



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

TRABAJO DE TITULACIÓN

**IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO CON
TECNOLOGÍA IOT QUE ACCIONE LOS DISPOSITIVOS ELECTRÓNICOS Y
QUE SUPERVISE EL ESTADO DE LOS SENSORES DE SEGURIDAD.**

MODALIDAD

TRABAJO DE TESIS

AUTOR

MILTON DANIEL NAVARRETE CHILAN

TUTOR

ING. LUIS CHUQUIMARCA, PHD.

LA LIBERTAD – ECUADOR

2025 - 2026

AGRADECIMIENTO

A las personas que estuvieron guiándome por el sendero del estudio en cada ciclo académico, por esa motivación y buenas vibra para seguir con mi propósito de poder crecer profesionalmente y llegar a ser el ingeniero de la familia, a todos ustedes infinitas gracias. Este espacio es para ustedes.

NAVARRETE MILTON

DEDICATORIA

Agradecido con Dios por cada día que me brinda y poder culminar mi calidad de estudiante de esta prestigiosa facultad, dedico este trabajo de titulación a mi familia, docentes, compañeros y al futuro equipo que le darán uso a mi implementación como muestra de afecto e impulsando al crecimiento de la tecnología en sus vidas.

NAVARRETE MILTON

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **Implementación de un sistema automatizado con tecnología IoT que accione los dispositivos electrónicos y que supervise el estado de los sensores de seguridad**, elaborado por el egresado Navarrete Chilan Milton Daniel, de la carrera de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.



Ing. Luis Enrique Chuquimarca Jiménez, PhD.

TRIBUNAL DEL GRADO



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**WASHINGTON DANIEL
TORRES GUIN**

Ing. Washington Torres Guin, Mgtr.
DECANO DE LA FACULTAD



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**JOSE MIGUEL SANCHEZ
AQUINO**

Ing. José Sánchez Aquino, Mgtr.
DIRECTOR DE LA CARRERA



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**JUNIOR RAFAEL
FIGUEROA OLMEDO**

Ing. Junior Figueroa Olmedo, Mgtr.
DOCENTE DE ÁREA



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**LUIS ENRIQUE
CHUQUIMARCA JIMENEZ**

Ing. Luis Chuquimarca Jiménez, PhD.
DOCENTE TUTOR



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**MARIA MARGARITA
RIVERA GONZALEZ**

Abg. María Rivera González, Mgtr.
SECRETARIO GENERAL

RESUMEN

La tecnología del Internet de las Cosas (IoT) ha experimentado un crecimiento sostenido en los últimos años, extendiendo su aplicación a diversos sectores, tales como el industrial, laboral, residencial, automotriz y educativo. En este contexto, el ámbito educacional adquiere especial relevancia, dado que las aulas tradicionales tienden a volverse obsoletas frente a la incorporación de soluciones tecnológicas innovadoras que transforman los espacios de aprendizaje en aulas inteligentes.

Actualmente, las aulas de vanguardia se caracterizan por disponer de cobertura total de red Wi-Fi, lo que permite la conectividad permanente de dispositivos eléctricos y electrónicos, los cuales pueden ser monitoreados y controlados de forma remota desde cualquier ubicación física mediante el uso de internet. Esta evolución tecnológica contribuye a optimizar la gestión de los recursos, mejorar la eficiencia energética y fortalecer la experiencia educativa.

En virtud de lo expuesto, el presente estudio se desarrolla en aulas de una institución de educación superior, donde se propone la implementación de un sistema basado en IoT que permita visualizar en tiempo real el estado de los equipos eléctricos a través de una plataforma web, posibilitando su encendido y apagado remoto. El principal desafío radica en establecer una comunicación confiable entre los dispositivos y la interfaz de control, haciendo uso eficiente de las tecnologías propias del Internet de las Cosas.

Adicionalmente, el sistema incorpora un panel táctil centralizado, diseñado para el control local de los equipos y la visualización inmediata de su estado operativo dentro del aula. Todo el sistema se apoya en una interfaz gráfica moderna, intuitiva y didáctica, orientada a facilitar la interacción del usuario y a garantizar una gestión tecnológica eficiente acorde a los estándares actuales de las aulas inteligentes.

Palabras clave: IoT, Monitoreo, Control de forma remota, Plataforma web.

ABSTRACT

The Internet of Things (IoT) technology has experienced sustained growth in recent years, expanding its application across various sectors such as the industrial, occupational, residential, automotive, and educational fields. In this context, the educational sector has gained particular relevance, as traditional classrooms are increasingly becoming obsolete in the face of innovative technological solutions that transform learning environments into smart classrooms.

Currently, state-of-the-art classrooms are characterized by having full Wi-Fi network coverage, which enables continuous connectivity of electrical and electronic devices. These devices can be monitored and controlled remotely from any physical location using internet connectivity. This technological evolution contributes to optimizing resource management, improving energy efficiency, and enhancing the overall educational experience.

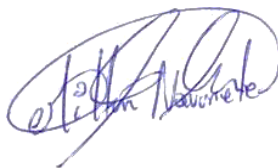
Based on the above, the present study is conducted in six classrooms of a higher education institution, where the implementation of an IoT-based system is proposed. This system allows real-time visualization of the operational status of electrical equipment through a web-based platform, enabling remote switching on and off each device. The main challenge lies in establishing reliable communication between the devices and the control interface, making efficient use of Internet of Things technologies.

Additionally, the system incorporates a centralized touch panel, designed for local equipment control and immediate visualization of their operational status within the classroom. The entire system is supported by a modern, intuitive, and user-friendly graphical interface, aimed at facilitating user interaction and ensuring efficient technological management in accordance with current smart classroom standards.

Keywords: IoT, Monitoring, Remote control, web platform.

DECLARACIÓN DEL AUTOR

Milton Daniel Navarrete Chilan, egresado de la Universidad Estatal Península de Santa Elena, Facultad de Sistemas y Telecomunicaciones, de la carrera Electrónica y Telecomunicaciones, la responsabilidad del contenido recae en mí y el patrimonio intelectual de la misma le corresponde a la Universidad Estatal Península de Santa Elena.



Milton Daniel Navarrete Chilan

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
APROBACIÓN DEL TUTOR	iv
TRIBUNAL DEL GRADO	v
RESUMEN	6
ABSTRACT.....	7
DECLARACIÓN DEL AUTOR	8
INTRODUCCIÓN	17
CAPÍTULO I	18
1.1. ANTECEDENTES	18
1.2. DESCRIPCIÓN DEL PROYECTO	18
1.3. OBJETIVOS	20
1.3.1. OBJETIVO GENERAL.....	20
1.3.2. OBJETIVOS ESPECÍFICOS	20
1.4. JUSTIFICACIÓN	21
1.5. ALCANCE DEL PROYECTO.....	22
1.6. METODOLOGÍA	23
1.6.1. Investigación exploratoria.....	23
1.6.2. Investigación experimental	23

1.6.3.	Investigación bibliográfica.....	23
1.7.	RESULTADOS ESPERADOS.....	23
CAPÍTULO II.....		25
2.1.	MARCO CONTEXTUAL.....	25
2.2.	MARCO CONCEPTUAL.....	26
2.2.1.	Caracterización del entorno de implementación.....	26
2.2.2.	Internet de las cosas.....	27
2.2.2.2	Arquitectura del IoT.....	28
2.2.3.	Plataforma de gestión y control IoT.....	30
2.2.4.	Infraestructura de comunicación en sistemas tecnológicos.....	31
2.2.5.	Protocolo de comunicación.....	31
2.2.6.	Gateway o puerto de enlace.....	33
2.2.7.	Interfaz de usuario.....	34
2.2.8.	Dispositivos electrónicos.....	34
2.2.9.	Normativa para las instalaciones eléctricas.....	45
2.3.	MARCO TEÓRICO.....	46
CAPÍTULO III.....		49
DESARROLLO DE LA PROPUESTA.....		49
3.1.	Componentes lógicos.....	49
3.1.1.	IDE Arduino 2.0.0.....	49

3.1.2.	Fritzing.....	50
3.1.3.	KiCad	50
3.1.4.	Autodesk Fusion 360	51
3.1.5.	Portal web creado por el servidor del ESP32.....	51
3.2.	Componentes físicos	52
3.2.1.	Tarjeta de desarrollo Esp32	52
3.2.2.	Sensores de movimientos.....	53
3.2.3.	HMI.....	54
3.2.4.	Regulador de voltaje LM2596	54
3.2.5.	Módulos relé 4 canales.....	55
3.2.6.	Diodo Led IR	55
3.2.7.	Transistor 2N4401	56
3.2.8.	Sensor de temperatura.....	56
3.2.9.	Sensor de contacto magnético.....	57
3.3.	Diseño de la propuesta	58
3.3.1.	Diagrama de conexiones	58
3.3.2.	Diseño del circuito en PCB.....	59
3.3.3.	Diseño estructural 3D	61
3.3.4.	Consumo del dispositivo.....	63
3.3.5.	Costo de implementación.....	64

3.4	Diseño del circuito PCB.....	65
3.4.1.	Diseño y ruteo del circuito impreso.....	65
3.4.2.	Diseño y modelado de la carcasa.....	66
CAPÍTULO IV.....		67
RESULTADOS.....		67
4.1.	Comparativa y selección de componentes	67
4.2.	Funciones del sistema automatizado.....	68
4.2.1.	Conexión a red WiFi.....	68
4.2.2.	Latencia del dispositivo en la red.....	70
4.2.3.	Densidad de redes WiFi	71
4.2.4.	Interfaces de usuario	72
CONCLUSIONES		82
RECOMENDACIONES.....		83
BIBLIOGRAFÍA		84
ANEXOS		90
Anexo 1. Código principal		90
Anexo 2. Código de recuperación de credenciales WiFi		93
Anexo 3. Código de control de corriente y HMI.....		94
Anexo 4. Código de codificación de señales IR.....		104
Anexo 6. Código del servidor		108

Anexo 7. Código de lectura de sensores	117
Anexo 8. Código de configuración a red y punto de acceso	120

ÍNDICE DE FIGURAS

Figura 1. Esquema de conexión del control.....	19
Figura 2. Conexión inalámbrica para datos con el web server.	19
Figura 3. Conexión al canal de Telegram.	20
Figura 4. Funcionamiento del sensor PIR.....	36
Figura 5. Sensor de contacto.....	37
Figura 6. Sensor de corriente.	38
Figura 7. Principio de la luz.....	39
Figura 8. Función del aire acondicionado.....	39
Figura 9. Estructura interna de la cerradura eléctrica.	41
Figura 10. Esquemático regulador Buck LM2596.....	42
Figura 11. Funcionamiento del IR emisor.	43
Figura 12. Ilustración del funcionamiento del ESP32.	44
Figura 13. Funcionamiento interno del relé.....	45
Figura 14. IDE Arduino versión 2.0.0.	49
Figura 15. Fritzing.	50
Figura 16. Kicad.....	50

Figura 17. Fusion 360.	51
Figura 18. Interfaz de usuario del sistema automatizado.	51
Figura 19. Regulador de voltaje LM2596.	55
Figura 20. Módulo relé de 4 canales.	55
Figura 21. Led infrarrojo IR.	56
Figura 22. Transistor 2N4401.	56
Figura 23. Sensor de contacto magnético.	58
Figura 24. Esquema de conexiones del sistema automatizado.	59
Figura 25. Sistema de alimentación.	59
Figura 26. Conexión de sensores.	60
Figura 27. Conexiones con la tarjeta de desarrollo ESP32.	61
Figura 28. Carcasa de protección del hardware.	62
Figura 29. Carcasa modelada en 3D.	62
Figura 30. Medidas de las tapas.	63
Figura 31. Medidas de la carcasa.	63
Figura 32. Ruteado de la PCB.	65
Figura 33. Renderizado de la caja.	66
Figura 34. Módulo didáctico final del proyecto.	67
Figura 35. Inicio de punto de acceso.	68
Figura 36. Configuración inicial desde la dirección IP.	69

Figura 37. Conexión exitosa a red Wifi.	69
Figura 38. Monitor serial: inicio de punto de acceso.	70
Figura 39. Gráfico de latencia del dispositivo en la red.	71
Figura 40. Menú de inicio.	73
Figura 41. Interfaz de modo de ahorro de energía.	73
Figura 42. Control de luces en el HMI.	74
Figura 43. Control de luces en el servidor web.	74
Figura 44. Error de encendido del aire acondicionado.	75
Figura 45. Configuración de señal del aire acondicionado.	77
Figura 46. Estado encendido del aire acondicionado.	78
Figura 47. Estado de apago del proyector.	79
Figura 48. Estado de encendido del proyector.	80
Figura 49. Notificaciones del estado del aula.	80
Figura 50. Gestión de seguridad de equipos electrónicos.	81

ÍNDICE DE TABLAS

Tabla 1. Características de tarjetas de desarrollo.	52
Tabla 2. Características de sensores de presencia.	53
Tabla 3. Pantallas LCD.	54
Tabla 4. Sensores de temperatura.	57

Tabla 5. Consumo total del hardware.	64
Tabla 6. Costo de implementación de la propuesta.	64
Tabla 7. Latencia del dispositivo en red.	70
Tabla 8. Potencia de señales WiFi-escaneadas	72
Tabla 9. Códigos de control IR de diferentes fabricantes.	75
Tabla 10. Transición de encendido y apagado de aire acondicionado.....	78

INTRODUCCIÓN

El Internet de las Cosas se fundamenta en la capacidad de los dispositivos para recopilar, procesar y compartir información mediante redes de comunicación, posibilitando el control a distancia de equipos que se encuentran físicamente alejados del usuario. Entre las primeras aplicaciones de esta tecnología se destacan la automatización de sistemas de iluminación, calefacción, ventilación, aire acondicionado, seguridad, entretenimiento y electrodomésticos, los cuales han contribuido en gran manera a mejorar la operatividad y la comodidad en distintos entornos.

En la actualidad, el IoT ha ampliado su campo de aplicación, extendiéndose a sectores como el residencial, industrial, de la salud, automotriz y educativo. Su presencia se ha vuelto cada vez más cotidiana, evidenciada en soluciones como sensores de presencia para la apertura automática de puertas, control de dispositivos mediante comandos de voz y sistemas inteligentes capaces de adaptarse al comportamiento del usuario. Esta tecnología permite no solo la automatización de procesos, sino también la recopilación y análisis de datos en tiempo real, facilitando la toma de decisiones informadas y optimizando el uso de los recursos disponibles.

En este contexto y dentro de este marco tecnológico, el sector educativo enfrenta el desafío de modernizar sus infraestructuras y adoptar soluciones que permitan una gestión adecuada de los recursos. Las aulas tradicionales tienden a volverse obsoletas frente a la necesidad de espacios inteligentes que integren tecnología, conectividad y control automatizado. En respuesta a esta problemática, el presente trabajo propone la implementación de un sistema automatizado basado en Internet de las Cosas para el control y monitoreo de equipos eléctricos y electrónicos en aulas. El sistema planteado permite visualizar en tiempo real el estado de los equipos mediante una interfaz gráfica, facilitando su encendido y apagado desde un único punto de control, sin necesidad de interactuar físicamente con cada interruptor o dispositivo. La solución integra software, hardware embebido y servicios en la red, empleando microcontroladores con conectividad inalámbrica que interactúan con plataformas web accesibles desde navegadores o aplicaciones móviles.

CAPÍTULO I

1.1. ANTECEDENTES

A finales de la década de los 90 surgió el término internet de las cosas (IoT), usado para describir la interconexión de los objetos físicos a través de redes de comunicación con el fin de realizar monitoreo e intercambio de información [1]. Este término fue propuesto por Kevin Ashton durante la época en la que trabajaba para la empresa Procter & Gamble [2].

Bajo este enfoque, el internet de las cosas se aplica en distintos entornos mediante la automatización y supervisión remota de dispositivos y sistemas, incluyendo iluminación, climatización, sistemas de seguridad, entretenimiento y electrodomésticos [3].

En la actualidad, el IoT se aplica en diversos entornos como hogares inteligentes, oficinas, industrias, salud y educación, ampliando progresivamente su presencia en distintos ámbitos del día a día. Estos avances favorecen la automatización y la eficiencia de los procesos [4].

De acuerdo con MagazIEEE Ecuador, la adopción del internet de las cosas en el Ecuador alcanza aproximadamente al 20% de la población, porcentaje que se encuentra condicionado por factores como el acceso a servicios de conectividad y la disponibilidad de tecnologías, en este contexto, el IoT se aplica principalmente mediante dispositivos y aplicaciones que permiten la interacción y el control de sistemas conectados a internet en entornos cotidianos [5].

1.2. DESCRIPCIÓN DEL PROYECTO

Implementación de un sistema automatizado para el control de los equipos eléctricos y electrónicos que se encuentran distribuidos en las seis aulas, el cual permita que, a través de una interfaz gráfica de usuario, encender o apagar los equipos desde un mismo lugar sin tener que acudir a cada interruptor o botón de encendido distribuidos dentro de las aulas.

El sistema está basado en software, hardware embebido e internet de las cosas; con respecto al software; uso de protocolos IoT como peticiones https por método get/post para el control mediante la red. El hardware, comprende el uso de la placa de desarrollo esp32 que permite conectarse a una red inalámbrica, y, a su vez, a un servidor dedicado que permita la interacción remota entre persona y máquina [6].

La implementación de este sistema automatizado enlaza los aparatos eléctricos y electrónicos de las seis aulas a un web server, a través de la red inalámbrica. Que mediante dispositivos móviles

como smartphone, laptops u ordenadores y una interfaz gráfica, el usuario controla y monitorea a cada uno de los aparatos eléctricos, tales como: luminarias, ventiladores, proyectores y aires acondicionados.

En la Figura 1, se muestra el esquema de conexión para el control de aparatos eléctricos basados en la tarjeta de desarrollo Esp32, modulo relé para el control de encendido – apagado y un diodo emisor de luz para el control por luz infrarroja dedicados a los aparatos que poseen mandos a distancia.

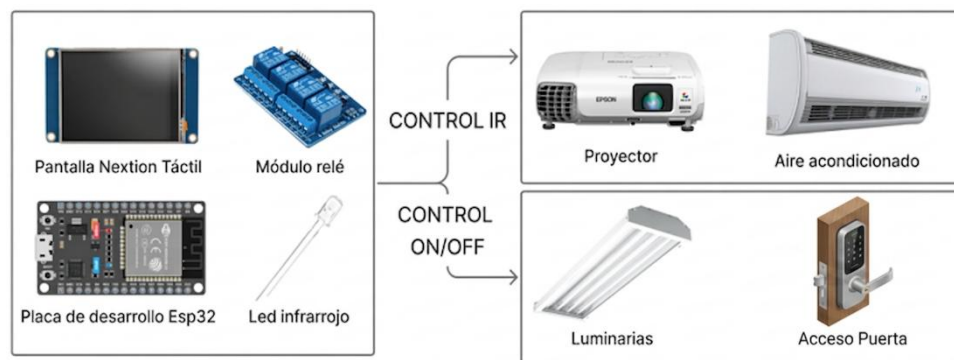


Figura 1. Esquema de conexión del control.

En la Figura 2, hace referencia al esquema de conexión y la comunicación inalámbrica para obtener los datos a través de la red, mediante el uso de protocolos de internet de las cosas a un servidor que crea la propia tarjeta de desarrollo. De esta manera, el usuario responsable puede acceder al sistema automatizado para el monitoreo y control de las aulas.

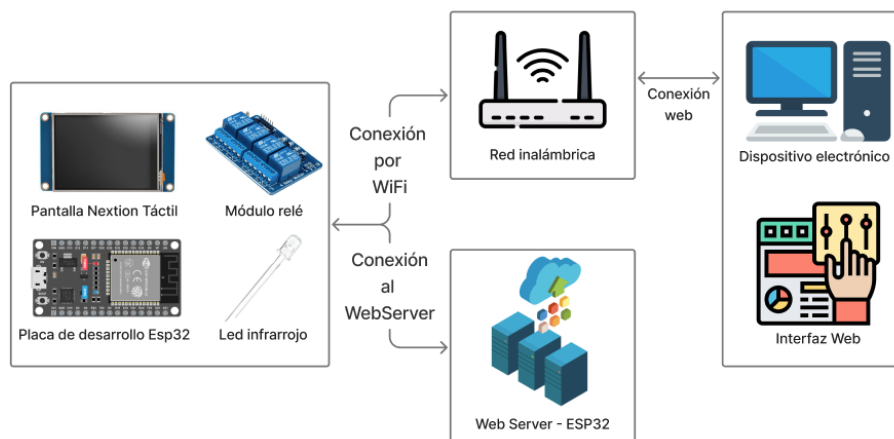


Figura 2. Conexión inalámbrica para datos con el web server.

Como complemento al sistema automatizado, se implementa una conexión a un canal de Telegram como en la Figura 3, para recibir notificaciones del estado de disponibilidad del aula de clases, en el caso de estar ocupado o si alguien está cerca de la puerta de entrada como parte de seguridad. Además, el sistema envía notificaciones cuando el aire acondicionado no se enciende por falta de energía eléctrica, y recomendaciones de encender del aire acondicionado cuando están por encima de la temperatura adecuada. También recibe notificación cuando se enciende o se apaga el proyector. Todo esto para tener un monitoreo del aula en tiempo real.

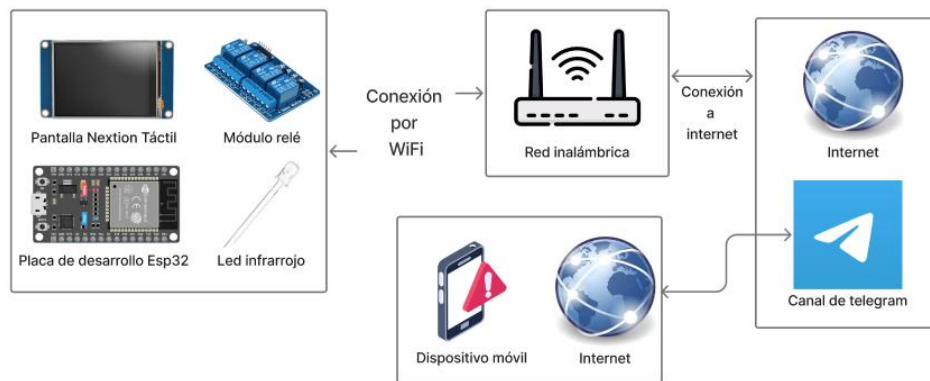


Figura 3. Conexión al canal de Telegram.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Implementar un sistema de accionamiento y monitoreo basados en tecnología IoT utilizando una placa de desarrollo y sensores electrónicos para la sofisticación domótica del aula.

1.3.2. OBJETIVOS ESPECÍFICOS

- Realizar una comparativa de los componentes electrónicos que se integran en el sistema para el accionamiento de los equipos eléctricos y el censado de puertas y ventanas de cada aula.
- Implementar los sensores por contacto, de temperatura y de presencia en el aula para el monitoreo dentro de cada aula.
- Implementar los diferentes actuadores para el accionamiento de los dispositivos eléctricos como lámparas, cerradura eléctrica de puerta y el sistema de accionamiento por señal RF

para el encendido y apagado del aire acondicionado y proyector, ubicados dentro de cada aula.

- Desarrollar un sistema automatizado utilizando una tarjeta de desarrollo para la conexión de sensores y actuadores con acceso a la nube de los parámetros adquiridos mediante comunicación Master y Slave para gestionar dispositivos de forma centralizada y eficiente.
- Desarrollar una interfaz para gestionar y monitorear los parámetros del sistema en tiempo real de la nube.
- Evaluar el sistema con referente al accionamiento y la latencia de comunicación que existe entre la nube y el equipo centralizado para elaborar un cuadro de prueba de error.

1.4. JUSTIFICACIÓN

En las aulas de clase, el uso de equipos como proyectores, luminarias, computadoras o sistemas de climatización se ha vuelto parte habitual del proceso de enseñanza. Estos recursos ayudan de manera significativa el aprendizaje, sin embargo, su manejo diario no siempre resulta eficiente.

En las instituciones educativas, el manejo de los equipos eléctricos y electrónicos suele estar a cargo de una persona responsable, generalmente el técnico docente, quien debe preparar las aulas antes del inicio de las clases. Sin embargo, cuando varios docentes comienzan sus actividades al mismo tiempo, se presenta una dificultad, debido a que el encargado debe desplazarse por diferentes pabellones para encender los proyectores y otros equipos necesarios. Esta situación puede generar retrasos y limitar la eficiencia en la preparación de los espacios de aprendizaje. Además, el tiempo que dedica a esta actividad podría aprovecharse en otras tareas propias de sus funciones. Por otra parte, una vez finalizadas las clases, es frecuente que algunos equipos permanezcan encendidos, ya sea por descuido o falta de control, lo que impide que se realice un apagado adecuado.

Otro factor considerable que ocurre es cuando los dispositivos eléctricos y electrónicos se quedan encendidos al terminar las clases, un consumo innecesario de energía eléctrica. Por ello, resulta importante fomentar un uso eficiente de la energía eléctrica, evitando desperdicios y promoviendo buenas prácticas en su utilización.

Además, la relevancia de este proyecto está enfocada en que, aparte de reducir el costo de la planilla de luz cada mes, su implementación no representa un gasto elevado. Esto se debe a que los componentes que se utilizarán son de bajo costo en comparación con los de tipo industrial.

1.5. ALCANCE DEL PROYECTO

La implementación de la propuesta tecnológica abarca en el diseño experimental y diseño de evaluación de programa del sistema capaz de ejecutar todo lo descrito en los objetivos, para ello es necesario la comprensión y el estudio de las partes electrónicas involucradas como el sistema, los módulos de comunicación, los sensores en las diferentes áreas, y la codificación que va a interpretar las señales de los equipos dentro de las aulas.

Con la identificación de las ubicaciones de los equipos eléctricos y electrónicos dentro de las aulas se plantea la ubicación de los elementos que integran el sistema de automatización, por ejemplo, los sensores de presencia estará ubicado en la parte frontal dentro del aula arriba de la pizarra enfocando a los pupitres, los sensores por contacto serán ubicados entre la ventana y el marco de la ventana, los actuadores como el botón para apertura de puerta será ubicado dentro del aula cerca de la puerta, también los diodos emisor de señal IR para el encendido del proyector será ubicado a 10cm de distancia frente al proyector, la tarjeta de desarrollo se ubicará dentro de un case blanco y este se instalará en un punto poco visible dentro del aula.

La implementación del proyecto consiste en usar canaletas y cables eléctricos para la conducción de señales, también optar por trabajos civiles para que la implementación sea limpia, todo esto permite que los equipos eléctricos y electrónicos se conecten de forma adecuada hasta el panel de control que contiene el hardware que administra todo el sistema automatizado.

El control de encendido y apagado de los equipos electrónicos se realiza mediante una conexión estable a la red inalámbrica y manualmente. Para el control manual que es mediante el HMI y para el control a través la interfaz web mediante la red. Estos mecanismos se encuentran sincronizados mediante el servidor dedicado que se crea en la placa de desarrollo del Esp32, es decir, que los estados de los dispositivos conectados se actualizarán de forma automática.

Para el monitoreo de las aulas de clases, se utiliza un canal de Telegram únicamente para recibir notificaciones en tiempo real. El sistema automatizado mediante su algoritmo consume la API de

Telegram, el cual implica utilizar los recursos de la red social para el uso mensajería instantánea. Las notificaciones contemplan los estados del aire acondicionado, proyector, temperatura del ambiente y el estado del aula en caso de estar ocupado o disponible.

1.6. METODOLOGÍA

Los pasos que seguir para llevar a cabo la presente propuesta: primero se da comienzo por la etapa de investigación bibliográfica, exploratoria y experimental.

1.6.1. Investigación exploratoria

Esta investigación es fundamental para el desarrollo de la presente propuesta, pues el control de aires acondicionados y proyectores requiere un análisis profundo de los protocolos de comunicación empleados por sus respectivos controles remotos.

1.6.2. Investigación experimental

Este tipo de investigación es fundamental para el correcto funcionamiento de los tipos de control que posee el sistema automatizado. Sobre todo, en el envío de IR que emite el diodo infrarrojo. Además, para la conexión con el canal de Telegram mediante la placa de desarrollo que dispone el proyecto para el sistema de notificaciones en tiempo real de los estados en los dispositivos eléctricos y electrónicos en las aulas de clases.

1.6.3. Investigación bibliográfica

Este tipo de investigación es necesaria para obtener la mayor cantidad de información relacionada con la presente propuesta, entre ellos están: tesis de grado o maestrías, artículos en revistas de alto impacto, artículos, conferencias científicas, entre otros. La revisión bibliográfica permite tener un enfoque general en las tendencias tecnológicas para el uso de protocolos de comunicación e internet de las cosas.

1.7. RESULTADOS ESPERADOS

Los resultados esperados de este proyecto se relacionan con el cumplimiento de los objetivos planteados:

- Se espera establecer una comparativa técnica y económica de los componentes electrónicos que conforma nuestro sistema de automatizado como son: sensores, actuadores, módulo de comunicación y dispositivos de control compatible con la tarjeta ESP32 del mismo modo que sean compatibles, funcionales y de bajo costo.
- Se espera implementar y verificar el funcionamiento de los sensores de presencia temperatura y contacto magnético, permitiendo obtener información en tiempo real sobre la ocupación del aula y el estado de la puerta y ventana. Los datos obtenidos servirán como base para establecer acciones automáticas, alertas y condiciones de seguridad dentro del aula.
- Se espera implementar los actuadores necesarios para controlar los principales dispositivos eléctricos y electrónicos del aula incluyendo luminaria, cerradura eléctrica, aire acondicionado y proyector. El sistema deberá permitir tanto como el accionamiento local mediante una pantalla HMI como el control mediante la interfaz web.
- Se espera de desarrollar un sistema automatizado basado en ESP32 que integre sensores, actuadores y comunicación inalámbrica mediante una arquitectura centralizada. Los datos en la nube serán gestionados mediante WEBSOCKET que permite establecer una comunicación bidireccional.
- Se espera desarrollar una interfaz de usuario mediante la pantalla HMI y una interfaz web Que permita visualizar y controlar el tiempo real los diferentes elementos del aula.
- Se espera evaluar el funcionamiento integral del sistema mediante pruebas de accionamiento, monitoreo, comunicación y latencia, verificando la respuesta de los sensores, actuadores, pantalla HMI e interfaz web. Las pruebas permitirán determinar la confiabilidad del sistema y demostrar el funcionamiento prolongado para qué sirva como referencia para futuras mejoras e implementación en otros espacios educativos.

CAPÍTULO II

2.1. MARCO CONTEXTUAL

En la actualidad, el avance de las tecnologías digitales y la transformación de los servicios han impulsado la adopción de sistemas inteligentes orientados a la automatización ha permitido la integración entre dispositivos, personas y sistemas, configurando entornos cada vez más conectados y eficientes [7].

En este escenario de transformación tecnológica, el Internet de las cosas (IoT) ha adquirido un papel relevante dentro del mercado de servicios, extendiendo su aplicación a diversos entornos como hogares con sistemas de domótica, oficinas, la industria, el sector de la salud y la educación. Esta tecnología constituye una evolución del Internet tradicional, caracterizada por su expansión progresiva hacia múltiples áreas laborales mediante la interconexión inteligente de objetos capaces de obtener, recopilar, almacenar y procesar información, con el fin de interactuar con personas u otros sistemas y dar lugar a entornos inteligentes [8].

La tecnología IoT posibilita que las máquinas ejecuten tareas monótonas sin necesidad de intervención humana. Los dispositivos IoT pueden contribuir a mejorar la experiencia educativa, haciéndola más eficiente y envolvente. Mediante la utilización de dispositivos conectados a la nube. Este aumento en la adopción de tecnología conectada ofrece la oportunidad de que los profesores optimicen su tiempo y recursos, ya que el IoT automatiza algunos procesos que anteriormente se llevaban a cabo manualmente [9].

Las soluciones basadas en Internet de las cosas permiten la supervisión remota, el registro de datos en tiempo real y la generación de alertas mediante dispositivos conectados a plataformas digitales. Estas características favorecen la implementación de sistemas de monitoreo y control en distintos entornos, demostrando la viabilidad del uso de sensores y redes IoT para la gestión de información y la toma de decisiones automatizadas, lo que respalda su adopción en espacios académicos y sistemas de control inteligente [10].

En un espacio educativo físico, como una sala de clases, que emplea diariamente diversos dispositivos eléctricos, como proyectores, luces, computadoras, pizarras digitales, ventiladores y aire acondicionado, ha evidenciado ser un componente valioso para mejorar la efectividad del

aprendizaje en los estudiantes. Estos dispositivos son manipulados por personas encargadas dentro espacio educativo, como tal las aulas de estudio superior, por lo tanto, implementar un sistema IoT en aulas de estudio superior proporcionará un entorno más seguro, eficiente y automatizado, mejorando la experiencia educativa y facilitando la gestión de recursos.

En un espacio educativo físico, como una sala de clases, se emplean diariamente diversos dispositivos eléctricos y electrónicos, tales como proyectores, luminarias, computadoras, pizarras digitales, ventiladores y sistemas de aire acondicionado. Estos dispositivos forman parte fundamental de los sistemas electrónicos utilizados para el desarrollo de las actividades académicas y se caracterizan por su diversidad y función dentro de los circuitos electrónicos [11]. Por lo tanto, en las aulas se presenta un mayor consumo energético en proyectores, ventiladores, la iluminaria, computadoras y pizarras electrónicas. Es por ello por lo que la implementación de un sistema automatizado basado con tecnología IoT para el control y supervisión de electrónicos ayudara a una disminución en los gastos asociados al consumo de electricidad, porque actualmente carecen de un mecanismo automatizado que fomente la eficiencia energética en las aulas.

2.2. MARCO CONCEPTUAL

2.2.1. Caracterización del entorno de implementación

El entorno físico de implementación corresponde a aulas académicas de educación superior, las cuales constituyen la unidad funcional donde se integrará el sistema automatizado basado en tecnología IoT. En estos espacios se encuentran distribuidos equipos eléctricos y electrónicos como luminarias, proyectores, ventiladores, sistemas de aire acondicionado, computadoras y pizarras digitales, los cuales forman parte de la infraestructura operativa del aula y requieren gestión eficiente para optimizar su uso y consumo energético. Desde el punto de vista técnico, el aula dispone de una red eléctrica convencional que alimenta los distintos dispositivos mediante circuitos independientes, cuyo accionamiento se realiza de forma manual. Además, intervienen variables ambientales como temperatura, ocupación y estado de puertas y ventanas, que pueden ser supervisadas mediante sensores electrónicos. La integración de estos elementos constituye la base física sobre la cual se implementa el sistema IoT propuesto para el control y monitoreo automatizado del espacio académico [12].

2.2.2. Internet de las cosas

El Internet de las Cosas (IoT, por sus siglas en inglés) representa una evolución en la conectividad digital, donde los objetos físicos se integran a la red mediante el uso de sensores, actuadores y protocolos de comunicación. A diferencia de la Internet convencional (enfocada en la interacción humano-humano o humano-computadora), el IoT facilita la comunicación máquina a máquina [13]. A diferencia de la computación convencional, el IoT facilita la comunicación máquina a máquina (M2M), permitiendo que los dispositivos recolecten y transfieran datos a través de una red inalámbrica con una intervención humana mínima [14].

Este ecosistema comprende cualquier objeto equipado con sensores, software y tecnologías de comunicación que le permitan intercambiar datos con otros sistemas a través de Internet. Según la literatura técnica, un sistema de IoT funcional consta de cuatro pilares: sensores para la captura de datos, conectividad para la transmisión, procesamiento en la nube para la analítica y una interfaz de usuario para la interacción final [15] [16].

2.2.2.1 Topologías de Red en el Ecosistema IoT

La disposición física y lógica de los nodos dentro de una red IoT es un factor determinante para la resiliencia, el consumo energético y la capacidad de expansión del sistema [17]. Dependiendo de los requerimientos de cobertura y redundancia, se suelen implementar las siguientes configuraciones:

- **Topología de Estrella:** Es la configuración más extendida en aplicaciones de domótica y redes de corto alcance. En este esquema, todos los dispositivos finales se conectan de forma directa a un nodo central o *Gateway*. Su principal ventaja es la simplicidad en la gestión y el bajo consumo de energía en los nodos finales, aunque presenta la vulnerabilidad de un punto único de falla: si el concentrador central falla, toda la red queda inoperativa [18].
- **Topología de Malla:** En esta arquitectura, los dispositivos no solo emiten sus propios datos, sino que actúan como repetidores para los demás nodos. Esta característica permite que, ante la falla de un nodo o una obstrucción en la señal, el dato busque automáticamente una ruta alternativa para llegar a su destino. Es la base de protocolos

industriales robustos como Zigbee y Z-Wave, destacando por su alta confiabilidad y gran alcance de cobertura [19].

- **Topología de Árbol:** Esta variante organiza los nodos en niveles jerárquicos definidos. Un nodo raíz central se conecta a varios concentradores de nivel intermedio, los cuales a su vez gestionan grupos de sensores finales. Esta estructura es ideal para proyectos de gran escala, como el monitoreo industrial o ciudades inteligentes, ya que facilita la segmentación de la red y la escalabilidad jerárquica del sistema [20].

2.2.2.2 Arquitectura del IoT

Para garantizar la interoperabilidad y el flujo eficiente de datos, un sistema IoT se estructura bajo un modelo de capas que segmenta las funciones desde el hardware hasta el usuario final. El estándar más aceptado en el ámbito de la ingeniería es el modelo de tres capas, el cual permite una gestión modular del sistema [21]:

- **Capa de Percepción (Física):** Constituye la base del ecosistema. Está compuesta por el hardware encargado de interactuar con el mundo físico. Incluye sensores para captar magnitudes (temperatura, humedad, presión) y actuadores que ejecutan acciones físicas basadas en los datos procesados. También integra tecnologías de identificación como RFID para distinguir cada objeto de forma única en la red [22].
- **Capa de Red (Transmisión):** Es la encargada de transferir la información recolectada hacia los centros de procesamiento. Utiliza diversos protocolos según el alcance y el consumo energético requerido, tales como redes de área personal (PAN), redes locales (LAN) o redes de área amplia de bajo consumo (LPWAN) como LoRaWAN. En esta capa, la conectividad 5G desempeña un papel crucial al permitir una alta densidad de dispositivos con baja latencia [23].
- **Capa de Aplicación:** Es la interfaz final donde el usuario interactúa con la información procesada. En este nivel se encuentran los sistemas de monitoreo, aplicaciones móviles y paneles de control (dashboards) que traducen los datos técnicos en información comprensible para la toma de decisiones en sectores como la agricultura inteligente o la industria 4.0.

Beneficios:

La adopción de sistemas de IoT conlleva una serie de beneficios significativos, entre los que destacan:

- **Eficiencia Energética:** reducción del consumo energético mediante el control automático de luces y sistemas de climatización.
- **Optimización de Recursos:** mejora en la utilización de recursos como la iluminación y la climatización mediante la automatización basada en la ocupación

2.2.2.1. Normativas del internet de las cosas

- **Y.4000/Y.2060** - Descripción general de IoT: Esta recomendación establece un marco de referencia fundamental para el Internet de las cosas (IoT) y las aplicaciones de máquina a máquina (M2M). En el contexto del IoT, numerosos dispositivos están interconectados para recopilar, transmitir y procesar datos, lo que permite una amplia gama de aplicaciones en áreas como la industria, el hogar inteligente, la salud, la agricultura y la logística. Esta normativa resalta la importancia de estandarizar los protocolos de comunicación, los modelos de datos y las arquitecturas de sistemas para garantizar la interoperabilidad y la seguridad en los entornos IoT [24].
- **Y.405.0/Y.2069** - Términos y definiciones para IoT: Este estándar proporciona una lista de términos y definiciones específicos relacionados con el Internet de las cosas. Esto es crucial para establecer un lenguaje común y garantizar la comprensión uniforme de los conceptos dentro de la comunidad IoT. Incluye términos como "dispositivo IoT", "plataforma de IoT", "sensor", "actuador", entre otros, que son fundamentales para la comunicación efectiva en este campo [24].
- **Y.4100/Y.2066** - Requisitos comunes de IoT: Este estándar clasifica las demandas de los sistemas de IoT en diferentes categorías, incluyendo requisitos de seguridad y protección de la privacidad. Establece los requisitos funcionales para la captura, almacenamiento, transferencia y procesamiento de datos en entornos IoT, así como los roles de los participantes involucrados. Además, presenta casos de uso que ilustran cómo se aplican estos requisitos en situaciones del mundo real [24].

- **Y.4103/F.748.0** - Requisitos comunes para aplicaciones de IoT: Enfocado específicamente en las aplicaciones de IoT, este estándar detalla los requisitos necesarios para garantizar el funcionamiento efectivo y seguro de las aplicaciones IoT en diversas áreas. Esto incluye aspectos como la interoperabilidad entre dispositivos, la gestión eficiente de datos y la seguridad de extremo a extremo [24][25].
- **Y.4552/Y.2078** - Modelos de soporte de aplicaciones de IoT: Este estándar presenta tres modelos de soporte de aplicaciones (configurables, adaptables, confiables) junto con sus fundamentos. Estos modelos complementan los requisitos establecidos en Y.4103/F.748.0, proporcionando una estructura para el desarrollo y despliegue de aplicaciones IoT que sean flexibles, robustas y confiables [25].
- **Y.4111/Y.2076** - Requisitos basados en la semántica y marco del IoT: Este estándar define requisitos para las capacidades de seguridad en el contexto del IoT, haciendo uso de tecnologías semánticas para la toma de decisiones relacionadas con la seguridad. Propone la anotación semántica, la gestión de políticas de seguridad y el control de acceso como mecanismos para mejorar la seguridad en los sistemas IoT [25].

2.2.3. Plataforma de gestión y control IoT

Una plataforma de Internet de las Cosas es el entorno tecnológico que permite la administración, integración y supervisión centralizada de los dispositivos conectados dentro de un sistema automatizado. A diferencia del concepto general de IoT, que describe la interconexión de objetos inteligentes, la plataforma constituye la capa de gestión encargada del registro de dispositivos, almacenamiento y procesamiento de datos, autenticación de usuarios y ejecución de reglas de control. Estas plataformas facilitan la comunicación bidireccional entre sensores, actuadores y unidades de control mediante protocolos como HTTP o MQTT, permitiendo la visualización de información en tiempo real a través de interfaces web o aplicaciones móviles. De esta manera, la plataforma IoT actúa como el núcleo coordinador del sistema, garantizando interoperabilidad, escalabilidad y control remoto eficiente dentro de un entorno integrado [26].

2.2.4. Infraestructura de comunicación en sistemas tecnológicos

Los sistemas tecnológicos modernos, como los basados en el microcontrolador ESP32, dependen de una infraestructura de comunicación eficiente para facilitar la transferencia de datos. Esta infraestructura se compone de dos elementos principales: los módulos de comunicación y los sistemas de comunicación.

Los módulos de comunicación son componentes específicos diseñados para gestionar la conectividad dentro de una red. En el ecosistema del ESP32, estos módulos están integrados directamente en el silicio (SoC), permitiendo la transmisión de información mediante interfaces de hardware como UART, SPI e I2C, y protocolos lógicos que fomentan la interoperabilidad. Estos módulos proporcionan los medios físicos y lógicos para la conexión entre sensores y actuadores, destacando especialmente por su pila de protocolos integrada para comunicación inalámbrica [27].

Por otro lado, los sistemas de comunicación en el ESP32 constituyen un conjunto de dispositivos destinados a la transmisión y recepción de señales de voz, datos, audio y video. Este sistema es fundamentalmente híbrido, ya que el ESP32 actúa como un nodo central que integra:

- Transmisores y receptores: Radios de radiofrecuencia (RF) internos que operan en la banda de 2.4 GHz.
- Medios de transmisión: El canal inalámbrico, empleando tecnologías como Wi-Fi (802.11 b/g/n) para conectividad a Internet y Bluetooth (v4.2 BR/EDR y BLE) para comunicación de bajo consumo con dispositivos móviles o mallas de sensores [28].

En conjunto, la integración de estos módulos y sistemas en el ESP32 permite establecer una infraestructura de comunicación robusta y eficiente. Gracias a su capacidad de procesamiento y su flexibilidad en el uso de diversos medios de transmisión, el ESP32 logra una conectividad cohesiva y segura, ideal para aplicaciones de Internet de las Cosas (IoT) y sistemas tecnológicos complejos.

2.2.5. Protocolo de comunicación

Un protocolo de comunicación es un conjunto de reglas y estándares que determinan cómo dos o más dispositivos deben intercambiar información de manera eficiente y efectiva. Estas reglas establecen el formato de los mensajes, la secuencia de operaciones, el control de errores, la

autenticación y otras características necesarias para una comunicación exitosa entre los sistemas. Los protocolos de comunicación son esenciales para garantizar la interoperabilidad y la fiabilidad en entornos de redes y sistemas distribuidos [29].

2.2.2.1. Sistema Master – Slave

El modelo maestro-esclavo representa un protocolo de comunicación en el que un dispositivo, denominado maestro, tiene el control sobre uno o varios dispositivos conocidos como esclavos. En este tipo de comunicación, el maestro toma la iniciativa, mientras que los esclavos solo responden a solicitudes [30]. En el contexto del ESP32, este modelo es fundamental cuando el microcontrolador actúa como el "cerebro" central que coordina periféricos externos.

El ESP32 asume el papel de administrador de la red y actúa como mediador mediante protocolos de comunicación serial integrados en su hardware:

- **SPI (Serial Peripheral Interface):** El ESP32 opera como maestro para comunicarse a alta velocidad con pantallas LCD, tarjetas SD o sensores complejos. Aquí, el ESP32 genera la señal de reloj y selecciona específicamente a qué esclavo hablar mediante una línea de selección (CS).
- **I2C (Inter-Integrated Circuit):** El ESP32 gestiona múltiples esclavos (como sensores de temperatura o acelerómetros) utilizando solo dos hilos, asignando una dirección única a cada dispositivo para realizar preguntas periódicas y recolectar datos [31].

A pesar de sus ventajas de organización jerárquica, este modelo en el ESP32 presenta limitaciones. Una desventaja crítica es la dependencia del maestro; si el ESP32 falla o entra en un ciclo de error, toda la red de sensores y actuadores queda paralizada. Además, la comunicación puede ser lenta en protocolos como I2C si el maestro debe consultar a demasiados dispositivos en secuencia.

A continuación, se mencionan algunos protocolos de comunicación empleados en este modelo y su relación con el ESP32:

- **Modbus:** Protocolo serial ampliamente usado en la industria. El ESP32, mediante sus puertos UART, puede configurarse como un maestro Modbus para enviar comandos de

lectura o escritura a PLCs o sensores industriales, adaptando los niveles de voltaje mediante transceptores (como el RS485).

- **CAN (Controller Area Network):** Aunque el texto original menciona que es multi-maestro, el ESP32 incluye un controlador CAN interno (ISO 11898-1). Esto le permite integrarse en redes automotrices donde puede actuar como un nodo que solicita datos específicos a otros módulos del vehículo.
- **PROFIBUS:** Estándar industrial donde el ESP32 puede integrarse como un nodo esclavo o maestro de bajo costo utilizando pasarelas (gateways) de comunicación, facilitando la interconexión entre sistemas de automatización robustos y redes de datos modernas.

En conjunto, el ESP32 aprovecha este modelo para garantizar una gestión centralizada y eficiente, permitiendo que un solo chip coordine una infraestructura tecnológica compleja de manera cohesiva [31].

2.2.6. Gateway o puerto de enlace

Los sistemas tecnológicos modernos, como los basados en el microcontrolador ESP32, requieren una infraestructura de comunicación robusta para gestionar el flujo de datos. En este contexto, el ESP32 no solo actúa como un nodo de procesamiento, sino que desempeña una función crítica como puerto de enlace o intermediario. Este procedimiento permite que el dispositivo funcione como un puente entre dos redes que, de lo contrario, no podrían establecer comunicación debido a discrepancias en los protocolos empleados, direcciones IP o características físicas de cada red [18].

En el caso específico del ESP32, esta capacidad de enlace se manifiesta en su arquitectura de red híbrida:

- **Interoperabilidad de Protocolos:** El ESP32 puede recibir datos de una red de sensores locales mediante protocolos de bajo consumo como Bluetooth Low Energy (BLE) o ESP-NOW y, simultáneamente, retransmitir dicha información hacia la nube o un servidor central a través de una red Wi-Fi.
- **Traducción de Datos:** Actúa como el punto donde se resuelven las diferencias jerárquicas y lógicas, permitiendo que dispositivos esclavos en una red local se integren a infraestructuras globales de Internet de las Cosas (IoT).

- **Gestión de Redes:** Gracias a su pila TCP/IP integrada, el dispositivo gestiona el direccionamiento y la seguridad necesarios para que los componentes de la red interna se comuniquen de manera cohesiva y segura con redes externas de mayor jerarquía [32].

Esta integración asegura que la infraestructura de comunicación sea eficiente, permitiendo que el ESP32 centralice la transferencia de datos y garantice la conectividad en entornos tecnológicos heterogéneos.

2.2.7. Interfaz de usuario

Es el medio a través del cual un usuario interactúa con un sistema informático, software o dispositivo. La interfaz de usuario proporciona los elementos visuales y las herramientas que permiten al usuario comunicarse y operar la funcionalidad del sistema de manera intuitiva y eficiente. Su objetivo principal es facilitar la interacción entre el usuario y la máquina. Una interfaz de usuario busca mejorar la usabilidad y la experiencia del usuario, considerando factores como la accesibilidad, la claridad de la información presentada y la eficiencia, facilita la interacción, minimiza la posibilidad de errores. Una interfaz de usuario intuitiva, accesible a través de aplicaciones móviles o plataformas web, permite a los usuarios monitorear el estado de las aulas, recibir alertas y ajustar configuraciones según sea necesario [33].

2.2.8. Dispositivos electrónicos

Los dispositivos electrónicos son sistemas que utilizan tecnología electrónica para realizar funciones específicas mediante el control y manipulación de señales eléctricas o campos electromagnéticos. Estos pueden operar en entornos analógicos o digitales y se aplican en múltiples áreas, desde la electrónica de consumo hasta la automatización industrial, constituyendo la base tecnológica de los sistemas inteligentes modernos [34]. En un sistema automatizado, los dispositivos electrónicos permiten la adquisición de información, el procesamiento de datos y la ejecución de acciones sobre el entorno, integrando componentes de control, monitoreo y comunicación.

2.2.2.1. Sensores

Los sensores son dispositivos electrónicos que permiten detectar cambios en magnitudes físicas del entorno y transformarlos en señales eléctricas que pueden ser transmitidas y procesadas dentro de un sistema de comunicación o control [35].

- **Sensores de presencia:** Estos dispositivos están diseñados para identificar la presencia de objetos o personas en su entorno mediante la captura y evaluación de radiación térmica. Funcionan al convertir los cambios en la imagen térmica de su área de detección en señales eléctricas cuantificables. Al identificar alteraciones en la radiación térmica, activan o desactivan dispositivos conectados, como luces y otros equipos electrónicos, en respuesta a la presencia detectada. La principal diferencia entre los detectores de movimiento y los sensores de presencia reside en la sensibilidad de sus componentes.

Los sensores de presencia son dispositivos diseñados para detectar la presencia de personas u objetos mediante la identificación de variaciones en la radiación térmica del entorno. Estos sensores convierten los cambios en la energía infrarroja en señales eléctricas que pueden ser procesadas por sistemas electrónicos para activar o desactivar dispositivos asociados. Debido a su alta sensibilidad, pueden dividir su área de detección en múltiples zonas, permitiendo registrar movimientos muy sutiles dentro del espacio monitoreado [36].

Para este proyecto se utiliza el sensor PIR. Su funcionamiento se basa en un cristal piroeléctrico que detecta la radiación infrarroja (ver Figura 4). Utiliza un lente de Fresnel que divide el área en sectores; cuando un cuerpo con calor atraviesa estos sectores, el sensor genera un pulso digital de 3.3V hacia el ESP32.

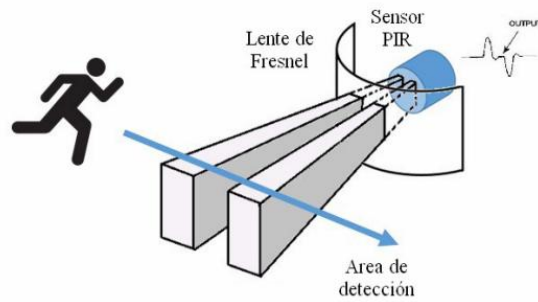


Figura 4. Funcionamiento del sensor PIR

- **Sensores por contacto:** Representan dispositivos especializados en la seguridad residencial o comercial, diseñados para detectar aperturas no autorizadas en puertas y ventanas. Estos dispositivos se instalan directamente en las superficies de las puertas y ventanas y se activan cuando se produce una separación entre el sensor y un imán emparejado. Estos sensores pueden encontrarse en configuraciones inalámbricas o cableadas según la aplicación requerida [37]. La característica destacada de estos dispositivos radica en su sencillez de instalación, ya que muchos sensores se pueden fijar con cinta adhesiva, evitando así la necesidad de perforar agujeros en las superficies. Estos sensores son versátiles y están disponibles en configuraciones inalámbricas o cableadas para adaptarse a diversas necesidades. La facilidad de instalación es una característica destacada, ya que muchos sensores se pueden montar con cinta adhesiva, lo que simplifica el proceso y elimina la necesidad de perforar agujeros en las superficies.

el sistema utiliza sensores de contacto magnético de 10cm basados en la tecnología Reed Switch. El funcionamiento interno consiste en un interruptor contenido en una cápsula de vidrio con dos láminas metálicas. En estado de reposo (puerta cerrada), el campo magnético del imán mantiene las láminas unidas, cerrando el circuito. Al abrir la puerta, el imán se aleja, las láminas se separan por su propia tensión mecánica y el circuito se abre, enviando un cambio de nivel lógico (de HIGH a LOW o viceversa) al pin de entrada del microcontrolador para activar la alerta (ver Figura 5).

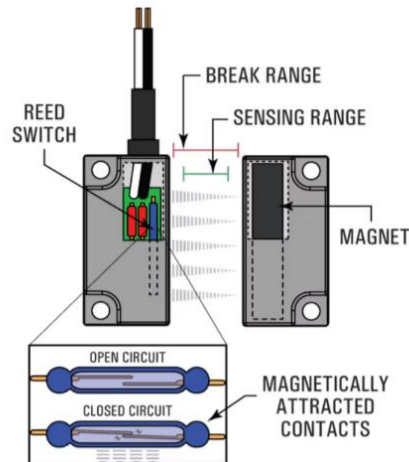


Figura 5. Sensor de contacto.

- Sensor de corriente:** Es un dispositivo que detecta la corriente eléctrica y la convierte en una señal de voltaje que puede ser fácilmente medida. Esta señal de voltaje es proporcional a la corriente que fluye a través del camino donde está ubicado el sensor. Existen diversos tipos de sensores, cada uno diseñado para funcionar en diferentes rangos de corriente y condiciones ambientales. Uno de los sensores más comunes es la resistencia de detección de corriente, que básicamente convierte la corriente en voltaje a través de una resistencia. La elección de la tecnología del sensor es crucial, dado que cada una puede ofrecer características específicas para diferentes aplicaciones.

Los sensores de corriente pueden utilizar tecnología de efecto Hall en circuito abierto o cerrado. En un sensor de circuito cerrado, se emplea una bobina que genera un campo magnético opuesto al campo producido por la corriente que se está midiendo. Aquí, un sensor Hall actúa como un dispositivo de detección y la señal de salida está directamente relacionada con la corriente medida.

Por otro lado, la Figura 6, en un sensor de bucle abierto, el flujo magnético generado por la corriente se enfoca en un circuito magnético y se detecta mediante un dispositivo Hall. La señal del dispositivo Hall proporciona una representación precisa e instantánea de la corriente primaria en este tipo de sensor. se utiliza específicamente el sensor de corriente no invasivo SCT-013. Este modelo se basa en el principio de un transformador de corriente (CT).

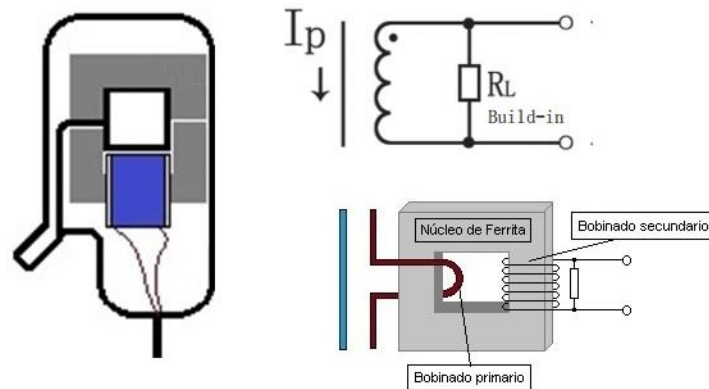


Figura 6. Sensor de corriente.

2.2.2.2. Actuadores

Tienen la capacidad de influir en un proceso automatizado, alterando los estados de un sistema. Su principal tarea consiste en inducir el movimiento de los elementos según las instrucciones provenientes de la unidad de control. El actuador recibe directrices de un regulador o controlador y produce una salida requerida para activar un componente de control final, transformando la energía de entrada en una forma utilizable para llevar a cabo una acción [38].

2.2.2.3. Luminarias

Son dispositivos que funcionan mediante la conversión entre energía eléctrica por medio de electrones al recibir una intensidad de corriente haciendo que ocurra un salto de orbita produciendo un choque de campos generando un efecto de energía llamado fotón, también la luz cambia de color cuando se encuentra dentro de envase encapsulado relleno de gases como el tungsteno. En efecto hoy en día tener un aula con un sistema de iluminación nos permite mantener el entorno del aula con claridad evitando mantener espacio con zonas oscuras o con sombras para evitar problemas cognitivos o fatiga visual al momento del cambio de visión entre la pizarra y el cuaderno, esto es muy importante al momento de llevar una buena percepción (ver Figura 7) [39].

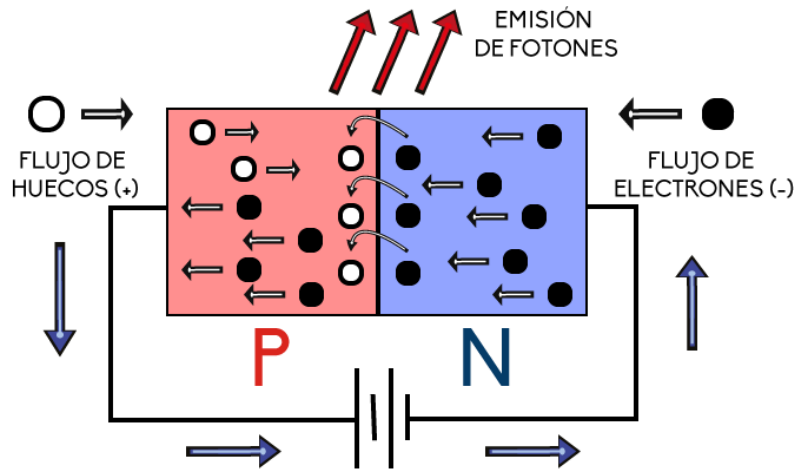


Figura 7. Principio de la luz.

2.2.2.4. Aire acondicionado

Su principio físico es generar aire frío desde su entorno teniendo en cuenta que va a utilizar aire caliente, físicamente aplica los conceptos de termodinámica en donde usa los estados de la materia entre líquido a gas un compuesto llamado refrigerante, todo lo realiza un motor compresor diseñado para trabajar sin descanso haciendo que por varillas de cobre circule el gas refrigerante hasta el evaporador haciendo cambiar la temperatura ambiente y volviendo un espacio confortable (ver Figura 8) [40].

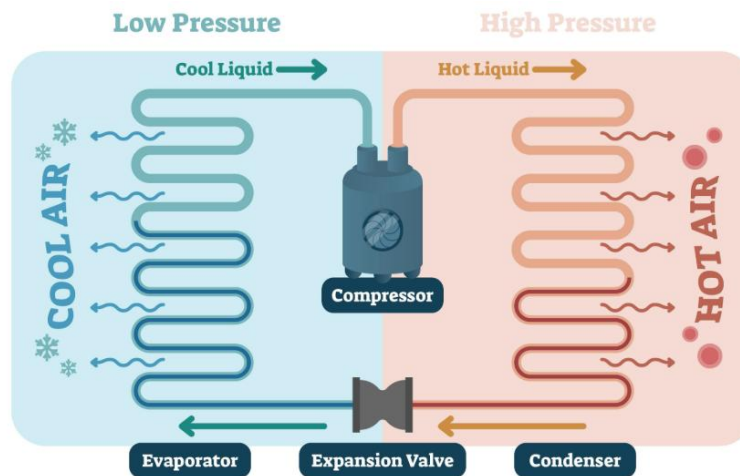


Figura 8. Función del aire acondicionado.

2.2.2.5. Cerradura electrónica

Una cerradura electrónica para puerta es un dispositivo de seguridad avanzado diseñado para controlar el acceso a un espacio mediante métodos electrónicos en lugar de llaves mecánicas tradicionales. A diferencia de las cerraduras convencionales, las cerraduras electrónicas utilizan componentes electrónicos, como sensores, circuitos y actuadores, para permitir o denegar el acceso de manera automatizada y, a menudo, más segura.

Estas cerraduras pueden variar en diseño y funcionalidad, pero suelen incluir características como teclados numéricos, lectores de tarjetas, escáneres de huellas dactilares o incluso tecnologías de reconocimiento facial. Los usuarios autorizados pueden desbloquear la cerradura introduciendo un código, utilizando una tarjeta de proximidad, escaneando su huella dactilar, entre otras opciones, dependiendo del tipo específico de cerradura electrónica.

La cerradura electrónica para puerta ofrece ventajas como la conveniencia de no necesitar una llave física, la capacidad de gestionar y auditar registros de acceso, así como la posibilidad de integrarse con sistemas de seguridad más amplios. Además, algunas cerraduras electrónicas permiten la programación de horarios de acceso y ofrecen notificaciones en tiempo real sobre la actividad de la cerradura [43].

Estas cajas suelen fabricarse con materiales como metal o plástico, siendo este último el más común. Están meticulosamente diseñadas para mantener las conexiones eléctricas a salvo de posibles accidentes y para evitar que los cables estén expuestos de manera peligrosa para las personas. Además, suelen estar construidas con materiales termoplásticos para proteger los conductos contra la entrada de agua o la exposición a la humedad.

funciona mediante un solenoide de 12V. Su funcionamiento técnico se basa en una bobina de cobre que, al recibir un pulso eléctrico, genera un campo electromagnético que retrae un perno metálico hacia el interior del cuerpo de la cerradura. Cuando la corriente se interrumpe, un resorte interno empuja el perno nuevamente a su posición original, manteniendo la puerta bloqueada (ver Figura 9).

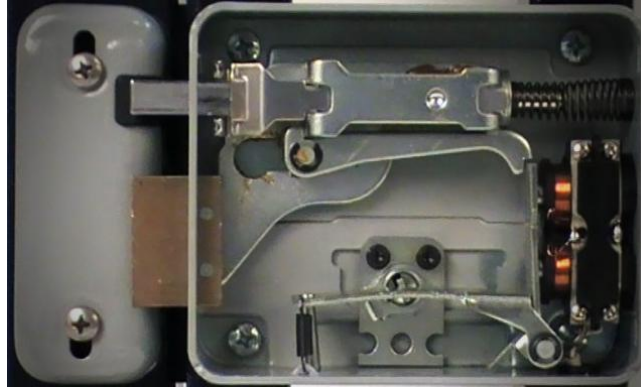


Figura 9. Estructura interna de la cerradura eléctrica.

2.2.2.6. Regulador de voltaje

Es un componente electrónico que permite regular o reducir el voltaje proveniente de una fuente de alimentación con un voltaje mayor en un circuito. Este dispositivo cuenta con la capacidad de ajustar el voltaje de salida mediante un potenciómetro multivuelta, brindando una alta precisión en la regulación [40].

Se trata de un módulo convertidor de potencia DC-DC, diseñado para manejar cargas de hasta 3 A con eficiencia, siendo compatible con plataformas como Arduino UNO, otras placas base y módulos básicos. En situaciones donde la corriente de salida supera los 2.5 A o la potencia de salida excede los 10 Watts, se recomienda añadir un disipador de calor para mantener un rendimiento óptimo [40].

En la Figura 10 muestra el regulador Buck LM2596, un convertidor conmutado que reduce el voltaje de entrada (12V) a un nivel seguro para el microcontrolador (5V). Su funcionamiento se basa en la conmutación de alta frecuencia, donde un interruptor interno abre y cierra el paso de corriente velozmente; junto con un inductor y un capacitor, 'suaviza' el voltaje para mantener una salida constante y eficiente, ajustada mediante su potenciómetro multivuelta.

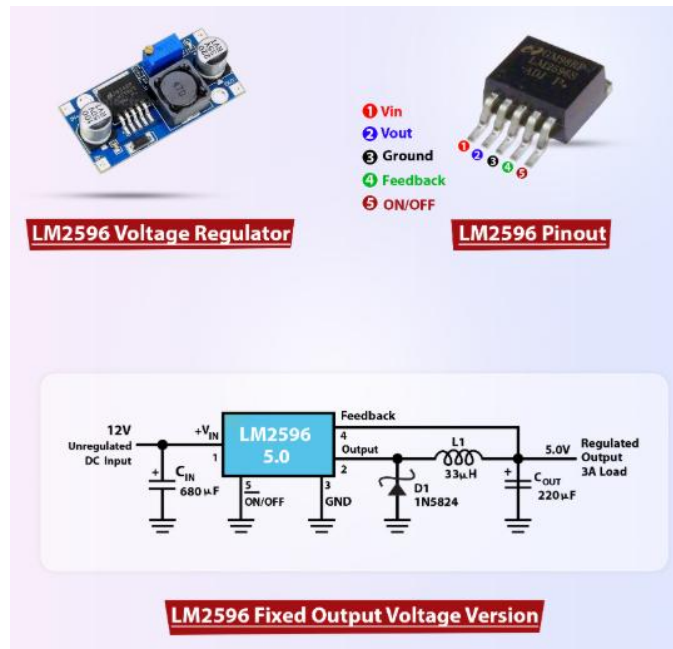


Figura 10. Esquemático regulador Buck LM2596.

2.2.2.7. Diodos

Un diodo infrarrojo es un componente semiconductor especializado que emite o detecta luz en la región infrarroja del espectro electromagnético. Constituido por una unión p-n de materiales semiconductores, como silicio o germanio, el diodo infrarrojo difiere de los diodos convencionales al estar diseñado específicamente para operar en el rango infrarrojo [41].

Cuando se aplica una corriente directa al diodo infrarrojo, los electrones se desplazan de la región de tipo n hacia la región de tipo p, mientras que los huecos se mueven en dirección opuesta. En la unión p-n, la recombinación de electrones y huecos libera energía en forma de luz infrarroja. Esta propiedad hace que los diodos infrarrojos sean útiles como emisores en sistemas de telecomunicaciones o aplicaciones de visión nocturna.

La Figura 11 muestra cuando se expone a luz infrarroja, el diodo puede actuar como detector, generando pares electrón-hueco que inducen una corriente medible. Esta capacidad se utiliza en sensores de proximidad, controles remotos y sistemas de detección de movimiento [41].

diodos IR emisores para actuar como un control remoto universal. Su funcionamiento se basa en la electroluminiscencia: al recibir una señal modulada del ESP32, el diodo convierte los pulsos

eléctricos en ráfagas de luz infrarroja imperceptibles al ojo humano. Estas ráfagas siguen un protocolo específico (como NEC o Sony) que permite comandar equipos externos como aires acondicionados o televisores desde el sistema de control.

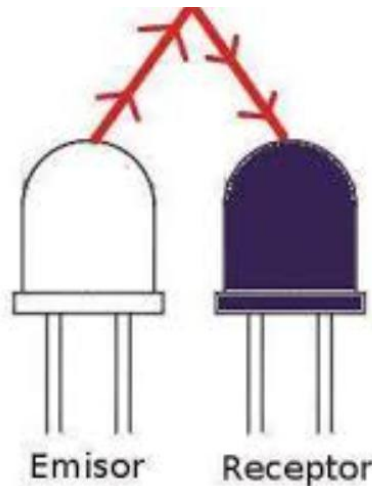


Figura 11. Funcionamiento del IR emisor.

2.2.2.8. Microcontroladores

En el mercado, hay diversas opciones disponibles al considerar la compra de un microcontrolador. Marcas reconocidas a nivel mundial ofrecen dispositivos con una mayor capacidad de procesamiento y diversas características, incluyendo combinaciones de periféricos integrados. Actualmente, a estas opciones se pueden agregar otras plataformas de desarrollo que contribuyen a reducir el tiempo de implementación, englobando el microcontrolador junto con la fuente, el grabador y la placa de expansión [41].

Un chip integrado tiene la capacidad de llevar a cabo instrucciones que han sido previamente almacenadas en su memoria, con el propósito de controlar dispositivos de entrada/salida. En el interior del chip, se incluyen una unidad central de procesamiento (CPU) y sus unidades de almacenamiento de memoria RAM y ROM. Algunos microcontroladores pueden aprovechar tecnologías inalámbricas como Wifi y Bluetooth, ya que la transmisión y recepción de datos a través de redes inalámbricas se está volviendo cada vez más común en este tipo de componentes.

Para este proyecto se ha seleccionado el ESP32, un microcontrolador de alto rendimiento tipo SoC (System on Chip). Su funcionamiento destaca por integrar procesadores de doble núcleo que

permiten la ejecución paralela de tareas, facilitando la gestión de la pila de protocolos Wi-Fi y Bluetooth sin interrumpir la lectura en tiempo real de los sensores periféricos conectados a sus pines de entrada y salida (ver Figura 12).

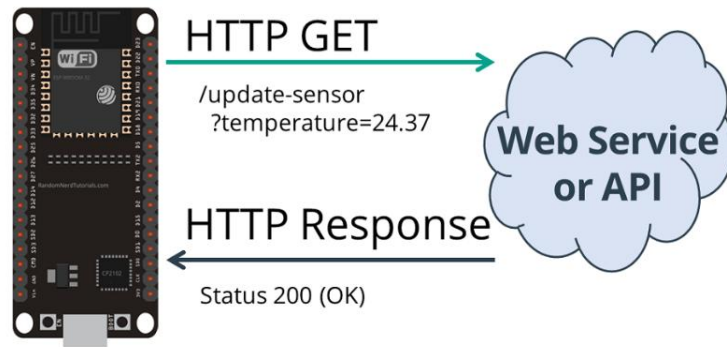


Figura 12. Ilustración del funcionamiento del ESP32.

2.2.2.9. Relé

Un relé es un dispositivo electromecánico que actúa como interruptor accionado eléctricamente. Funciona mediante una bobina que, al recibir corriente, genera un campo magnético actuando como un electroimán. Este campo magnético atrae los contactos del relé, cambiando su posición y permitiendo abrir o cerrar circuitos eléctricos. Con contactos como el normalmente abierto (NO) y normalmente cerrado (NC), el relé se utiliza en circuitos de control y secundarios. Los relés pueden tener diferentes cantidades de contactos y se utilizan en diversas aplicaciones, desde circuitos simples hasta relés de potencia diseñados para tensiones elevadas y conmutación de corrientes más altas. Su versatilidad y capacidad para controlar dispositivos eléctricos hacen que los relés sean fundamentales en la ingeniería eléctrica y la automatización (ver Figura 13) [42].

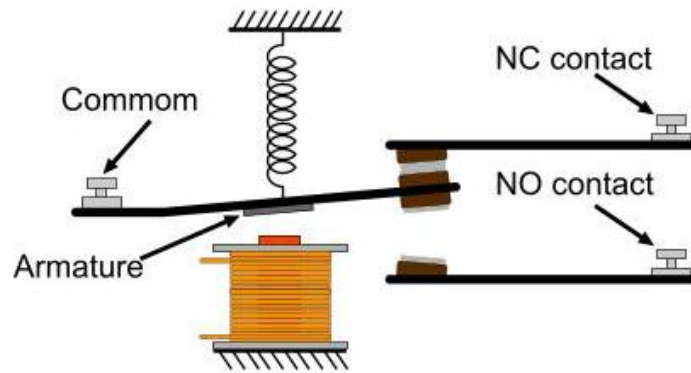


Figura 13. Funcionamiento interno del relé.

2.2.2.10. Convertidor AC-DC

Un convertidor AC-DC, también conocido como rectificador, es un dispositivo electrónico que convierte corriente alterna (AC) en corriente continua (DC). La corriente alterna es la forma de electricidad que se suministra a través de la red eléctrica convencional, mientras que la corriente continua es la que utilizan la mayoría de los dispositivos electrónicos para su funcionamiento. El proceso de conversión se realiza utilizando un puente rectificador, que consta de diodos semiconductores que permiten el paso de la corriente en una dirección mientras bloquean la corriente en la dirección opuesta, lo que resulta en una corriente continua en la salida del rectificador [44].

Los convertidores AC-DC se utilizan en una amplia variedad de aplicaciones, desde fuentes de alimentación para dispositivos electrónicos hasta sistemas de control de motores industriales. Son componentes fundamentales en la electrónica moderna, ya que permiten la operación de dispositivos que requieren corriente continua en entornos donde la corriente alterna es la forma de suministro de energía predominante [45].

2.2.9. Normativa para las instalaciones eléctricas

La normativa técnica que regula las cajas eléctricas o de conexión establece que estas deben funcionar como receptáculos críticos diseñados para aislar las uniones entre cables y conductos, garantizando no solo la eficiencia en la distribución de energía, sino también la seguridad y organización necesaria para resguardar el entorno contra riesgos de arcos eléctricos o posibles

cortocircuitos [46]. Bajo este marco legal y técnico, las instalaciones fijas deben cumplir estrictamente con la norma internacional IEC 60670, desarrollada por la International Electrotechnical Commission, la cual dicta los requisitos técnicos de diseño para envolventes destinadas a accesorios de baja tensión en entornos domésticos y comerciales. Este estándar internacional asegura la integridad del sistema eléctrico mediante la evaluación de criterios de protección mecánica frente a impactos, una alta estabilidad térmica bajo condiciones de carga y, fundamentalmente, la implementación de materiales con propiedades auto extingüibles probados mediante ensayos de hilo incandescente; estas características garantizan que cualquier falla térmica o combustión interna no se propague hacia los materiales de construcción, protegiendo de manera efectiva tanto la integridad estructural de la edificación como la seguridad física del usuario final [47].

2.3. MARCO TEÓRICO

Para llevar a cabo el presente estudio se tuvo en consideración las siguientes tesis, artículos que sirvieron como base para fundamentar y desarrollar la propuesta:

El monitoreo y control de las variables ambientales en espacios arquitectónicos cerrados constituye un aspecto fundamental para garantizar condiciones adecuadas de confort y seguridad en entornos educativos. La integración de tecnologías basadas en el Internet de las Cosas (IoT) permite la adquisición y transmisión de datos en tiempo real relacionados con parámetros como temperatura, humedad y calidad del aire, facilitando la toma de decisiones automatizadas dentro del sistema. Mediante el uso de microcontroladores programables, sensores electrónicos y plataformas de comunicación inalámbrica, es posible supervisar continuamente el comportamiento de los equipos y del entorno, optimizando el funcionamiento de los dispositivos y mejorando la eficiencia energética del espacio. Asimismo, la correcta configuración de los módulos de control y la comunicación entre dispositivos IoT resulta esencial para garantizar la fiabilidad del sistema, permitiendo el almacenamiento y análisis de la información recopilada para futuras acciones de mejora. Este tipo de soluciones tecnológicas demuestra la importancia de la automatización y del monitoreo inteligente de variables ambientales en aulas de clase y otros espacios cerrados, contribuyendo a mantener rangos adecuados de temperatura y humedad, así como a mejorar la experiencia de los usuarios dentro del entorno controlado [47].

El trabajo “Diseño e implementación de un prototipo escalable con infraestructura IoT para la medición y visualización de la calidad del aire usando tecnología open source para el edificio de la Carrera de Ingeniería en Networking y Telecomunicaciones”, desarrollado en la Universidad de Guayaquil en 2020 por Padilla Herrera y León Heredia, presenta un sistema basado en sensores y microcontroladores para el monitoreo de variables ambientales en espacios cerrados. La investigación demuestra la viabilidad del uso de tecnologías IoT para la recolección y visualización de datos en tiempo real dentro de entornos educativos, constituyendo un antecedente relevante para proyectos de supervisión y control automatizado en aulas y edificios inteligentes [48]. Este proyecto utiliza dispositivos como el Arduino y el módulo ESP8266 WIFI para recolectar datos sobre la calidad del aire, especialmente en un entorno educativo. La implementación de sensores como el MQ-2, capaz de detectar metano, butano y humo, alinea con la necesidad de monitorear variables específicas, similar al estudio, por lo que se explica con precisión la configuración e información relevante de estos dispositivos.

“Sistema de telegestión para el control del consumo eléctrico mediante una red IOT en la Facultad de Ingeniería en Sistemas, electrónica e industrial de la Universidad Técnica de Ambato”. El proyecto de titulación del estudiante Iván Ricardo Atiaja Soria, describe la implementación de un sistema de telegestión para controlar el consumo eléctrico en la Facultad, la utilización de la herramienta de programación gráfica Node RED subraya la importancia de interfaces visuales en la implementación de sistemas IoT, ofreciendo una forma intuitiva de recopilar y visualizar datos. La presencia de placas de desarrollo como Arduino y ESP-32 para la conexión de sensores y el control de dispositivos refuerza la aplicabilidad de estas plataformas en proyectos de automatización, resaltando la necesidad de programación y control preciso de componentes electrónicos. La incorporación de sensores BH1750 y SCT050 para medir luminosidad y consumo energético, respectivamente, coincide con la importancia otorgada a la medición precisa de variables. Además, el uso de herramientas como ZeroTier para la gestión externa subraya la relevancia de la conectividad en sistemas de control automatizado, proporcionando un ejemplo concreto de acceso remoto a datos. En conjunto, este proyecto ofrece un caso concreto que respalda y enriquece los principios y conceptos abordados en la investigación principal sobre IOT [49].

En la tesis titulada “Desarrollo de un prototipo de control de aforo en la Iglesia Czestochowa mediante el uso de sensores de movimiento con alarma y bloqueo de acceso automático de

puertas”. Publicado por la Universidad de Guayaquil en el año 2022, proyecto de titulación de los estudiantes Hermida Garófalo Gisell Alexandra, Vera Márgary Kent Bruce, para la Facultad de Ciencias Matemáticas y Físicas, carrera de Ingeniería en Networking y Telecomunicaciones. Esta investigación aborda el desarrollo de un prototipo mediante sensores de movimiento y una metodología en cascada. El sistema de control de aforo se compone de sensores FC-51 y PIR HC-SR501, conectados a una placa Arduino mediante un código de programación en lenguaje C++, mostrando la aplicabilidad de dispositivos electrónicos en la automatización de procesos. La inclusión de Arduino IoT Cloud para el monitoreo y control remoto resalta la importancia de la conectividad en sistemas IoT. El uso eficiente de sensores de movimiento e infrarrojo, junto con una alarma que advierte sobre el límite de aforo, demuestra la aplicación práctica y económica de tecnologías para la detección de personas en entornos concurridos. La visualización de datos en tiempo real a través de una pantalla LCD y la notificación mediante alarma y bloqueo de acceso a puertas aportan a la eficacia del monitoreo y control de aforo, destacando la importancia de sistemas automatizados en la gestión de situaciones específicas, como la pandemia. La metodología en cascada utilizada en el desarrollo del proyecto refuerza la planificación estructurada y la importancia de las etapas iniciales en la relación de resultados esperados [50].

CAPÍTULO III

DESARROLLO DE LA PROPUESTA

Los componentes son muy importantes para el desarrollo de la tesis, cada componente ya sea lógico o físico cumple un rol de desempeño para lograr toda la implementación, a continuación, se detalla de cada componente con su comparativa respectiva.

3.1. Componentes lógicos

La parte lógica es el inicio del desarrollo de la tesis, los softwares utilizados son indispensable para la programación y la automatización.

3.1.1. IDE Arduino 2.0.0

Es la versión mejorada de IDE Arduino manteniendo su código abierto con su ventaja de ser más eficaz en rapidez que su versión anterior, la mejor versión cuenta con la opción de autocompletado, para una mejor codificación [10]. IDE Arduino también es funcional para Windows, Mac OS X y Linux, para versión de 64 bits (ver Figura 14).



Figura 14. IDE Arduino versión 2.0.0.

3.1.2. Fritzing

Este software ayuda a la idealización lógica de los elementos físicos conectados entre la placa de desarrollo, sensores y actuadores como en la Figura 15, una de las cualidades es la extensa comunidad por lo que en la web se logra descargar las librerías de los componentes electrónicos que no están disponibles por defecto en el software.



Figura 15. Fritzing.

3.1.3. KiCad

Es una herramienta de software libre y de código abierto utilizada para el diseño de esquemas electrónicos y placas de circuito impreso (PCB). Esta plataforma permite desarrollar de manera integral proyectos electrónicos, desde la creación del diagrama esquemático hasta el ruteo y la generación de archivos de fabricación (Gerber). En el presente proyecto, KiCad se emplea para el diseño del circuito electrónico del sistema IoT, facilitando la correcta interconexión de los componentes, la verificación eléctrica del diseño y la elaboración de la placa que aloja el hardware embebido (ver Figura 16).



Figura 16. Kicad.

3.1.4. Autodesk Fusion 360

Es una plataforma integral de diseño asistido por computadora (CAD) que permite el modelado paramétrico y tridimensional de piezas y ensamblajes mecánicos. En el presente proyecto, Fusion 360 se utiliza para diseñar y modelar la carcasa del sistema, garantizando precisión dimensional, correcta disposición de los componentes electrónicos y optimización del material (ver Figura 17).



Figura 17. Fusion 360.

3.1.5. Portal web creado por el servidor del ESP32

Se implementa una página web generada por el servidor web embebido en la placa de desarrollo ESP32, destinada a la visualización y gestión de la información obtenida por el sistema IoT. Este portal permite al usuario acceder, mediante un navegador web, a los datos provenientes de los sensores y al estado de los dispositivos conectados, facilitando el monitoreo del sistema a través de una red local. La estructura de la página se desarrolla utilizando HTML como en la Figura 18, para la organización del contenido y CSS para la definición del diseño visual, permitiendo presentar la información de forma clara y organizada dentro del entorno web creado por el propio microcontrolador.



Figura 18. Interfaz de usuario del sistema automatizado.

3.2. Componentes físicos

Los componentes que intervienen son desde sensores, actuadores, microcontroladores, interruptores, relé, entre otros. Varios de ellos representan para parte de control de accionar y otros están enfocados en la comunicación por mando a distancia y por conexión lineal.

3.2.1. Tarjeta de desarrollo Esp32

Se realiza la mención de 3 modelos de microcontroladores fabricados por Expressis System, y se realiza la comparativa y con ello las diferencia entre sus características principales.

La tarjeta de desarrollo Esp32 de la empresa Expressis System, es ideal para el desarrollo del proyecto por sus características y funciones. Entre las funciones más destacadas es la incorporación de Wifi para aplicaciones de internet de las cosas y por su hardware de bajo consumo (ver Tabla 1).

Tabla 1. Características de tarjetas de desarrollo.

PARAMETRO TECNICOS	Arduino Mega	Esp8266	Esp32
Imagen de referencia			
Microcontrolador	ATMega	Esp8266	Esp32
Voltaje de operación	5V	3.3V	3.3V
Voltaje de alimentación	7V – 12V	7V – 12V	7V – 12V
Wi-fi	No	Si	Si
EEPROM	4 kb	4 kb	24Kb
Velocidad de reloj	16MHz	16MHz	16MHz

3.2.2. Sensores de movimientos

Los sensores de movimientos son los que detectan la presencia de un objeto ya sea con vida o inerte, es un dispositivo electrónico que permite censar un área y enviar una señal de dato para ser notificado. Las señales pueden ser ondas ultrasónicas o luz infrarroja. Dentro del proyecto se necesita que sistema automatizado tenga un alcance que pueda ajustarse, por lo que en la Tabla 2, se presentan varios sensores y sus características.

- Sensor infrarrojo
- Sensor presencia
- Sensor ultrasónico (analógico/ digital)

Tabla 2. Características de sensores de presencia.

PARAMETRO TECNICOS	HC-SR04	SHARP GP2Y0A02YK0F	PIR HC-SR501
MODELO			
Voltaje de operación	5V	4.5V – 5.5V	3.3V – 5V
Corriente de consumo	25mA	30mA	50uA
Ángulo efectivo	15 grados	25 grados	110 grados
Frecuencia de operación	40KHz	30MHz	50MHz
Pines de salida	4	3	3
Distancia de medida	2cm x 400cm	20cm x 150cm	300cm x 700cm
Corrientes de pines entrada/salida	50mA	40mA	40mA

El sensor escogido para el desarrollo de la tesis es el sensor PIR HC-SR501 por demostrar un amplio rango en grados de detección de objeto en movimiento.

3.2.3. HMI

En general, mantener una interfaz para que el usuario tenga la flexibilidad de monitorear los sensores o accionar un control a distancia es indispensable en proyectos de automatización. Por lo que, dentro de la propuesta, se considera implementar una pantalla que permita la interacción entre el sistema automatizado y el usuario. A continuación, en la Tabla 3 se consideran pantallas de varios modelos y características.

Tabla 3. Pantallas LCD.

PARAMETRO TECNICOS	LCD 16x2	TFT LCD TOUCH	NEXTION LCD
MODELO			
Voltaje de operación	3.3V – 5V	5.0V	5V
Corriente de consumo	25 mA	100 mA	90 mA
Tipo de comunicación	I2C	SPI	SERIAL

La pantalla que se utiliza en la tesis es la NEXTION LCD por ser multicolor, por su estructura, tamaños y poseer la funcionabilidad del táctil.

3.2.4. Regulador de voltaje LM2596

El regulador LM2596 que se muestra en la Figura 19, es un convertidor de voltaje DC-DC tipo reductor (buck) ampliamente utilizado en aplicaciones electrónicas que requieren una conversión estable de niveles de tensión. En el presente proyecto, el LM2596 se emplea para reducir el voltaje

de entrada proveniente de la fuente de alimentación a valores adecuados y estables para el correcto funcionamiento del ESP32 y los demás componentes electrónicos del sistema IoT.

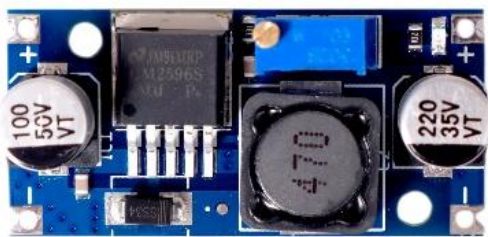


Figura 19. Regulador de voltaje LM2596.

3.2.5. Módulos relé 4 canales

El módulo relé de 4 canales que se muestra en la Figura 20, es un dispositivo de interfaz electromecánica que permite controlar cargas eléctricas de mayor potencia a partir de señales de bajo voltaje generadas por un microcontrolador. En el presente proyecto, este módulo se utiliza para el encendido y apagado de equipos eléctricos como iluminación y otros dispositivos instalados en las aulas.

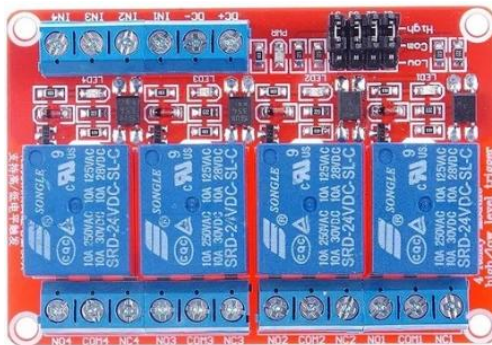


Figura 20. Módulo relé de 4 canales.

3.2.6. Diodo Led IR

El diodo LED infrarrojo (IR) que se presenta en la Figura 21, es un dispositivo optoelectrónico utilizado en el proyecto para emitir señales infrarrojas que permiten la comunicación y el control remoto de equipos electrónicos. Este componente convierte la señal eléctrica proveniente del microcontrolador, previamente amplificada, en radiación infrarroja invisible al ojo humano.



Figura 21. Led infrarrojo IR.

3.2.7. Transistor 2N4401

El transistor 2N4401 que se presenta en la Figura 22, es un transistor bipolar de unión (BJT) tipo NPN, utilizado en el presente proyecto para amplificar la señal infrarroja (IR) generada por el microcontrolador antes de su transmisión. Su función principal es aumentar la corriente de salida hacia el diodo emisor infrarrojo, permitiendo que la señal IR alcance una mayor potencia y alcance para el control de los equipos electrónicos, tal como el proyecto o aire acondicionado.

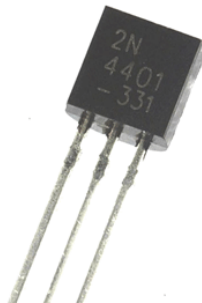


Figura 22. Transistor 2N4401.

3.2.8. Sensor de temperatura

Los sensores de temperatura son dispositivos electrónicos empleados en el proyecto para medir y monitorear las condiciones térmicas del aula en tiempo real, proporcionando información para la gestión del ambiente y la seguridad de los equipos eléctricos. Estos sensores convierten la temperatura en señales eléctricas que pueden ser interpretadas por el ESP32, permitiendo su visualización en la plataforma web y en el panel táctil local. A continuación, en la siguiente

Tabla 4.

Tabla 4. Sensores de temperatura.

PARAMETRO TECNICOS	LM35	LM35 SONDA	DHT11
MODELO			
Voltaje de operación	3.3V – 5V	3.3V – 5V	3.3V – 5V
Corriente de consumo	60 uA	60 uA	2.5 mA
Tipo de comunicación	Análogo	Análogo	Single bus
Precisión	$\pm 0.5^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$	$\pm 2^{\circ}\text{C}$

Se escoge el modelo de sensor DHT11 por el tipo de comunicación de bus simple y porque la forma de censado va de acorte al espacio cerrado del aula.

3.2.9. Sensor de contacto magnético

El sensor de contacto para ventana que se presenta en la Figura 23, es un dispositivo de tipo magnético utilizado en el proyecto para detectar el estado de apertura o cierre de ventanas, aportando una capa adicional de seguridad al aula inteligente. Su funcionamiento se basa en la proximidad entre un imán y un interruptor, lo que permite generar una señal digital que es procesada por el ESP32.



Figura 23. Sensor de contacto magnético.

3.3. Diseño de la propuesta

3.3.1. Diagrama de conexiones

En la Figura 24, se presenta de manera general el esquema de conexiones del sistema automatizado, donde se evidencia la correcta integración de los sensores, la pantalla HMI y los actuadores previamente descritos con la tarjeta de desarrollo ESP32. Asimismo, se aprecia que todo el hardware del sistema es alimentado mediante baterías de litio, las cuales permiten un modo de operación de emergencia, garantizando la continuidad del funcionamiento ante posibles cortes del suministro eléctrico en el campus universitario.

Los sensores no se conectan directamente a la placa de circuito impreso, debido a que varios de ellos requieren ser ubicados estratégicamente dentro del aula para un correcto funcionamiento. Tal es el caso del sensor de temperatura, el cual debe captar las condiciones ambientales del entorno, y del sensor de contacto, que se instala en la ventana mediante una bornera de dos pines y cableado externo. De igual manera, los sensores de presencia se colocan en puntos específicos del aula para garantizar una detección adecuada. En contraste, el sensor de corriente se encuentra integrado directamente a la placa, ya que su operación requiere la incorporación de resistencias y capacitores que forman parte del acondicionamiento de la señal y del diseño electrónico del sistema. (ver Figura 26).

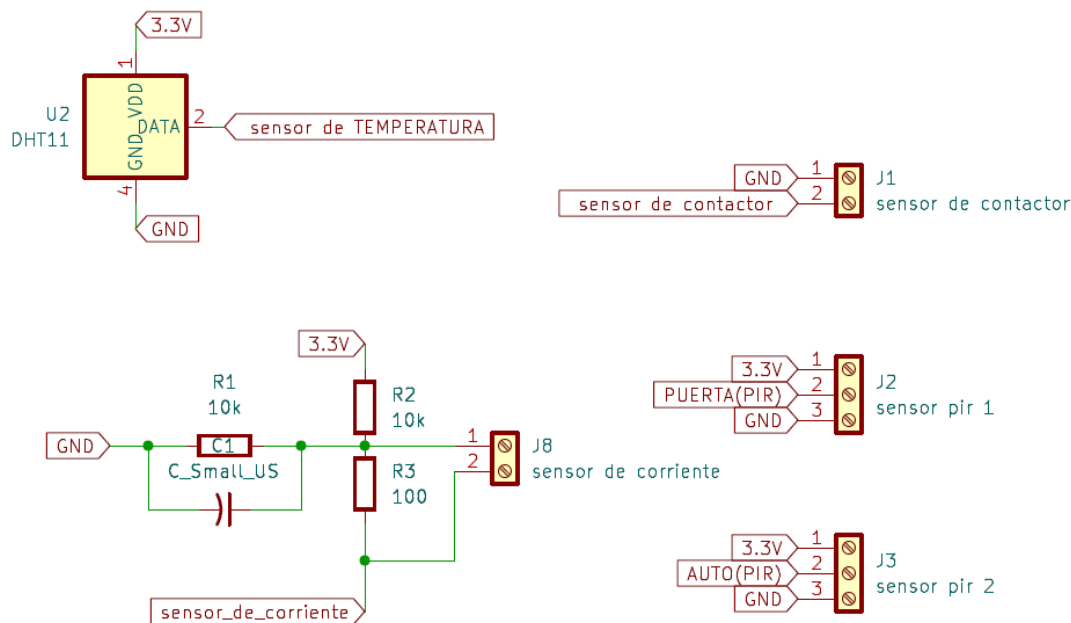


Figura 26. Conexión de sensores.

Para la integración de la tarjeta de desarrollo ESP32 en la placa de circuito impreso (PCB) se emplean espadines hembra, lo que facilita su montaje, reemplazo y mantenimiento. De manera complementaria, se utilizan espadines macho para la conexión de otros componentes del sistema, tales como la pantalla HMI, el módulo relé, el LED indicador del estado del sistema automatizado y el diodo infrarrojo (IR). Esta metodología de conexión mejora la modularidad del diseño, optimiza el ensamblaje del hardware y permite una mayor flexibilidad ante futuras modificaciones o ampliaciones del sistema. (ver Figura 27).

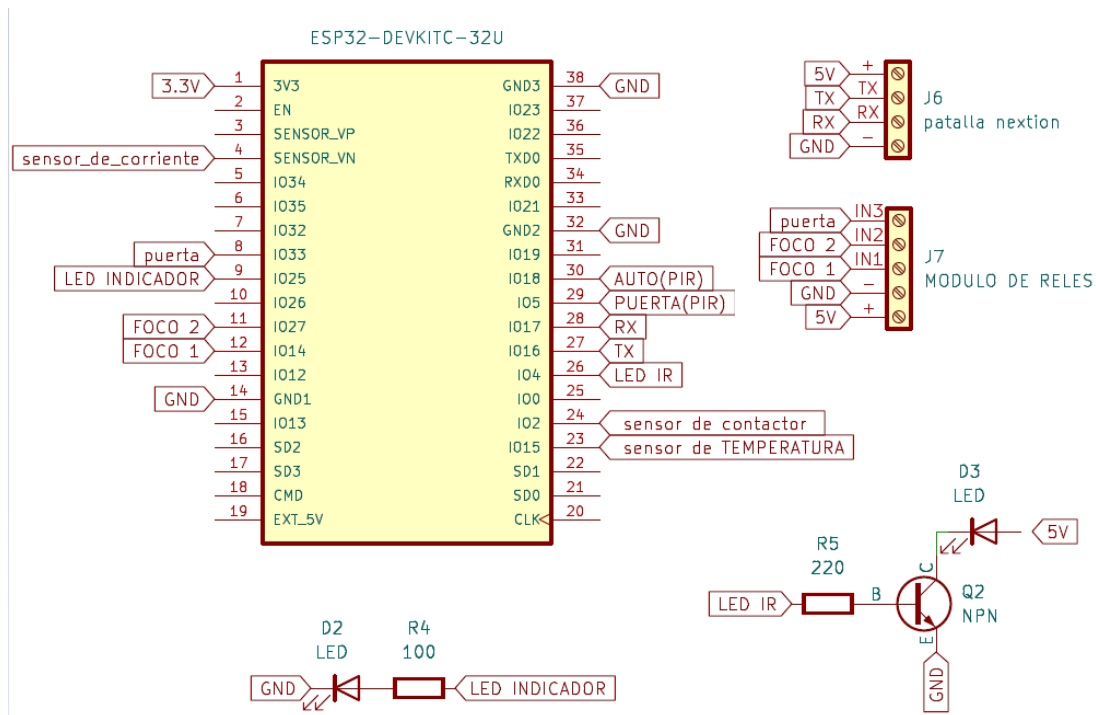


Figura 27. Conexiones con la tarjeta de desarrollo ESP32.

3.3.3. Diseño estructural 3D

La carcasa destinada a la protección del circuito impreso ha sido diseñada con el propósito de ser compacta, resistente y de fácil manipulación. Su fabricación se realiza mediante impresión 3D utilizando material PLA, lo que garantiza un adecuado equilibrio entre durabilidad, bajo peso y facilidad de fabricación. La estructura dispone de compartimientos diferenciados, donde el compartimiento principal alberga la placa de circuito impreso, permitiendo además la salida ordenada del cableado correspondiente a los sensores instalados en distintas áreas del aula. Adicionalmente, cuenta con un compartimiento secundario destinado al alojamiento seguro de baterías de litio tipo 18650. Esto último permite que el hardware del dispositivo sea de fácil manipulación y mantenimiento (ver Figura 28, 29, 30, 31).

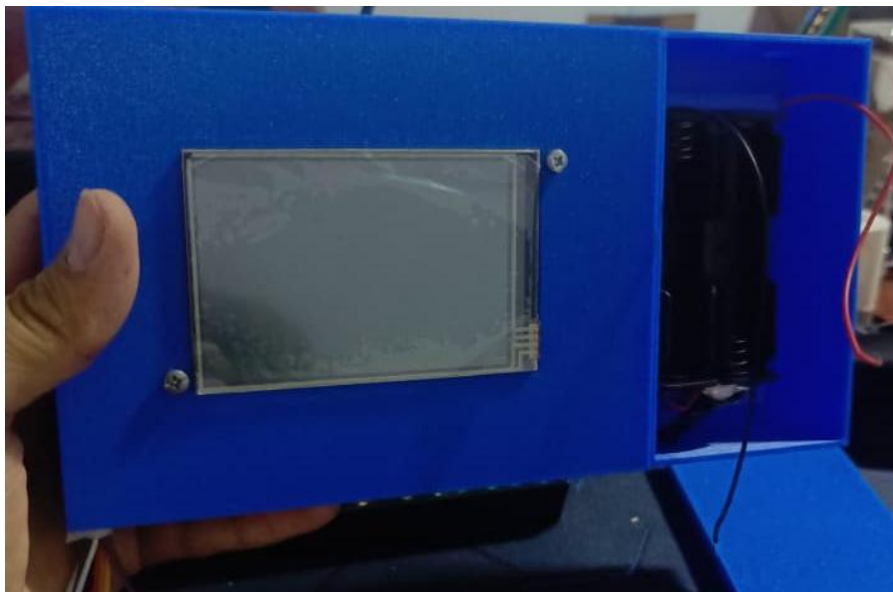


Figura 28. Carcasa de protección del hardware.

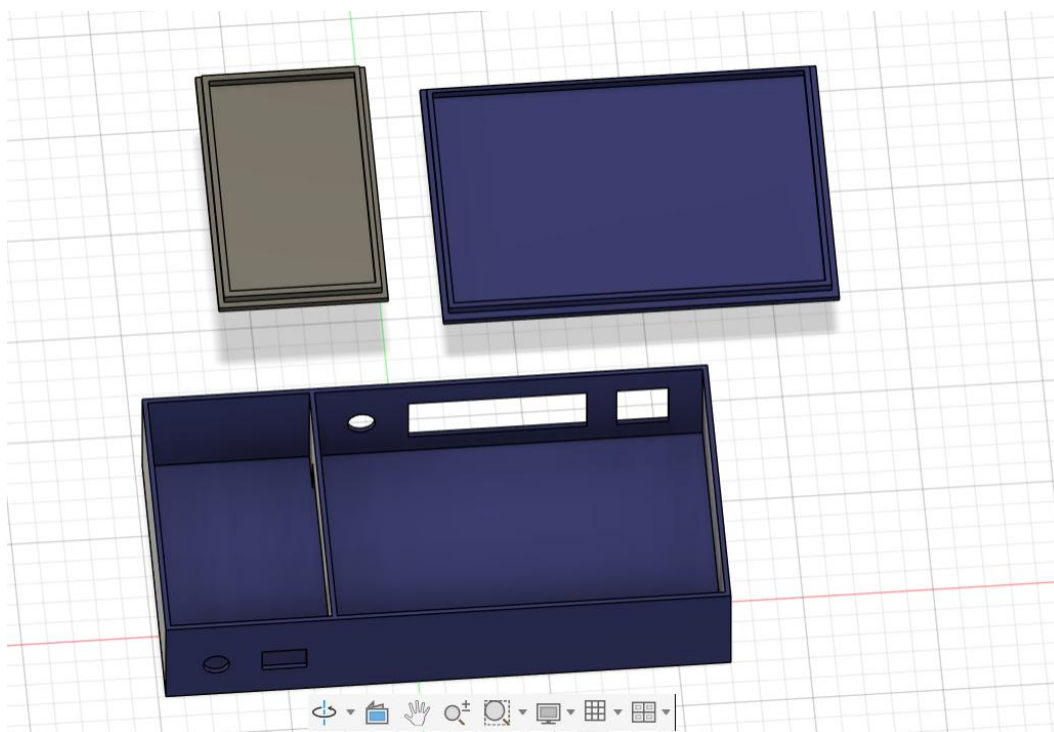


Figura 29. Carcasa modelada en 3D.

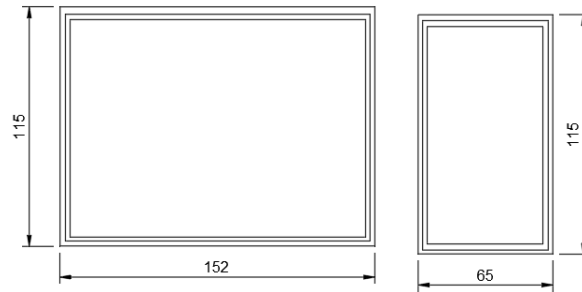


Figura 30. Medidas de las tapas.

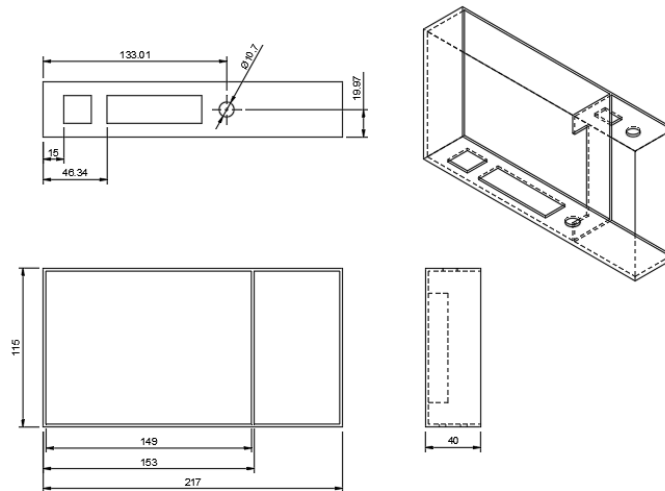


Figura 31. Medidas de la carcasa.

3.3.4. Consumo del dispositivo

A continuación, en la

Tabla 5 se presenta el consumo energético de cada uno de los dispositivos electrónicos que conforman el sistema automatizado propuesto. El valor total obtenido corresponde a un consumo aproximado, ya que no todos los componentes permanecen activos de forma continua. Algunos dispositivos presentan picos de consumo únicamente durante su activación, como es el caso del módulo relé de 4 canales. De manera similar, la pantalla HMI no se mantiene encendida permanentemente, puesto que el sistema ha sido configurado para entrar en modo de bajo consumo cuando no existe interacción por parte del usuario. Este enfoque permite optimizar el uso de energía y mejorar la eficiencia general del sistema.

Tabla 5. Consumo total del hardware.

Dispositivo	Consumo de corriente
ESP32	500 mA
Sensor PIR	6 mA
Sensor de corriente no invasivo SCT-013-030	0 mA
DHT11	2.5 mA
Sensor de contacto	5 mA
Módulo relé 4 canales	70 mA
Pantalla Nextion	90 mA
LED IR	20 mA
Cerradura eléctrica	250 mA
Total	943.5 mA

3.3.5. Costo de implementación

A continuación, en la Tabla 6 se presenta el cálculo del costo de implementación del sistema automatizado para las seis aulas consideradas en el proyecto. En este análisis no se contemplan trabajos civiles, ya que se busca minimizar intervenciones estructurales; sin embargo, sí se incluye un presupuesto destinado al uso de canaletas sobrepuestas, las cuales permiten una instalación ordenada, segura y adaptable del cableado necesario para el correcto funcionamiento del sistema.

Tabla 6. Costo de implementación de la propuesta.

Cant.	Descripción	P. Unit.	P. Total
6	ESP32	\$ 30,00	\$ 180,00
6	Pantalla Nextion 2.8"	\$ 45,00	\$ 270,00
6	Reguladores LM2596	\$ 6,00	\$ 36,00
6	Sensor de temperatura DHT11	\$ 30,00	\$ 30,00
6	Diodos IR	\$ 2,00	\$ 24,00

6	Módulo SCT-013	\$	10,00	\$	60,00
6	Sensor contacto magnético 10cm	\$	4,00	\$	24,00
6	Sensor de presencia PIR	\$	4,00	\$	24,00
6	Adaptadores de pared 12V/1A	\$	12,00	\$	72,00
6	Carcasa impresa en 3D	\$	40,00	\$	240,00
6	Relé 4 canales	\$	8,00	\$	48,00
2	Bobinas de cable UTP C5E 50M	\$	30,00	\$	60,00
40	Canaletas de 2" x 3m	\$	1,00	\$	40,00
6	Cerradura electrónica para puerta	\$	20,00	\$	160,00
1	Componentes electrónicos varios	\$	100,00	\$	100,00
6	Circuito impreso PCB	\$	100,00	\$	100,00
Total				\$	1468,00

3.4 Diseño del circuito PCB

3.4.1. Diseño y ruteo del circuito impreso.

En la Figura 32, muestra el ruteo de las pistas de cobre en la placa con sus respectivas terminales para la perforación y colocación de los elementos electrónicos.

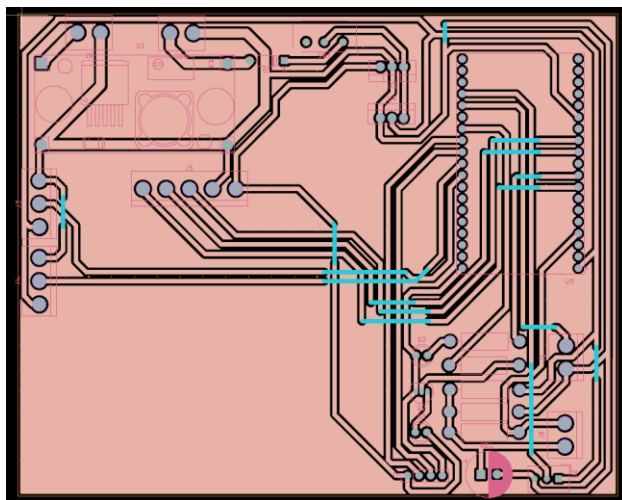


Figura 32. Ruteado de la PCB.

3.4.2. Diseño y modelado de la carcasa.

En la Figura 33, muestra el modelo compacto para la integración de la placa, pantalla y fuentes de regulación de manera que cumpla la estética.

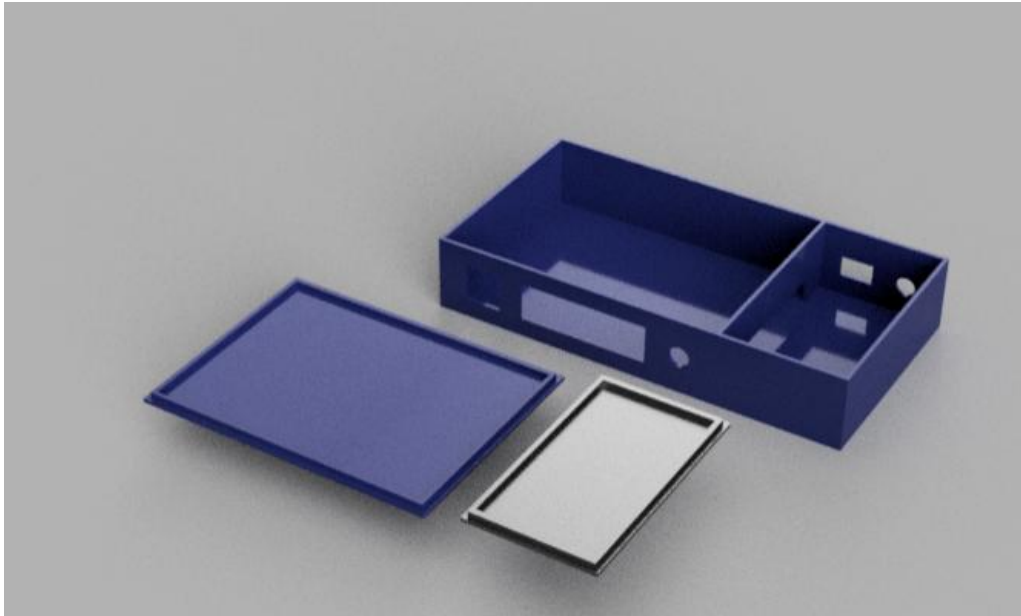


Figura 33. Renderizado de la caja.

CAPÍTULO IV

RESULTADOS

4.1. Comparativa y selección de componentes

La presente propuesta se caracteriza por su escalabilidad tanto a nivel de hardware como de software, gracias al uso de componentes electrónicos de código abierto. Esta elección representa una ventaja de suma importancia, ya que se apoya en una amplia comunidad de desarrollo, lo que facilita el acceso a documentación, soporte y mejoras continuas. Además, los costos de implementación resultan considerablemente menores en comparación con soluciones comerciales, las cuales, desde una perspectiva de desarrollo, suelen ser poco flexibles y limitadas en su capacidad de adaptación a las necesidades específicas del proyecto.

A continuación, en la Figura 34, se muestra un módulo didáctico que dispone de todos los componentes detallados en capítulos anteriores. Se utilizaron canaletas para el tendido eléctrico, se implementó una cerradura eléctrica que simula el acceso al aula. Además, de los sensores de presencia que notifican la disponibilidad del aula, como otros accesorios de suma importancia que le dan realce al proyecto: como es el sensor de temperatura y de contacto magnético para las ventanas.

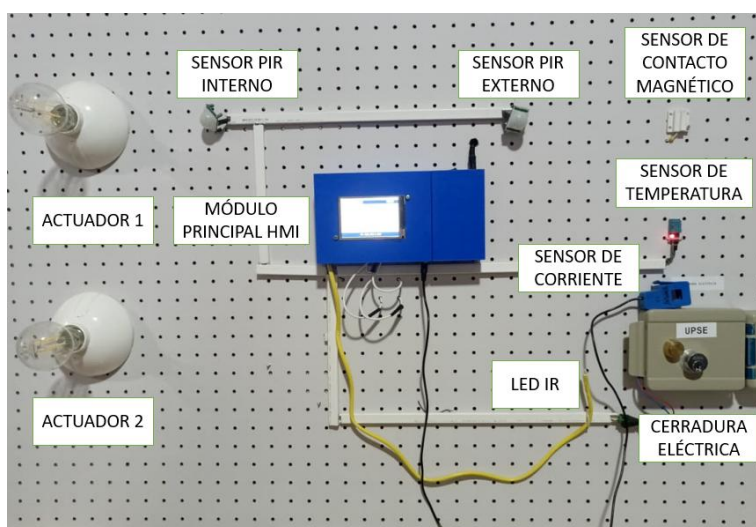


Figura 34. Módulo didáctico final del proyecto.

4.2. Funciones del sistema automatizado

El sistema automatizado tiene varias funciones por lo que a continuación se desglosará y describirá cada una de ellas.

4.2.1. Conexión a red WiFi

Para el inicio del sistema, es indispensable que el dispositivo cuente con una conexión estable a una red Wi-Fi, ya que esta garantiza la correcta comunicación y funcionamiento del sistema automatizado. Para ello, el ESP32 recupera las credenciales de la red inalámbrica almacenadas en su memoria no volátil mediante el sistema SPIFFS. Este proceso contempla dos escenarios de operación.

En el primer escenario, cuando el dispositivo no encuentra credenciales previamente registradas, crea automáticamente un punto de acceso como en la Figura 35, que permite al usuario, desde cualquier dispositivo inteligente, configurar la red Wi-Fi de manera sencilla.



Figura 35. Inicio de punto de acceso.

El usuario debe ingresar a la red inalámbrica de nombre “ESP32_Setup”, el cual no dispone de contraseña para su acceso y desde un navegador ingresar a la dirección IP 192.168.4.1, donde aparecerá una página de configuración como en la Figura 36. Una vez que se almacenen las nuevas credenciales, el dispositivo cada vez que se encienda se conectará automáticamente a esa red sin necesidad de volver a estar configurando.



The screenshot shows a web browser interface. At the top, a status bar indicates 'Sin conexión a Internet'. Below it, the address bar shows '192.168.4.1'. The main content area is titled 'Conectar a WiFi'. It contains a 'Red WiFi:' label, a dropdown menu with '(-59dBm)' selected, a 'Contraseña:' label, and an empty password input field. There is a checkbox labeled 'Mostrar contraseña' and a dark blue button labeled 'Guardar y Conectar'.

Figura 36. Configuración inicial desde la dirección IP.

En el segundo escenario, cuando las credenciales ya han sido almacenadas, el dispositivo se conecta de forma automática a la red configurada cada vez que se inicia, como se muestra en la Figura 37. Es importante que el dispositivo se mantenga conectado, por lo que se realiza un test cada 1 minuto para verificar la conexión.



Figura 37. Conexión exitosa a red Wifi.

Además, como modo desarrollador se tiene el monitor serial en la IDE Arduino mediante una conexión por USB podemos monitorear el estado de la red en los escenarios descritos con anterioridad, en la Figura 38, se muestra el monitor serial donde se crea el punto de acceso en el caso de no tener la red wifi disponible.

```
Credenciales cargadas:
SSID: TeamDevEc
Conectando a WiFi guardado: TeamDevEc
.....
Fallo la conexión. Iniciando AP...
Iniciando Punto de Acceso...
AP IP: 192.168.4.1
Escaneando redes WiFi...
```

Figura 38. Monitor serial: inicio de punto de acceso.

4.2.2. Latencia del dispositivo en la red

La medición de la latencia en el sistema automatizado es clave para analizar el tiempo de respuesta del enlace entre el dispositivo basado en ESP32 y la red de comunicación Wi-Fi sobre la cual se realiza el control de los equipos que se encuentran en el aula. Así, se llevaron a cabo un total de 100 pruebas de ping de forma consecutiva y se recogieron los resultados que pueden consultarse en la Tabla 7, donde se indica el valor de latencia en milisegundos en cada intento de intercambio con el dispositivo conectado. Estos resultados nos permitirán entender cómo se estabiliza y cómo responde la red en todo el proceso de intercambio de información.

Tabla 7. Latencia del dispositivo en red.

Ping #	Latencia (ms)	Ping #	Latencia (ms)	Ping #	Latencia (ms)	Ping #	Latencia (ms)	Ping #	Latencia (ms)
1	18	21	15	41	20	61	17	81	11
2	9	22	8	42	17	62	22	82	12
3	15	23	11	43	12	63	16	83	14
4	18	24	19	44	15	64	21	84	16
5	22	25	15	45	15	65	12	85	22
6	21	26	24	46	16	66	25	86	13
7	11	27	21	47	10	67	22	87	15
8	25	28	24	48	11	68	9	88	22
9	10	29	17	49	11	69	11	89	25
10	16	30	9	50	16	70	20	90	8
11	14	31	19	51	18	71	12	91	23
12	14	32	8	52	18	72	18	92	21

13	21	33	17	53	11	73	17	93	11
14	15	34	9	54	24	74	20	94	24
15	25	35	13	55	9	75	20	95	18
16	19	36	10	56	21	76	22	96	14
17	24	37	25	57	21	77	23	97	9
18	8	38	19	58	24	78	20	98	17
19	19	39	21	59	18	79	20	99	18
20	21	40	11	60	24	80	8	100	16

Del mismo modo, la Figura 39, correspondiente podría aportarnos información gráfica que muestre la evolución de la latencia a lo largo de las pruebas realizadas, evidenciar la respuesta del sistema a través de la evolución de lag. Este avance resulta relevante, dado que una latencia baja puede asegurar que las acciones de control y de monitoreo del sistema IoT queden registradas en el corto periodo que necesita el aula inteligente.

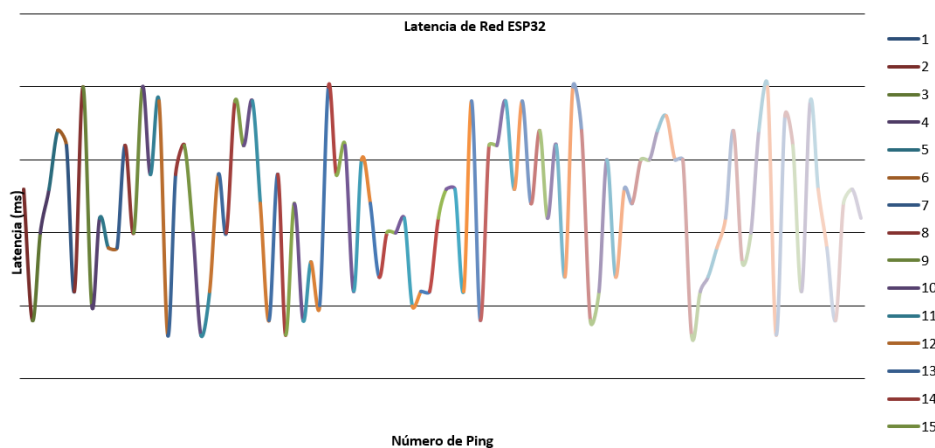


Figura 39. Gráfico de latencia del dispositivo en la red.

4.2.3. Densidad de redes WiFi

Para el desarrollo del presente proyecto, ha sido necesario el análisis de la densidad de redes Wi-Fi en el entorno, en la medida que el sistema automatizado, soporte del ESP32, depende parcialmente del uso de una conexión inalámbrica para el monitoreo y control de los dispositivos eléctricos. En base a la identificación de la cantidad de redes existentes en el área, puede saberse el grado de saturación del espectro, así como las interferencias entre los canales, con sus consiguientes efectos sobre la calidad de la señal, la latencia o la fiabilidad de la comunicación.

Considerando que el sistema se implementa en un campus universitario, con muchos puntos de acceso y dispositivos enlazados de manera simultánea, todo análisis resulta relevante para asegurar la existencia de una comunicación estable y segura en el funcionamiento del sistema automatizado. En este contexto, en la Tabla 8 se presenta un muestreo de señales en un rango de distancia de 1 a 5.5 metros.

Tabla 8. Potencia de señales WiFi-escaneadas

Muestra	Distancia (m)	Potencia Estimada (dBm)
1	1.00	-34
2	1.50	-39
3	2.00	-43
4	2.50	-45
5	3.00	-48
6	3.50	-49
7	4.00	-59
8	4.50	-63
9	5.00	-66
10	5.50	-77

4.2.4. Interfaces de usuario

A continuación, se registra la forma correcta de todas las funciones que contiene la pantalla táctil y como opera para el monitoreo de todos los actuadores y sensores de seguridad.

4.2.4.1. Pantalla de inicio

Como todo sistema tecnológico, las interfaces de usuario deben ser claras y amigables, de modo que faciliten la interacción con el usuario. Por esta razón, el sistema dispone de dos pantallas de inicio. La primera corresponde a la pantalla principal del sistema automatizado, la cual integra un menú de control que permite gestionar de manera sencilla los distintos dispositivos, tal como se muestra en la Figura 40. Por su parte, en la Figura 41, se presenta una pantalla de protección y ahorro de energía, diseñada para activarse cuando no existe interacción del usuario, lo que contribuye a la eficiencia energética y a la prolongación de la vida útil de la pantalla.



Figura 40. Menú de inicio.



Figura 41. Interfaz de modo de ahorro de energía.

4.2.2.2. Control de luminarias

El sistema automatizado, tal como se explicó en capítulos anteriores, dispone de dos modos de control: a través de la pantalla HMI y mediante el servidor web. Ambos modos se encuentran completamente sincronizados, de manera que cualquier cambio en el estado de los dispositivos conectados se refleja de forma inmediata en ambas interfaces. El control local mediante la HMI se presenta en la Figura 42, mientras que el control remoto a través del servidor web se muestra en la Figura 43.



Figura 42. Control de luces en el HMI.

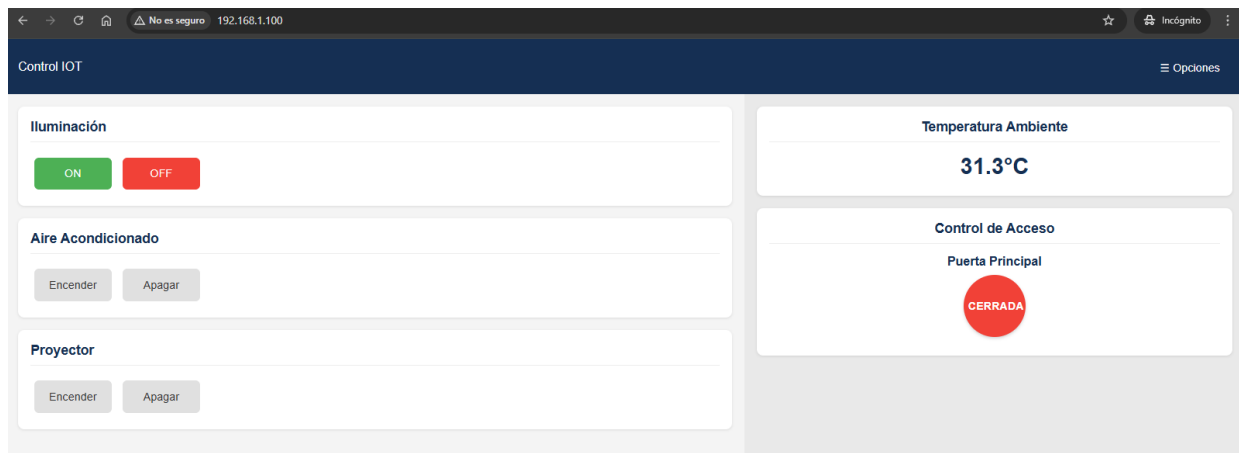


Figura 43. Control de luces en el servidor web.

4.2.2.3. Control seguro de encendido y apagado de A/C

El encendido y apagado seguro del aire acondicionado es importante para el ahorro de energía eléctrica, por lo que se tiene dos mecanismos para evitar que se encienda el aire acondicionado en situaciones donde se encuentren las ventanas abiertas o en su defecto que el aire acondicionado no se encuentre energizado. Como se aprecia en la Figura 44, se muestra una notificación de advertencia de conexión, asimismo, el botón de acción no se resalta porque la acción no fue efectiva.

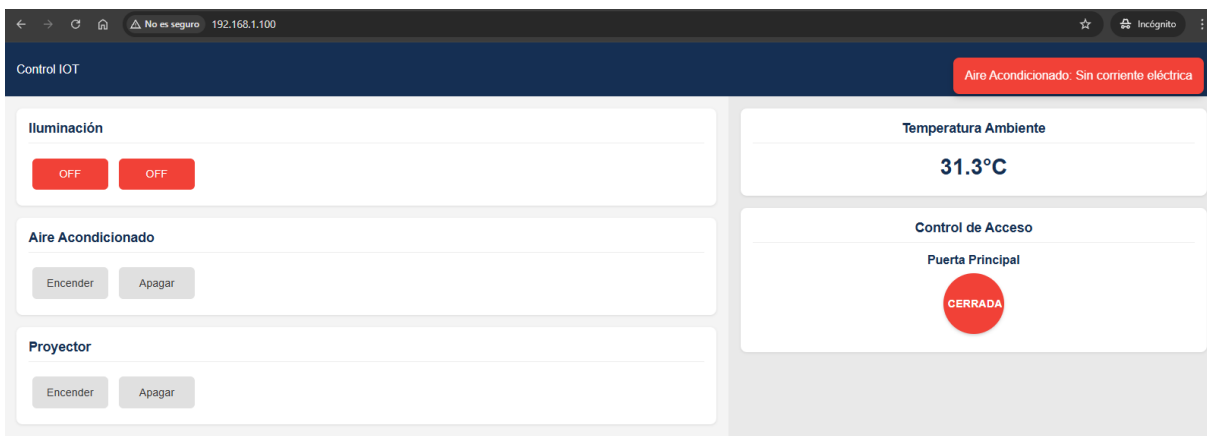


Figura 44. Error de encendido del aire acondicionado.

Los fabricantes de aires acondicionados no emplean un protocolo único de comunicación por infrarrojo, lo que da lugar a una amplia variedad de controles remotos y señales IR en el mercado. Frente a esta diversidad, el sistema automatizado propuesto ha sido diseñado con un enfoque flexible y adaptable, permitiendo la configuración de las señales infrarrojas directamente a través del monitor serial de la IDE de Arduino. Esta característica facilita la compatibilidad con distintas marcas y modelos de equipos. A continuación Tabla 9, se presenta un listado de protocolos IR correspondientes a diversos fabricantes de aires acondicionados disponibles en el mercado.

Tabla 9. Códigos de control IR de diferentes fabricantes.

Código	Fabricante	Código	Fabricante	Código	Fabricante
10	LG	54	VESTEL_AC	85	HITACHI_AC344
15	COOLIX	55	TECO	86	CORONA_AC
16	DAIKIN	57	TCL112AC	89	SANYO_AC
18	KELVINATOR	59	MITSUBISHI_HEAVY_88	90	VOLTAS
20	MITSUBISHI_AC	60	MITSUBISHI_HEAVY_152	92	TRANS_COLD

24	GREE	61	DAIKIN216	93	TECHNIBEL _AC
27	ARGO	62	SHARP_AC	94	MIRAGE
28	TROTEC	63	GOOD WEATHER	96	PANASONIC _AC32
32	TOSHIBA_AC	65	DAIKIN160	98	ECOCLIM
33	FUJITSU_AC	66	NEOCLIMA	100	TRUMA
34	MIDEA	67	DAIKIN176	101	HAIER_ AC176
38	HAIER_AC	68	DAIKIN128	102	TEKNO POINT
40	HITACHI_AC	69	AMCOR	103	KELON
41	HITACHI_AC1	70	DAIKIN152	104	TROTEC_ 3550
44	HAIER_AC_ YRW02	71	MITSUBISHI 136	105	SANYO_ AC88
45	WHIRLPOOL_AC	72	MITSUBISHI 112	108	RHOSS
46	SAMSUNG_AC	73	HITACHI_ AC424	109	AIRTON
48	ELECTRA_AC	78	DAIKIN64	111	HITACHI_ AC264
49	PANASONIC_AC	79	AIRWELL	113	HITACHI _AC296
51	LG2	80	DELONGHI _AC	115	HAIER_ AC160
53	DAIKIN2	84	CARRIER_ AC64	120	BOSCH144

Para realizar la configuración, es necesario enviar el valor “1” a través del monitor serial, tal como se muestra en la Figura 45. Al recibir este comando, la placa de desarrollo despliega automáticamente el listado de protocolos, de acuerdo con lo presentado en la Tabla 9. Posteriormente, y en función de la marca del aire acondicionado a controlar, el usuario debe ingresar nuevamente el código correspondiente, permitiendo así seleccionar y configurar el protocolo IR adecuado para el correcto funcionamiento del sistema.

Este proceso de configuración es fundamental y debe repetirse cada vez que se requiera, ya que con el paso del tiempo es posible que los aires acondicionados de las aulas sean reemplazados por distintos motivos, como mantenimiento, desgaste o actualización de equipos. Gracias a esta mecánica, el sistema se mantiene flexible y fácil de adaptar, permitiendo ajustar la configuración a nuevos dispositivos sin complicaciones y asegurando la continuidad del funcionamiento del sistema automatizado

```
--- Selección protocolo AIRE ---  
Selecciona un protocolo ingresando su número:  
[10] LG  
[15] COOLIX  
[16] DAIKIN  
[18] KELVINATOR  
[20] MITSUBISHI_AC  
[24] GREE  
[27] ARGO  
[28] TROTEC  
[32] TOSHIBA_AC  
[33] FUJITSU_AC  
[34] MIDEA  
[38] HAIER_AC  
[40] HITACHI_AC  
[41] HITACHI_AC1  
[44] HAIER_AC_YRW02  
[45] WHIRLPOOL_AC
```

Figura 45. Configuración de señal del aire acondicionado.

Finalmente, en el escenario ideal de funcionamiento, el aire acondicionado se encenderá correctamente y, de manera simultánea, el estado del equipo se actualizará en la interfaz HMI, reflejándose mediante el cambio del ícono correspondiente, tal como se muestra en la Figura 46. Esto permite al usuario confirmar visualmente que la acción ha sido ejecutada con éxito y que el sistema se encuentra operando de forma adecuada.



Figura 46. Estado encendido del aire acondicionado.

4.2.2.4. Latencia de comunicación con el aire acondicionado

Se realizaron diversas pruebas de funcionamiento para verificar el encendido y apagado de aires acondicionados de diferentes marcas, con el objetivo de evaluar la correcta transmisión de las señales infrarrojas generadas por el sistema automatizado. Estas pruebas permitieron analizar el comportamiento del sistema frente a distintos equipos y comprobar su compatibilidad. En la Tabla 10 se presentan los resultados obtenidos, donde se detallan las pruebas relacionadas con el encendido y apagado de los aires acondicionados, así como el tiempo de transición registrado en cada caso, lo que permite valorar la rapidez de la respuesta del sistema.

Tabla 10. Transición de encendido y apagado de aire acondicionado.

Prueba #	Marca	Protocolo	Código	Estado inicial	Estado final	Tiempo de transición
1	Indurama	LG2/Mirage	51	Off	On	1
				On	Off	8
2	RCA	LG2/Mirage	51	Off	On	1
				On	Off	7
3	Longtime	LG2/Mirage	51	Off	On	1
				On	Off	8
4	TCL	Midea	89	Off	On	1
				On	Off	8

5	LG	LG	10	Off	On	1
				On	Off	8
6	Innova	Gree	24	On	Off	1
				On	Off	7

4.2.2.5. Control de proyector

El control de encendido y apagado del proyector también se basa en el uso de protocolos infrarrojos (IR), los cuales, en cierta medida, se encuentran alineados con los presentados en la Tabla 7. En este caso, la configuración se realiza enviando el número “2” a través del monitor serial, seguido del código correspondiente al protocolo, de acuerdo con la marca y modelo del proyector. Este procedimiento permite que el sistema automatizado se adapte fácilmente a futuros cambios dentro del aula de clase, como la sustitución de equipos. En la Figura 47 se muestra el estado del proyector cuando se encuentra apagado, mientras que en la Figura 48 se presenta su estado una vez encendido.



Figura 47. Estado de apago del proyector.



Figura 48. Estado de encendido del proyector.

4.2.2.6. Notificaciones en tiempo real

El sistema automatizado incorpora un canal de notificaciones a través de la red social Telegram, utilizando su API de mensajería instantánea para establecer una comunicación directa con el usuario. Gracias a este enlace, el sistema es capaz de informar en tiempo real sobre distintos eventos y estados de funcionamiento. En la Figura 49, se muestran ejemplos de notificaciones generadas por diversas funciones del sistema, como el encendido automático del aire acondicionado cuando el sensor de temperatura detecta que se ha superado un valor predefinido. De igual manera, se notifican el estado del proyector y las advertencias que se generan cuando no es posible ejecutar determinadas acciones debido a las condiciones del aula.

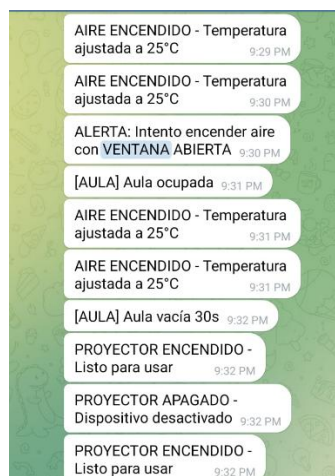


Figura 49. Notificaciones del estado del aula.

En la Figura 50, se presentan otras funciones relevantes del sistema, como la detección de presencia cuando una persona se aproxima a la puerta del aula. Esta metodología contribuye a mejorar la seguridad de los equipos electrónicos, ya que permite generar alertas oportunas ante situaciones inusuales. Asimismo, se muestran advertencias de seguridad relacionadas con el encendido y apagado del aire acondicionado, lo que permite al usuario estar informado sobre el estado del sistema y actuar de manera preventiva.

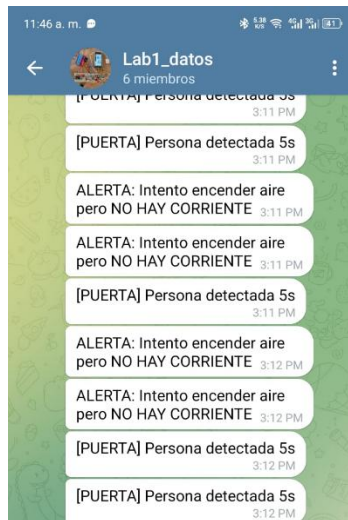


Figura 50. Gestión de seguridad de equipos electrónicos.

CONCLUSIONES

- La comparativa realizada entre los distintos componentes electrónicos permitió seleccionar dispositivos de bajo costo y de amplia compatibilidad, lo que garantiza la viabilidad económica. Esta selección fue determinante para asegurar un sistema automatizado seguro y escalable.
- La implementación de sensores de presencia, contacto magnético y temperatura permitió supervisar en tiempo real el estado del aula, detectando ocupación, apertura de puertas y ventanas, así como condiciones ambientales. Esto fortaleció la idea de un sistema automatizado seguro y eficiente.
- Los actuadores implementados, junto con los módulos relé y el sistema de control por señal infrarroja, permitieron el encendido y apagado seguro de luminarias, puerta de ingreso, proyectores y aires acondicionados. El sistema demostró una respuesta adecuada tanto desde el control local HMI como desde la interfaz web.
- El uso de una arquitectura basada en comunicación Master-Slave y servicios de IoT permitió una gestión centralizada de sensores y actuadores. La comunicación inalámbrica mostró una latencia mínima y una alta confiabilidad, validando el correcto intercambio de datos entre el hardware y la plataforma web.
- El desarrollo de una interfaz gráfica amigable, tanto en la pantalla HMI como en la plataforma web, facilitó la interacción del usuario con el sistema. La visualización en tiempo real de los estados del aula y las notificaciones mediante Telegram demostraron ser herramientas ideales para el monitoreo continuo.
- Las pruebas realizadas evidenciaron un correcto funcionamiento del sistema en términos de accionamiento y latencia de comunicación. Los resultados reflejaron una alta confiabilidad operativa, lo que confirma la viabilidad del sistema para su aplicación en entornos reales.
- Se logró implementar de manera satisfactoria un sistema de accionamiento y monitoreo basado en tecnología IoT, utilizando una tarjeta de desarrollo ESP32 y diversos sensores electrónicos. El sistema demostró ser funcional, confiable y adaptable a entornos educativos, contribuyendo a la modernización de la infraestructura tecnológica institucional.

RECOMENDACIONES

- Se recomienda realizar una evaluación periódica de los componentes electrónicos utilizadas en el sistema de automatización, considerando criterios de compatibilidad, costo, actualización de versión, por consumo energético, y por fiabilidad de marca.
- Se recomienda realizar revisiones o pruebas de calibración de los sensores de presencia, temperatura, contacto magnético, y medición de flujo de corriente, con el propósito de garantizar el correcto censado de los datos obtenidos.
- Se recomienda realizar pruebas periódicas de los actuadores como es la cerradura electrónica, luminarias, proyector y aire acondicionado.
- Se recomienda mantener una configuración organizada de los parámetros del sistema, credenciales de red y protocolos de comunicación utilizado por el ESP32 como emplean del mecanismo de almacenamiento estructurado como archivo JSON en la memoria del dispositivo.
- Se recomienda conservar la interfaz de usuario orientada objeto de manera sea sencilla de manipular tanto en la pantalla HMI como en la interfaz web.
- Se recomienda seguir con las pruebas prolongadas del sistema en condiciones reales demostrando su correcto funcionamiento ante un reinicio, desconexión del Wi-Fi, cortes de energía, y funcionamiento simultáneo. Estas pruebas permitirán identificar posibles problemas de latencia, sincronización, accionamiento, pérdida de información y servirán como base para futuras mejorar el sistema.

BIBLIOGRAFÍA

- [1] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A literature review,” *Journal of Computer and Communications*, vol. 3, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.
- [2] A. Rejeb *et al.*, “Unleashing the power of internet of things and blockchain: A comprehensive analysis and future directions,” *Internet of Things and Cyber-Physical Systems*, vol. 4. 2024. doi: 10.1016/j.iotcps.2023.06.003.
- [3] C. R. Espinoza Carrión, M. J. Pérez Espinoza, and M. B. Peralta Mocha, “El Internet de las cosas: antecedentes, conceptualización y riesgos,” *Memorias del Segundo Congreso Internacional de Ciencia y Tecnología UTMACH*, Universidad Técnica de Machala, Machala, Ecuador, 2016.
- [4] Y. Huang, *Contributions to the Resilience Management in the Internet of Things*, Ph.D. dissertation, Universidad Politécnica de Madrid, Madrid, Spain, 2015.
- [5] J. E. M. Díaz, “Cybersecurity and Internet of Things. Outlook for this decade,” *Computacion y Sistemas*, vol. 26, no. 3, 2022, doi: 10.13053/CYS-26-3-3925.
- [6] F. Martínez Tenorio, C. Daza Bolaños, and E. A. Benachi Arce, “Prototipo basado en IoT y Arduino para validar la disponibilidad de estacionamientos en la sede norte de la UNIAJC,” *Sapientia*, vol. 14, no. 28, 2022, doi: 10.54278/sapientia.v14i28.119.
- [7] A. C. Cruz, J. A. A. Jiménez, and J. G. Domínguez, “Monitoreo de un sistema fotovoltaico con registro de variables para abastecimiento eléctrico a dispositivos móviles haciendo uso de (IoT) / Monitoramento de um sistema fotovoltaico com registro de variáveis para fornecimento de eletricidade a dispositivos móveis usando (IoT),” *Brazilian Journal of Development*, vol. 8, no. 6, 2022, doi: 10.34117/bjdv8n6-061.
- [8] O. Q. Muñoz, *Internet de las cosas (IoT)*, 2019. [Online]. Available: <https://books.google.com.ec/books?id=vnnEDwAAQBAJ>

- [9] D. Buenaño, “¿Tiene el Internet de las cosas algo que decir en el campo educativo?,” Forbes Ecuador, 2022. [Online]. Available: <https://www.forbes.com.ec>
- [10] M. E. Alomoto Tomalá, Diseño e implementación de un sistema de monitoreo domiciliario para la gestión integral de la salud cardiovascular, trabajo de titulación, Universidad Estatal Península de Santa Elena, La Libertad, Ecuador, 2024. [Online]. Available: <https://repositorio.upse.edu.ec/>
- [11] EcuRed, “Dispositivos electrónicos,” s. f. [Online]. Available: https://www.ecured.cu/Dispositivos_electr%C3%B3nicos
- [12] J. F. Torres Madrigal and L. D. Badilla Oviedo, "Análisis de la implementación de hardware educativo basado en Internet de las cosas (IoT) para la mediación de experiencias de educación ambiental: una revisión sistemática," *Innovaciones Educativas*, vol. 27, no. 43, Jul./Dec. 2025. doi: 10.22458/ie.v27i43.5474.b
- [13] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347-2376, 2015.
- [14] Overview of the Internet of Things, ITU-T Recommendation Y.2060, June 2012.
- [15] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 6th ed. Pearson, 2021.
- [16] W. Stallings, *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional, 2017.
- [17] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Pearson, 2017.
- [18] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787-2805, 2010.
- [19] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *J. Electr. Comput. Eng.*, vol. 2017, pp. 1-25, 2017.

- [20] M. A. Zambrano-Asanza et al., "Unmanned Aerial Vehicles Systems over IoT: A Comprehensive Survey," *IEEE Access*, vol. 7, pp. 140220-140239, 2019.
- [21] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125-1142, 2017.
- [22] R. H. Weber, "Internet of Things – New security and privacy challenges," *Comput. Law Secur. Rev.*, vol. 26, no. 1, pp. 23-30, 2010.
- [23] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617-1655, 2016.
- [24] B. R. García Yagual, "Diseño e implementación de un sistema de control de acceso para dispositivos de seguridad basado en tecnología IoT," Tesis de grado, ESPOL, Ecuador, 2023. [Online]. Available: <https://www.dspace.espol.edu.ec/bitstream/123456789/57584>
- [25] ITU-T, "Requirements, support models and semantic framework for Internet of Things (IoT) applications," Recomendaciones ITU-T Y.4103/F.748.0, Y.4552/Y.2078 y Y.4111/Y.2076, Unión Internacional de Telecomunicaciones, 2018–2019. [Online]. Available: <https://www.itu.int/rec/T-REC-Y>
- [26] M. E. Alomoto Tomalá, Prototipo de un sistema domótico configurable a través de comandos de voz y mensajes de texto, Trabajo de titulación, Universidad Estatal Península de Santa Elena, La Libertad, Ecuador, 2019. [Online]. Available: <https://repositorio.upse.edu.ec/>
- [27] R. R. R. Párraga, «Análisis y mejoramiento de la eficiencia energética en una universidad ubicada en la Provincia del Guayas»,» GUAYAQUIL – ECUADOR, 2021.
- [28] TechTarget, "Plataforma de gestión de datos (DMP)," Available: <https://www.computerweekly.com/es/definicion/Plataforma-de-gestion-de-datos-o-DMP>
- [29] A R. Zurawski, *Industrial Communication Technology Handbook*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2015. doi:10.1201/b19518.

- [30] TechTarget, “Plataforma de gestión de datos (DMP),” ComputerWeekly.es, 14-jun-2021. [En línea]. Disponible en: <https://www.computerweekly.com/es/definicion/Plataforma-de-gestion-de-datos-o-DMP>.
- [31] M. Bevan, “What is usability?,” *Interactions*, vol. 17, no. 4, pp. 32–35, 2010. doi:10.1145/1806491.1806493
- [32] B. R. García Yagual y L. A. Mancheno Poveda, Diseño e implementación de un sistema de control de acceso para dispositivos de seguridad basado en tecnología IoT, Proyecto integrador, Facultad de Ingeniería en Electricidad y Computación (FIEC), Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil, Ecuador, 2023. [En línea]. Disponible en: <http://www.dspace.espol.edu.ec/handle/123456789/57584>.
- [33] Universidad Nacional Autónoma de México, Sistemas de comunicaciones, cap. 3, [En línea]. Disponible en: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/861/A6.pdf>
- [34] J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, 5th ed. Springer, 2016. doi: 10.1007/978-3-319-19303-8
- [35] R. H. Diez, «Diseño de un sistema de sensores y gestion inteligente para el Campus Sur de la UPM,» [Online]. Available: https://oa.upm.es/53830/1/TFG_PABLO_VALENZUELA_MOLINA.pdf.
- [36] J. P. Bentley, *Principles of Measurement Systems*, 4th ed. Harlow, U.K.: Pearson, 2005. doi: 10.1007/978-1-4419-2911-2.
- [37] Velázquez Martí, “Reguladores DC-DC fotovoltaicos PWM,” Universitat Politècnica de València, [Online]. Available: <https://riunet.upv.es/bitstream/handle/10251/168554/Vel%C3%A1zquez%20-%20Reguladores%20DC-DC%20Fotovoltaicos.pdf>
- [38] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, 2nd ed. New York, NY, USA: Springer, 2001. doi: 10.1007/978-1-4614-5524-0.

- [39] Pini, “The fundamentals of proximity sensors: selection and use in industrial automation,” DigiKey, Apr. 21, 2021. [Online]. Available: <https://www.digikey.com/es/articles/the-fundamentals-of-proximity-sensors-selection-and-use-industrial-automation>.
- [40] A. Luque and S. Hegedus, Handbook of Photovoltaic Science and Engineering, 2nd ed. Chichester, U.K.: Wiley, 2011. doi:10.1002/9780470974705.
- [41] J. Valvano, Embedded Systems: Introduction to ARM Cortex-M Microcontrollers. [Online]. Available: <https://users.ece.utexas.edu/~valvano/>
- [42] A. P. París, “Relés electromagnéticos y electrónicos,” Revista Vivat Academia, 2003. [Online]. Available: <https://dialnet.unirioja.es/descarga/articulo/8825390.pdf>
- [43] Ó. Cano Grau, *Proyecto de diseño de la instalación eléctrica y domótica de una casa unifamiliar aislada*, Trabajo de fin de grado, Universitat Politècnica de Catalunya (UPC), Barcelona, España, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/355097>
- [44] K. Li, S.-C. Tan y R. S. Y. Hui, “On beat frequency oscillation of two-stage wireless power receivers,” arXiv preprint, 2020. [Online]. Available: <https://arxiv.org/abs/2004.13420>
- [45] K. Li, S.-C. Tan y R. S. Y. Hui, “On beat frequency oscillation of two-stage wireless power receivers,” arXiv preprint, 2020. [Online]. Available: <https://arxiv.org/abs/2004.13420>
- [46] International Electrotechnical Commission, “IEC 60670-1: Boxes and enclosures for electrical accessories for household and similar fixed electrical installations – Part 1: General requirements,” IEC, 2015.
- [47] J. A. Rodríguez Peña and C. D. Vásquez Mosquera, “Implementación de un sistema de control y monitoreo de las variables ambientales que influyen en el confort de espacios arquitectónicos cerrados (aulas de clase),”

- Trabajo de grado, Institución Universitaria Antonio José Camacho, Facultad de Ingenierías, 2023. [Online]. Available: <https://repositorio.uniajc.edu.co/entities/publication/465b7837-e889-4e88-8e2f-abf96fefa705>
- [48] J. A. Padilla Herrera and J. A. León Heredia, “Diseño e implementación de un prototipo escalable con infraestructura IoT para la medición y visualización de la calidad del aire usando tecnología open source para el edificio de la Carrera de Ingeniería en Networking y Telecomunicaciones,” Trabajo de titulación, Univ. de Guayaquil, Ecuador, 2020. [Online]. Available: <http://repositorio.ug.edu.ec/handle/redug/49504>
- [49] I. R. Atiaja Soria, Sistema de telegestión para el control del consumo eléctrico mediante una red IoT en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato. Trabajo de titulación de grado, Univ. Técnica de Ambato, Ecuador, 2023. [Online]. Available: <https://repositorio.uta.edu.ec/handle/123456789/38413>
- [50] A. L. Albert, Electrónica y dispositivos electrónicos. Barcelona, España: Editorial Reverté, 2020. [Online]. Available: https://api.pageplace.de/preview/DT0400.9788429190281_A44162109/preview-9788429190281_A44162109.pdf

ANEXOS

Anexo 1. Código principal

```
#include <WiFi.h>
#include <SPIFFS.h>
#include <HardwareSerial.h>
#include <DHT.h>
#include "EmonLib.h"
EnergyMonitor emon1;
#include "credentials.h"
#include "pages.h"
#include "wifi_manager.h"
#include "device_control.h"
#include "ir_control.h"
#include "sensores_entrada.h"

#define DHTPIN 15
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int STATUS_LED = 25;
#define currCalibration 0.52 // Factor de calibración
const int currentPin = 34; // GPIO34 en ESP32

// Variables para control de temperatura
float ultimaTemperatura = 0;
bool alertaTemperaturaEnviada = false;
const float TEMPERATURA_UMBRAL = 30.0; // 30°C
const unsigned long INTERVALO_ALERTA = 300000; // 5 minutos entre alertas
unsigned long ultimaAlertaTiempo = 0;

void setup() {
  Serial.begin(115200);
  nextionSerial.begin(9600, SERIAL_8N1, 16, 17);
  pinMode(STATUS_LED, OUTPUT);
  digitalWrite(STATUS_LED, HIGH);

  if (!SPIFFS.begin(true)) {
    Serial.println("SPIFFS error - Modo AP");
    startAP();
    return;
  }

  dht.begin();
  setupRelays();
```

```

startWiFi();
setupWebSockets();
irStatusCallback = [](const String& device, bool success, const String& message) {
    String json = "{\"type\":\"ir_status\",\"device\":\"" + device +
        "\",\"success\":\"" + (success ? "true" : "false") +
        "\",\"message\":\"" + message + "\"}";
    websocket.broadcastTXT(json);
};

setupSensores();

// Sensor ventana
if (isVentanaCerrada()) {
    Serial.println("Ventana cerrada - Sistema seguro");
    sendText("tSeguridad", "SISTEMA SEGURO");
} else {
    Serial.println("Ventana abierta - Monitorizando");
    sendText("tSeguridad", "VENTANA ABIERTA");
    ventanaAbierta = true;
    tiempoVentanaAbierta = millis();
}

emon1.current(currentPin, currCalibration);

Serial.println("Control IR listo: [1] AIRE [2] PROYECTOR");
Serial.println(" Sistema de seguridad de ventana activado");
Serial.println(" Alerta temperatura activada (>30°C)");
}

// Temperatura
void verificarTemperatura(float temperatura) {
    if (isnan(temperatura)) {
        return;
    }

    if (temperatura > TEMPERATURA_UMBRAL) {
        if (!alertaTemperaturaEnviada || (millis() - ultimaAlertaTiempo > INTERVALO_ALERTA)) {
            Serial.println("ALERTA: Temperatura alta detectada: " + String(temperatura, 1) + "°C");

            // Enviar mensaje a Telegram
            String mensaje = "ALERTA DE TEMPERATURA: " + String(temperatura, 1) +
                "°C detectada. Recomendado encender el aire acondicionado.";
            enviarMensajeTelegram(mensaje.c_str());

            alertaTemperaturaEnviada = true;
            ultimaAlertaTiempo = millis();
        }
    }
}

```

```

    // Mostrar alerta en Nextion
    sendText("tAlertaTemp", "TEMP ALTA: " + String(temperatura, 1) + "C");
}
} else {
    // Temperatura normal, resetear alerta
    if (alertaTemperaturaEnviada) {
        Serial.println("Temperatura normalizada: " + String(temperatura, 1) + "°C");

        if (millis() - ultimaAlertaTiempo > INTERVALO_ALERTA) {
            String mensaje = "Temperatura normalizada: " + String(temperatura, 1) + "°C";
            enviarMensajeTelegram(mensaje.c_str());
            ultimaAlertaTiempo = millis();
        }

        alertaTemperaturaEnviada = false;
        sendText("tAlertaTemp", ""); // Limpiar alerta
    }
}

ultimaTemperatura = temperatura;
}

void loop() {
    loopWebSockets();
    handleSerialIR();
    handleNextionMessages();
    syncNextionWithRealState();
    loopSensores();

    static unsigned long lastSensorRead = 0;
    static unsigned long lastCheck = 0;
    static unsigned long lastBlink = 0;
    static unsigned long lastCurrentCheck = 0;

    if (millis() - lastCurrentCheck > 5000) {
        lastCurrentCheck = millis();
        tieneCorrienteAire();
    }

    if (millis() - lastSensorRead > 2000) {
        lastSensorRead = millis();
        float temperatura = dht.readTemperature();

        if (!isnan(temperatura)) {
            sendSensorData(temperatura, 0);
        }
    }
}

```

```

String temp = String(temperatura, 1);
sendText("Temp", temp);

    verificarTemperatura(temperatura);
}
}

if (WiFi.status() != WL_CONNECTED) {
    if (millis() - lastBlink > 500) {
        lastBlink = millis();
        digitalWrite(STATUS_LED, !digitalRead(STATUS_LED));
    }
}

if (millis() - lastCheck > 30000) {
    lastCheck = millis();
    if (WiFi.status() != WL_CONNECTED && !WiFi.softAPgetStationNum()) {
        ESP.restart();
    }
}
}
}

```

Anexo 2. Código de recuperación de credenciales WiFi

```

#pragma once
#include <ArduinoJson.h>
#include <SPIFFS.h>

String credFile = "/credentials.json";

bool saveCredentials(String ssid, String password) {
    DynamicJsonDocument doc(512);
    doc["ssid"] = ssid;
    doc["password"] = password;
    File file = SPIFFS.open(credFile, FILE_WRITE);
    if (!file) {
        Serial.println("Error abriendo archivo para guardar");
        return false;
    }
    serializeJson(doc, file);
    file.close();
    Serial.println("Credenciales guardadas exitosamente");
    return true;
}

```

```

bool loadCredentials(String &ssid, String &password) {
    if (!SPIFFS.exists(credFile)) {
        Serial.println("No existe archivo de credenciales");
        return false;
    }
    File file = SPIFFS.open(credFile, FILE_READ);
    if (!file) {
        Serial.println("Error abriendo archivo de credenciales");
        return false;
    }
    DynamicJsonDocument doc(512);
    DeserializationError error = deserializeJson(doc, file);
    file.close();
    if (error) {
        Serial.println("Error leyendo JSON");
        return false;
    }
    ssid = doc["ssid"].as<String>();
    password = doc["password"].as<String>();
    Serial.println("Credenciales cargadas:");
    Serial.println("SSID: " + ssid);
    return true;
}

void deleteCredentials() {
    if (SPIFFS.exists(credFile)) {
        SPIFFS.remove(credFile);
        Serial.println("Credenciales borradas");
    } else {
        Serial.println("No habia credenciales para borrar");
    }
}

```

Anexo 3. Código de control de corriente y HMI

```

#pragma once
#include <Arduino.h>
#include <WebSocketsServer.h>
#include <SoftwareSerial.h>
#include "ir_control.h"
#include "sensores_entrada.h"

void updateNextionImage(const String& objectName, int imageOn, int imageOff, bool state);
void syncNextionWithRealState();
void sendText(String component, String text);
void sendToNextion(String command);

```

```

extern void enviarMensajeTelegram(const char* mensaje);
extern float ultimaTemperatura;
extern bool alertaTemperaturaEnviada;
extern const float TEMPERATURA_UMBRAL;
extern const unsigned long INTERVALO_ALERTA;
double corrienteRMS = 0;
const double UMBRAL_CORRIENTE = 0.016;

WebSocketsServer websocket(81);
typedef void (*IRCallback)(const String& device, bool success, const String& message);
extern IRCallback irStatusCallback;

extern HardwareSerial nextionSerial;

// Configuración de hardware
const uint8_t RELAY_PINS[] = {14, 27, 33};
const char* RELAY_NAMES[] = {"foco1", "foco2", "puerta"};
const uint8_t RELAY_COUNT = 3;

#define SENSOR_MAGNETICO 2

// Estados del sistema de seguridad
bool seguridadActivada = true;
bool ventanaAbierta = false;
unsigned long tiempoVentanaAbierta = 0;
const unsigned long TIEMPO_ESPERA_APAGADO = 15000; // 15 segundos

bool relayStates[RELAY_COUNT] = {false, false, false};
bool aireState = false;
bool proyectorState = false;
bool puertaAbierta = false;

IRCallback irStatusCallback = nullptr;
unsigned long puertaOpenTime = 0;
const unsigned long PUERTA_AUTO_CLOSE_TIME = 15000;

bool isVentanaCerrada() {
    return digitalRead(SENSOR_MAGNETICO) == LOW;
}

void cambiarPaginaNextion(const String& nombrePagina) {
    String comando = "page " + nombrePagina;
    sendToNextion(comando);
}

void activarAdvertenciaDispositivos() {

```

```

    cambiarPaginaNextion("pageMensaje");
    Serial.println("Mostrando pageMensaje por 15 segundos");
}

bool tieneCorrienteAire() {
    double sumaCorriente = 0;
    int lecturas = 3;

    for (int i = 0; i < lecturas; i++) {
        double Irms = emon1.calcIrms(1480);
        sumaCorriente += Irms;
        delay(100);
    }

    double corrientePromedio = sumaCorriente / (lecturas);
    bool hayCorriente = (corrientePromedio > UMBRAL_CORRIENTE);

    Serial.print("Corriente: ");
    Serial.print(corrientePromedio, 3);
    Serial.print(" A - ");
    Serial.println(hayCorriente ? "HAY CORRIENTE" : "SIN CORRIENTE");

    return hayCorriente;
}

void controlAire(bool state) {
    if (!state) {
        Serial.println("Solicitando APAGAR aire...");

        bool success = controlIR_Aire(false);

        int intentos = 0;
        while (tieneCorrienteAire() && intentos < 3) {
            Serial.println("Aún hay corriente, reenviando señal de apagado...");
            delay(1000);
            controlIR_Aire(false);
            intentos++;
        }

        if (!tieneCorrienteAire()) {
            aireState = false;
            Serial.println("Aire APAGADO correctamente - Sin corriente detectada");
        } else {
            Serial.println("Aire podría seguir encendido - Hay corriente persistente");
        }
    }
}

```

```

updateNextionImage("aire", 19, 20, false);
websocket.broadcastTXT("{\"device\":\"aire\\\", \"state\":false}");

if (irStatusCallback) {
    irStatusCallback("aire", true, "Aire apagado");
}
return;
}

if (!tieneCorrienteAire()) {
    Serial.println("NO HAY CORRIENTE - No se puede encender aire");
    sendText("txtEnergia", "Alerta: Sin Energia");
    activarAdvertenciaDispositivos();

    if (irStatusCallback) {
        irStatusCallback("aire", false, "Sin corriente eléctrica");
    }
    enviarMensajeTelegram("ALERTA: Intento encender aire pero NO HAY CORRIENTE");
    return;
}

if (!isVentanaCerrada()) {
    Serial.println("VENTANA ABIERTA - No se puede encender aire");
    if (irStatusCallback) {
        irStatusCallback("aire", false, "Ventana abierta");
    }
    enviarMensajeTelegram("ALERTA: Intento encender aire con VENTANA ABIERTA");
    sendText("tError", "Alerta: Venta Abierta");
    return;
}

Serial.println("Encendiendo aire - Condiciones OK");
updateNextionImage("aire", 19, 20, true);

// Enviar señal IR para encender
bool success = controlIR_Aire(true);

if (tieneCorrienteAire()) {
    aireState = true;
    Serial.println("Aire ENCENDIDO correctamente - Con corriente detectada");
} else {
    aireState = false;
    Serial.println("Aire NO encendió - Sin corriente después de señal");
    success = false;
}
}

```

```

    websocket.broadcastTXT("{\"device\":\"aire\",\"state\":\"" + String(aireState ? "true" : "false") +
    "}");

    if (irStatusCallback) {
        irStatusCallback("aire", success, success ? "Aire encendido" : "Error al encender");
    }
}

void verificarTemperatura(float temperatura);

void apagarAirePorSeguridad() {
    Serial.println(" SEGURIDAD: Apagando aire automáticamente.");

    bool success = controlIR_Aire(false);

    int intentos = 0;
    while (tieneCorrienteAire() && intentos < 3) {
        Serial.println("Reenviando señal de seguridad...");
        delay(1000);
        controlIR_Aire(false);
        intentos++;
    }

    aireState = false;
    updateNextionImage("aire", 19, 20, false);
    websocket.broadcastTXT("{\"device\":\"aire\",\"state\":false}");
    String mensaje = "SEGURIDAD: Aire apagado automáticamente. Ventana abierta más de 15
segundos";
    enviarMensajeTelegram(mensaje.c_str());
    if (irStatusCallback) {
        irStatusCallback("aire", true, "Apagado por seguridad");
    }
}

void testearSeguridadAire() {
    static unsigned long ultimoTest = 0;

    if (millis() - ultimoTest < 1000) return;
    ultimoTest = millis();

    if (!aireState) return;

    bool ventanaCerradaAhora = isVentanaCerrada();

    if (!ventanaCerradaAhora) {

```

```

    if (!ventanaAbierta) {
        ventanaAbierta = true;
        tiempoVentanaAbierta = millis();
    }

    if (millis() - tiempoVentanaAbierta >= 15000) {
        controlAire(false);
    }

} else {
    if (ventanaAbierta) {
        ventanaAbierta = false;
        tiempoVentanaAbierta = 0;
    }
}
}

void updateNextionImage(const String& objectName, int imageOn, int imageOff, bool state) {
    nextionSerial.write(0xFF); nextionSerial.write(0xFF); nextionSerial.write(0xFF);
    nextionSerial.print(objectName + ".pic=");
    nextionSerial.print(state ? imageOn : imageOff);
    nextionSerial.write(0xFF); nextionSerial.write(0xFF); nextionSerial.write(0xFF);
}

void syncNextionWithRealState() {
    updateNextionImage("foco1", 17, 18, relayStates[0]);
    updateNextionImage("foco2", 17, 18, relayStates[1]);
    updateNextionImage("puerta", 30, 31, relayStates[2]);

    updateNextionImage("proyector", 33, 34, proyectorState);
    updateNextionImage("aire", 19, 20, aireState);
}

void setupWebSockets() {
    websocket.begin();
    websocket.onEvent([](uint8_t num, WStype_t type, uint8_t * payload, size_t length) {
        if (type == WStype_CONNECTED) {
            for (int i = 0; i < RELAY_COUNT; i++) {
                String json = "{\"device\":\"" + String(RELAY_NAMES[i]) + "\",\"state\":\"" +
                    String(relayStates[i] ? "true" : "false") + "\"}";
                websocket.sendTXT(num, json);
            }

            websocket.sendTXT(num, "{\"device\":\"aire\",\"state\":\"" + String(aireState ? "true" : "false")
+ "\"}");

```

```

    websocket.sendTXT(num, "{\"device\\\": \"proyector\\\", \"state\\\": \"" + String(proyectorState ?
    "true" : "false") + "\"}");

    syncNextionWithRealState();
}
});
}

void controlRelay(uint8_t index, bool state) {
    if (index >= RELAY_COUNT) return;

    digitalWrite(RELAY_PINS[index], state ? LOW : HIGH);
    relayStates[index] = state;

    switch(index) {
    case 0: // Foco 1
        updateNextionImage("foco1", 17, 18, state);
        break;
    case 1: // Foco 2
        updateNextionImage("foco2", 17, 18, state);
        break;
    case 2: // Puerta
        updateNextionImage("puerta", 30, 31, state);
        break;
    }

    if (index == 2) {
        puertaAbierta = state;
        puertaOpenTime = state ? millis() : 0;
        Serial.println(state ? "Puerta ABIERTA (15s)" : "Puerta CERRADA");
    }

    String json = "{\"device\\\": \"\" + String(RELAY_NAMES[index]) + "\", \"state\\\": \"" +
        String(state ? "true" : "false") + "\"}";
    websocket.broadcastTXT(json);
}

void checkPuertaAutoClose() {
    if (puertaAbierta && puertaOpenTime > 0) {
        unsigned long elapsedTime = millis() - puertaOpenTime;

        if (elapsedTime >= PUERTA_AUTO_CLOSE_TIME) {
            controlRelay(2, false);
        }
    }
}
}

```

```

// Funciones específicas
void controlFoco1(bool state) { controlRelay(0, state); }
void controlFoco2(bool state) { controlRelay(1, state); }
void controlPuerta(bool state) { controlRelay(2, state); }

void controlProyector(bool state) {
    if (state && !isVentanaCerrada()) {
        if (irStatusCallback) {
            irStatusCallback("proyector", false, "Sensor de seguridad abierto");
        }
        return;
    }

    proyectorState = state;
    updateNextionImage("proyector", 33, 34, state);

    if (irStatusCallback) {
        irStatusCallback("proyector", true, state ? "Encendiendo..." : "Apagando...");
    }
    bool success = controlIR_Proyector(state);
    if (irStatusCallback) {
        if (success) {
            irStatusCallback("proyector", true, state ? "Proyector encendido" : "Proyector apagado");
        } else {
            irStatusCallback("proyector", false, "Error de señal");
        }
    }
    websocket.broadcastTXT("{\"device\":\"proyector\",\"state\":\"" + String(state ? "true" : "false")
+ "\"}");
}

void setupRelays() {
    for (int i = 0; i < RELAY_COUNT; i++) {
        pinMode(RELAY_PINS[i], OUTPUT);
        digitalWrite(RELAY_PINS[i], HIGH);
    }
    pinMode(SENSOR_MAGNETICO, INPUT_PULLUP);

    ventanaAbierta = !isVentanaCerrada();
    if (ventanaAbierta) {
        tiempoVentanaAbierta = millis();

        String mensaje = "SISTEMA INICIADO - Ventana encontrada ABIERTA. Aire bloqueado";
        enviarMensajeTelegram(mensaje.c_str());
    } else {

```

```

        Serial.println("Ventana cerrada al iniciar sistema");
    }

}

void sendSensorData(float temperatura, float humedad) {
    String json = "{\"temperatura\": " + String(temperatura) + ", \"humedad\":0}";
    websocket.broadcastTXT(json);
}

String getDeviceStates() {
    String json = "{\"foco1\": " + String(relayStates[0] ? "true" : "false") +
        ", \"foco2\": " + String(relayStates[1] ? "true" : "false") +
        ", \"puerta\": " + String(relayStates[2] ? "true" : "false") +
        ", \"aire\": " + String(aireState ? "true" : "false") +
        ", \"proyector\": " + String(proyectorState ? "true" : "false") + "}";
    return json;
}

void loopWebSockets() {
    websocket.loop();
    checkPuertaAutoClose();
    testearSeguridadAire();
}

void handleNextionMessages(){
    if (nextionSerial.available()) {
        String received = "";

        while (nextionSerial.available()) {
            byte b = nextionSerial.read();

            if (b == '\n') break;
            if (b >= 0x20 && b <= 0x7E)
                received += char(b);
        }

        received.trim();

        if (received.length() > 0) {
            Serial.print("Nextion: ");
            Serial.println(received);

            if (received == "0001") {
                String ipStr = WiFi.localIP().toString();
                String comando = "tIP.txt=\"" + ipStr + "\""; // 'tIP' es el nombre del campo de texto
            }
        }
    }
}

```

```

        nextionSerial.print(comando);
        nextionSerial.write(0xFF);
        nextionSerial.write(0xFF);
        nextionSerial.write(0xFF);
        syncNextionWithRealState();
    }
    else if (received == "0002") {
        controlFoco1(true);
    }
    else if (received == "0003") {
        controlFoco1(false);
    }
    else if (received == "0004") {
        controlFoco2(true);
    }
    else if (received == "0005") {
        controlFoco2(false);
    }
    else if (received == "0006") {
        controlProyector(true);
    }
    else if (received == "0007") {
        controlProyector(false);
    }
    else if (received == "0008") {
        controlAire(true);
    }
    else if (received == "0009") {
        controlAire(false);
    }
    else if (received == "0010") {
        controlPuerta(true);
    }
    else if (received == "0011") {
        controlPuerta(false);
    }
    else {
        Serial.println("Código desconocido");
    }
}
}

```

```

static unsigned long lastSync = 0;
if (millis() - lastSync > 2000) {
    syncNextionWithRealState();
    lastSync = millis();
}

```

```

}
}

```

Anexo 4. Código de codificación de señales IR

```

#pragma once
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRac.h>
#include <IRutils.h>
#include <ArduinoJson.h>
#include <SPIFFS.h>

const uint16_t kIrLed = 4;
IRac ac(kIrLed);

decode_type_t protocoloAC = decode_type_t::UNUSED;
decode_type_t protocoloProyector = decode_type_t::UNUSED;
String input = "";
int modoControl = 0;

const char* archivoAC = "/protocolo_ac.json";
const char* archivoProyector = "/protocolo_proyector.json";

void guardarProtocolo(decode_type_t protocolo, const char* archivo) {
    DynamicJsonDocument doc(64);
    doc["protocolo"] = (int)protocolo;
    File file = SPIFFS.open(archivo, FILE_WRITE);
    if (file) {
        serializeJson(doc, file);
        file.close();
        Serial.println("Protocolo guardado correctamente.");
    } else {
        Serial.println("Error al guardar protocolo.");
    }
}

decode_type_t cargarProtocolo(const char* archivo) {
    if (!SPIFFS.exists(archivo)) {
        Serial.println("Archivo no existe.");
        return decode_type_t::UNUSED;
    }

    File file = SPIFFS.open(archivo, FILE_READ);
    if (!file) {
        Serial.println("Error abriendo archivo.");
    }
}

```

```

    return decode_type_t::UNUSED;
}

DynamicJsonDocument doc(64);
DeserializationError error = deserializeJson(doc, file);
file.close();

if (error) {
    Serial.println("Error al leer JSON.");
    return decode_type_t::UNUSED;
}

return static_cast<decode_type_t>(doc["protocolo"].as<int>());
}

void mostrarProtocolos() {
    Serial.println(F("Selecciona un protocolo ingresando su número:"));
    for (int i = 1; i < kLastDecodeType; i++) {
        decode_type_t tipo = static_cast<decode_type_t>(i);
        if (ac.isProtocolSupported(tipo)) {
            Serial.printf("[%d] %s\n", i, typeToString(tipo).c_str());
        }
    }
    Serial.println("Escribe el número del protocolo y presiona Enter:");
}

void configurarAC(bool encender, decode_type_t protocolo) {
    ac.next.protocol = protocolo;
    ac.next.model = 1;
    ac.next.mode = stdAc::opmode_t::kCool;
    ac.next.celsius = true;
    ac.next.degrees = 25;
    ac.next.fanspeed = stdAc::fanspeed_t::kMedium;
    ac.next.swingv = stdAc::swingv_t::kOff;
    ac.next.swingh = stdAc::swingh_t::kOff;
    ac.next.light = false;
    ac.next.beep = false;
    ac.next.econo = false;
    ac.next.filter = false;
    ac.next.turbo = false;
    ac.next.quiet = false;
    ac.next.sleep = -1;
    ac.next.clean = false;
    ac.next.clock = -1;
    ac.next.power = encender;
}

```

```

// ----- CAMBIO ---

void enviarSenal(decode_type_t protocolo, bool encender) {
    if (!ac.isProtocolSupported(protocolo)) {
        Serial.println("Protocolo no válido o no soportado.");
        return;
    }

    ac.next.protocol = protocolo;
    ac.next.model = 1;
    ac.next.mode = stdAc::opmode_t::kCool;
    ac.next.celsius = true;
    ac.next.degrees = 25;
    ac.next.fanspeed = stdAc::fanspeed_t::kMedium;
    ac.next.swingv = stdAc::swingv_t::kOff;
    ac.next.swingh = stdAc::swingh_t::kOff;
    ac.next.light = false;
    ac.next.beep = false;
    ac.next.econo = false;
    ac.next.filter = false;
    ac.next.turbo = false;
    ac.next.quiet = false;
    ac.next.sleep = -1;
    ac.next.clean = false;
    ac.next.clock = -1;
    ac.next.power = encender;

    Serial.print("ENVIANDO SEÑAL IR - Protocolo: ");
    Serial.print(typeToString(protocolo).c_str());
    Serial.print(" - Acción: ");
    Serial.println(encender ? "ENCENDER" : "APAGAR");

    ac.sendAc();

    delay(100);

    Serial.println("Señal IR enviada completamente");
}

bool controlIR_Aire(bool state) {
    decode_type_t protocolo = cargarProtocolo(archivoAC);
    if (protocolo == decode_type_t::UNUSED) {
        Serial.println("Protocolo AC no válido");
        return false;
    }
}

```

```

    }
    enviarSenal(protocolo, state);
    return true;
}

bool controlIR_Proyector(bool state) {
    decode_type_t protocolo = cargarProtocolo(archivoProyector);
    if (protocolo == decode_type_t::UNUSED) {
        Serial.println("Protocolo proyector no válido");
        return false;
    }
    enviarSenal(protocolo, state);
    return true;
}

void handleSerialIR() {
    while (Serial.available()) {
        char c = Serial.read();
        if (c == '\n' || c == '\r') {
            if (modoControl == 0) {
                if (input == "1") {
                    modoControl = 1;
                    Serial.println("\n--- Selección protocolo AIRE ---");
                    mostrarProtocolos();
                } else if (input == "2") {
                    modoControl = 2;
                    Serial.println("\n--- Selección protocolo PROYECTOR ---");
                    mostrarProtocolos();
                } else {
                    Serial.println("Comando inválido. Usa 1 para AIRE, 2 para PROYECTOR.");
                }
            } else {
                int id = input.toInt();
                decode_type_t candidato = static_cast<decode_type_t>(id);
                if (id > 0 && id < kLastDecodeType && ac.isProtocolSupported(candidato)) {
                    if (modoControl == 1) guardarProtocolo(candidato, archivoAC);
                    if (modoControl == 2) guardarProtocolo(candidato, archivoProyector);
                    Serial.printf("Protocolo %s guardado.\n", typeToString(candidato).c_str());
                } else {
                    Serial.println("Protocolo inválido. Intenta otra vez.");
                    mostrarProtocolos();
                }
            }
            modoControl = 0;
        }
        input = "";
    } else {

```

```

        input += c;
    }
}
}

```

Anexo 6. Código del servidor

```

#pragma once
#include "device_control.h"

String wifiConfigPage() {
    return R"rawliteral(
<!DOCTYPE html><html lang=es>
<head>
<meta charset=UTF-8>
<meta name=viewport content="width=device-width,initial-scale=1.0">
<title>Configurar WiFi</title>
<style>
body{font-family:Arial,sans-serif;background:#f0f0f0;display:flex;flex-direction:column;align-
items:center;justify-content:center;height:100vh}
form{background:white;padding:2rem;border-radius:10px;box-shadow:0      0      10px
rgba(0,0,0,0.1);width:300px}
input,select{width:100%;padding:0.5rem;margin:0.5rem 0;box-sizing:border-box}
button{background:#102f54;color:white;border:none;padding:0.7rem;width:100%;border-
radius:5px;font-size:1rem;margin-top:1rem}
.show-password{margin:0.5rem 0;display:flex;align-items:center}
.show-password input{width:auto;margin-right:0.5rem}
</style>
</head>
<body>
<form action=/save method=POST>
<h2>Conectar a WiFi</h2>
<label for=ssid>Red WiFi:</label>
<select name=ssid id=ssid>%OPTIONS%</select>
<label for=password>Contraseña:</label>
<input type=password name=password id=password>
<div class=show-password>
<input type=checkbox id=showPassword onchange=togglePassword()>
<label for=showPassword>Mostrar contraseña</label>
</div>
<button type=submit>Guardar y Conectar</button>
</form>
<script>
function togglePassword(){
    const p=document.getElementById('password');
    const s=document.getElementById('showPassword');

```

```

    p.type=s.checked?'text':'password';
  }
</script>
</body>
</html>
)rawliteral";
}

```

```

String mainPage() {
    return R"rawliteral(
<!DOCTYPE html><html lang=es>
<head>
<meta charset=UTF-8>
<meta name=viewport content="width=device-width,initial-scale=1.0">
<title>Control IOT</title>
<style>
body {margin:0;font-family:Arial,sans-serif;background-color:#f4f4f4;display:flex;flex-
direction:column;min-height:100vh}
.navbar {background-color:#102f54;padding:1rem;display:flex;justify-content:space-
between;align-items:center;color:white}
.main-container {display:flex;flex:1;flex-direction:row}
.control-panel {flex:3;padding:1rem}
.status-panel {flex:2;padding:1rem;background-color:#e9e9e9;display:flex;flex-
direction:column;gap:1rem}
.btn {background-color:#e0e0e0;color:#333;border:none;padding:0.8rem
1rem;margin:0.3rem;border-radius:6px;cursor:pointer;transition:all
0.3s;font-
size:0.9rem;width:100px}
.btn.on {background-color:#4CAF50;color:white}
.btn.off {background-color:#f44336;color:white}
.btn-circle {width:80px;height:80px;border-radius:50%;border:none;font-size:0.9rem;font-
weight:bold;cursor:pointer;transition:all 0.3s;margin:0.5rem auto;display:block;background-
color:#e0e0e0;color:#333;box-shadow:0 2px 4px rgba(0,0,0,0.2)}
.btn-circle.on {background-color:#4CAF50;color:white;transform:scale(1.05)}
.btn-circle.off {background-color:#f44336;color:white}
.section {background:white;border-radius:8px;padding:1rem;margin-bottom:1rem;box-shadow:0
1px 3px rgba(0,0,0,0.1)}
.access-section {background:white;border-radius:8px;padding:1rem;box-shadow:0 1px 3px
rgba(0,0,0,0.1);text-align:center}
.sensor-section {background:white;border-radius:8px;padding:1rem;box-shadow:0 1px 3px
rgba(0,0,0,0.1);text-align:center}
h2 {color:#102f54;border-bottom:1px solid #eee;padding-bottom:0.5rem;margin-top:0;font-
size:1.1rem}
.access-title {font-size:1rem;font-weight:bold;margin:0.3rem 0;color:#102f54}
.sensor-value {font-size:1.8rem;font-weight:bold;margin:0.3rem 0;color:#102f54}
.dropdown {position:relative;display:inline-block}

```

```

.dropdown-content{display:none;position:absolute;right:0;background-color:#f9f9f9;min-
width:140px;box-shadow:0      4px      8px      rgba(0,0,0,0.2);z-index:1;border-
radius:4px;overflow:hidden}
.dropdown-content.show{display:block!important}
.dropdown-content a{color:#333;padding:10px 12px;text-decoration:none;display:block;text-
align:left;transition:background-color 0.3s;font-size:0.9rem}
.dropdown-content a:hover{background-color:#102f54;color:white}
.dropdown-btn{background-color:#102f54;color:white;border:none;padding:0.6rem
1rem;border-radius:6px;cursor:pointer;font-size:0.9rem}
.modal{display:none;position:fixed;z-
index:10;left:0;top:0;width:100%;height:100%;background-color:rgba(0,0,0,0.4)}
.modal-content{background-color:#fff;margin:20%      auto;padding:15px;width:85%;max-
width:350px;border-radius:8px;text-align:center}
.toast{visibility:hidden;min-width:150px;background-color:#333;color:#fff;text-
align:center;border-radius:4px;padding:8px;position:fixed;z-
index:100;bottom:20px;left:50%;transform:translateX(-50%);transition:visibility      0s,opacity
0.5s;opacity:0;font-size:0.9rem}
.toast.show{visibility:visible;opacity:1}
#notification-container {
  position: fixed;
  top: 20px;
  right: 20px;
  z-index: 1000;
  display: flex;
  flex-direction: column;
  gap: 10px;
}
.ir-notification {
  padding: 15px;
  border-radius: 8px;
  color: white;
  box-shadow: 0 4px 8px rgba(0,0,0,0.2);
  max-width: 300px;
  animation: slideIn 0.3s forwards, fadeOut 0.3s 2.7s forwards;
}
.ir-notification.success {
  background-color: #4CAF50;
}
.ir-notification.error {
  background-color: #f44336;
}
.ir-notification.sending {
  background-color: #2196F3;
}
@keyframes slideIn {
  from {

```

```

    opacity: 0;
    transform: translateX(100px);
  }
  to {
    opacity: 1;
    transform: translateX(0);
  }
}
@keyframes fadeOut {
  from {
    opacity: 1;
    transform: translateX(0);
  }
  to {
    opacity: 0;
    transform: translateX(100px);
  }
}
@media (max-width:768px){
.main-container{flex-direction:column}
.control-panel,.status-panel{flex:none;width:100%;padding:0.5rem}
.btn{width:90px;padding:0.6rem 0.8rem;font-size:0.8rem;margin:0.2rem}
.btn-circle{width:70px;height:70px;font-size:0.8rem}
.section,.access-section,.sensor-section{padding:0.8rem;margin-bottom:0.8rem}
h2{font-size:1rem}
.sensor-value{font-size:1.5rem}
.dropdown-content{min-width:120px}
.dropdown-content a{padding:8px 10px;font-size:0.8rem}
#notification-container {
  top: 10px;
  right: 10px;
  left: 10px;
}
.ir-notification {
  max-width: none;
}
}
@media (max-width:1024px) and (min-width:769px){
.btn{width:110px}
.btn-circle{width:90px;height:90px}
.section,.access-section,.sensor-section{padding:1.2rem}
}
@media (max-width:480px){
.navbar{padding:0.8rem;flex-direction:column;gap:0.5rem}
.btn{width:80px;padding:0.5rem 0.6rem;font-size:0.75rem}
.btn-circle{width:60px;height:60px;font-size:0.7rem}

```

```

.sensor-value{font-size:1.3rem}
.dropdown-btn{padding:0.5rem 0.8rem;font-size:0.8rem}
}
</style>
<script>
let deviceStates={foco1:null,foco2:null,puerta:null,aire:null,proyector:null};
let sensorValues={temperatura:0};
let socket;
let notificationCounter = 0;

function showIRNotification(device, success, message) {
  // Crear contenedor si no existe
  let container = document.getElementById('notification-container');
  if (!container) {
    container = document.createElement('div');
    container.id = 'notification-container';
    document.body.appendChild(container);
  }

  const notificationId = 'notif-' + (notificationCounter++);
  const notification = document.createElement('div');
  notification.id = notificationId;
  notification.className = 'ir-notification ' +
    (success ? 'success' :
      (message === 'Enviando señal...' ? 'sending' : 'error'));

  const deviceName = device === 'aire' ? 'Aire Acondicionado' : 'Proyector';
  notification.textContent = deviceName + ': ' + message;

  // Insertar al principio para que las nuevas aparezcan arriba
  container.insertBefore(notification, container.firstChild);

  // Auto-remove after 3 seconds
  setTimeout(() => {
    const element = document.getElementById(notificationId);
    if (element && element.parentNode) {
      element.parentNode.removeChild(element);
    }
  }, 3000);
}

function toggleDropdown(){
  document.getElementById('dropdownContent').classList.toggle('show');
}

window.onclick=function(e){

```

```

if(!e.target.matches('.dropdown-btn')&&!e.target.closest('.dropdown-content')){
  document.getElementById('dropdownContent').classList.remove('show');
}
}

```

```

function closeDropdown(){
  document.getElementById('dropdownContent').classList.remove('show');
}

```

```

function initWebSocket(){
  const wsUri='ws://'+location.hostname+':81/ws';
  console.log('Conectando:',wsUri);
  socket=new WebSocket(wsUri);

```

```

  socket.onopen=function(){
    console.log('WebSocket conectado');
    fetch('/getStates').then(r=>r.json()).then(data=>{
      deviceStates=data;
      updateButtonStates();
    });
  };

```

```

  socket.onmessage=function(e){
    try{
      const d=JSON.parse(e.data);

      // Manejar notificaciones IR
      if(d.type==='ir_status'){
        showIRNotification(d.device,d.success,d.message);
      }
      // Manejar actualizaciones de estado de dispositivos
      else if(d.device){
        deviceStates[d.device]=d.state;
        updateButtonStates();
      }
      // Manejar datos de sensores
      else if(d.temperatura!==undefined){
        sensorValues.temperatura=d.temperatura;
        updateSensorValues();
      }
    }catch(err){
      console.error('Error parsing message:',err);
    }
  };

```

```

  socket.onerror=function(err){

```

```

    console.error('WebSocket error:',err);
  };

  socket.onclose=function(){
    setTimeout(initWebSocket,2000);
  };
}

function updateButtonStates(){
  document.querySelectorAll('button[data-device]').forEach(btn=>{
    const d=btn.dataset.device;
    const s=deviceStates[d];
    if(s!==null){
      if(d==='puerta'){
        btn.classList.toggle('on',s);
        btn.classList.toggle('off',!s);
        btn.textContent=s?'ABIERTA':'CERRADA';
      }else{
        btn.classList.toggle('on',s);
        btn.classList.toggle('off',!s);
        btn.textContent=s?'ON':'OFF';
      }
    }
  });
}

function updateSensorValues(){
  document.getElementById('tempValue').textContent=sensorValues.temperatura.toFixed(1)+'°C';
}

function controlDevice(device,state=null){
  const s=state!==null?state:!deviceStates[device];
  fetch('/control?device='+device+'&action='+s?'on':'off').then(r=>{
    if(r.ok){
      deviceStates[device]=s;
      updateButtonStates();
    }
  });
}

function confirmReset(){
  closeDropdown();
  document.getElementById('resetModal').style.display='block';
}

function cancelReset(){

```

```

document.getElementById('resetModal').style.display='none';
}

function showToast(m){
const t=document.getElementById('toast');
t.textContent=m;
t.className='toast show';
setTimeout(()=>{t.className='toast'},3000);
}

function doReset(){
fetch('/reset').then(r=>{
if(r.ok)showToast('WiFi reiniciado');
else showToast('Error reiniciando WiFi');
cancelReset();
}).catch(()=>{
showToast('Error de conexión');
cancelReset();
});
}

window.onload=function(){
initWebSocket();
updateSensorValues();
document.querySelectorAll('button[data-device]').forEach(btn=>{
btn.textContent='...';
btn.classList.remove('on','off');
});
};
</script>
</head>
<body>
<div class=navbar>
<div>Control IOT</div>
<div class=dropdown>
<button class=dropdown-btn onclick=toggleDropdown()>≡ Opciones</button>
<div id=dropdownContent class=dropdown-content>
<a href=# onclick=confirmReset()>Reset WiFi</a>
</div>
</div>
</div>

<div class=main-container>
<div class=control-panel>
<div class=section>
<h2>Iluminación</h2>

```

```

<button class=btn data-device=foco1 onclick="controlDevice('foco1')">...</button>
<button class=btn data-device=foco2 onclick="controlDevice('foco2')">...</button>
</div>

<div class=section>
<h2>Aire Acondicionado</h2>
<button class=btn onclick="controlDevice('aire',true)">Encender</button>
<button class=btn onclick="controlDevice('aire',false)">Apagar</button>
</div>

<div class=section>
<h2>Proyector</h2>
<button class=btn onclick="controlDevice('proyector',true)">Encender</button>
<button class=btn onclick="controlDevice('proyector',false)">Apagar</button>
</div>
</div>

<div class=status-panel>
<div class=sensor-section>
<h2>Temperatura Ambiente</h2>
<div class=sensor-value id=tempValue>0°C</div>
</div>

<div class=access-section>
<h2>Control de Acceso</h2>
<div class=access-title>Puerta Principal</div>
<button class=btn-circle data-device=puerta onclick="controlDevice('puerta')">...</button>
</div>
</div>
</div>

<div id=resetModal class=modal>
<div class=modal-content>
<p>¿Estás seguro de resetear el WiFi?</p>
<button class=btn onclick=cancelReset()>Cancelar</button>
<button class=btn onclick=doReset()>Aceptar</button>
</div>
</div>

<div id=toast class=toast></div>
</body>
</html>
)rawliteral";
}

```

Anexo 7. Código de lectura de sensores

```
#pragma once
#include <Arduino.h>
#include <WiFiClientSecure.h>
#include "device_control.h"

const          char*          TELEGRAM_BOT_TOKEN          =
"5676698504:AAEbOC1YEWlj7_flySsn3XmZlu8yk_NsFow";
const char* TELEGRAM_CHAT_ID = "-4953975409";
const char* TELEGRAM_SERVER = "api.telegram.org";
extern const char* TELEGRAM_CHAT_ID;
extern const char* TELEGRAM_SERVER;

#define PIR_PUERTA_PIN 5    // Sensor en la ENTRADA del curso
#define PIR_AULA_PIN 18    // Sensor dentro del AULA

#define TIEMPO_ALERTA_PUERTA 5000 // 15 segundos para puerta
#define TIEMPO_AULA_VACIA 10000   // 30 segundos para aula

// Estados de los sensores
struct SensorEstado {
    bool estadoAnterior;
    unsigned long tiempoDetectado;
    bool alertaEnviada;
    const char* nombre;
};

SensorEstado puertaEstado = {false, 0, false, "PUERTA"};
SensorEstado aulaEstado = {false, 0, false, "AULA"};

WiFiClientSecure client;

void enviarMensajeTelegram(const char* mensaje) {
    if (WiFi.status() != WL_CONNECTED) return;

    static char request[512];
    snprintf(request, sizeof(request),
        "POST /bot%s/sendMessage HTTP/1.1\r\n"
        "Host: api.telegram.org\r\n"
        "Content-Type: application/x-www-form-urlencoded\r\n"
        "Content-Length: %d\r\n"
        "\r\n"
        "chat_id=%s&text=%s",
        TELEGRAM_BOT_TOKEN,
        strlen("chat_id=") + strlen(TELEGRAM_CHAT_ID) +
        strlen("&text=") + strlen(mensaje),
```

```

        TELEGRAM_CHAT_ID, mensaje);

client.setInsecure();

if (client.connect(TELEGRAM_SERVER, 443)) {
    client.print(request);
    delay(100);
    client.stop();
}
}

void setupSensores() {
    pinMode(PIR_PUERTA_PIN, INPUT);
    pinMode(PIR_AULA_PIN, INPUT);

    Serial.println("\n[SENSORES] Inicializados");
    Serial.printf("Puerta:    %ds,    Aula:    %ds\n",    TIEMPO_ALERTA_PUERTA/1000,
    TIEMPO_AULA_VACIA/1000);

    client.setInsecure();
}

void manejarSensor(SensorEstado &sensor, int pin, unsigned long tiempoEspera, bool esPuerta)
{
    bool estadoActual = digitalRead(pin);

    if (esPuerta) {
        if (estadoActual && !sensor.estadoAnterior) {
            Serial.printf("[%s] Iniciando conteo %ds\n", sensor.nombre, tiempoEspera/1000);
            sensor.tiempoDetectado = millis();
            sensor.alertaEnviada = false;
        }

        if (estadoActual && sensor.tiempoDetectado > 0 && !sensor.alertaEnviada) {
            if (millis() - sensor.tiempoDetectado >= tiempoEspera) {
                String mensaje = "[" + String(sensor.nombre) + "] Persona detectada " +
                    String(tiempoEspera/1000) + "s";
                Serial.println(mensaje);
                enviarMensajeTelegram(mensaje.c_str());
                sensor.alertaEnviada = true;
            }
        }
    }

    if (!estadoActual && sensor.estadoAnterior) {
        sensor.tiempoDetectado = 0;
        sensor.alertaEnviada = false;
    }
}

```

```

    }
} else {
    if (estadoActual && !sensor.estadoAnterior) {
        sensor.tiempoDetectado = millis();
        if (sensor.alertaEnviada) {
            String mensaje = "[" + String(sensor.nombre) + "] Aula ocupada";
            enviarMensajeTelegram(mensaje.c_str());
            sensor.alertaEnviada = false;
        }
    }
}

if (sensor.tiempoDetectado > 0) {
    unsigned long tiempoInactividad = millis() - sensor.tiempoDetectado;

    if (tiempoInactividad >= tiempoEspera && !sensor.alertaEnviada) {
        String mensaje = "[" + String(sensor.nombre) + "] Aula vacía " +
            String(tiempoEspera/1000) + "s";
        enviarMensajeTelegram(mensaje.c_str());
        sensor.alertaEnviada = true;
    }
}
}

sensor.estadoAnterior = estadoActual;
}

void debugSensores() {
    static unsigned long lastDebug = 0;

    if (millis() - lastDebug > 2000) {
        lastDebug = millis();

        Serial.printf("SENS      -      P:%d      A:%d",      digitalRead(PIR_PUERTA_PIN),
            digitalRead(PIR_AULA_PIN));

        if (aulaEstado.tiempoDetectado > 0) {
            unsigned long t = millis() - aulaEstado.tiempoDetectado;
            if (t < TIEMPO_AULA_VACIA) {
                Serial.printf(" | Aula:%lus", (TIEMPO_AULA_VACIA - t)/1000);
            }
        }
        Serial.println();
    }
}

void loopSensores() {

```

```

    manejarSensor(puertaEstado, PIR_PUERTA_PIN, TIEMPO_ALERTA_PUERTA, true);
    manejarSensor(aulaEstado, PIR_AULA_PIN, TIEMPO_AULA_VACIA, false);
    debugSensores();
}

```

Anexo 8. Código de configuración a red y punto de acceso

```

#pragma once
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <ArduinoJson.h>
#include "credentials.h"
#include "pages.h"
#include "device_control.h"
#include <HardwareSerial.h>

extern const int STATUS_LED; // Declaración externa del pin
extern HardwareSerial nextionSerial;

AsyncWebSocket ws("/ws");
AsyncWebServer server(80);
String networks = ""; // Almacena redes escaneadas

// Configuración de comunicación con Nextion
HardwareSerial nextionSerial(2); // UART2 en ESP32 (RX=16, TX=17)

void sendToNextion(String command) {
    nextionSerial.print(command);
    nextionSerial.write(0xFF);
    nextionSerial.write(0xFF);
    nextionSerial.write(0xFF);
    delay(20);
}

// Función para enviar texto a componentes específicos
void sendText(String component, String text) {
    sendToNextion(component + ".txt=\"" + text + "\"");
}

// Función para actualizar la pantalla con el estado de conexión
void updateConnectionStatus(bool connected, String ssid = "", String ip = "") {
    if (connected) {
        sendToNextion("page pageMenu");
        sendText("pageMenu.tIP", ip);
        sendText("pageMenu.tSSID", ssid);
    }
}

```

```

    } else {
        sendToNextion("page pageWiFi");
        sendText("t0", "Modo Configuración");
        sendText("t1", "ESP32_Setup");
        sendText("t2", WiFi.softAPIP().toString());
        sendText("t3", "Conectese a esta red");
    }
}

// Escanea redes una vez al iniciar (no bloqueante)
void scanNetworks() {
    Serial.println("Escaneando redes WiFi...");
    sendText("t0", "Escaneando redes...");
    networks = "";
    int n = WiFi.scanNetworks(false, true); // Escaneo asíncrono
    for (int i = 0; i < n; i++) {
        networks += "<option value=\"" + WiFi.SSID(i) + "\">" + WiFi.SSID(i) + " (" + WiFi.RSSI(i) +
        "dBm)</option>";
    }
}

// Configura el punto de acceso
void startAP() {
    WiFi.disconnect(true);
    delay(100);

    digitalWrite(STATUS_LED, HIGH);

    Serial.println("Iniciando Punto de Acceso...");
    WiFi.softAP("ESP32_Setup", "12345678");
    Serial.print("AP IP: ");
    Serial.println(WiFi.softAPIP());

    updateConnectionStatus(false);

    scanNetworks();

    // Manejador para página de configuración
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
        String page = wifiConfigPage();
        page.replace("%OPTIONS%", networks);
        request->send(200, "text/html", page);
    });

    // Manejador para guardar credenciales
    server.on("/save", HTTP_POST, [](AsyncWebServerRequest *request) {

```

```

if (request->hasParam("ssid", true) && request->hasParam("password", true)) {
    String ssid = request->getParam("ssid", true)->value();
    String password = request->getParam("password", true)->value();

    if (saveCredentials(ssid, password)) {
        request->send(200, "text/html", "<h1>Credenciales guardadas. Reiniciando...</h1>");
        sendText("t0", "Reiniciando...");
        delay(100);
        ESP.restart();
    } else {
        request->send(500, "text/plain", "Error al guardar");
        sendText("t3", "Error al guardar");
    }
} else {
    request->send(400, "text/plain", "Datos incompletos");
    sendText("t3", "Datos incompletos");
}
});

server.begin();
}

// Configura el servidor en modo estación
void startServer() {
    // Página principal
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
        request->send(200, "text/html", mainPage());
    });

    // Reset WiFi
    server.on("/reset", HTTP_GET, [](AsyncWebServerRequest *request) {
        deleteCredentials();
        request->send(200, "text/html", "<h1>Credenciales borradas. Reiniciando...</h1>");
        sendText("t0", "Reiniciando...");
        delay(100);
        ESP.restart();
    });

    // Manejador para control de dispositivos
    server.on("/control", HTTP_GET, [](AsyncWebServerRequest *request) {
        if (request->hasParam("device") && request->hasParam("action")) {
            String device = request->getParam("device")->value();
            String action = request->getParam("action")->value();
            bool state = (action == "on");

            if (device == "foco1") controlFoco1(state);
        }
    });
}

```

```

else if (device == "foco2") controlFoco2(state);
else if (device == "puerta") controlPuerta(state);
else if (device == "aire") controlAire(state);
else if (device == "proyector") controlProyector(state);

// Notificar a todos los clientes WebSocket sobre el cambio de estado
String json = "{\"device\":\"" + device + "\",\"state\":\"" + (state ? "true" : "false") + "\"}";
ws.textAll(json);

sendToNextion(device + ".val=" + (state ? "1" : "0"));

request->send(200, "application/json", "{\"status\":\"ok\"}");
} else {
    request->send(400, "text/plain", "Parámetros incorrectos");
}
});

// Manejador para obtener estados
server.on("/getStates", HTTP_GET, [](AsyncWebServerRequest *request) {
    String json = "{}";
    json += "\"foco1\":\"" + String(relayStates[0] ? "true" : "false") + "\", ";
    json += "\"foco2\":\"" + String(relayStates[1] ? "true" : "false") + "\", ";
    json += "\"puerta\":\"" + String(relayStates[2] ? "true" : "false") + "\", ";
    json += "\"aire\":\"" + String(aireState ? "true" : "false") + "\", ";
    json += "\"proyector\":\"" + String(proyectorState ? "true" : "false") + "\"}";
    request->send(200, "application/json", json);
});

server.begin();
irStatusCallback = [](const String& device, bool success, const String& message) {
    String json = "{\"type\":\"ir_status\",\"device\":\"" + device +
        "\",\"success\":\"" + (success ? "true" : "false") +
        "\",\"message\":\"" + message + "\"}";
    ws.textAll(json);
};

}

// Gestiona la conexión WiFi
void startWiFi() {
    nextionSerial.begin(9600, SERIAL_8N1, 16, 17);
    delay(1000);
    sendToNextion("page pageWiFi");
    sendText("t0", "Conectando...");

    String ssid, password;

```

```

if (loadCredentials(ssid, password)) {
  Serial.println("Conectando a WiFi guardado: " + ssid);
  sendText("t1", ssid);
  sendText("t2", "Conectando...");
  sendText("t3", "");

  WiFi.begin(ssid.c_str(), password.c_str());

  unsigned long startTime = millis();
  int progress = 0;

  while (WiFi.status() != WL_CONNECTED && millis() - startTime < 10000) {
    digitalWrite(STATUS_LED, !digitalRead(STATUS_LED)); // Parpadeo rápido

    progress = min(100, (int)((millis() - startTime) / 100));
    sendToNextion("j0.val=" + String(progress));

    String dots = "";
    for (int i = 0; i < (millis() / 500 % 3); i++) dots += ".";
    sendText("t0", "Conectando" + dots);

    delay(250);
    Serial.print(".");
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nConectado! IP: " + WiFi.localIP().toString());
    digitalWrite(STATUS_LED, LOW);

    sendToNextion("j0.val=100");
    sendText("t0", "Conectado!");
    sendText("t1", WiFi.SSID());
    sendText("t2", WiFi.localIP().toString());
    sendText("t3", "");

    delay(2000);

    startServer();
    setupWebSockets();

    updateConnectionStatus(true, WiFi.SSID(), WiFi.localIP().toString());
  } else {
    digitalWrite(STATUS_LED, HIGH);
    Serial.println("\nFallo la conexión. Iniciando AP...");
    sendText("t0", "Error conexion");
  }
}

```

```
    sendText("t3", "Iniciando AP...");  
    delay(1000);  
    startAP();  
  }  
  } else {  
    Serial.println("No hay credenciales guardadas.");  
    sendText("t0", "Modo Configuración");  
    sendText("t3", "No hay credenciales");  
    delay(1000);  
    startAP();  
  }  
}
```