la dirección I2C del módulo para garantizar el funcionamiento adecuado del programa o sketch. (Módulo Expansor IO PCF8574, s. f.-b)

## **CAPÍTULO III**

### **DESARROLLO**

### 3.1 Levantamiento de requerimientos al cliente

Tabla 6 Primer requerimiento

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Numero de requisito</b>   | <b>1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Nombre del requisito</b>  | Abrir cortina.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Requisito del usuario</b> | Sistema que permita la apertura de la cortina industrial.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Prioridad</b>             | Alta.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Descripción</b>           | El usuario podrá abrir la cortina mediante el sistema de automatización vinculado con la cortina industrial.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Actores</b>               | Cliente.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Precondiciones</b>        | El sistema debe de estar cargado a la placa.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Flujo normal</b>          | <ol style="list-style-type: none"> <li>1. El usuario habilita el permiso de abrir la cortina (Botón 1).</li> <li>2. El usuario abre la cortina dentro del tiempo estimado.</li> <li>3. La cortina se abre por un tiempo destinado.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Flujo alternativo</b>     | <ol style="list-style-type: none"> <li>1. El usuario habilita el permiso de abrir la cortina (Botón 1). <ol style="list-style-type: none"> <li>1.1 El usuario no abre la cortina dentro del tiempo estimado</li> <li>1.2 La cortina se bloquea <ol style="list-style-type: none"> <li>1.2.1 Si el usuario abre la cortina sin habilitar el permiso se activará la alarma.</li> <li>1.2.2 La cortina se cierra.</li> <li>1.2.3 El usuario solicita que se desactive la alarma.</li> </ol> </li> <li>1.3 El usuario habilito nuevamente el permiso.</li> </ol> </li> <li>2. El usuario abre la cortina dentro del tiempo estimado.</li> <li>3. La cortina se abre por un tiempo destinado.</li> </ol> |

Tabla 7 Segundo requerimiento

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Numero de requisito</b>   | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Nombre del requisito</b>  | Cerrar cortina.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Requisito del usuario</b> | Sistema que permita la apertura de la cortina industrial.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Prioridad</b>             | Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Descripción</b>           | El usuario podrá cerrar la cortina mediante el sistema de automatización vinculado con la cortina industrial.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Actores</b>               | Cliente                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Precondiciones</b>        | El sistema debe de estar cargado a la placa.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Flujo normal</b>          | <ol style="list-style-type: none"> <li>1. El tiempo destinado de abertura de la cortina esta por acabar.</li> <li>2. La cortina se cierra dentro del tiempo destinado de abertura.</li> <li>3. En espera de que se habilite el permiso de abertura.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Flujo alternativo</b>     | <ol style="list-style-type: none"> <li>1. El tiempo destinado de abertura de la cortina esta por acabar. <ol style="list-style-type: none"> <li>1.1 La cortina no se cierra dentro del tiempo destinado de abertura. <ol style="list-style-type: none"> <li>1.1.1 Se activa la alarma.</li> </ol> </li> </ol> </li> <li>2. El usuario solicita más tiempo de abertura dentro del margen destinado a la primera abertura. <ol style="list-style-type: none"> <li>2.1 El tiempo de abertura cambia más tiempo.</li> <li>2.2 El tiempo agregado está por terminar. <ol style="list-style-type: none"> <li>2.2.1 El usuario baja la cortina dentro del tiempo agregado. <ol style="list-style-type: none"> <li>2.2.1.1 El sistema cambia a Paso 3.</li> </ol> </li> <li>2.2.2 El usuario no baja la cortina dentro del tiempo agregado <ol style="list-style-type: none"> <li>2.2.2.1 Se activa la alarma.</li> <li>2.2.2.2 Se baja cortina.</li> <li>2.2.2.3 El usuario solicita que se desactive la alarma.</li> </ol> </li> </ol> </li> </ol> </li> <li>3. En espera de que se habilite el permiso de abertura.</li> </ol> |

### 3.2 Crear un diseño esquemático detallado del sistema de automatización

Al realizar el esquema tomando en cuenta los escenarios planteados, queda de la siguiente manera.

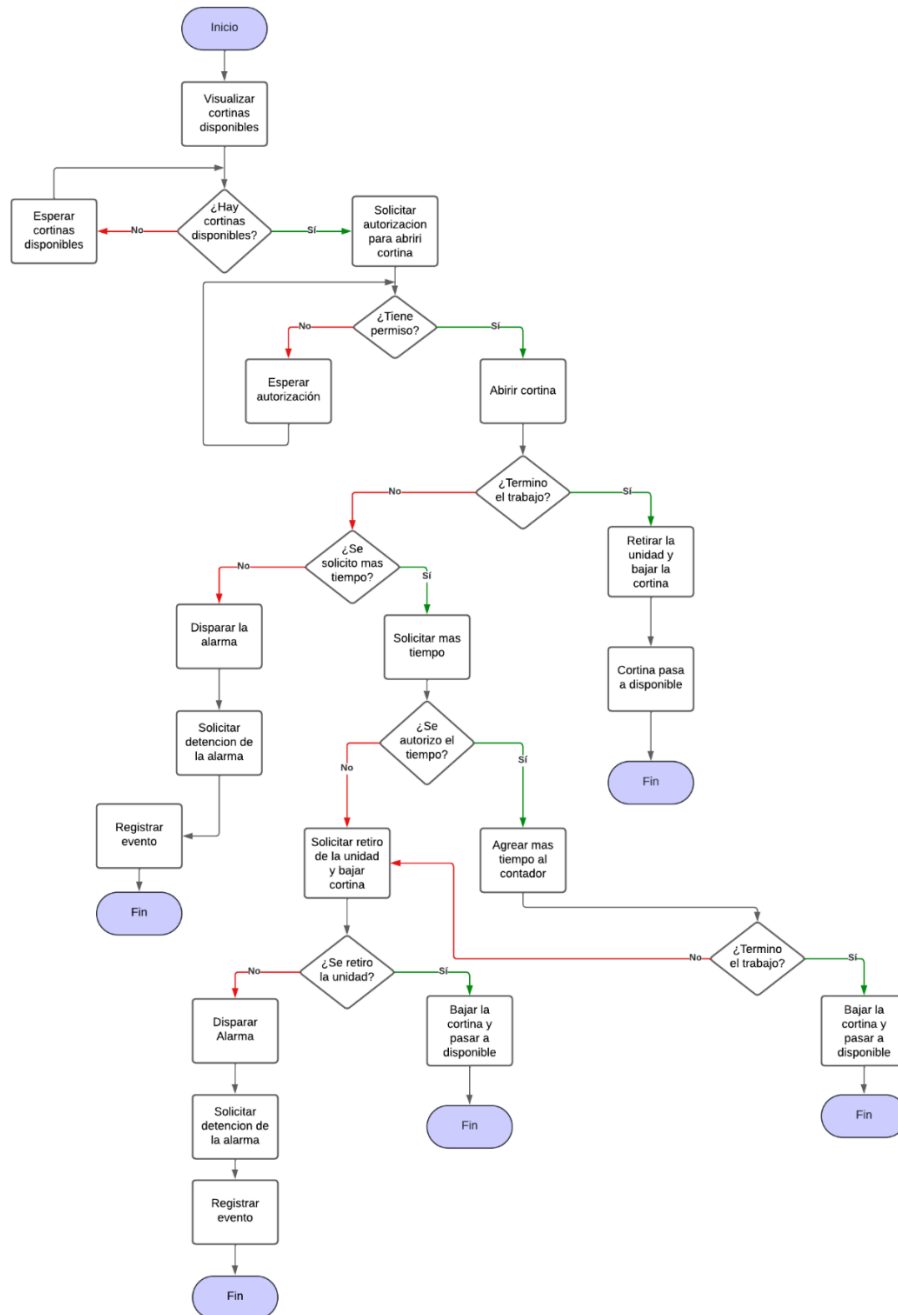


Imagen 18 Diagrama de flujo del proceso Fuente: Propia

### 3.3 Configuración del entorno de trabajo

Para comenzar a trabajar con Arduino (C++) se debe de descargar el instalador, para ello, se deben de seguir los siguientes pasos.

1. Buscar en el navegador “Arduino IDE”.
2. Buscar la opción de Descargas.
3. Seleccionar el sistema operativo a utilizar.
4. Descargar el ejecutable del programa

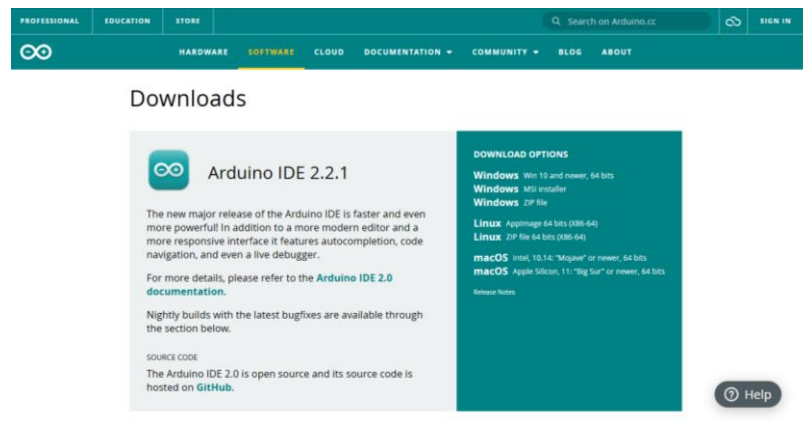


Imagen 19 Pantalla de descarga de Arduino IDE Fuente: Propia

Una vez descargado el archivo se debe de buscar en la Carpeta predeterminada de descargas, una vez ubicado de debe de instalar de la siguiente manera.

1. Seleccionar el archivo ejecutable.
2. Clic derecho.
3. Seleccionar “Ejecutar como administrador”.
4. Esperar a que se inicie la instalación.

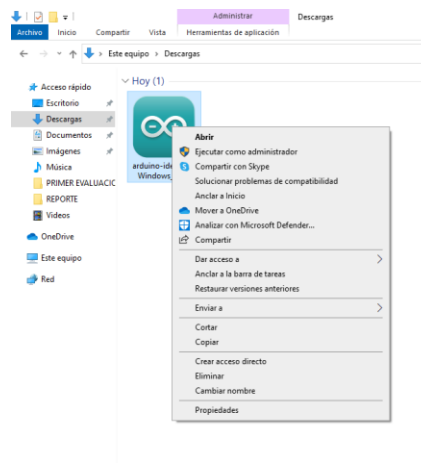


Imagen 20 Instalación de Arduino IDE Fuente: Propia

Al momento de instalar los pasos son los siguientes:

1. Aceptar el Acuerdo de licencia.
2. Escoger los usuarios que pueden trabajar con el software (Todos).
3. Seleccionar la carpeta de instalación. (Dejar la que está por defecto).
4. Terminar la instalación.
5. Al abrir se terminarán de descargar los paquetes necesarios.

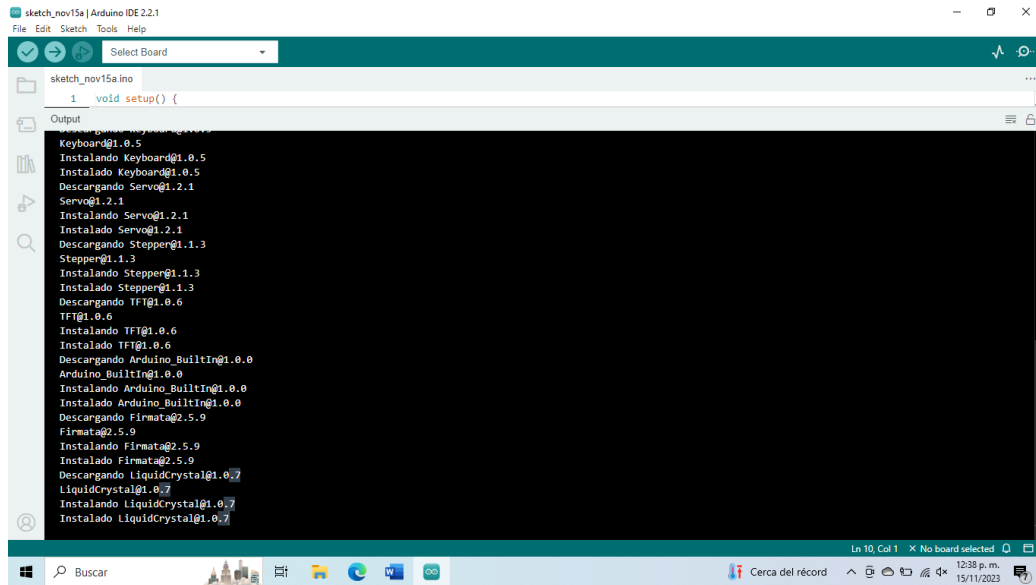


Imagen 21 Instalación de paquetes complementarios Fuente: Propia

Dentro de Arduino IDE se debe buscar “Boards Manager” se debe buscar la opción de ESP32, para ello se deben de seguir los siguientes pasos:

1. Abrir “Tools”.
2. Abrir la opción “Board”.
3. Seleccionar “Board Manager”.
4. Ingresar en el buscador “ESP32” e instalar la segunda opción (Espressif Systems).
5. Esperar a que el paquete se instale.



Imagen 22 Paquete ESP32 instalado Fuente: Propia

### 3.4 Programación en Simulador Wokwi

Para iniciar con la programación del proyecto, se utilizar un simulador de ESP32 nombrado Wokwi, para esto se debe tener una cuenta del mismo o bien utilizar la cuenta de GitHub para que los proyectos generados se puedan guardar.

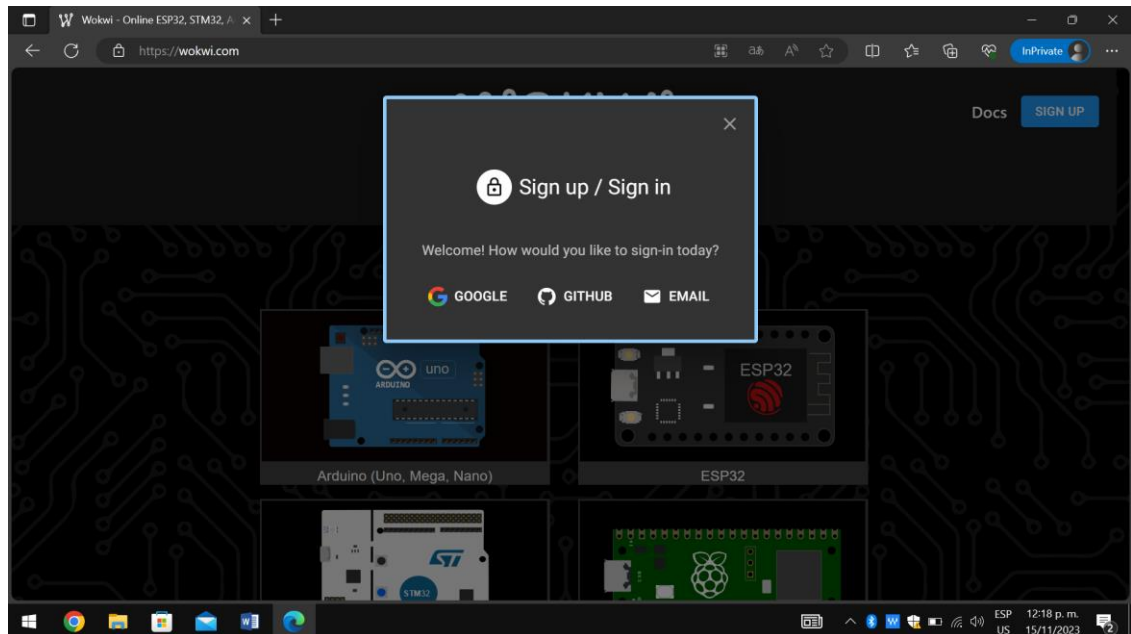


Imagen 23 Pantalla inicio de sesión Wokwi Fuente: Propia

Al entrar se debe ubicar la sección de “Starter Templates” y seleccionar la opción “ESP32”

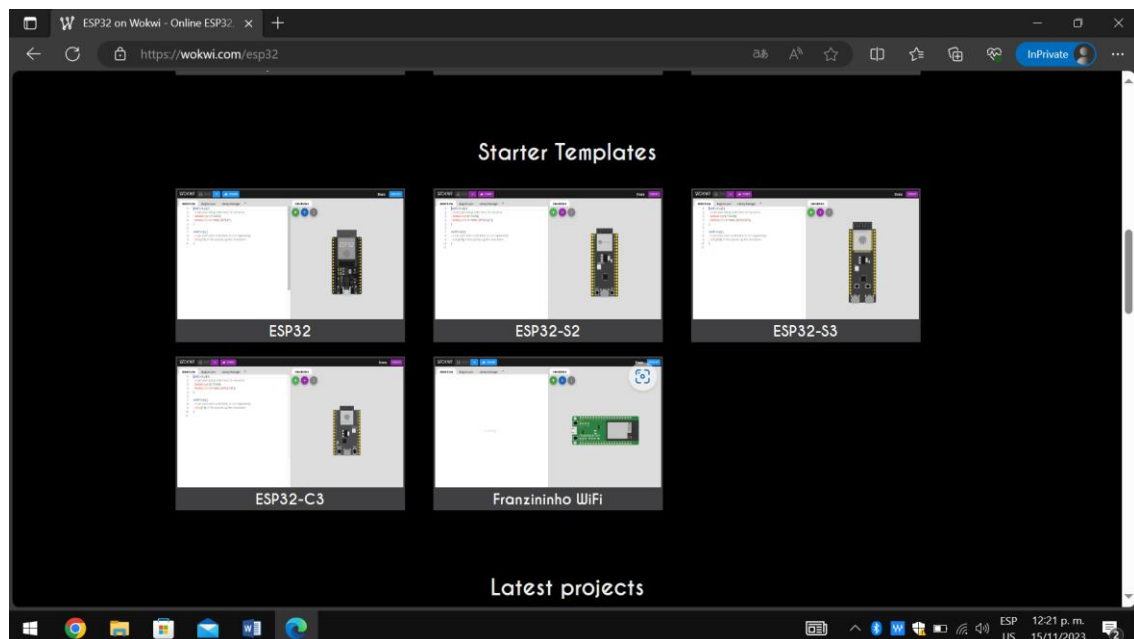


Imagen 24 Templates ejemplo de ESP32 Fuente: Propia

La pantalla se divide en 2, en la parte izquierda funciona el código y del lado derecho se visualiza el emulador del microcontrolador ESP32.

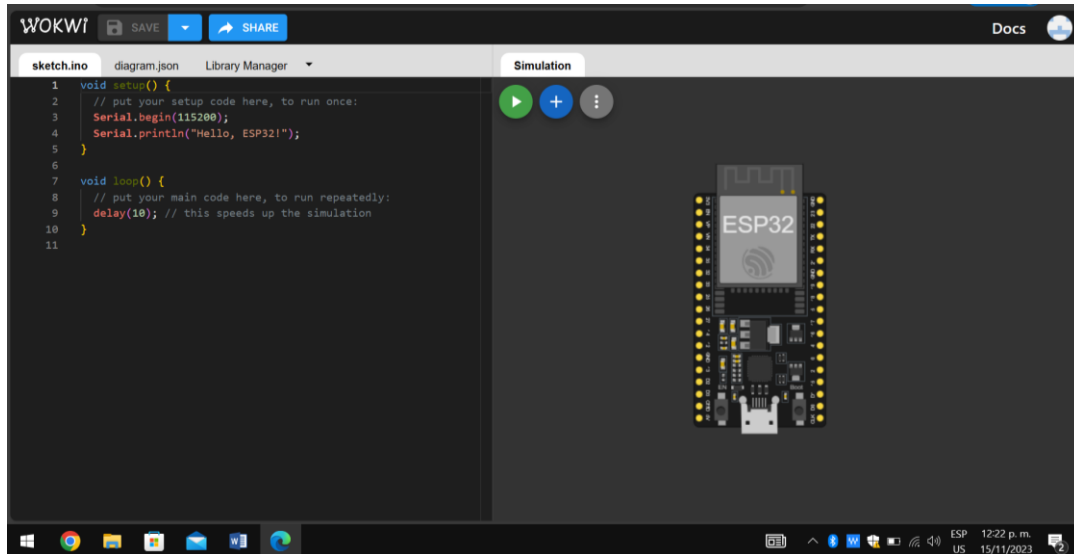


Imagen 25 Pantalla principal Wokwi Fuente: Propia

Para comenzar con la emulación se debe de recrear el siguiente circuito eléctrico, que incluye:

- 1 Microcontrolador ESP32
- 2 Relevadores
- 10 leds
- 10 resistencias de 330 ohms
- 2 Switches
- 1 buzzer

De esta manera se conforma:

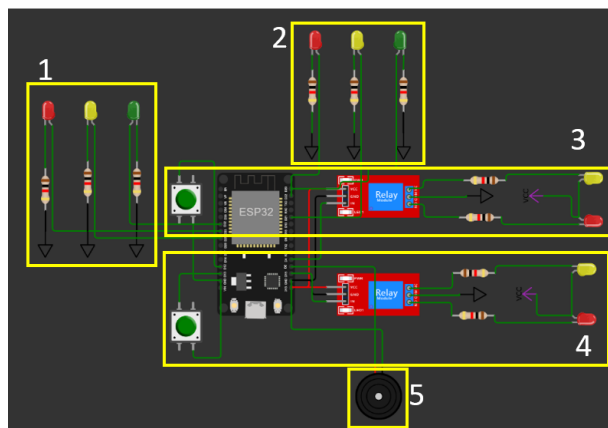


Imagen 26 Circuito en simulador Wokwi Fuente: Propia

Donde:

1. Es el semáforo perteneciente a la cortina 1.
2. Es el semáforo perteneciente a la cortina 2.
3. Es el botón que activa al relevador de la cortina 1.
4. Es el botón que activa al relevador de la cortina 2.
5. Buzzer.

Una vez realizado el circuito en el simulador, se empieza con la lógica, para ello se debe de llamar a la librería de Arduino, para que las funciones que se coloquen más adelante no tengas fallas.

```
#include <Arduino.h>
```

Se definen las variables para las cortinas (Para la cortina 2 son las mismas variables con el detalle de que se sustituye el 1 por el 2 y cambia el número de pin)

```
const int botonPin1 = 12; // GPIO D12 para el primer botón
const int relePin1 = 4;    // GPIO D4 para el primer relé
const int semaforoRojo1 = 33; // GPIO D33 para el LED rojo del semáforo del
primer relé
const int semaforoVerde1 = 32; // GPIO D32 para el LED verde del semáforo del
primer relé
const int semaforoAmarillo1 = 25; // GPIO D31 para el LED amarillo del semáforo
del primer relé
const int zumbadorPin = 2; // GPIO D2 para el zumbador compartido por ambos
semáforos
```

Se declaran las variables de estado y el tiempo de inicio de cada uno de los relés, posteriormente se declaran las variables para el control del proceso de los semáforos, su estado inicial es falso.

```
bool estadoRele1 = LOW; // Estado inicial del primer relé (Comúnmente Cerrado)
bool estadoRele2 = LOW; // Estado inicial del segundo relé (Comúnmente Cerrado)

unsigned long tiempoInicio1 = 0; // Tiempo de inicio del primer relé activado
unsigned long tiempoInicio2 = 0; // Tiempo de inicio del segundo relé activado

bool procesoActivo1 = false; // Variable para controlar si un proceso del primer
relé está activo
bool procesoActivo2 = false; // Variable para controlar si un proceso del
segundo relé está activo
```

Después, dentro de la función `setup()`, se configuran los pines como entradas y salidas respecto a la función que tiene dentro del circuito, además se establecen los estados iniciales de los leds y relés.

```
void setup() {  
    pinMode(botonPin1, INPUT_PULLUP); // Configura el primer botón como entrada  
    con pull-up  
    pinMode(relePin1, OUTPUT);          // Configura el primer relé como salida  
    pinMode(semaforoRojo1, OUTPUT);    // Configura el LED rojo del semáforo del  
    primer relé como salida  
    pinMode(semaforoVerde1, OUTPUT);   // Configura el LED verde del semáforo del  
    primer relé como salida  
    pinMode(semaforoAmarillo1, OUTPUT); // Configura el LED amarillo del semáforo  
    del primer relé como salida  
    digitalWrite(relePin1, estadoRele1); // Establece el estado inicial del primer  
    relé  
    digitalWrite(semaforoRojo1, HIGH); // Inicialmente, enciende el LED rojo del  
    semáforo 1  
    digitalWrite(semaforoVerde1, LOW); // Inicialmente, apaga el LED verde del  
    semáforo  
    digitalWrite(semaforoAmarillo1, LOW); // Inicialmente, apaga el LED amarillo  
    del semáforo 1  
  
    pinMode(botonPin2, INPUT_PULLUP); // Configura el segundo botón como entrada  
    con pull-up  
    pinMode(relePin2, OUTPUT);          // Configura el segundo relé como salida  
    pinMode(semaforoRojo2, OUTPUT);    // Configura el LED rojo del semáforo del  
    segundo relé como salida  
    pinMode(semaforoVerde2, OUTPUT);   // Configura el LED verde del semáforo del  
    segundo relé como salida  
    pinMode(semaforoAmarillo2, OUTPUT); // Configura el LED amarillo del semáforo  
    del segundo relé como salida  
    digitalWrite(relePin2, estadoRele2); // Establece el estado inicial del  
    segundo relé  
    digitalWrite(semaforoRojo2, HIGH); // Inicialmente, enciende el LED rojo del  
    semáforo 2  
    digitalWrite(semaforoVerde2, LOW); // Inicialmente, apaga el LED verde del  
    semáforo 2  
    digitalWrite(semaforoAmarillo2, LOW); // Inicialmente, apaga el LED amarillo  
    del semáforo 2  
  
    pinMode(zumbadorPin, OUTPUT); // Configura el zumbador como salida  
    digitalWrite(zumbadorPin, LOW); // Apaga el zumbador inicialmente  
  
}
```

Se crea una función nombrada **cambiarEstadoRelay**, la función de esta es cambiar de estado al relevador, ayudando a optimizar la programación.

```
void cambiarEstadoRelay(int relePin, bool &estadoRelay) {  
    estadoRelay = !estadoRelay;  
    digitalWrite(relePin, estadoRelay);  
}
```

Dentro del loop se leen los estados de los botones para determinar cuándo se debe de iniciar un proceso en alguno de los semáforos.

```
void loop() {  
    int estadoBoton1 = digitalRead(botonPin1);  
    int estadoBoton2 = digitalRead(botonPin2);
```

Se controla un semáforo mediante la detección de la presión del botón 1. Si el semáforo no está actualmente en proceso y se detecta que el botón está presionado, el código enciende el LED amarillo del semáforo, hace parpadear este LED durante 5 segundos, apaga el LED amarillo, cambia el estado de un relé, registra el tiempo actual y marca el proceso como activo. Además, apaga el LED rojo y enciende el LED verde del semáforo, indicando así una transición al estado permitido. Para el semáforo 2 es la misma lógica, con las variables correspondientes.

```
    if (!procesoActivo1) {  
        if (estadoBoton1 == LOW) {  
            digitalWrite(semaforoAmarillo1, HIGH); // Enciende el LED amarillo del  
semaforo 1  
            for (int i = 0; i < 5000; i += 500) {  
                digitalWrite(semaforoAmarillo1, !digitalRead(semaforoAmarillo1)); //  
Parpadea el LED amarillo  
                delay(500); // Retardo para el parpadeo  
            }  
            digitalWrite(semaforoAmarillo1, LOW); // Apaga el LED amarillo del  
semaforo 1  
            cambiarEstadoRelay(relePin1, estadoRele1);  
            tiempoInicio1 = millis();  
            procesoActivo1 = true;  
            digitalWrite(semaforoRojo1, LOW); // Apaga el LED rojo del semáforo 1  
            digitalWrite(semaforoVerde1, HIGH); // Enciende el LED verde del semáforo  
1  
        }  
    }  
}
```

Esta porción de código gestiona el procedimiento de cierre en el semáforo 1. Cuando el relé está en estado alto (HIGH) y ha transcurrido al menos 10 segundos desde el inicio del proceso (registrado en tiempoInicio1), se ejecutan varias acciones. Se altera el estado del relé, se desactiva el proceso, se apaga el LED verde y se enciende el LED rojo del semáforo 1. Además, se activa una alarma mediante la activación de un zumbador durante 5 segundos antes de apagarlo. Se repite la misma lógica para el segundo relevador con las variables correspondientes.

```
// Verifica si ha pasado el tiempo de activación del primer relé (10 segundos)
if (estadoRele1 == HIGH && (millis() - tiempoInicio1) >= 10000) {
    cambiarEstadoRelay(relePin1, estadoRele1);
    procesoActivo1 = false;
    digitalWrite(semaforoRojo1, HIGH); // Enciende el LED rojo del semáforo 1
    digitalWrite(semaforoVerde1, LOW); // Apaga el LED verde del semáforo 1

    // Activa la alarma
    digitalWrite(zumbadorPin, HIGH);
    delay(5000); // Sonar la alarma durante 5 segundos
    digitalWrite(zumbadorPin, LOW); // Apaga la alarma
}
```

En esta parte del código se controla la lógica de parpadeo del LED verde en el semáforo 1. La condición verifica si el estado del relé es alto (HIGH), el proceso 1 está activo y ha transcurrido al menos 8 segundos desde el inicio del proceso. En caso afirmativo, el código alterna el estado del LED verde del semáforo en un patrón de parpadeo con un retardo de 500 milisegundos entre cada cambio de estado. Se repite la lógica para el semáforo verde 2, con las variables correspondientes. (Se puede ver el código completo en Anexos 1.3)

```
// Controla la alerta verde (parpadeo) durante los últimos 2 segundos antes de
apagar el LED rojo del primer rele
if (estadoRele1 == HIGH && procesoActivo1 && (millis() - tiempoInicio1) >=
8000) {
    digitalWrite(semaforoVerde1, !digitalRead(semaforoVerde1)); // Parpadea el
LED verde del semáforo 1
    delay(500); // Retardo para el parpadeo
}
```

Al cargar el programa en el simulador, y ejecutarlo, el primer estado que debe de aparecer es los leds rojos encendidos, esto indica que la cortina se encuentra disponible.

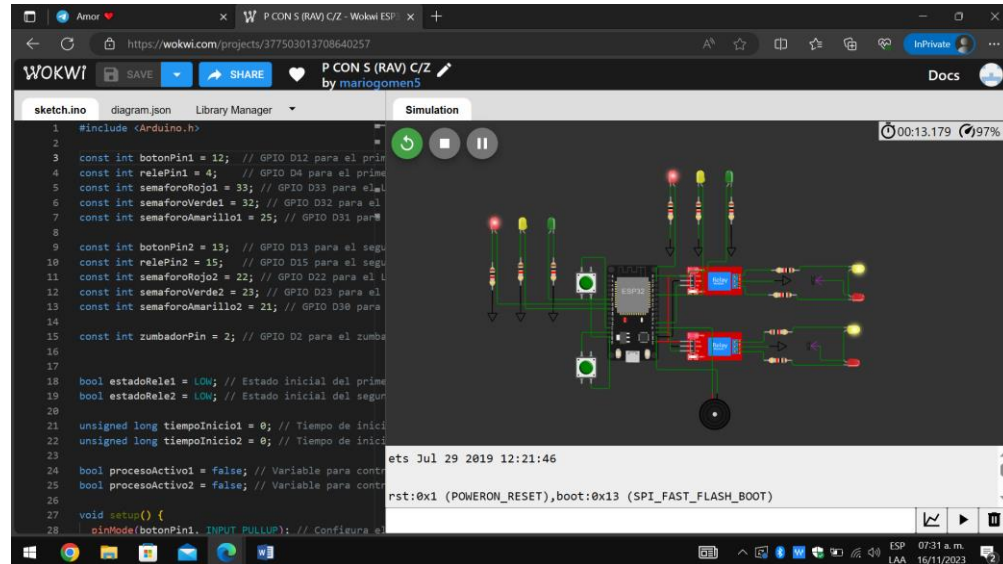


Imagen 27 Simulación activa estado 1 Fuente: Propia

Al cambiar de estado, se puede observar que la luz amarilla se enciende, indicando la apertura de la cortina.

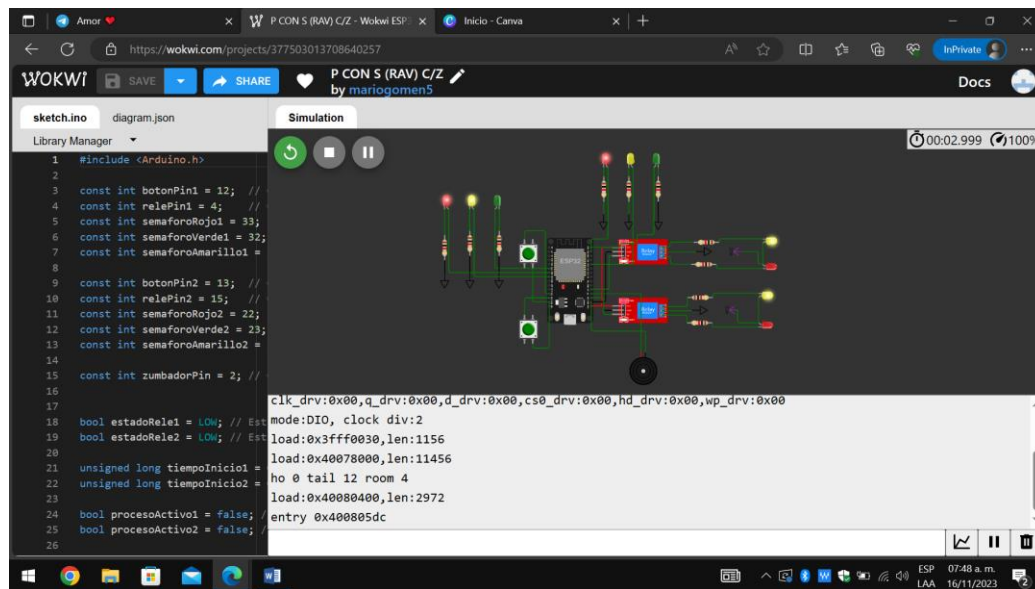


Imagen 28 Simulación activa estado 2 Fuente: Propia

El ultimo estado es el led verde, lo que indica que el relevador cambio de estado, indica que la cortina se abrió.

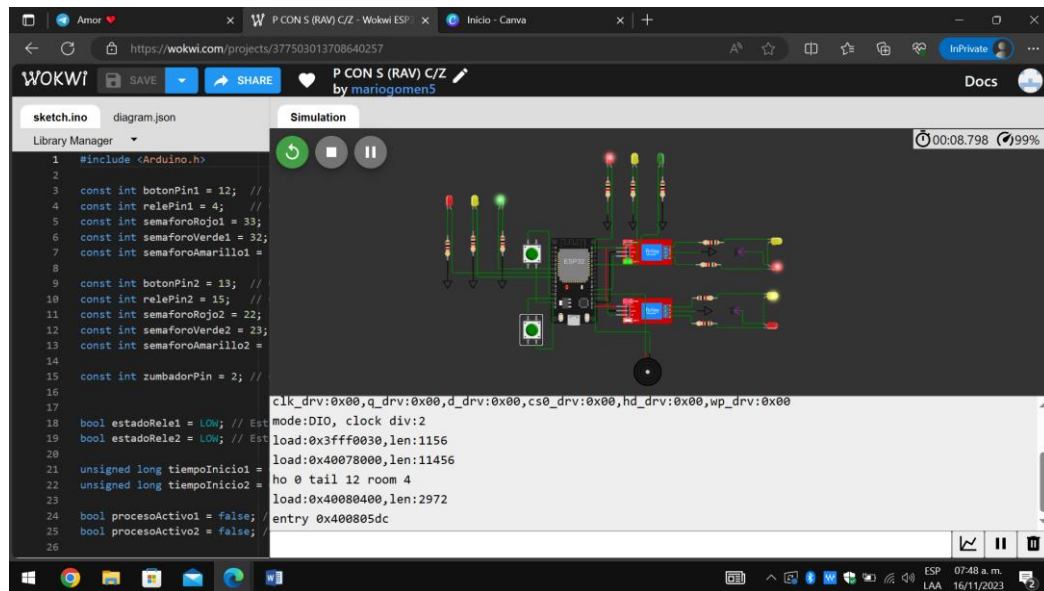


Imagen 29 Simulación activa estado 3 Fuente: Propia

### 3.5 Programación placa KC868 con Blynk

Para iniciar la conexión con Blynk se debe de configurar Arduino IDE. Para instalar la librería necesaria se debe de seguir los siguientes pasos:

1. Abrir la opción “Library Manager”.
2. Buscar la librería “Blynk”.
3. Esperar a que termine la descarga de la librería.

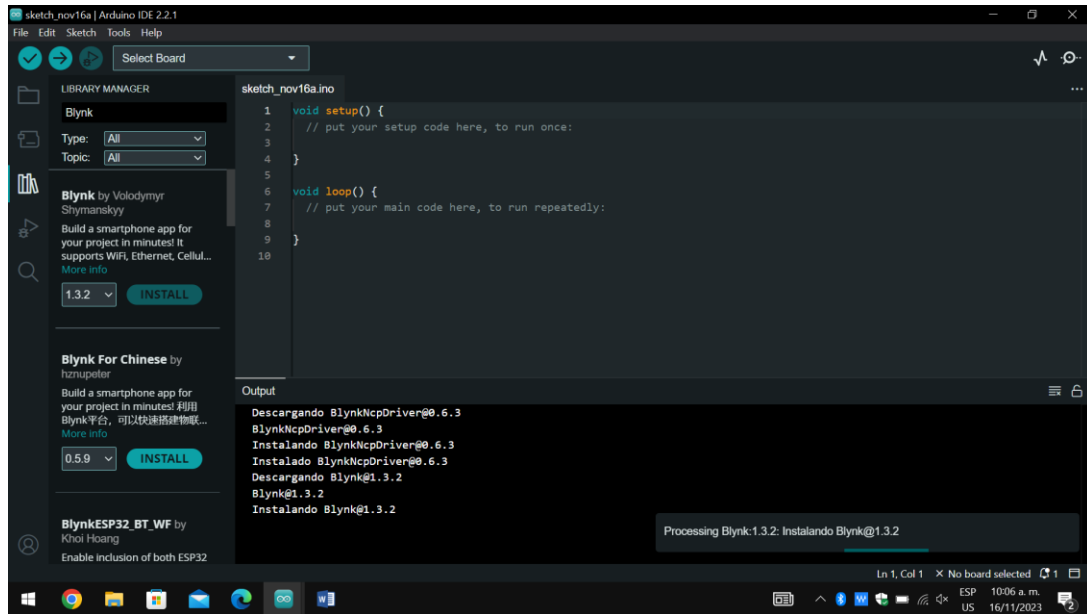


Imagen 30 Instalación de paquete Blynk Fuente: Propia

Para abrir el ejemplo se necesita:

1. Abrir la opción “File”.
2. Escoger la opción “Examples”.
3. Seleccionar la opción “Blynk”
4. Seleccionar “Boards Wifi”.
5. Escoger la opción “ESP32 Wifi”

Se abrirá un sketch en el cual viene la información que se requiere para crear la conexión con el microcontrolador ESP32 y la aplicación Blynk.

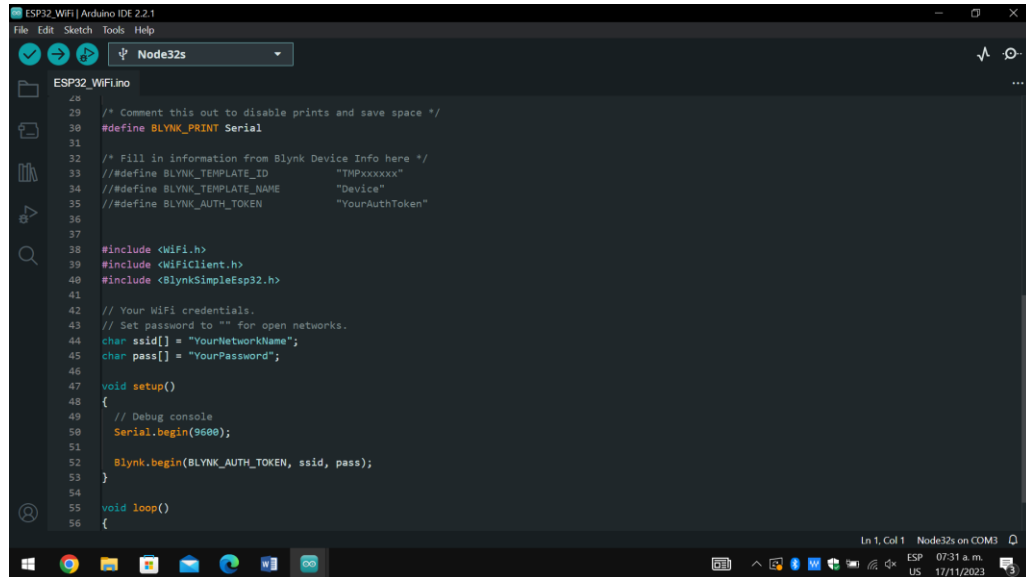


Imagen 31 Sketch ejemplo conexión blynk Fuente: Propia

Al tener listo el sketch, se debe de abrir la página de Blynk, una vez logeado, se debe de crear un nuevo template, para esto se deben de seguir los siguientes pasos:

1. Seleccionar la opción “Templates”
2. Seleccionar “New Template”
3. Rellenar la información necesaria:
  - Nombre.
  - Hardware (ESP32).
  - Connection Type (Wifi).
4. Guardar el Template

Posteriormente el Template creado se debe de cargar a un Datastream para esto se debe de buscar la opción “DataStream” y seguir los siguientes pasos:

1. Seleccionar la opción “New Datastreams”.
2. Seleccionar “Dital Pin”.
3. Ingresar la información solicitada:
  - Name.
  - Alias.
  - El número de pin.
  - El tipo de salida (Output o Input).

4. Seleccionar “Create”.
5. El datastream se guarda.

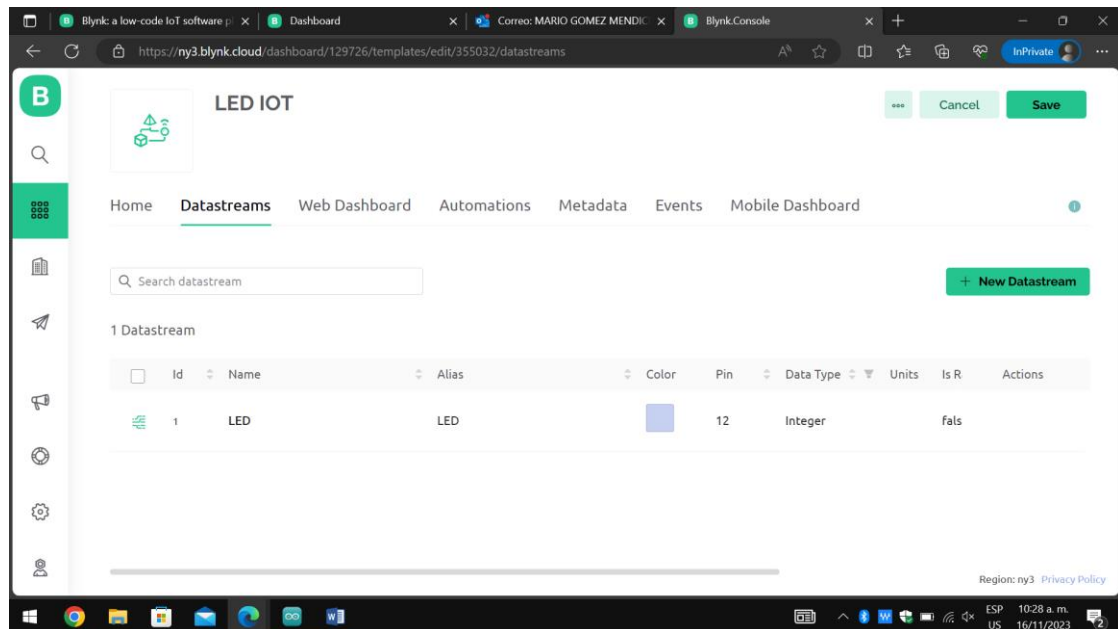


Imagen 32 Pantalla principal Datastream Fuente: Propia

Se debe seleccionar la opción “Web Dashboard” y arrastrar el Widget “Switch” (se encuentra dentro del apartado de “Widget Box”) a la sección “Add new widget”.

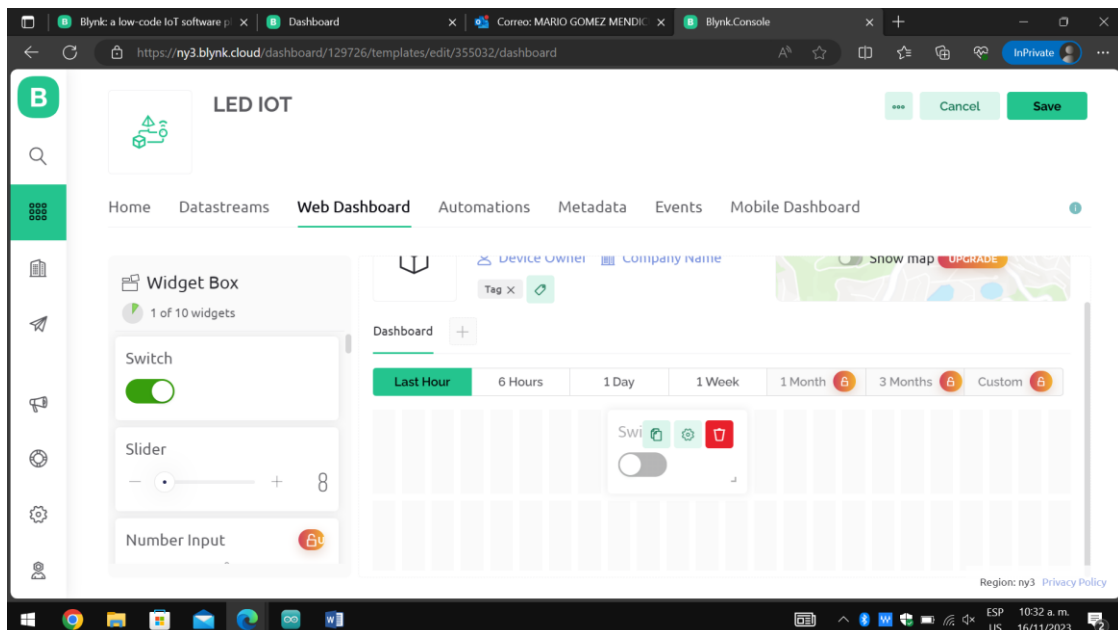


Imagen 33 Pantalla principal Web Dashboard Fuente: Propia

Para configurar el widget, se selecciona el icono de tuerca dentro del widget colocado, lo que se debe configurar es:

- Título (Nombre).
- Escoger el Datastream (El creado anteriormente).
- Valores (ON/OFF).
- Mostrar las etiquetas.
- Posición.
- Color del widget

Al finalizar la configuración se debe seleccionar el botón “Save” para guardar los cambios realizados.

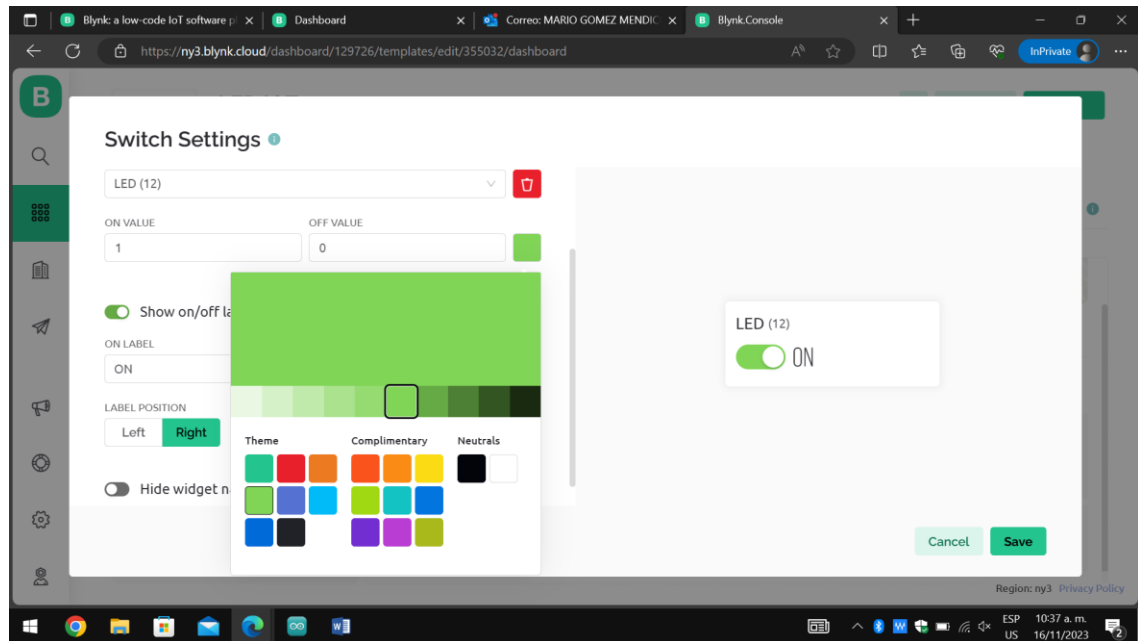


Imagen 34 Configuración del widget Fuente: Propia

Al finalizar este proceso, se debe de crear un nuevo device (Servicio), se debe de dirigir a la opción Search y dentro de esta escoger la opción “New Device” después seleccionar la opción “From Template” ahí se debe seleccionar el template anteriormente creado y crearlo.

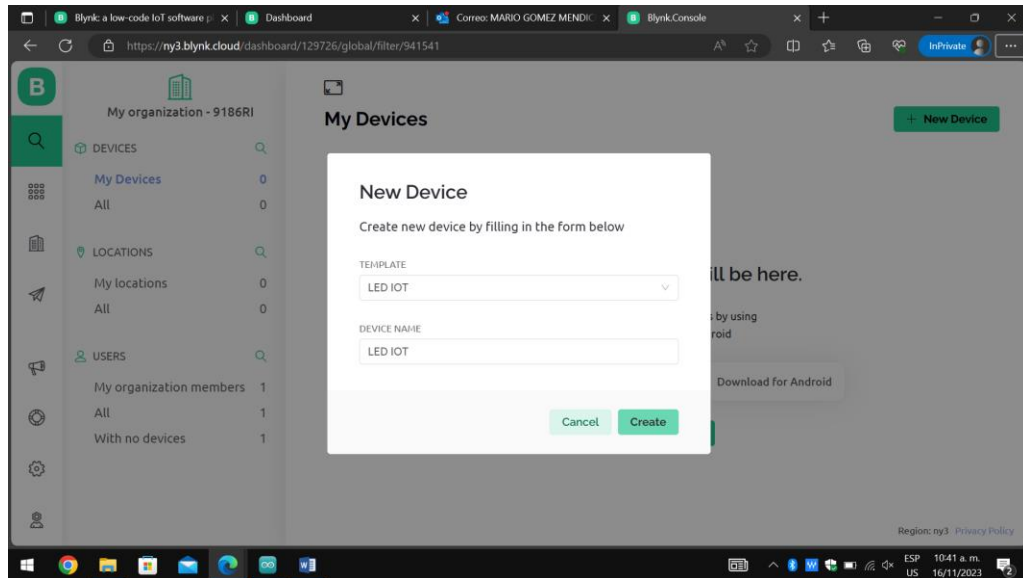


Imagen 35 Pantalla principal nuevo servicio Fuente: Propia

Al crear el device, el sistema brindara la información necesaria para la conexión con Arduino IDE.

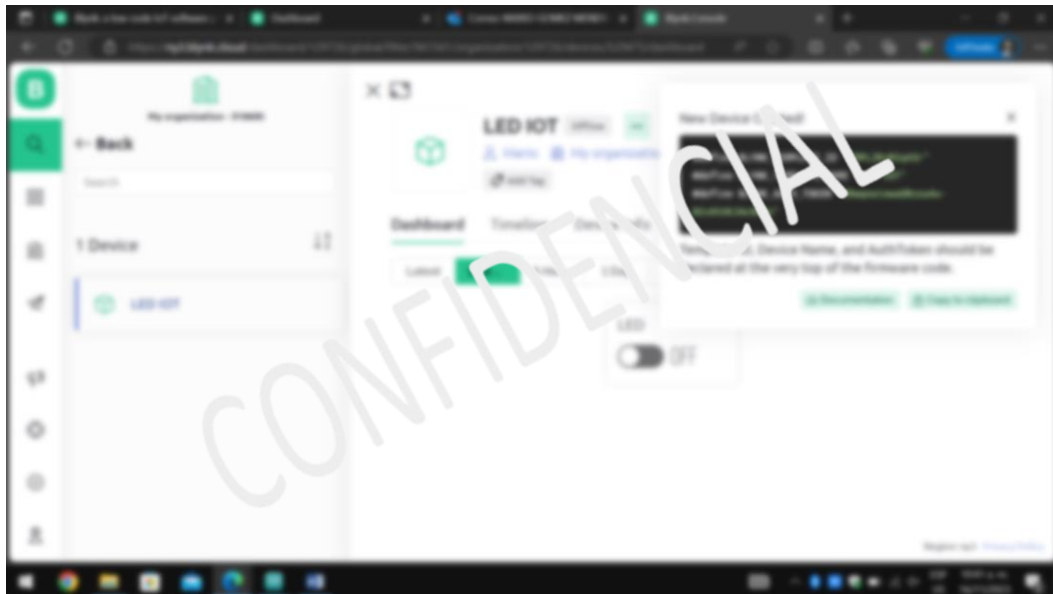


Imagen 36 Información Token de Blynk Fuente: Propia

Al tener la información, se pega dentro del sketch de Arduino IDE, para iniciar la conexión con la aplicación.

```
#define BLYNK_TEMPLATE_ID "CONFIDENCIAL"  
#define BLYNK_TEMPLATE_NAME "CONFIDENCIAL"  
#define BLYNK_AUTH_TOKEN "CONFIDENCIAL"
```

Se llaman a las bibliotecas necesarias.

```
#include <WiFi.h>  
#include <WiFiClient.h>  
#include <BlynkSimpleEsp32.h>
```

Se deben de ingresar las credenciales de la red de wifi en la que está conectado el dispositivo.

```
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "CONFIDENCIAL";  
char pass[] = "CONFIDENCIAL";
```

Se inicia una comunicación serial y se inicia la comunicación con los parámetros específicos para la conexión (Se puede ver el código completo en Anexos 1.4).

```
void setup()  
{  
  Serial.begin(9600);  
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);  
}  
  
void loop()  
{  
  Blynk.run();  
}
```

Al conectar la placa al PC y compilar el programa, se puede observar el monitor serial que la conexión fue exitosa



*Imagen 37 Conexión exitosa a Blynk Fuente: Propia*

### 3.6 Programación placa KC868 con WhatsApp

Para comenzar la comunicación con WhatsApp, debe ingresar a la aplicación “ThingESP”, una vez logeado, se selecciona la opción “Add New Project” e ingresar la información solicitada:

- Nombre del proyecto.
- Credencial del proyecto (Guardar esta información por es relevante más adelante).

El proyecto se creará.



Imagen 38 Pantalla proyecto creado en ThingESP Fuente: Propia

Se direccionará a la página principal de la aplicación.

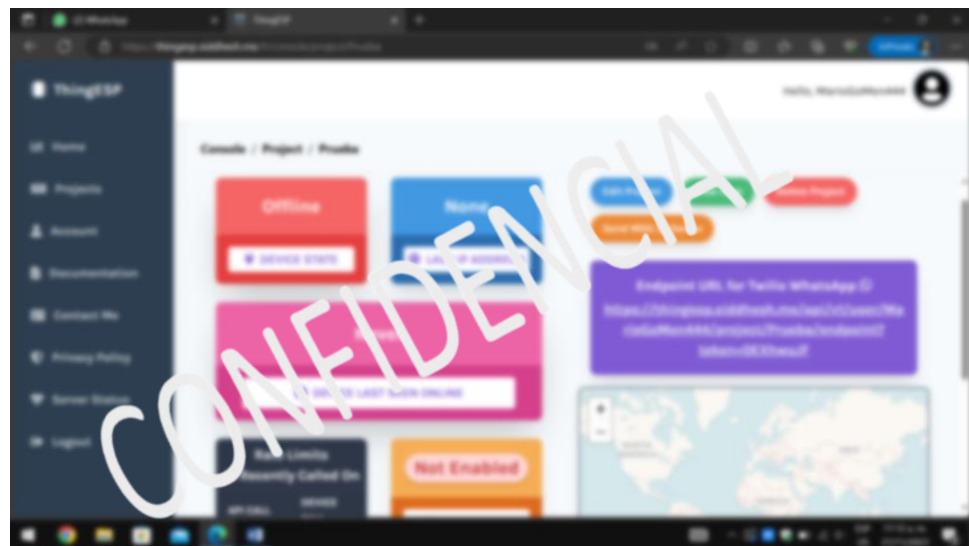
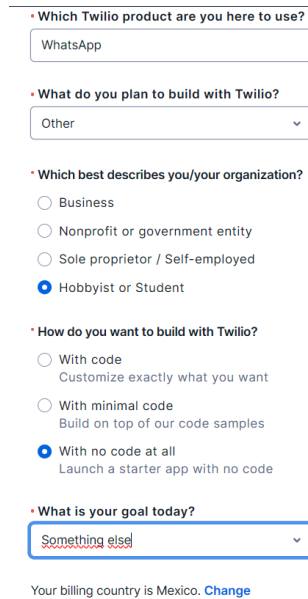


Imagen 39 Pantalla principal ThingESP Fuente: Propia

Posteriormente debe de entrar a la página de Twilio, una vez logeado seleccionar la opción “Create New Account”, al entrar en esta opción seguir los siguientes pasos.

- Asignar un nombre.
- Agregar el número de WhatsApp a vincular.
- Ingresar el número de verificación enviado al número asociado.
- Contestar la encuesta de la siguiente manera:



• Which Twilio product are you here to use?  
WhatsApp

• What do you plan to build with Twilio?  
Other

• Which best describes you/your organization?  
☐ Business  
☐ Nonprofit or government entity  
☐ Sole proprietor / Self-employed  
☒ Hobbyist or Student

• How do you want to build with Twilio?  
☐ With code  
Customize exactly what you want  
☐ With minimal code  
Build on top of our code samples  
☒ With no code at all  
Launch a starter app with no code

• What is your goal today?  
Something else

Your billing country is Mexico. [Change](#)

Imagen 40 Encuesta Twilio Fuente: Propia

- Guardar en contactos el numero proporcionado en la página y mandar el siguiente mensaje “join cake-direct”



Imagen 41 Primer mensaje WhatsApp Fuente: Propia

Al terminar los pasos anteriores, se manda un mensaje de prueba dentro de la página de Twilio para comprobar la comunicación.

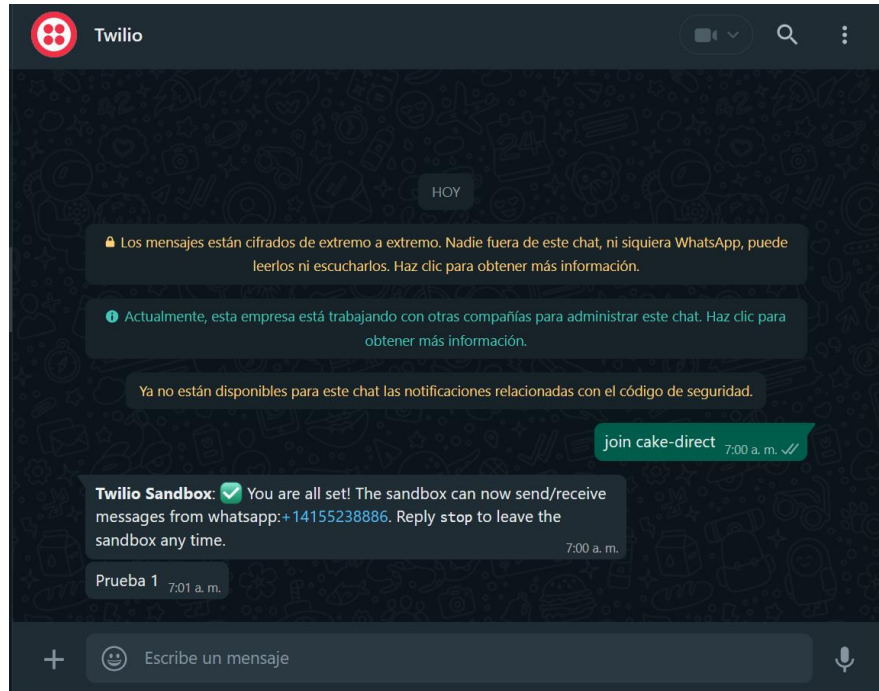


Imagen 42 Mensaje de prueba en WhatsApp Fuente: Propia

Al comprobar la comunicación, se debe de copiar la información brindada en la ventana de Python.



Imagen 43 Información conexión con Twilio Fuente: Propia

Se debe regresar a la ventana principal de ThingESP y copiar la información que se encuentra ubicada en la ventana morada y regresar a Twilio, dentro de la opción “When a message comes in” se pega la información y se verifica que el método sea POST y guardar los cambios.



Imagen 44 Pantalla Sandbox Configuration Twilio Fuente: Propia

Al finalizar esto, se debe regresar a la ventana de ThinESP y escoger la opción “Edit Project”, dentro de esta, se debe de habilitar la opción “Device calls” y pegar la información guardada de la ventana de Python y guardar los cambios.

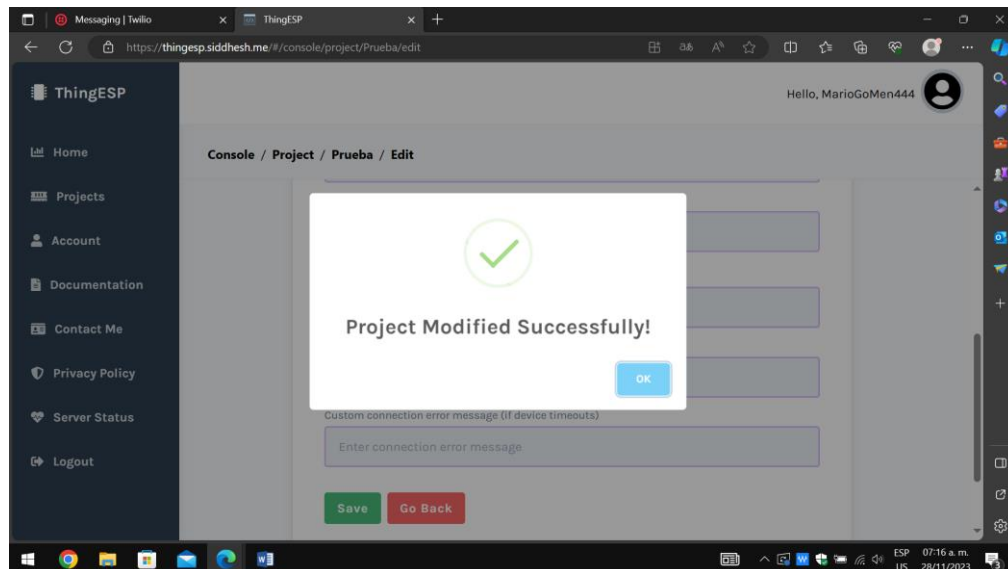


Imagen 45 Pantalla proyecto modificado Fuente: Propia

Al hacer esto los servicios estarán activos, ahora, se deberá de subir el código para estar en línea con la aplicación. Se llaman a las bibliotecas necesarias y se ingresan las credenciales de ThingESP.

```
#include <WiFi.h>
#include <ThingESP.h>

ThingESP32 thing("Nombre usuario", "Nombre proyecto", "Token"); //Conecta el
dispositivo con la plataforma
```

Se declaran variables.

```
int LED = 2;

unsigned long previousMillis = 0;
const long INTERVAL = 6000;
```

Se inicializa la comunicación serial, los modos de las variables y credenciales de conexión.

```
void setup()
{
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
  thing.SetWiFi("SIDD ", "PASSWORD");
  thing.initDevice();
}
```

Y se crea la función “HandleResponse” para realizar las consultas dentro de la conexión de WhatsApp (Se puede ver el código completo en Anexos 1.5).

```
String HandleResponse(String consulta) //HandleResponse: gestiona la respuesta
de la llamada y si la llamada se realiza correctamente, analiza la respuesta
{
  if (consulta == "prendido") {
    digitalWrite(LED, 1);
    return "REALIZADO LUZ ENCENDIDA";
  }

  else if (consulta == "apagado") {
    digitalWrite(LED, 0);
    return "REALIZADO LUZ APAGADA";
  }

  else if (consulta == "estatus")
```

```
    return digitalRead(LED) ? "led encendida" : "led apagada";  
    else return "invalido";  
}  
  
void loop()  
{  
    thing.Handle(); //llama al cliente para la transferencia de datos  
}
```

Al compilar a la placa, se activa la línea y se establece la comunicación.



Imagen 46 Pantalla principal ThingESP en línea Fuente: Propia

Se comprueba la conexión.

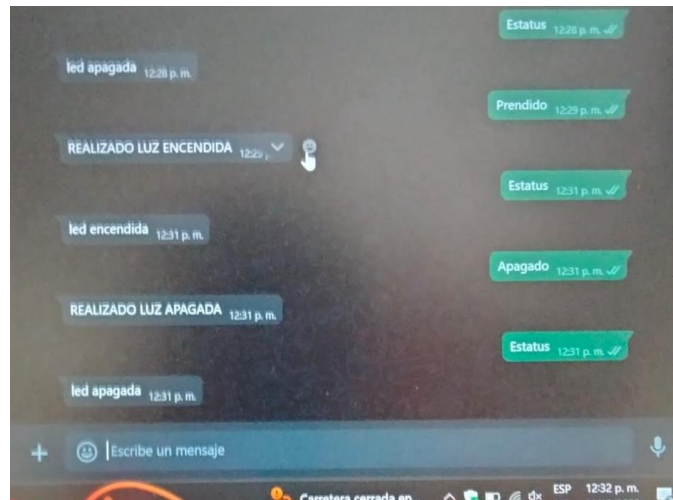


Imagen 47 Prueba conexión con WhatsApp Fuente: Propia

### 3.7 Programación placa KC868 con Telegram

Para conectar a telegram, primero se debe de crear un Bot, se hará uso de “BotFather”, el cual pedirá la siguiente información al utilizar el comando /newbot.

- Nombre del bot.
- Nombre de usuario (Terminación “\_bot”).

Al ingresar estos datos, BotFather brindara un Token, este sirve para la comunicación en la programación. Se debe de crear un archivo .h en el cual se ingresa las credenciales de la red wifi y ahí se ingresa el Token.



Se llaman las librerías necesarias.

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "CTBot.h"
```

Se inicializan las variables y se llama al archivo .h creado anteriormente.

```
CTBot miBot;
CTBotInlineKeyboard miTeclado;

#include "token_ejemplo.h"
uint8_t led = 2;
```

La función para la comunicación con telegram se llama TBMessage msg crea un teclado y la las consultas (Se puede ver el código completo en Anexos 1.6).

```
TBMessage msg;

if (miBot.getNewMessage(msg)) {
  if (msg.messageType == CTBotMessageTypeText) {
    if (msg.text.equalsIgnoreCase("opciones")) {
      miBot.sendMessage(msg.sender.id, "Cambiar Led", miTeclado);
    }
    else {
      miBot.sendMessage(msg.sender.id, "prueba 'opciones'");
    }
  }
}
```

```
} else if (msg.messageType == CTBotMessageQuery) {  
  Serial << "Mensaje: " << msg.sender.firstName;  
  if (msg.callbackQueryData.equals("encender")) {  
    Serial.println(" Encender");  
    digitalWrite(led, HIGH);  
    miBot.endQuery(msg.callbackQueryID, "Led Encendido", true);  
  } else if (msg.callbackQueryData.equals("apagar")) {  
    Serial.println(" Apagar");  
    digitalWrite(led, LOW);  
    miBot.endQuery(msg.callbackQueryID, "Led Apagado");  
  }  
}
```

Al compilar el programa en la placa e inicializar el bot, debe de responder con la primera opción establecida en la programación.

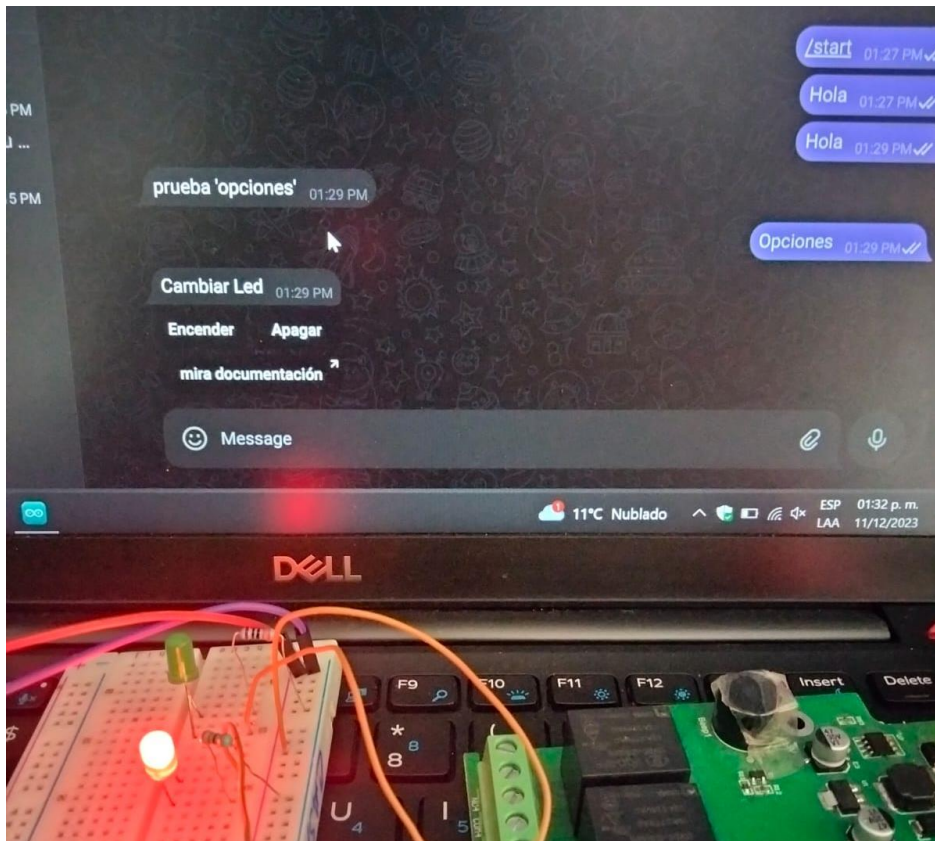


Imagen 48 Conexión con telegram Fuente: Propia

Si se presiona uno de los botones, el bot es capaz de mandar un mensaje alertando de la acción realizada.

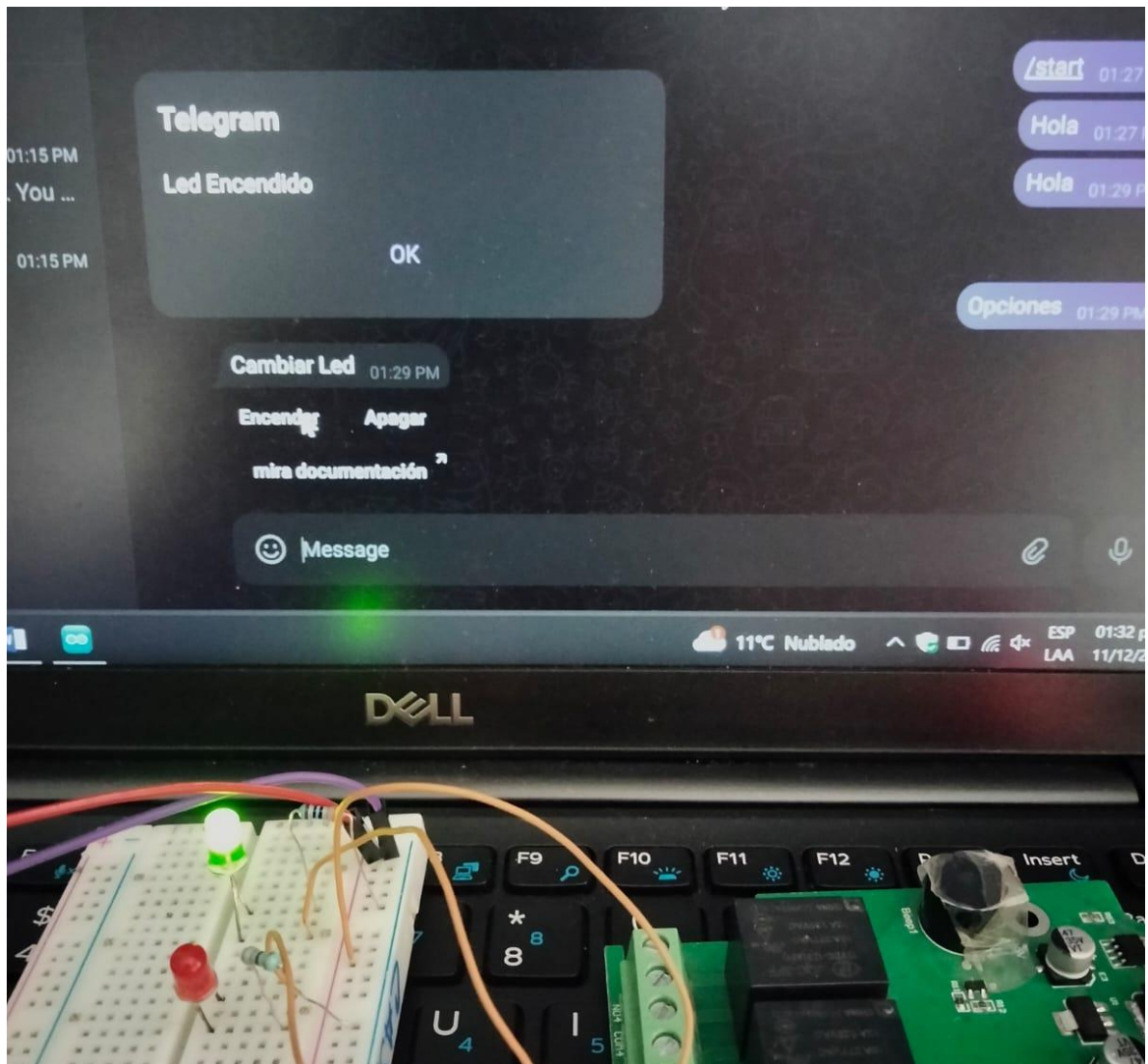


Imagen 49 Respuesta de telegram Fuente: Propia

### 3.8 Creación de la placa

Con las retroalimentaciones obtenidas en juntas para avances, se obtuvo la necesidad de crear un PCB con características específicas para satisfacer las necesidades del cliente, en una placa se encuentra la posibilidad de:

- Hasta 7 cortinas disponibles
- 1 alarma general
- 1 alarma interna
- 10 entradas digitales

Por lo que se diseñó la siguiente PCB.

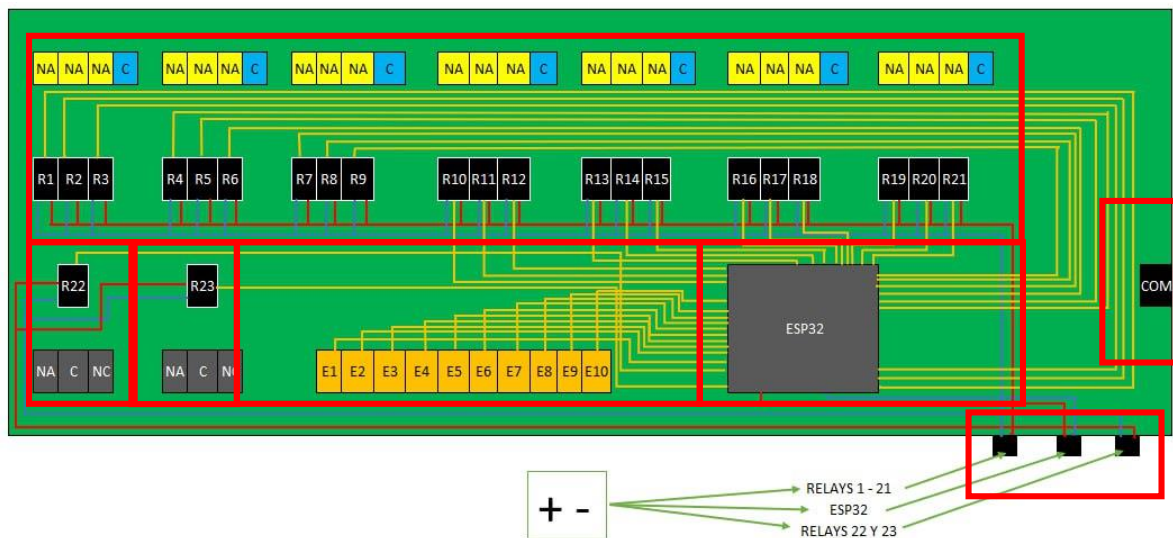


Imagen 50 PCB diseñada para el proyecto Fuente: Propia

La cual cuenta con:

- 23 relevadores
  1. 21 destinados a los focos y
  2. 1 alarma interna
  3. 1 alarma externa
  4. 10 entradas digitales
  5. 1 microcontrolador ESP32,
  6. Entrada de comunicación tipo C
  7. 3 entradas de energía para alimentar los relevadores de los focos, de las alarmas y alimentar al microcontrolador ESP32

### 3.9 Programación con placa KC868 E16S

Para entregar avances a los clientes, se adquirió la placa KC868 E16S, que es la placa más cercana a la creada con anterioridad.

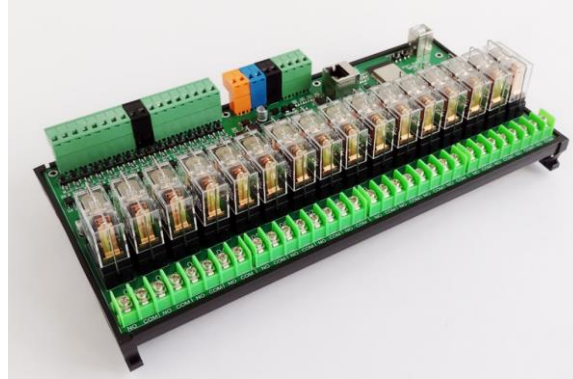


Imagen 51 Placa KC868 E16S Fuente: [https://es.aliexpress.com/item/1005004794816007.html?gps-id=platformRecommendH5ForSpider&pvid=2d8ec439-7269-428e-aaf9-c0a7fef06905&t=gps-id:platformRecommendH5ForSpider,pvid:2d8ec439-7269-428e-aaf9-c0a7fef06905,ttp\\_buckets:66](https://es.aliexpress.com/item/1005004794816007.html?gps-id=platformRecommendH5ForSpider&pvid=2d8ec439-7269-428e-aaf9-c0a7fef06905&t=gps-id:platformRecommendH5ForSpider,pvid:2d8ec439-7269-428e-aaf9-c0a7fef06905,ttp_buckets:66)

Esta placa función con 4 expansores de pines PCF854. Los cuales se declaran de la siguiente manera:

- PCF8574:U27 (relay1-8): 100 001 = 0x21
- PCF8574:U28 (relay9-16): 100 101 = 0x25
- PCF8574:U23 (DI1-8): 100 010 = 0x22
- PCF8574:U24 (DI9-16): 100 100 = 0x24

Los primeros dos PCF8574 pertenecen a los 16 relevadores y los otros pertenecen a las 16 entradas digitales o botones con los que cuenta la placa. Para funciones del proyecto la placa se divide la siguiente manera:

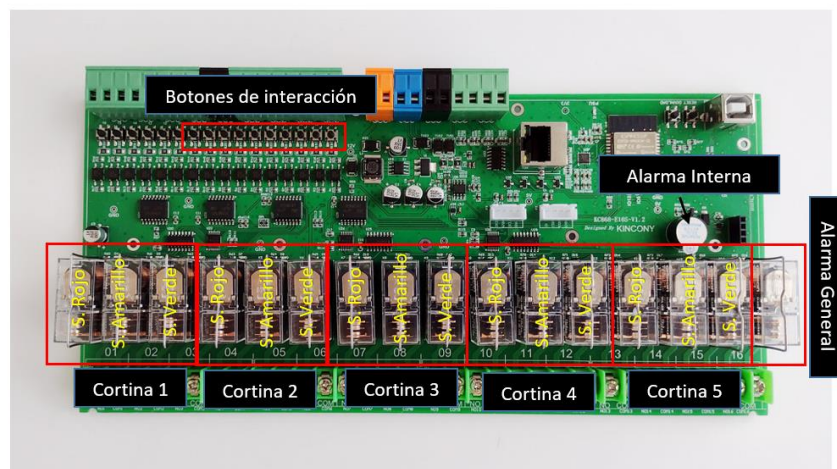


Imagen 52 Organización placa KC868 E16S para el proyecto Fuente: Propia

Al inicio se nombra la librería para establecer una comunicación I2C, posteriormente se definen las variables para la primera cortina, teniendo en consideración la regla para los PCF8574.

```
//Librerias
#include <Wire.h>

//Se defines las variables
#define PCF8574_U27_ADDRESS 0x21 // R1-8
#define PCF8574_U28_ADDRESS 0x25 // R9-16
#define PCF8574_U23_ADDRESS 0x22 // DI 1=8

//primer hilo
#define RELAY1_BIT 0
#define RELAY2_BIT 1
#define RELAY3_BIT 2

//primer hilo
#define BUTTON1_BIT 0
#define BUTTON2_BIT 1

//General
#define RELAY16_BIT 7 //Alarma general
#define BUZZER_PIN 2 //Alarma interna
```

Se definen funciones de tiempo.

```
//Funciones extra
unsigned long tiempoInicio = 0;
bool rele1Encendido = false;
bool boton2Presionado = false;
bool buzzerEncendido = false;
```

Se definen los estados iniciales y los pines del ESP32 para comunicación I2C (SDA Y SCL).

```
void setup() {
    //Inicia comunicacion I2C y definen estados iniciales
    Wire.begin(16, 15);
    pinMode(BUZZER_PIN, OUTPUT);
    cambiarEstadoBuzzer(LOW);
    cambiarEstadoRelay1(RELAY1_BIT, HIGH);
}
```

Se crean las variables de lectura para las entradas digitales.

```
int estadoBoton1 = leerEstadoBoton(BUTTON1_BIT, PCF8574_U23_ADDRESS);  
int estadoBoton2 = leerEstadoBoton(BUTTON2_BIT, PCF8574_U23_ADDRESS);
```

Se inicializa el primer relevador encendido (La luz roja).

```
//Se inicializa primer relevador (Luz roja está encendida)  
cambiarEstadoRelay1(RELAY1_BIT, rele1Encendido);
```

Este código comprueba si el botón 1 no está siendo presionado (estado LOW) y si el relé1 no está encendido. En caso afirmativo, ejecuta una serie de acciones:

- Apaga dos relés (RELAY1\_BIT y RELAY3\_BIT).
- Registra el tiempo actual utilizando la función millis().
- Establece rele1Encendido en true, indicando que el relé1 está activado en ese momento.

```
if (estadoBoton1 == LOW && !rele1Encendido) {  
    cambiarEstadoRelay1(RELAY1_BIT, LOW);  
    cambiarEstadoRelay1(RELAY3_BIT, LOW);  
    tiempoInicio = millis();  
    rele1Encendido = true;  
}
```

En esta parte del código, se verifica si el botón 2 no está siendo presionado (estado LOW). Si esta condición se cumple, se activa la variable boton2Presionado. Por otro lado, si boton2Presionado ya estaba activa (lo cual indica que el botón 2 fue presionado anteriormente), se desactiva y se registra el tiempo actual utilizando la función millis().

```
if (estadoBoton2 == LOW) {  
    boton2Presionado = true;  
} else if (boton2Presionado) {  
    boton2Presionado = false;  
    tiempoInicio = millis();  
}
```

Se verifica si el boton2 está siendo oprimido, si es el caso, el contador cambia de 10 segundos a 20 segundos.

```
    if (rele1Encendido && millis() - tiempoInicio > (boton2Presionado ? 20000 : 10000)) {  
        cambiarEstadoRelay1(RELAY3_BIT, HIGH);  
        cambiarEstadoRelay1(RELAY1_BIT, HIGH);  
        rele1Encendido = false;  
    }
```

Esta sección controla la alarma y el parpadeo con respecto a las condicionales.

```
if(boton2Presionado && rele1Encendido == false) {  
    cambiarEstadoBuzzer(HIGH);  
    buzzerEncendido = true;  
    activarAlarm(RELAY16_BIT);  
    parpadearRelay(RELAY2_BIT);  
    delay(50);  
} else if(boton2Presionado && millis() - tiempoInicio > 20000){  
    cambiarEstadoBuzzer(HIGH);  
    buzzerEncendido = true;  
    delay(50);  
}  
else {  
    if (buzzerEncendido) {  
        cambiarEstadoBuzzer(LOW);  
        buzzerEncendido = false;  
        desactivarAlarm(RELAY16_BIT);  
    }  
}
```

Cuando falten 5 segundos para que acabe el contador de 20 segundos el parpadeo iniciara.

```
//Control parpadeo cuando faltan 15 segundos  
if(boton2Presionado && millis() - tiempoInicio > 15000){  
    parpadearRelay(RELAY2_BIT);  
    delay(50);  
}  
delay(50);  
} //FIN LOOP
```

Para leer el estado de los botones, se crea la siguiente función.

```
//Lee los bytes de comunicacion I2C
int leerEstadoBoton(int bit, int direccion) {
    Wire.requestFrom(direccion, 1);
    return bitRead(Wire.read(), bit);
}
```

Estas funciones sirven para cambiar de estado los relevadores.

```
//Cambia de estado el primer relevador
void cambiarEstadoRelay1(int bit, byte nuevoEstado) {

    Wire.requestFrom(PCF8574_U27_ADDRESS, 1);
    byte bits = Wire.read();
    bitWrite(bits, bit, nuevoEstado);
    Wire.beginTransmission(PCF8574_U27_ADDRESS);
    Wire.write(bits);
    Wire.endTransmission();
}

//Cambia de estado el segundo relevador
void cambiarEstadoRelay2(int bit, byte nuevoEstado) {
    Wire.requestFrom(PCF8574_U28_ADDRESS, 1);
    byte bits = Wire.read();
    bitWrite(bits, bit, nuevoEstado);
    Wire.beginTransmission(PCF8574_U28_ADDRESS);
    Wire.write(bits);
    Wire.endTransmission();
}
```

Las siguientes funciones son complementarias para el funcionamiento óptimo del código, cambia de estado las alarmas.

```
void cambiarEstadoBuzzer(byte nuevoEstado) {
    digitalWrite(BUZZER_PIN, nuevoEstado);
}

void parpadearRelay(int bit) {
    cambiarEstadoRelay1(bit, LOW);
    delay(500); // Ajusta el tiempo del parpadeo según tus necesidades
    cambiarEstadoRelay1(bit, HIGH);
}

void activarAlarm(int bit){
```

```

    if(boton2Presionado == true){
        cambiarEstadoRelay2(bit,LOW);
    }
}
void desactivarAlarm(int bit){
    if(boton2Presionado == false){
        cambiarEstadoRelay2(bit,HIGH);
    } }

```

Para las 4 cortinas faltante es la misma lógica, solo cambia el nombre de las variables y su asignación en PCF8574 (Se puede ver el código completo en Anexos 1.7).

Para el funcionamiento del código en la placa, se debe hacer las conexiones de la siguiente manera. (El circuito completo se puede ver en Anexos 1.8).

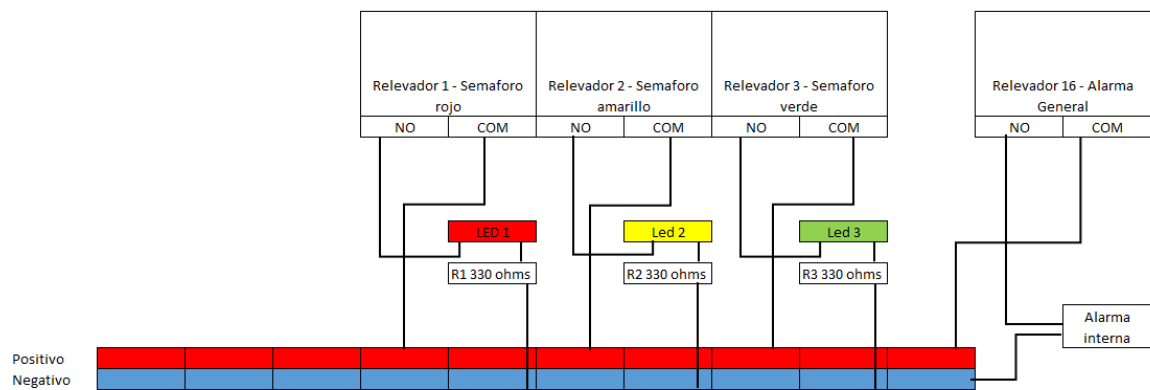


Imagen 53 Circuito para primer cortina y alarma general Fuente: Propia

## **CAPÍTULO IV**

### **RESULTADOS**

## 4.1 Resultados de la simulación

Al correr la simulación, el primer resultado son los semáforos en color rojo, los cuales indican que los relevadores se encuentran apagados, lo que indica que las cortinas se encuentran cerradas.

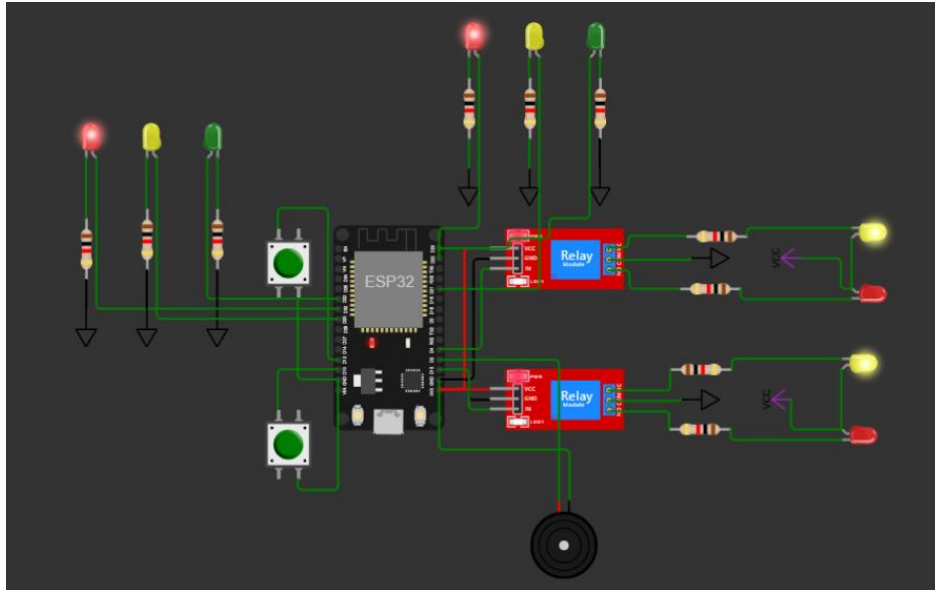


Imagen 54 Simulación cortinas cerradas Fuente: Propia

Posteriormente, al presionar el botón de apertura, el semáforo pasa a color amarillo perteneciente a este empezara a parpadear, indicando que hay permiso de apertura.

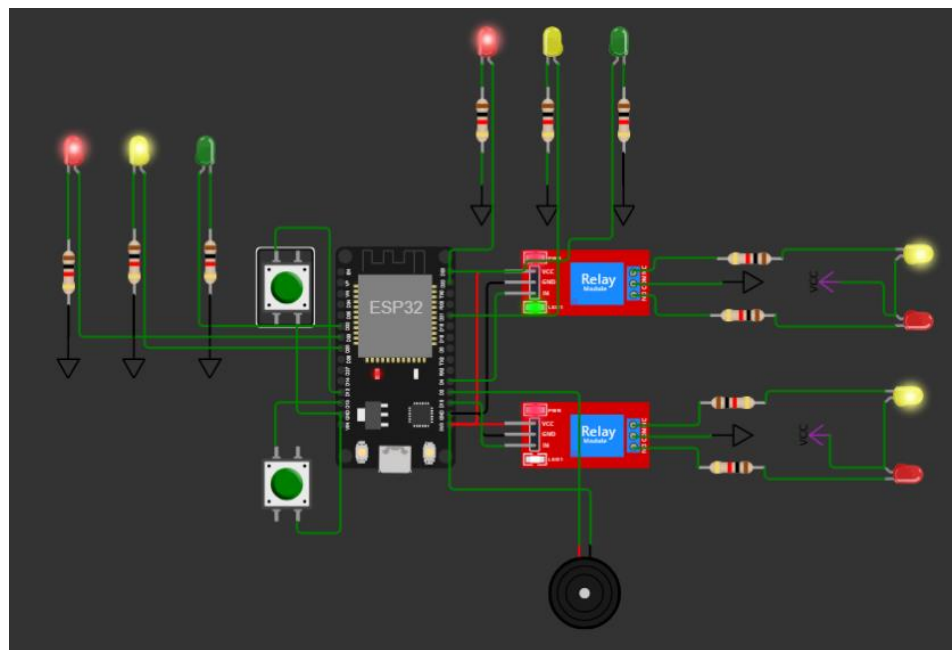


Imagen 55 Simulación permiso de apertura cerradas Fuente: Propia

Al terminar el tiempo de permiso de apertura el semáforo pasar a color verde, lo cual indica que los relevadores cambiaron de estado o que la cortina está abierta.

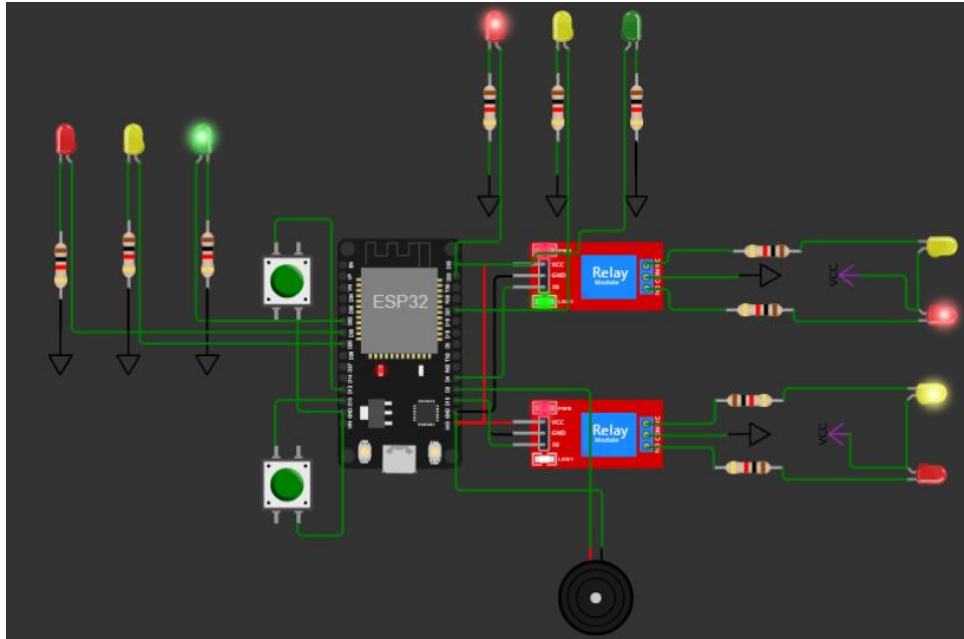


Imagen 56 Simulación cortina abierta cerradas Fuente: Propia

El mismo procedimiento pasa para la cortina 2.

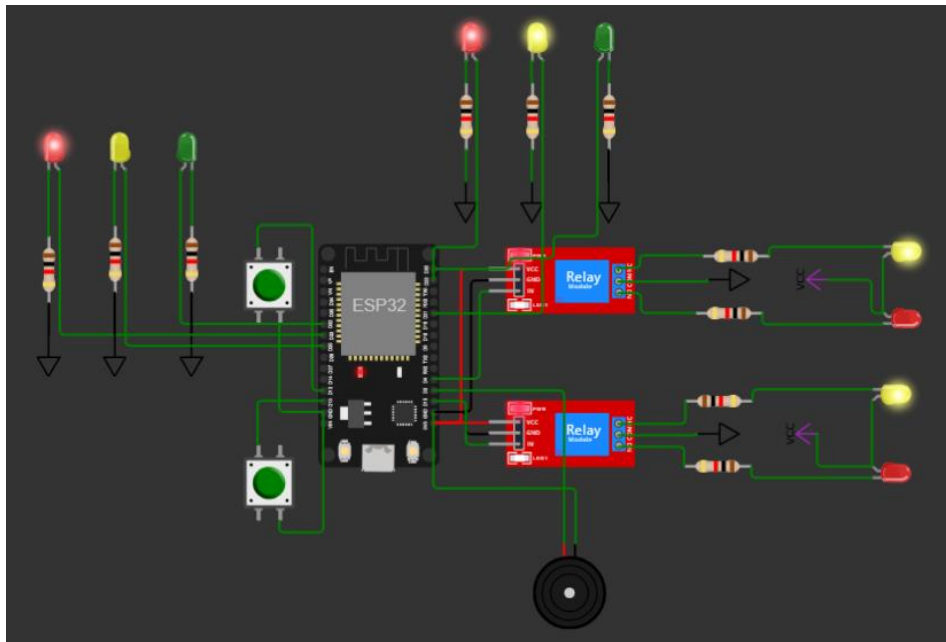


Imagen 57 Simulación semáforo amarillo cortina 2 cerradas Fuente: Propia

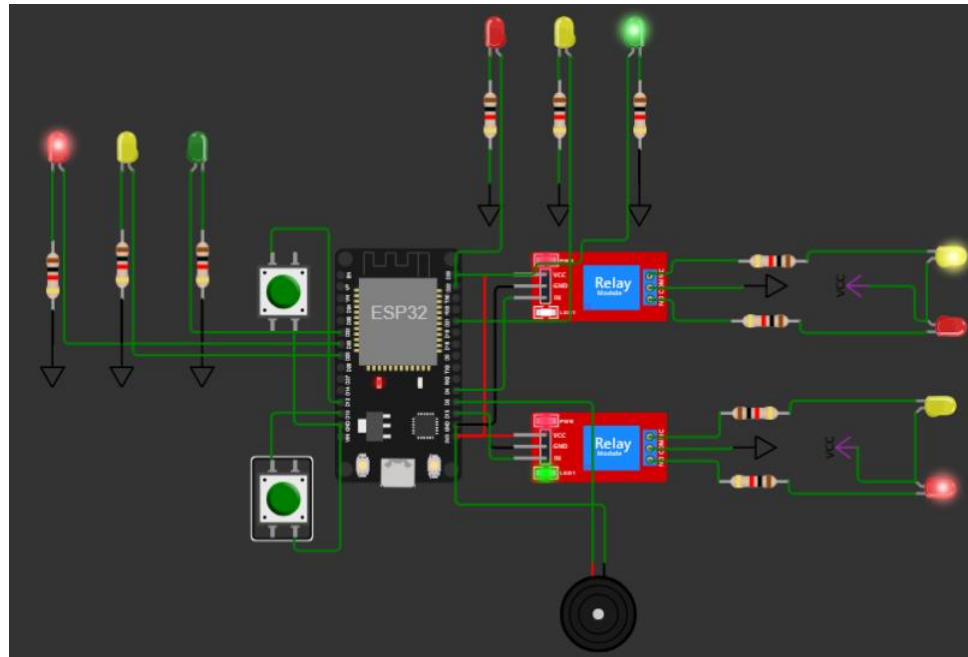


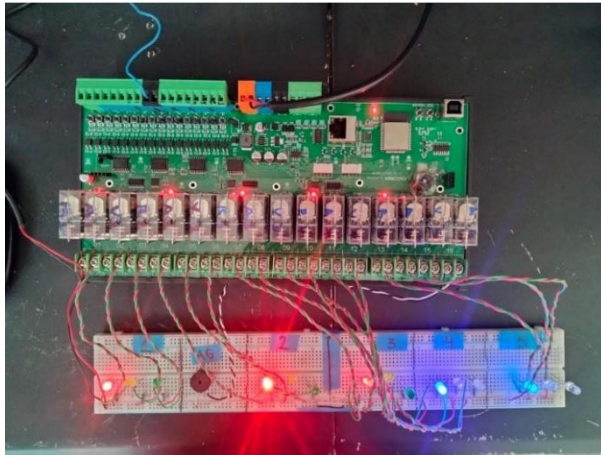
Imagen 58 Simulación semáforo verde cortina 2 cerradas Fuente: Propia

## 4.2 Resultados de la programación en la placa KC868 E16S

Por problemas de tiempo y logística la placa especificada para cumplir con las necesidades del proyecto no llegara en el tiempo estipulado de Residencias profesionales, sin embargo, las bases del proyecto se construyeron en la Placa KC868 E16S, teniendo como resultado los siguientes:

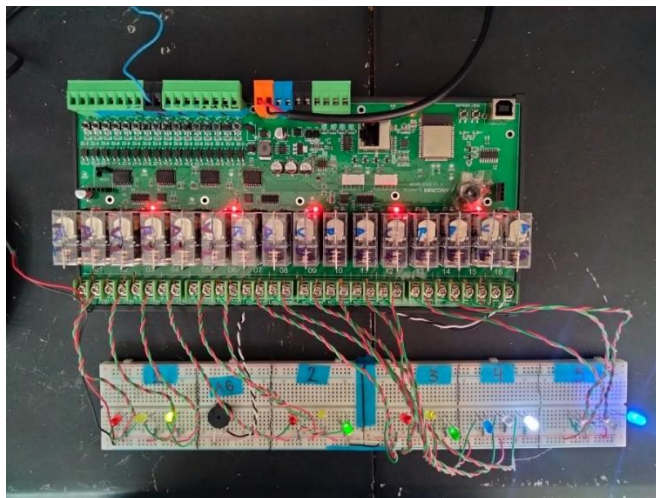
Teniendo en consideración que los tiempos utilizados son para fines prácticos, cuando se implemente se ajustarán a las necesidades del cliente.

La placa KC868 E16S soporta 5 cortinas, 1 alarma general y 1 alarma interna.



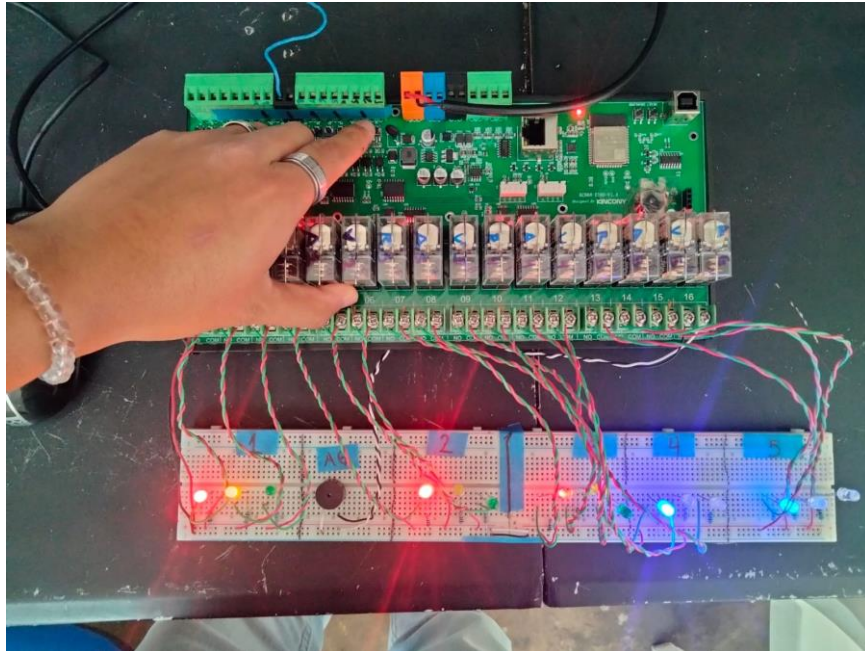
*Imagen 59 Simulación de cortina abajo led rojo encendido Fuente: Propia*

Al presionar los botones de permiso de apertura, los relevadores pertenecientes a la luz roja se apagan y se enciende los pertenecientes a la luz verde.



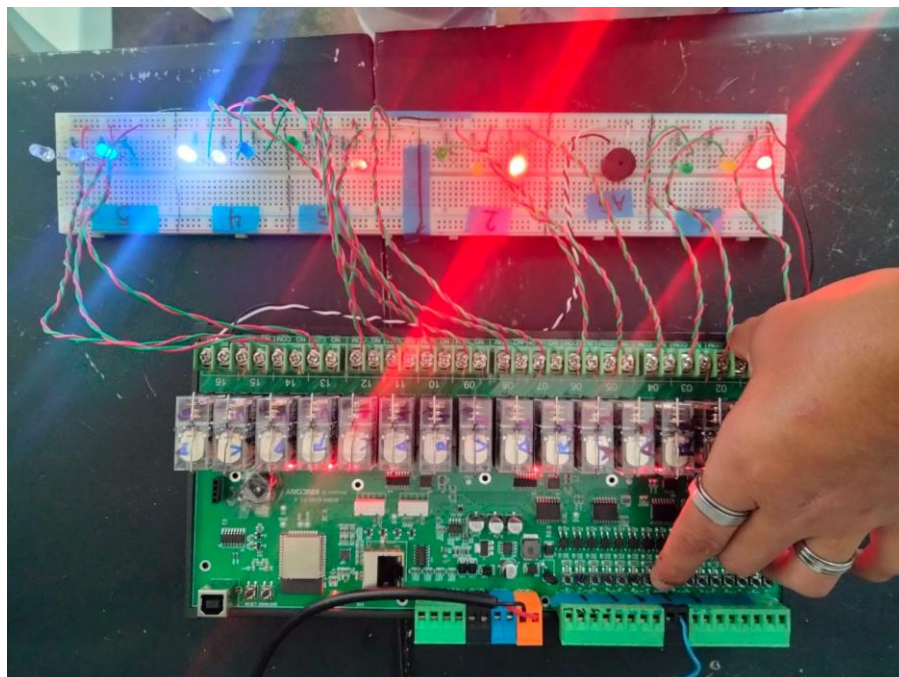
*Imagen 60 Simulación de cortina arriba led verde encendido Fuente: Propia*

Al presionar el botón de apertura sin dar permiso la alarma se activa y se activa el relevador de la luz amarilla correspondiente a la cortina que se intenta abrir.



*Imagen 61 Cortina 1 abierta sin permiso Led amarillo encendido Fuente: Propia*

Al habilitar la cortina y presionar el botón de apertura, cuando falten 5 segundos para que se acabe el tiempo el relevador de la luz amarilla iniciara a parpadear como aviso.



*Imagen 62 Cortina 4 abierta, alerta que se aproxima su cierre Fuente: Propia*

### 4.3 Diseño de ubicación de la placa KC868 E16S en cortinas

Para empezar a realizar prueba, se diseñó un esquema para 2 cortinas industriales, en el cual la placa se encuentra al centro para que la distancia del cableado sea el mismo, se conectan 2 semáforos junto con la alarma general.

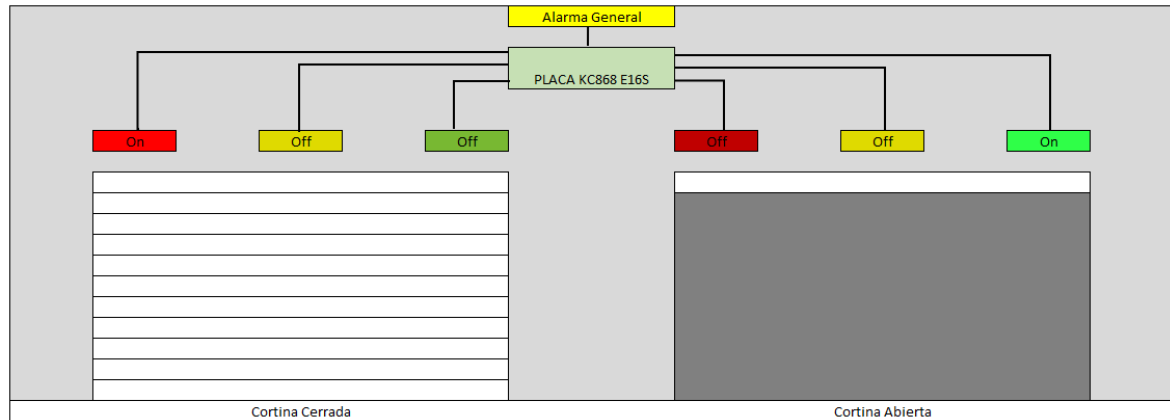


Imagen 63 Diseño de ubicación para dos cortinas Fuente: Propia

Posteriormente se realiza el diseño para conectar las 5 cortinas que tiene como límite la placa, junto con la alarma general, se decide colocar en medio para tener un mejor control en la distribución.

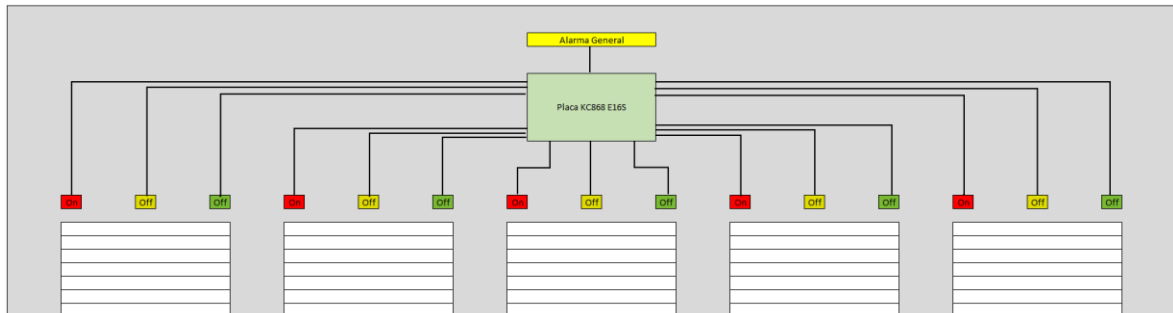
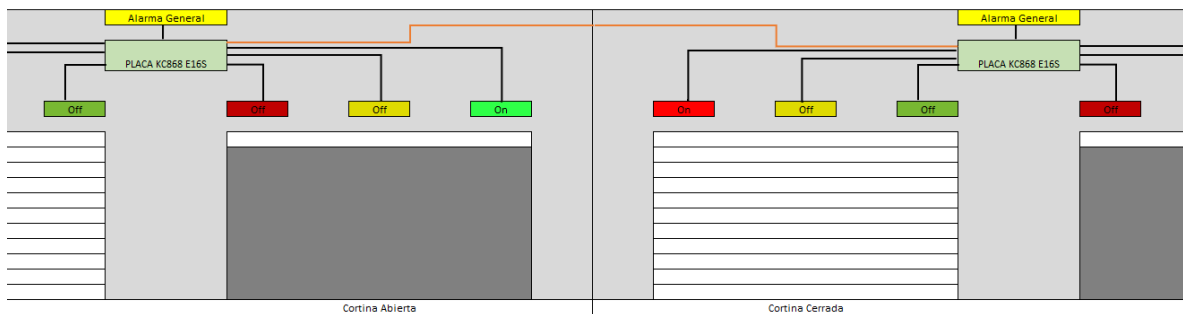


Imagen 64 Diseño de ubicación para cinco cortinas Fuente: Propia

Para la conexión entre 2 placas, se retoma el primer diseño, el cual se tiene considerado conectar vía Ethernet, se tomaron aspectos como:

- Velocidad y Rendimiento.
- Estabilidad de la Conexión.
- Menor Latencia.
- Menor Interferencia.
- Mayor Ancho de Banda.
- Menor Consumo de Energía.



*Imagen 65 Diseño de la comunicación con 2 placas Fuente: Propia*

#### 4.4 Cronograma esperado para implementación

Retomando los problemas de tiempo y logística de la placa, se crea un cronograma para la implementación y pruebas del proyecto, para apoyo con el seguimiento de las actividades faltantes a lo largo del proyecto, considerando que, si hay atrasos con alguna de las actividades, la entrega final puede postergarse.

*Tabla 8 Cronograma estimado para entrega de proyecto*

|                                                | Marzo | Abril | Mayo | Junio | Julio | Agosto | Septiembre |
|------------------------------------------------|-------|-------|------|-------|-------|--------|------------|
|                                                |       |       |      |       |       |        |            |
| Llegada de la placa solicitada                 |       |       |      |       |       |        |            |
| Configuración del código para nueva placa      |       |       |      |       |       |        |            |
| Implementación del código                      |       |       |      |       |       |        |            |
| Pruebas físicas internas                       |       |       |      |       |       |        |            |
| Implementación y pruebas en primeras cortinas. |       |       |      |       |       |        |            |
| Seguimiento de las demás cortinas.             |       |       |      |       |       |        |            |
| Entrega proyecto.                              |       |       |      |       |       |        |            |

### 4.5 Propuestas

Después de un análisis detallado del diseño esperado para configuración de red basada en Ethernet, se sugiere la consideración de utilizar conexiones Wi-Fi como una medida estratégica para optimizar la eficiencia dentro del entorno de trabajo. Los detalles a mejorar son:

#### **Simplicidad en la Instalación y Configuración:**

Eliminar la necesidad de cables Ethernet simplificará la instalación y la reconfiguración de dispositivos, reduciendo la complejidad logística y los costos asociados con la infraestructura física. La eliminación de cables redundantes contribuirá a una estética más limpia y accesible en el entorno de trabajo, promoviendo un espacio más ordenado y eficiente.

#### **Adaptabilidad a Cambios en la Distribución de Espacios:**

La conexión Wi-Fi ofrecerá una mayor flexibilidad para acomodar cambios en la disposición de estaciones de trabajo, permitiendo una adaptación rápida y eficiente a las necesidades específicas de cada proyecto.

#### **Eficiencia Mejorada:**

Con los avances tecnológicos en Wi-Fi, la velocidad y la confiabilidad de las conexiones inalámbricas han mejorado significativamente, lo que resultará en una mayor eficiencia operativa y una transferencia de datos más rápida.

Además, se propone hacer uso de una base de datos, para un manejo estructurado y organizado de los eventos generados, permitiendo guardar y/o recuperar los datos necesarios para poder realizar búsquedas, consultas, las cuales puede ser cruciales al realizar un análisis de datos y generar informes. Todo esto puede respaldar la toma de decisiones conforme a lo planeado dentro del proyecto.

Las bases de datos recomendadas son:

#### **Firebase Realtime Database o Firestore:**

Firebase, que es parte de la plataforma de desarrollo de Google, ofrece bases de datos en tiempo real y basadas en documentos (Firestore).

#### **SQLite:**

Es una base de datos SQL ligera y sin servidor que es fácil de integrar en proyectos embebidos.

## CONCLUSIONES

A pesar de alcanzar metas específicas, se identificaron notables limitaciones, como problemas logísticos en la entrega de materiales, destacando la necesidad de considerar ajustes o mejoras en proyectos posteriores.

Es esencial resaltar que, más allá de los objetivos establecidos, se obtuvieron resultados adicionales significativos, como el desarrollo de habilidades blandas, la experiencia en equipos multidisciplinarios, la capacidad para trabajar bajo presión y el estímulo del aprendizaje autodidacta. Estos hallazgos inesperados abren nuevas oportunidades para proyectos futuros.

Adicionalmente, se reconocen las excepciones y omisiones en los resultados, proporcionando valiosos conocimientos sobre las limitaciones inherentes al enfoque metodológico empleado.

A pesar de los desafíos encontrados, los resultados del proyecto contribuyen de manera sustancial al conocimiento del tema, estableciendo así una base sólida para futuras implementaciones o mejoras en proyectos similares.

## COMPETENCIAS DESARROLLADAS

Tabla 9 Competencias desarrolladas

| Materia:                                       | Competencia:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fundamentos de programación                    | <ul style="list-style-type: none"> <li>➤ Comprende y aplica los conceptos básicos, nomenclatura y herramientas para el diseño de algoritmos orientado a la resolución de problemas.</li> <li>➤ Conoce y aplica un lenguaje de programación para la resolución de problemas.</li> <li>➤ Conoce y aplica las estructuras condicionales y repetitivas de un lenguaje de programación para resolver problemas reales.</li> <li>➤ Conoce y aplica la modularidad en el desarrollo de programas para la optimización de los mismos y reutilización de código.</li> </ul>                                                                                                                                                        |
| Gestión de proyectos                           | <ul style="list-style-type: none"> <li>➤ Conoce y comprende el entorno de la gestión de proyectos.</li> <li>➤ Planifica un proyecto de software utilizando una metodología de trabajo para determinar su viabilidad.</li> <li>➤ Identifica y selecciona propuestas de proyecto de software para la presentación de un contrato al cliente.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                     |
| Ingeniería de software                         | <ul style="list-style-type: none"> <li>➤ Abstrae información del usuario final para elaborar el análisis de requerimientos del software a desarrollar.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Principios eléctricos y aplicaciones digitales | <ul style="list-style-type: none"> <li>➤ Soluciona problemas que engloben escenarios de circuitos eléctricos para calcular parámetros en base a leyes y teoremas.</li> <li>➤ Maneja instrumentos y equipos de medición eléctricos</li> <li>➤ Identifica las características de los dispositivos electrónicos utilizando hoja de especificaciones</li> <li>➤ Simula circuitos analógicos para su comprensión, demostrando los conceptos vistos en clase.</li> <li>➤ Aplica técnicas de simplificación para construir circuitos con el mínimo costo.</li> <li>➤ Comprende y aplica las herramientas básicas de análisis de los sistemas analógicos y digitales para resolver problemas del ámbito computacional.</li> </ul> |

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Programación lógica y funcional   | <ul style="list-style-type: none"> <li>➤ Conoce los principios lógicos y funcionales de la programación para aplicarlos en la resolución de problemas.</li> <li>➤ Identificar los paradigmas de los lenguajes de programación.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Sistemas programables             | <ul style="list-style-type: none"> <li>➤ Aplica microcontroladores en el diseño de interfaces hombre-máquina y máquina-máquina de sistemas programables.</li> <li>➤ Aplica principios físicos y comprende transductores y sensores</li> <li>➤ Identifica las características eléctricas de un microcontrolador.</li> <li>➤ Conoce la arquitectura interna del microcontrolador.</li> <li>➤ Comprende la estructura de registros del microcontrolador.</li> <li>➤ Analiza dispositivos de entrada/salida y puertos del microcontrolador.</li> <li>➤ Programa microcontroladores utilizando puertos de E/S.</li> <li>➤ Construye y comprueba circuitos con microcontrolador.</li> <li>➤ Diseña y aplica interfaces hombre-máquina y máquina-máquina</li> </ul> |
| Taller de investigación I         | <ul style="list-style-type: none"> <li>➤ Diseña un protocolo de investigación.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Taller de investigación II        | <ul style="list-style-type: none"> <li>➤ Analiza y conforma la actualización del protocolo de investigación para darle seguimiento.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Tópicos avanzados de programación | <ul style="list-style-type: none"> <li>➤ Diseña e implementa componentes y librerías para lograr la reutilización de código.</li> <li>➤ Crea subprogramas para resolver problemas concurrentes utilizando Multihilos.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Ingeniería de requerimientos      | <ul style="list-style-type: none"> <li>➤ Comprende el modelado de negocios, la importancia de la creación de modelos para entender, comunicar y analizar la complejidad de las organizaciones</li> <li>➤ Aplica los criterios para determinar la naturaleza de los Requerimientos.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## FUENTES DE INFORMACIÓN

- *About Espressif | Espressif Systems.* (s. f.). <https://www.espressif.com/en/company/about-espressif>
- Admin, & Admin. (2022, 15 febrero). KC868-A4 Hardware Design Details - Smart Home Automation | KinCony. *Smart Home Automation | KinCony - Home Automation DIY.*  
<https://www.kincony.com/kc868-a4-hardware-design-details.html>
- Aronowitz, A. (2021). *Programación LUA: La guía para principiantes definitiva para aprender Lua paso a paso.* Independently Published.
- Carmenate, J. G. (2022, 7 marzo). ESP32 Wifi + Bluetooth en un solo lugar. *Programarfacil Arduino y Home Assistant.*  
[https://programarfacil.com/esp8266/esp32/#Conclusion\\_sobre\\_el\\_chip\\_ESP32](https://programarfacil.com/esp8266/esp32/#Conclusion_sobre_el_chip_ESP32)
- Castillo, J. A. (2023, 9 octubre). *Qué es una PCB o placa de circuito impreso. uso, cómo se fabrica.* Profesional Review. <https://www.profesionalreview.com/2019/02/11/pcb-que-es/>
- *Comparación entre un microcontrolador y un microprocesador. Diferencia entre chips de computadora centralizados: AWS.* (s. f.). Amazon Web Services, Inc.  
<https://aws.amazon.com/es/compare/the-difference-between-microprocessors-microcontrollers/>
- *El tutorial de Python.* (s. f.). Python documentation. <https://docs.python.org/es/3/tutorial/>
- *Electrónica Rayte SL.* (s. f.). <https://rayte.com/blog/post/5-tipos-de-arduino>
- *ESP-32 - Sigma Electrónica.* (2023, 9 octubre). Sigma Electrónica.  
<https://www.sigmaelectronica.net/producto/esp-32/>
- *ESP32 ¿Qué es y para qué sirve? - programador novato.* (2022, 9 septiembre). Programador Novato. <https://www.programadornovato.com/esp32-que-es-y-para-que-sirve/#Caracteristicas>
- *Espressif.* (s. f.). <https://www.eurotronix.com/es/fabricantes/espressif>

- Fundación Aquae. (2020, 30 octubre). *¿Sabes qué es un arduino y para qué sirve?* - Fundación AQUAE. <https://www.fundacionaquae.org/wiki/sabes-arduino-sirve/>
- Gallardo, A. R., Herrera-Morales, J. R., Carrillo, S. S., Andrade-Aréchiga, M., & Michel, E. M. R. (2023). *Internet de las cosas* (1.<sup>a</sup> ed.). enfoque academico. <https://doi.org/10.53897/li.2023.0001.ucol>
- Garrote, D. (2022). IOT en la industria 4.0 ¿Qué papel juega actualmente? *Nuclio Digital School*. <https://nuclio.school/el-iot-en-la-industria-4/#Aplicaciones-de-IoT-en-diferentes-sectores>
- Ghostapi. (2023, 26 marzo). Historia del Internet a través de su línea de tiempo - Historia sobre. *LetsTrip*. <https://historiasobre.com/historia-del-internet-a-traves-de-su-linea-de-tiempo/>
- Gutiérrez, Á. P. H. (2022). *El lenguaje de programación Python*.
- *Historia del Internet de las cosas*. (2021, 8 octubre). BBVA.CH. <https://www.bbva.ch/noticia/historia-del-internet-de-las-cosas/>
- Kim, D. (2023). *Lua para principiantes: Aprenda a programar con ejemplos prácticos*. Independently Published.
- Lequerica, J. R. (2016). *Arduino para jóvenes y no tan jóvenes*. ANAYA MULTIMEDIA.
- *Manual de referencia de LUA 5.1*. (s. f.). <https://www.lua.org/manual/5.1/es/manual.html>
- Matuk, U.-. J. (2019, 14 mayo). *Esto era COMDEX, la expo más influyente de la tecnología por casi dos décadas*. unocero. <https://www.unocero.com/entretenimiento/comdex-antecesor-ces/>
- Morales, J. A. R. (2022, 12 febrero). *ESP32 características y pines*. PASIÓN ELECTRÓNICA. <https://pasionelectronica.com/esp32-caracteristicas-y-pines/>
- *NodeMCU-32 30-pin ESP32 WiFi*. (s. f.). Naylamp Mechatronics - Perú. <https://naylampmechatronics.com/espressif-esp/384-nodemcu-32-30-pin-esp32-wifi.html>
- Pérez, F. E. V., & Areny, R. P. (2007). *Microcontroladores: fundamentos y aplicaciones con PIC*.

- Peterson, Z. (2023, 15 julio). *¿Qué es un PCB o placa de circuito impreso?* Altium.  
<https://resources.altium.com/es/p/what-is-a-pcb>
- Santos, M. D. (2023, 24 marzo). *Lenguajes de programación compatibles con ESP32 - Polaridad.es*. Polaridad.es. <https://polaridad.es/lenguajes-de-programacion-compatibles-con-esp32/>
- Timetoast. (1969). *Evolución e historia del Internet (1969-2020) Timeline*. Timetoast timelines.  
<https://www.timetoast.com/timelines/evolucion-e-historia-del-internet-1969-2020>
- Usategui, J. M. A., & Martínez, I. A. (2003). *Microcontroladores PIC: diseño práctico de aplicaciones: primera parte, el PIC 16F84, lenguajes PBASIC y Ensamblador*.

## ANEXOS

### Anexo 1. Autorización de código



Intelligent Buildings & Security

México D.F. a 15 de abril de 2024.

**At'n: A quien corresponda**

Tecnológico de Estudios Superiores de Coacalco.

A solicitud de su estudiante el Sr. Mario Gómez Mendicuti de la carrera de **Ingeniería en sistemas computacionales** con matrícula **201920425** contratado para el departamento de innovación y desarrollo, referente a la publicación del código de programación de uno de nuestros proyectos en su proyecto de titulación de ingeniería, nos vemos en la penosa necesidad de No autorizar la publicación de todo el código, debido a que es un sistema de seguridad y el sistema podría quedar vulnerable tanto ataques cibernéticos como de hardware, sin embargo podemos permitir la publicación de ciertas partes del código, sin mencionar el cliente ni el lugar de la instalación, además de no permitir las fotos de la instalación final, podemos autorizar la publicación del código de un hilo de la programación de la versión Beta, con el funcionamiento de los subsistemas que componen el proyecto.

Sin más por el momento quedo de usted para cualquier duda o comentario.



Intelligent Buildings & Security

DISTRIBUIDORA DE TECNOLOGÍA  
ESPECIALIZADA CJ, S.A.P.I. DE C.V.

**R.F.C.: DTE151215UT0**

Laurel casa 1, Los Reyes Ixtacala,  
Tlalnepantla, Edo. de México. C.P. 54090

**Carlos Méncos García.**  
**Director de Operaciones**  
**ASPI Systems.**

## Anexo 2. Glosario

- **Expansor de pines:** Dispositivo que permite aumentar el número de pines disponibles en un microcontrolador para conectar más dispositivos. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Internet:** Red global de computadoras que se conectan entre sí para compartir información y recursos. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Internet de las cosas:** Concepto que se refiere a la interconexión de dispositivos físicos a través de Internet para recopilar y compartir datos. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Microcontrolador:** Dispositivo integrado que incluye un procesador, memoria y periféricos, utilizado para controlar otros dispositivos. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Ohms:** Unidad de medida de la resistencia eléctrica en un circuito. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **PCB (Placa de Circuito Impreso):** Tablero que sostiene componentes electrónicos y conecta circuitos mediante trazas conductoras. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Relevador:** Dispositivo electromagnético que controla la apertura o cierre de un circuito eléctrico. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Resistencia:** Componente electrónico que limita el flujo de corriente en un circuito. (Gran Diccionario Enciclopédico Ilustrado, 1981).
- **Sistema embebido:** Sistema informático integrado en un dispositivo más grande, diseñado para realizar funciones específicas. (Gran Diccionario Enciclopédico Ilustrado, 1981).

### Anexo 3. Código para simulador

```
#include <Arduino.h>

const int botonPin1 = 12; // GPIO D12 para el primer botón
const int relePin1 = 4;   // GPIO D4 para el primer relé
const int semaforoRojo1 = 33; // GPIO D33 para el LED rojo del semáforo del
primer relé
const int semaforoVerde1 = 32; // GPIO D32 para el LED verde del semáforo del
primer relé
const int semaforoAmarillo1 = 25; // GPIO D31 para el LED amarillo del semáforo
del primer relé

const int botonPin2 = 13; // GPIO D13 para el segundo botón
const int relePin2 = 15;   // GPIO D15 para el segundo relé
const int semaforoRojo2 = 22; // GPIO D22 para el LED rojo del semáforo del
segundo relé
const int semaforoVerde2 = 23; // GPIO D23 para el LED verde del semáforo del
segundo relé
const int semaforoAmarillo2 = 21; // GPIO D30 para el LED amarillo del semáforo
del segundo relé

const int zumbadorPin = 2; // GPIO D2 para el zumbador compartido por ambos
semáforos

bool estadoRele1 = LOW; // Estado inicial del primer relé (Comúnmente Cerrado)
bool estadoRele2 = LOW; // Estado inicial del segundo relé (Comúnmente Cerrado)

unsigned long tiempoInicio1 = 0; // Tiempo de inicio del primer relé activado
unsigned long tiempoInicio2 = 0; // Tiempo de inicio del segundo relé activado

bool procesoActivo1 = false; // Variable para controlar si un proceso del primer
relé está activo
bool procesoActivo2 = false; // Variable para controlar si un proceso del
segundo relé está activo

void setup() {
    pinMode(botonPin1, INPUT_PULLUP); // Configura el primer botón como entrada
con pull-up
    pinMode(relePin1, OUTPUT);         // Configura el primer relé como salida
    pinMode(semaforoRojo1, OUTPUT);    // Configura el LED rojo del semáforo del
primer relé como salida
    pinMode(semaforoVerde1, OUTPUT);   // Configura el LED verde del semáforo del
primer relé como salida
```

```

    pinMode(semaforoAmarillo1, OUTPUT); // Configura el LED amarillo del semáforo
del primer relé como salida
    digitalWrite(relePin1, estadoRele1); // Establece el estado inicial del primer
relé
    digitalWrite(semaforoRojo1, HIGH); // Inicialmente, enciende el LED rojo del
semáforo 1
    digitalWrite(semaforoVerde1, LOW); // Inicialmente, apaga el LED verde del
semáforo
    digitalWrite(semaforoAmarillo1, LOW); // Inicialmente, apaga el LED amarillo
del semáforo 1

    pinMode(botonPin2, INPUT_PULLUP); // Configura el segundo botón como entrada
con pull-up
    pinMode(relePin2, OUTPUT);          // Configura el segundo relé como salida
    pinMode(semaforoRojo2, OUTPUT);     // Configura el LED rojo del semáforo del
segundo relé como salida
    pinMode(semaforoVerde2, OUTPUT);    // Configura el LED verde del semáforo del
segundo relé como salida
    pinMode(semaforoAmarillo2, OUTPUT); // Configura el LED amarillo del semáforo
del segundo relé como salida
    digitalWrite(relePin2, estadoRele2); // Establece el estado inicial del
segundo relé
    digitalWrite(semaforoRojo2, HIGH); // Inicialmente, enciende el LED rojo del
semáforo 2
    digitalWrite(semaforoVerde2, LOW); // Inicialmente, apaga el LED verde del
semáforo 2
    digitalWrite(semaforoAmarillo2, LOW); // Inicialmente, apaga el LED amarillo
del semáforo 2

    pinMode(zumbadorPin, OUTPUT); // Configura el zumbador como salida
    digitalWrite(zumbadorPin, LOW); // Apaga el zumbador inicialmente
}

void cambiarEstadoRelay(int relePin, bool &estadoRelay) {
    estadoRelay = !estadoRelay;
    digitalWrite(relePin, estadoRelay);
}

void loop() {
    int estadoBoton1 = digitalRead(botonPin1);
    int estadoBoton2 = digitalRead(botonPin2);

    if (!procesoActivo1) {
        if (estadoBoton1 == LOW) {

```

```

        digitalWrite(semaforoAmarillo1, HIGH); // Enciende el LED amarillo del
semáforo 1
        for (int i = 0; i < 5000; i += 500) {
            digitalWrite(semaforoAmarillo1, !digitalRead(semaforoAmarillo1)); //
Parpadea el LED amarillo
            delay(500); // Retardo para el parpadeo
        }
        digitalWrite(semaforoAmarillo1, LOW); // Apaga el LED amarillo del
semáforo 1

        cambiarEstadoRelay(relePin1, estadoRele1);
        tiempoInicio1 = millis();
        procesoActivo1 = true;
        digitalWrite(semaforoRojo1, LOW); // Apaga el LED rojo del semáforo 1
        digitalWrite(semaforoVerde1, HIGH); // Enciende el LED verde del semáforo
1
    }
}

if (!procesoActivo2) {
    if (estadoBoton2 == LOW) {
        digitalWrite(semaforoAmarillo2, HIGH); // Enciende el LED amarillo del
semáforo 2
        for (int i = 0; i < 5000; i += 500) {
            digitalWrite(semaforoAmarillo2, !digitalRead(semaforoAmarillo2)); //
Parpadea el LED amarillo
            delay(500); // Retardo para el parpadeo
        }
        digitalWrite(semaforoAmarillo2, LOW); // Apaga el LED amarillo del
semáforo 2

        cambiarEstadoRelay(relePin2, estadoRele2);
        tiempoInicio2 = millis();
        procesoActivo2 = true;
        digitalWrite(semaforoRojo2, LOW); // Apaga el LED rojo del semáforo 2
        digitalWrite(semaforoVerde2, HIGH); // Enciende el LED verde del semáforo
2
    }
}

// Verifica si ha pasado el tiempo de activación del primer relé (10 segundos)
if (estadoRele1 == HIGH && (millis() - tiempoInicio1) >= 10000) {
    cambiarEstadoRelay(relePin1, estadoRele1);
    procesoActivo1 = false;
    digitalWrite(semaforoRojo1, HIGH); // Enciende el LED rojo del semáforo 1
    digitalWrite(semaforoVerde1, LOW); // Apaga el LED verde del semáforo 1

```

```

        // Activa la alarma
        digitalWrite(zumbadorPin, HIGH);
        delay(5000); // Sonar la alarma durante 5 segundos
        digitalWrite(zumbadorPin, LOW); // Apaga la alarma
    }

    // Verifica si ha pasado el tiempo de activación del segundo relé (10 segundos)
    if (estadoRele2 == HIGH && (millis() - tiempoInicio2) >= 10000) {
        cambiarEstadoRelay(relePin2, estadoRele2);
        procesoActivo2 = false;
        digitalWrite(semaforoRojo2, HIGH); // Enciende el LED rojo del semáforo 2
        digitalWrite(semaforoVerde2, LOW); // Apaga el LED verde del semáforo 2

        // Activa la alarma
        digitalWrite(zumbadorPin, HIGH);
        delay(5000); // Sonar la alarma durante 5 segundos
        digitalWrite(zumbadorPin, LOW); // Apaga la alarma
    }

    // Controla la alerta verde (parpadeo) durante los últimos 2 segundos antes
    de apagar el LED rojo del primer rele
    if (estadoRele1 == HIGH && procesoActivo1 && (millis() - tiempoInicio1) >=
8000) {
        digitalWrite(semaforoVerde1, !digitalRead(semaforoVerde1)); // Parpadea el
LED verde del semáforo 1
        delay(500); // Retardo para el parpadeo
    }

    // Controla la alerta verde (parpadeo) durante los últimos 2 segundos antes
    de apagar el LED rojo del segundo rele
    if (estadoRele2 == HIGH && procesoActivo2 && (millis() - tiempoInicio2) >=
8000) {
        digitalWrite(semaforoVerde2, !digitalRead(semaforoVerde2)); // Parpadea el
LED verde del semáforo 2
        delay(500); // Retardo para el parpadeo
    }
}

```

## Anexo 4. Código para la conexión Blynk con KC868 A4

```
void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

## Anexo 5. Código para la conexión WhatsApp con KC868 A4

```
int LED = 2;

unsigned long previousMillis = 0;
const long INTERVAL = 6000;

void setup()
{
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
  thing.SetWiFi("red wifi", "Systems.01");
  thing.initDevice();
}

String HandleResponse(String consulta) //HandleResponse: gestiona la respuesta

{
  if (consulta == "prendido") {
    digitalWrite(LED, 1);
    return "REALIZADO LUZ ENCENDIDA";
  }

  else if (consulta == "apagado") {
    digitalWrite(LED, 0);
    return "REALIZADO LUZ APAGADA";
  }

  else if (consulta == "estatus")
    return digitalWrite(LED) ? "led encendida" : "led apagada";
  else return "invalido";
}

void loop()
{
  thing.Handle(); //llama al cliente para la transferencia de datos
}
```

**Anexo 6. Código para la conexión de Telegram con KC868 A4**

```
#include "CTBot.h"

CTBot miBot;
CTBotInlineKeyboard miTeclado;

#include "token_ejemplo.h"
uint8_t led = 2;

void setup() {
    Serial.begin(115200);

    //blynk

    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);

    //Telegram
    Serial.println("Iniciando Bot Telegram...");
    miBot.wifiConnect(ssid, password);
    miBot.setTelegramToken(token);

    if (miBot.testConnection()) {
        Serial.println("Bot Conectado");
    }
    else {
        Serial.println("Error con Bot");
    }

    pinMode(led, OUTPUT);
    digitalWrite(led, LOW);

    miTeclado.addButton("Encender", "encender", CTBotKeyboardButtonQuery);
    miTeclado.addButton("Apagar", "apagar", CTBotKeyboardButtonQuery);
}

void loop() {
    //blynk

    Blynk.run();

    //telegram
    TBMessage msg;
```

```
if (miBot.getNewMessage(msg)) {
  if (msg.messageType == CTBotMessageText) {
    if (msg.text.equalsIgnoreCase("opciones")) {
      miBot.sendMessage(msg.sender.id, "Cambiar Led", miTeclado);
    }
    else {
      miBot.sendMessage(msg.sender.id, "prueba 'opciones'");
    }
  } else if (msg.messageType == CTBotMessageQuery) {
    Serial << "Mensaje: " << msg.sender.firstName;
    if (msg.callbackQueryData.equals("encender")) {
      Serial.println(" Encender");
      digitalWrite(led, HIGH);
      miBot.endQuery(msg.callbackQueryID, "Led Encendido", true);
    } else if (msg.callbackQueryData.equals("apagar")) {
      Serial.println(" Apagar");
      digitalWrite(led, LOW);
      miBot.endQuery(msg.callbackQueryID, "Led Apagado");
    }
  }
}

delay(250);
}
```

## Anexo 7. Programa para la conexión con placa KC868 E16S

```
#include <Wire.h>

//Se defines la variables
#define PCF8574_U27_ADDRESS 0x21 // R1-8
#define PCF8574_U28_ADDRESS 0x25 // R9-16
#define PCF8574_U23_ADDRESS 0x22 // DI 1-8
#define PCF8574_U24_ADDRESS 0x24 // DI 9-16

//relevadores primer hilo
#define RELAY1_BIT 0
#define RELAY2_BIT 1
#define RELAY3_BIT 2

//relevadores segundo hilo
#define RELAY4_BIT 3
#define RELAY5_BIT 4
#define RELAY6_BIT 5

//relevadores tercer hilo
#define RELAY7_BIT 6
#define RELAY8_BIT 7
#define RELAY9_BIT 0

//relevadores cuarto hilo
#define RELAY10_BIT 1
#define RELAY11_BIT 2
#define RELAY12_BIT 3

//Relevadores quinto hilo
#define RELAY13_BIT 4
#define RELAY14_BIT 5
#define RELAY15_BIT 6

//botones primer hilo
#define BUTTON1_BIT 0
#define BUTTON2_BIT 1

//botones segundo hilo
#define BUTTON3_BIT 2
#define BUTTON4_BIT 3

//botones tercer hilo
#define BUTTON5_BIT 4
```

```
#define BUTTON6_BIT 5

//botones cuarto hilo
#define BUTTON7_BIT 6
#define BUTTON8_BIT 7

//botones quinto hilo
#define BUTTON9_BIT 0
#define BUTTON10_BIT 1

//GENERAL
#define RELAY16_BIT 7
#define BUZZER_PIN 2

//FUNCIONES DE TIEMPO DE 3 HILOS
unsigned long tiempoInicio = 0;
unsigned long tiempoInicio2 = 0;
unsigned long tiempoInicio3 = 0;
unsigned long tiempoInicio4 = 0;
unsigned long tiempoInicio5 = 0;

//Primer hilo
bool rele1Encendido = false;
bool boton2Presionado = false;
bool buzzerEncendido = false;

//segundo hilo
bool rele4Encendido = false;
bool boton4Presionado = false;
bool buzzerEncendido2 = false;

//tercer hilo
bool rele7Encendido = false;
bool boton6Presionado = false;
bool buzzerEncendido3 = false;

//cuarto hilo
bool rele10Encendido = false;
bool boton8Presionado = false;
bool buzzerEncendido4 = false;

//quinto hilo
bool rele13Encendido = false;
bool boton10Presionado = false;
bool buzzerEncendido5 = false;
```

```
void setup() {
    Wire.begin(16, 15);
    pinMode(BUZZER_PIN, OUTPUT);
    cambiarEstadoBuzzer(LOW);
    cambiarEstadoBuzzer2(LOW);
    cambiarEstadoBuzzer3(LOW);
    cambiarEstadoBuzzer4(LOW);
    cambiarEstadoBuzzer5(LOW);
    cambiarEstadoRelay1(RELAY1_BIT, HIGH);
    cambiarEstadoRelay4(RELAY4_BIT, HIGH);
    cambiarEstadoRelay7(RELAY7_BIT, HIGH);
    cambiarEstadoRelay10(RELAY10_BIT, HIGH);
    cambiarEstadoRelay13(RELAY13_BIT, HIGH);
}

void loop() {
    loop1();
    loop2();
    loop3();
    loop4();
    loop5();
}

void loop1() {

    int estadoBoton1 = leerEstadoBoton(BUTTON1_BIT, PCF8574_U23_ADDRESS);
    int estadoBoton2 = leerEstadoBoton(BUTTON2_BIT, PCF8574_U23_ADDRESS);

    //Se inicialia primer relevador (Luz roja esta encendida)
    cambiarEstadoRelay1(RELAY1_BIT, rele1Encendido);

    //
    if (estadoBoton1 == LOW && !rele1Encendido) {
        cambiarEstadoRelay1(RELAY1_BIT, LOW);
        cambiarEstadoRelay1(RELAY3_BIT, LOW);
        tiempoInicio = millis();
        rele1Encendido = true;
    }

    //
    if (estadoBoton2 == LOW) {
```

```
        boton2Presionado = true;
    } else if (boton2Presionado) {
        boton2Presionado = false;
        tiempoInicio = millis();
    }

    // Verifica si el BOTON2 esta siendo oprimido (Cambia 10 segundos a 20
segundos)
    if (rele1Encendido && millis() - tiempoInicio > (boton2Presionado ? 20000 :
10000)) {
        cambiarEstadoRelay1(RELAY3_BIT, HIGH);
        cambiarEstadoRelay1(RELAY1_BIT, HIGH);
        rele1Encendido = false;
    }

    //Control de la alarma y parpadeo
    if(boton2Presionado && rele1Encendido == false) {
        cambiarEstadoBuzzer(HIGH);
        buzzerEncendido = true;
        activarAlarm(RELAY16_BIT);
        parpadearRelay(RELAY2_BIT);
        delay(50);
    } else if(boton2Presionado && millis() - tiempoInicio > 20000){
        cambiarEstadoBuzzer(HIGH);
        buzzerEncendido = true;
        delay(50);
    }
    else {
        if (buzzerEncendido) {
            cambiarEstadoBuzzer(LOW);
            buzzerEncendido = false;
            desactivarAlarm(RELAY16_BIT);
        }
    }

    //Control parpadeo cuando faltan 15 segundos
    if(boton2Presionado && millis() - tiempoInicio > 15000){
        parpadearRelay(RELAY2_BIT);
        delay(50);
    }
    delay(50);
} //FIN LOOP

//Lee los bytes de comunicacion I2C
```

```
int leerEstadoBoton(int bit, int direccion) {
    Wire.requestFrom(direccion, 1);
    return bitRead(Wire.read(), bit);
}

//Cambia de estado el primer relevador
void cambiarEstadoRelay1(int bit, byte nuevoEstado) {

    Wire.requestFrom(PCF8574_U27_ADDRESS, 1);
    byte bits = Wire.read();
    bitWrite(bits, bit, nuevoEstado);
    Wire.beginTransmission(PCF8574_U27_ADDRESS);
    Wire.write(bits);
    Wire.endTransmission();
}

//Cambia de estado el segundo relevador
void cambiarEstadoRelay2(int bit, byte nuevoEstado) {
    Wire.requestFrom(PCF8574_U28_ADDRESS, 1);
    byte bits = Wire.read();
    bitWrite(bits, bit, nuevoEstado);
    Wire.beginTransmission(PCF8574_U28_ADDRESS);
    Wire.write(bits);
    Wire.endTransmission();
}

//Cambia de estado el buzzer
void cambiarEstadoBuzzer(byte nuevoEstado) {
    digitalWrite(BUZZER_PIN, nuevoEstado);
}

//Controla el parpadeo
void parpadearRelay(int bit) {
    cambiarEstadoRelay1(bit, LOW);
    delay(500); // Ajusta el tiempo del parpadeo según tus necesidades
    cambiarEstadoRelay1(bit, HIGH);
}

//Controla la activacion de alarma
void activarAlarm(int bit){
    if(boton2Presionado == true){
        cambiarEstadoRelay2(bit,LOW);
    }
}
```

```
    }  
  }  
  
  //Controla la desactivacion de la alarma  
  void desactivarAlarm(int bit){  
    if(boton2Presionado == false){  
      cambiarEstadoRelay2(bit,HIGH);  
    }  
  }  
}
```



## Anexo 9. Esquema eléctrico KC868 E16S

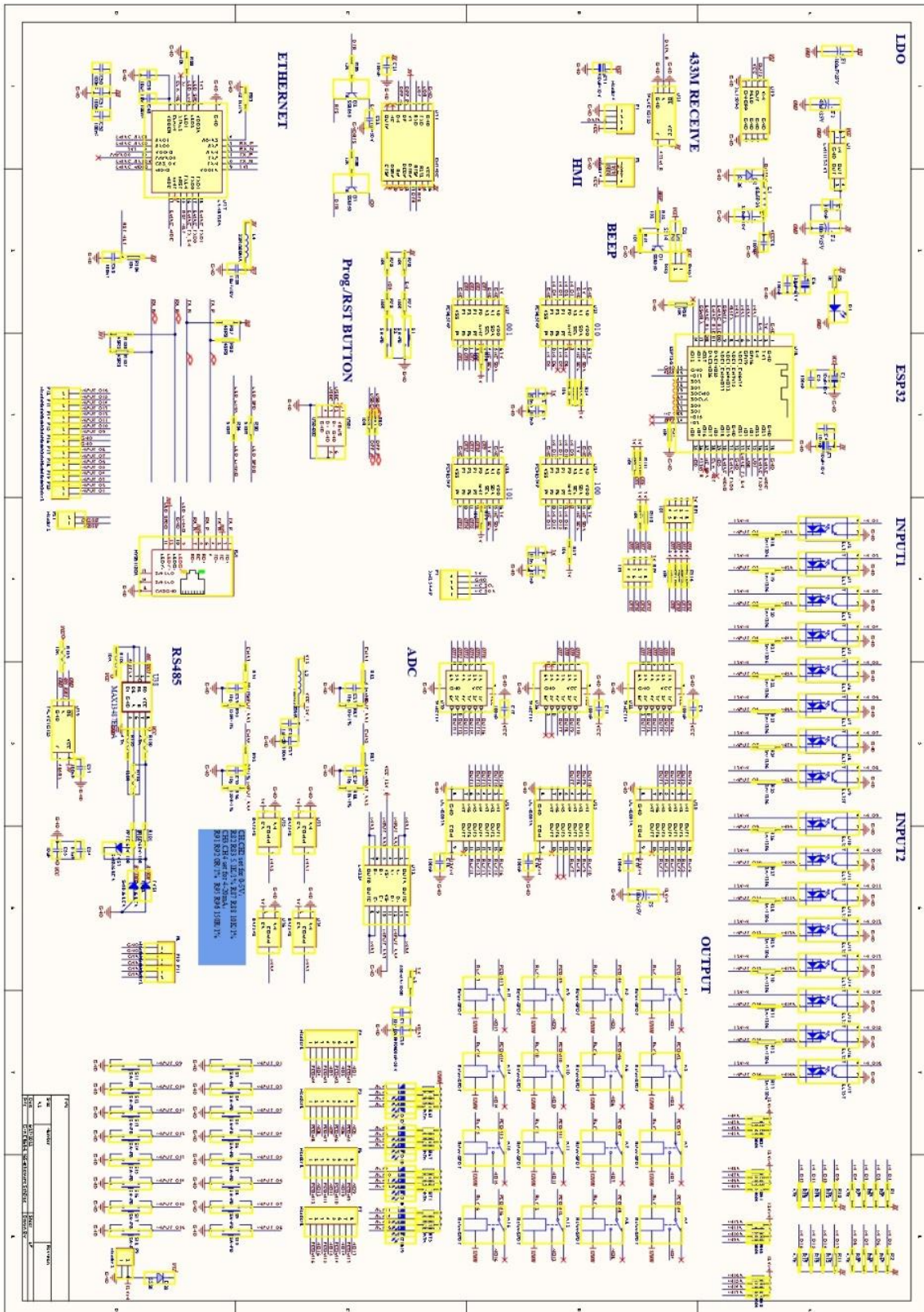


Imagen 68 Esquema Placa KC868 E16S Fuente:

<https://www.kincony.com/forum/showthread.php?tid=2809&highlight=E16S>

## Anexo 10. Esquema PCF8574

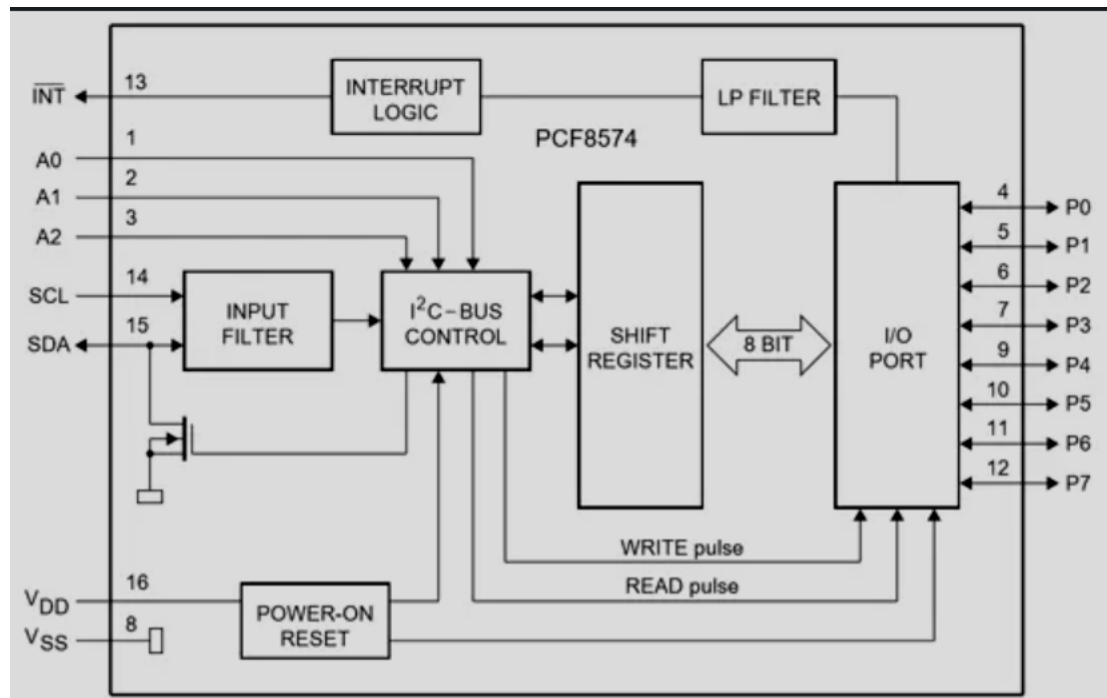


Imagen 69 Esquema PCF8574 Fuente: <https://www.luisllamas.es/mas-pines-digitales-con-arduino-y-el-expansor-es-i2c-pcf8574/>