



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA DOMÓTICO BASADO EN IOT PARA EL CENTRO INTEGRAL
DEL ADULTO MAYOR DE LA CIUDAD DE AMBATO**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

ÁREA: Electrónica y Comunicaciones

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y sistemas de control.

AUTOR: Jonathan Eduardo Moreta Tigse

TUTOR: Ing. Santiago Altamirano, Mg.

Ambato – Ecuador

septiembre – 2022

APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Titulación, con el tema: SISTEMA DOMÓTICO BASADO EN IOT PARA EL CENTRO INTEGRAL DEL ADULTO MAYOR DE LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Jonathan Eduardo Moreta Tigse, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo

Ambato, septiembre 2022.

Ing. Santiago Altamirano, Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA DOMÓTICO BASADO EN IOT PARA EL CENTRO INTEGRAL DEL ADULTO MAYOR DE LA CIUDAD DE AMBATO es absolutamente original, autentico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.



Jonathan Eduardo Moreta Tigse

C.C. 1804580700

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Jonathan Eduardo Moreta Tigse, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DOMÓTICO BASADO EN IOT PARA EL CENTRO INTEGRAL DEL ADULTO MAYOR DE LA CIUDAD DE AMBATO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, septiembre 2022.

Ing. Carlos Sánchez Rosero, Mg.

PRESIDENTE SUBROGANTE DEL TRIBUNAL

Ing. Santiago Manzano, Mg
PROFESOR CALIFICADOR


Ing. Pamela Castro, Mg
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022.



Jonathan Eduardo Moreta Tigse

C.C. 1804580700

AUTOR

DEDICATORIA

El presente trabajo de titulación se lo dedico a mi madre Mirian Tigse, por ser una mujer excepcional, siempre levantándose en cada fracaso e incentivándose tras cada triunfo en la vida. Motivándose cada día a alcanzar nuevas metas y ser un mejor hombre. Cada uno de sus consejos valen oro y nos han traído hasta este momento.

Jonathan Eduardo Moreta Tigse

AGRADECIMIENTO

Agradezco a Dios por otorgarme la sabiduría y haberme guiado en el camino para alcanzar esta meta profesional.

A mi madre Mirian, a mis hermanos Edison, Carolina y Michael, quienes en situaciones difíciles me brindaron su apoyo y quienes de una u otra forma con sus risas, juegos o bromas entre horas de estudio, me motivaron a alcanzar este logro, que también es de ustedes. Los amo demasiado.

Al ingeniero Santiago Altamirano por ser un gran apoyo y guía para alcanzar este objetivo, a mis docentes a lo largo de mi formación y a mis amigos de aula quienes también estimo durante esta etapa en la FISEI.

Jonathan Eduardo Moreta Tigse

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS.....	xii
RESUMEN EJECUTIVO	xvii
ABSTRACT	xviii
CAPITULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de Investigación.....	1
1.2 Antecedentes Investigativos.....	1
1.2.1 Contextualización del Problema	3
1.3 Fundamentación Teórica	4
1.3.1 Centro Integral del Adulto Mayor.....	4
1.3.2 Adulto Mayor	5
1.3.3 Tecnologías domóticas.....	8
1.3.4 Arquitecturas de control domótico.....	10
1.3.5 Internet de las Cosas (IoT).....	11
1.3.6 Aplicaciones del IoT	13
1.3.7 Tecnologías de comunicación inalámbrica	13
1.3.8 Estándar IEEE 802.11	14
1.3.9 Protocolo de comunicación MQTT.....	15
1.3.10 Servidor LAMP	18
1.3.11 Node Red.....	18
1.3.12 Servicios de mensajería móvil	19
1.3.13 Reconocimiento de Voz	20
1.3.14 Ordenadores de placa reducida	22
1.3.15 Microcontrolador ESP32.....	22
1.3.16 Sistemas de Control	27
1.4 Objetivos	28
1.4.1 Objetivo General	28
1.4.2 Objetivos específicos	28
CAPÍTULO II	29
METODOLOGÍA	29
2.1 Materiales	29
2.2 Métodos	29

2.2.1	Modalidad de investigación	29
2.2.2	Recolección de información.....	30
2.2.3	Procesamiento y análisis de datos	30
2.2.4	Desarrollo del proyecto	30
CAPITULO III.....		32
RESULTADOS Y DISCUSIÓN		32
3.1	Análisis y discusión de los resultados.....	32
3.2	Desarrollo de la propuesta.....	32
3.2.1	Beneficios de la Domótica	32
3.2.2	Requerimientos para la asistencia de adultos mayores	33
3.2.3	Requerimientos del sistema domótico basado en IoT.....	34
3.2.4	Diagrama de bloques del sistema.....	37
3.2.5	Identificación de variables a ser monitorizadas por el sistema.....	34
3.2.6	Identificación de variables a ser controladas por el sistema.	35
3.2.7	Sensores para el sistema domótico.....	35
3.2.8	Actuadores para el sistema domótico.....	36
3.2.9	Servidor en la nube.....	38
3.2.9.1	Implementación del servidor en la nube	39
3.2.9.2	Instalación de LAMP en el servidor	43
3.2.9.3	Broker Mosquitto en el servidor	44
3.2.9.4	Instalación de Node Red en el servidor	45
3.2.10	Elaboración de los módulos electrónicos.....	48
3.2.10.1	Módulo de temperatura y humedad	48
3.2.10.2	Módulo detector de humo	51
3.2.10.3	Módulo detector de movimiento.....	53
3.2.10.4	Módulo de Asistencia	56
3.2.10.5	Módulo de control de iluminación.....	58
3.2.10.6	Módulos de control del calefactor y ventilador	62
3.2.11	Programación de los módulos	65
3.2.12	Creación de la Base de datos.....	72
3.2.13	Configuración de Node Red.....	73
3.2.13.1	Conexión con la base de datos del servidor en la nube	73
3.2.13.2	Conexión con el broker MQTT del servidor en la nube.....	74
3.2.13.3	Conexión con bot de Telegram.....	75
3.2.13.4	Configuración de la interfaz del tablero de monitoreo	78
3.2.13.5	Configuración de la interfaz de tablero de control	82
3.2.13.6	Generación de Notificaciones Automáticas.....	84
3.2.14	Reporte de alertas.....	85
3.2.15	Reconocimiento de Voz	86
3.2.15.1	Instalación del sistema operativo en la Raspberry Pi.	87
3.2.15.2	Speech Recognition	88
3.2.15.3	PAHO-MQTT	89

3.2.15.4	Tkinter.....	90
3.2.15.5	Interfaz para el reconocimiento de voz y conexión con MQTT ...	90
3.2.16	Implementación del sistema.....	93
3.2.16.1	Infraestructura del área habitacional.....	93
3.2.16.2	Recursos tecnológicos de red disponibles en el centro.....	93
3.2.16.3	Despliegue de la Red Wifi.....	94
3.2.16.4	Instalación de los módulos de monitoreo	96
3.2.16.5	Instalación de los módulos de control.....	98
3.2.16.6	Estación de monitoreo y control por voz.....	101
3.2.16.7	Pruebas de Funcionamiento	103
3.2.16.8	Presupuesto del proyecto	114
CAPÍTULO IV	117
CONCLUSIONES Y RECOMENDACIONES	117
4.1	Conclusiones	117
4.2	Recomendaciones.....	118
REFERENCIAS BIBLIOGRÁFICAS	119
ANEXOS	124

ÍNDICE DE TABLAS

Tabla 1.	Diferentes Tecnologías Domóticas	9
Tabla 2.	Comparación de tecnologías inalámbricas	13
Tabla 3.	Versiones del estándar IEEE 802.11	14
Tabla 4.	Brokers MQTT.....	16
Tabla 5.	Características de los servicios de mensajería móvil	20
Tabla 6.	Ordenadores de placas reducidas	22
Tabla 7.	Comparación entre sensores de Temperatura y Humedad.....	24
Tabla 8.	Comparación entre sensores de Co2	25
Tabla 9.	Comparación entre sensores movimiento	26
Tabla 10.	Empresas proveedoras de servicios en la nube	39
Tabla 11.	Condiciones para el envío de Notificaciones.....	84
Tabla 12.	Órdenes por voz	92
Tabla 13.	Configuración de dirección IP	94
Tabla 14.	Evaluación del reconocimiento de voz en Raspberry pi.....	113
Tabla 15.	Porcentaje de fiabilidad del control por voz	113
Tabla 16.	Presupuesto de materiales del proyecto	114
Tabla 17.	Presupuesto Diseño y Ejecución del proyecto	115
Tabla 18.	Presupuesto total del Proyecto	115
Tabla 19.	Presupuesto referencial para tres áreas habitacionales del CIAM.....	116

ÍNDICE DE FIGURAS

Figura 1.	Domótica con diferentes funciones de control.....	7
Figura 2.	Arquitecturas de control domótico.....	11
Figura 3.	Arquitectura IoT.....	12
Figura 4.	Funcionamiento de MQTT.....	17
Figura 5.	Programación a través nodos y flujos en Node Red	19
Figura 6.	Sistema de reconocimiento de voz.....	21
Figura 7.	Microcontrolador ESP32 devkitc v4	23
Figura 8.	Diagrama de bloques del Sistema de lazo abierto.....	27
Figura 9.	Diagrama de bloques del Sistema de lazo cerrado.....	28
Figura 10.	Pulsador de asistencia	36
Figura 11.	Actuador tipo relé de 5V	37
Figura 12.	Diagrama de bloques del sistema domótico basado en IoT	38
Figura 13.	Detalles de la instancia creada en la capa gratuita de AWS.	40
Figura 14.	Asignación de IP elástica al servidor en la nube.....	40
Figura 15.	Dirección IP pública y elástica del servidor en la nube	41
Figura 16.	Reglas de tráfico de datos del servidor en la nube	41
Figura 17.	Generación del par de claves para el servidor en la nube	42
Figura 18.	Conexión con el servidor en la nube a través de SSH.....	42
Figura 19.	Autenticación al broker Mosquitto.....	45
Figura 20.	Node Red instalado en el servidor en la nube.....	46
Figura 21.	Generación de contraseña cifrada para node red.....	47
Figura 22.	Credenciales de acceso al panel principal de node red.	47
Figura 23.	Credenciales de acceso a páginas web de node red.	47
Figura 24.	Circuito esquemático del módulo de temperatura y humedad.	49
Figura 25.	Distribución de elementos y diagrama PCB del módulo de temperatura y humedad	49
Figura 26.	Montaje de componentes del módulo de temperatura y humedad.....	50
Figura 27.	Caja contenedora del módulo de temperatura y humedad	51
Figura 28.	Circuito esquemático del módulo de detección de humo (CO2).	51
Figura 29.	Distribución de elementos y diagrama PCB del módulo de detección de humo (CO2).	52

Figura 30.	Montaje de componentes del módulo de detección de humo.	52
Figura 31.	Caja contenedora del módulo de detección de humo.....	53
Figura 32.	Circuito esquemático del módulo de detección de movimiento.	54
Figura 33.	Distribución de elementos y diagrama PCB del módulo detector de movimiento.	54
Figura 34.	Montaje de componentes del módulo detector de movimiento.	55
Figura 35.	Caja contenedora del módulo de detección de humo.....	56
Figura 36.	Circuito esquemático del módulo de asistencia	56
Figura 37.	Distribución de elementos y diagrama PCB del módulo de asistencia	57
Figura 38.	Montaje de componentes del módulo detector de movimiento.	57
Figura 39.	Caja contenedora del módulo de asistencia	58
Figura 40.	Circuito esquemático para control de iluminación.....	59
Figura 41.	Distribución de componentes del módulo de control de iluminación..	60
Figura 42.	Diagrama PCB del módulo de control de iluminación	61
Figura 43.	Montaje de componentes del módulo de control de iluminación.	61
Figura 44.	Caja contenedora del módulo de control de iluminación.....	62
Figura 45.	Circuito esquemático de Módulo de control calefactor y ventilador ...	63
Figura 46.	Distribución de elementos y diagrama PCB	64
Figura 47.	Montaje de componentes del módulo detector de movimiento.	64
Figura 48.	Caja contenedora del módulo de control de calefactor y ventilador....	65
Figura 49.	Diagrama de flujo del módulo de temperatura y humedad.....	66
Figura 50.	Diagrama de flujo del módulo detector de humo.....	67
Figura 51.	Diagrama de flujo del módulo detector de movimiento.	68
Figura 52.	Diagrama de flujo del módulo de asistencia	69
Figura 53.	Diagrama de flujo del módulo de control de iluminación.	70
Figura 54.	Diagrama de flujo del módulo de control del calefactor.....	71
Figura 55.	Diagrama de flujo del módulo de control del ventilador.	72
Figura 56.	Base de datos para el reporte de alertas	73
Figura 57.	Funcionamiento de la base de datos con los campos de la tabla.....	73
Figura 58.	Configuración del nodo MySql.....	74
Figura 59.	Nodos “mqtt out” y “mqtt in”, para publicación y suscripción.	74
Figura 60.	Campos de configuración de conexión con el broker MQTT.....	75
Figura 61.	Campos de configuración de seguridad del broker MQTT.....	75

Figura 62.	Chatbot automatizado BotFather en Telegram	76
Figura 63.	Token generado del chatbot para el usuario “Sistema-CIAM”	77
Figura 64.	Configuración del nodo telegram.....	77
Figura 65.	Nodo “receiver” y ChatID del grupo de telegram.....	77
Figura 66.	Programación del nodo “function” para el envío de mensaje al grupo de Telegram	78
Figura 67.	Mensaje desde el chatbot al grupo de telegram	78
Figura 68.	Nodos para el monitoreo de temperatura	79
Figura 69.	Nodos para el monitoreo de humedad.....	79
Figura 70.	Nodos para el monitoreo de humo (co2).....	80
Figura 71.	Nodos para el monitoreo de movimiento	80
Figura 72.	Nodos para asistencia.....	81
Figura 73.	Interfaz con elementos para el monitoreo de variables.....	81
Figura 74.	Nodos de control manual del módulo de control del calefactor.....	82
Figura 75.	Nodos del control automático del módulo del calefactor.....	82
Figura 76.	Nodos de control manual del módulo de control del ventilador	82
Figura 77.	Nodos del control automático del módulo del ventilador	83
Figura 78.	Nodos de control para iluminación	83
Figura 79.	Interfaz con elementos del control y monitoreo de dispositivos.....	84
Figura 80.	Nodos de configuración para las notificaciones vía Telegram	85
Figura 81.	Control de activación/desactivación de notificaciones automáticas. ...	85
Figura 82.	Interfaz del reporte de alertas del Sistema domótico	86
Figura 83.	Diagrama de bloques para el control por voz.....	86
Figura 84.	Micrófono AE-MICU04 para el reconocimiento de voz [64].....	87
Figura 85.	Instalación del sistema operativo RaspiOS con Raspberry Pi Imager.	87
Figura 86.	Instalación de la librería SpeechRecognition en la Raspberry	88
Figura 87.	Instalación de la librería PAHO-MQTT en python.....	89
Figura 88.	Archivo para suscripción al broker MQTT del servidor en la nube	89
Figura 89.	Recibo de datos del topic “ciam” desde el servidor en la nube	90
Figura 90.	Interfaz gráfica para el reconocimiento de voz	91
Figura 91.	Inicialización del reconocimiento de voz.....	91
Figura 92.	Área habitacional designada para el proyecto.....	93
Figura 93.	Rack de comunicaciones	94

Figura 94.	Router TP-Link Archer C24 [66].....	95
Figura 95.	Configuración de IP estática en el router Wifi.....	95
Figura 96.	Instalación del Router Wifi en el área habitacional	96
Figura 97.	Distribución de los dispositivos	96
Figura 98.	Proceso de instalación de los módulos de temperatura, detección de movimiento y asistencia.....	97
Figura 99.	Módulos de monitoreo instalados	97
Figura 100.	Módulos de Asistencia y detección de humo instalados.....	98
Figura 101.	Conexión del módulo de control del calefactor	99
Figura 102.	Conexión del módulo de control del ventilador.....	99
Figura 103.	Mediciones de corriente en los interruptores para luces previa instalación del módulo de control de iluminación.	100
Figura 104.	Proceso de instalación del módulo de control de iluminación.....	101
Figura 105.	Módulo instalado para el control de luminarias.....	101
Figura 106.	Oficinas departamentales del Centro donde se ubica la Raspberry Pi	102
Figura 107.	Raspberry pi y micrófono para el reconocimiento de voz	102
Figura 108.	Topics de los dispositivos IoT del sistema.....	103
Figura 109.	Interfaz principal del sistema domótico	104
Figura 110.	Interfaz de activación de notificaciones automáticas vía Telegram...	104
Figura 111.	Notificación vía telegram, Temperatura superior a 28°C	105
Figura 112.	Notificación vía telegram, Temperatura superior a 160°C	105
Figura 113.	Mensajes de los nodos de Temperatura a la base de datos	106
Figura 114.	Medición de humedad relativa	106
Figura 115.	Notificación en interfaz (showdialog) indicador de Alerta de humo.	107
Figura 116.	Notificación vía Telegram y Actualización del reporte de alertas.....	107
Figura 117.	Funcionamiento del módulo detector de movimiento.....	107
Figura 118.	Notificación de asistencia en la interfaz web.....	108
Figura 119.	Notificación automática vía Telegram de asistencia y envío de alerta a la base de datos.....	108
Figura 120.	Funcionamiento de Calefacción automática	109
Figura 121.	Cambio de estado (apagado/encendido) “Calefacción automática” .	109
Figura 122.	Envío del mensaje de activación de calefacción automática la base de datos	109

Figura 123.	Funcionamiento de Calefacción automática	110
Figura 124.	Envío del mensaje de activación de la ventilación automática a la base de datos.....	111
Figura 125.	Estados del módulo de control de iluminación en la interfaz web.....	111
Figura 126.	Botón de encendido/apagado general.....	112
Figura 127.	Ejecución de órdenes a través del reconocimiento de voz.....	112

RESUMEN EJECUTIVO

El cuidado de adultos mayores año tras año representa grandes inversiones para su asistencia debido al proceso degenerativo que les impide sobrellevar una vida plena o realizar tareas cotidianas sencillas. La necesidad de contar con personas que estén a cargo de su cuidado también implica aplicar tecnologías que ayuden a su asistencia frente a eventos de riesgo. El presente proyecto desarrolla e implementa un sistema domótico basado en IoT para el Centro Integral del Adulto Mayor (CIAM) de la ciudad de Ambato, que bajo directrices del personal junto con las competencias del autor, logran determinar requerimientos específicos para el centro. El diseño y construcción de dispositivos IoT para el monitoreo de variables relacionadas con su entorno y el control de las instalaciones, se complementa con un servidor en la nube cuyos recursos están direccionados a brindar una comunicación a través del protocolo MQTT. El control automático también se integra en su configuración para brindar condiciones ambientales adecuadas a los adultos mayores, frente a valores de temperaturas no indicadas para su diario vivir. A través de una interfaz de usuario desarrollada en Node Red, el personal cuenta con la tecnología que apoya la asistencia de los adultos mayores, así como las notificaciones que el sistema genera al reconocer cambios en las variables medidas. El reconocimiento de voz se agrega al sistema para el control de los dispositivos mediante la utilización de Raspberry Pi. Conformando un sistema capaz de proveer la comunicación entre dispositivos y generar notificaciones automáticas mediante servicios de mensajería móvil.

Palabras clave: Domótica, internet de las cosas, MQTT, adultos mayores, nodos, dispositivos IoT, Raspberry Pi, reconocimiento de voz

ABSTRACT

Caring for older adults year after year represents large investments for their assistance due to the degenerative process that prevents them from leading a full life or performing simple daily tasks. The need to have people in charge of their care also implies applying technologies that help them assist in risk events. This project develops and implements a home automation system based on IoT for the Comprehensive Center for the Elderly (CIAM) of the city of Ambato, which under the guidelines of the staff together with the author's skills, manage to determine specific requirements for the center. The design and construction of IoT devices for the monitoring of variables related to their environment and the control of the facilities, is complemented by a server in the cloud whose resources are directed to provide communication through the MQTT protocol. Automatic control is also integrated into its configuration to provide adequate environmental conditions for older adults, in the face of temperature values not indicated for their daily lives. Through a user interface developed in Node Red, the staff has the technology that supports the assistance of older adults, as well as the notifications that the system generates when recognizing changes in the measured variables. Speech recognition is added to the system for device control using Raspberry Pi. Forming a system capable of providing communication between devices and generating automatic notifications through mobile messaging services.

Keywords: Home automation, internet of things, MQTT, older adults, nodes, IoT devices, Raspberry Pi, voice Recognition.

CAPITULO I

MARCO TEÓRICO

1.1 Tema de Investigación

“Sistema domótico basado en IoT para el Centro Integral del Adulto Mayor de la ciudad de Ambato”.

1.2 Antecedentes Investigativos

En base a una investigación realizada en diferentes repositorios a nivel nacional e internacional y artículos científicos publicados en diferentes revistas, se han encontrado diversos proyectos de investigación relacionados a la domótica y tecnología IoT, los de mayor relevancia se detallan a continuación:

El proyecto de Investigación “Diseño e implementación de un prototipo de seguridad para control domótico basado en IoT bajo ambientes de dispositivos móviles con Android” realizada en el año 2020 por Darío Wladimir Herrera Chávez, plantea la utilización de tecnologías relacionadas a la domótica y selecciona la plataforma Fire Base integrada a la plataforma Google Cloud. Esto le permite alojar datos en la nube y establecer la comunicación de forma bidireccional entre los diferentes elementos del sistema. En la arquitectura propuesta hace uso de bases de datos NoSQL en Fire Store mediante la integración de SDK a Python, permitiendo que la información recogida de sensores y actuadores se almacene datos en la nube y los presente en la interfaz de la aplicación Android con retardos de respuesta y ejecución relativamente bajos con un 92.16% de acciones completadas exitosamente. [1]

En el 2019, Miguel Vázquez Wilksen lleva a cabo una investigación denominada “Diseño y desarrollo de un sistema domótico de bajo costo” (Universidad Carlos III de Madrid), en la cual desarrolla una serie de módulos con NodeMCU (ESP8266) integrados a placas con diversos dispositivos electrónicos como actuadores o sensores. Estos componentes se comunican a un servidor HTTP, y se enlazan a la app

desarrollada en Android para el control de los principales sistemas de una vivienda como iluminación, enchufes, control de persiana, apertura de puertas, temperatura y humedad y control de la calefacción. Al final añade la unidad central para control del sistema completo. El autor demuestra la escalabilidad del sistema y la disponibilidad del mismo frente a fallos o errores, precisando el reducido costo del sistema implementado. [2]

En 2018, Jorge Jarne Brun presenta su investigación denominada “Smart Home usando IoT y Chatbots” (Universidad Complutense de Madrid), diseña un sistema de hogar inteligente, el cual utiliza como unidad central de procesamiento una Raspberry PI y la plataforma de IBM Bluemix con Node-RED. La programación la realiza a través de nodos o bloques, los cuales están interconectados entre ellos con una interfaz de navegador web construido a partir de node.js y una API que conecta con un servicio de mensajería móvil. Los datos recogidos por cada una de las placas se envían cada 5 segundos a través del protocolo MQTT al que se le ha asignado QoS para evitar el envío de duplicados hacia la nube. La ejecución de acciones y control de los dispositivos conectados a la red local de la casa, se logra enviando mensajes de texto a través de la aplicación de mensajería Telegram y el usuario recibe la información acerca del estado del dispositivo seleccionado. [3]

En 2017, Imanol Sanz Ruiz desarrolla la investigación “Diseño e Implementación de una aplicación móvil para el control de un sistema domótico” (Universidad Carlos III de Madrid). El trabajo presenta una arquitectura maestro- esclavo, con una aplicación y una Raspberry PI con sensores y actuadores. El nodo maestro se encarga de enviar ordenes en forma de peticiones al nodo esclavo, que realiza la acción especificada identificando alguno de los 3 tipos de órdenes, encendido/apagado, desactivación de elementos y recuperación de información, bajo un modelo vista-controlador para la gestión del sistema. Las órdenes enviadas desde la aplicación se ejecutan en un tiempo cercano a 2 segundos, con espera de aproximadamente 3 segundos adicionales para verse reflejadas en la interfaz y continuar con una nueva orden. [4]

En 2017, María Díaz Hernández presenta su trabajo de investigación denominado “Caracterización, implantación y evaluación de dispositivos domóticos para adultos

mayores mediante sistemas embebidos e internet de las cosas” (Universidad Distrital Francisco José de Caldas), en la cual analiza las potencialidades de diversos dispositivos para el control domótico como sensores y actuadores. Implementa un sistema de adquisición y supervisión de datos, con el fin de regular las condiciones y mantener un ambiente adecuado para el adulto mayor. Mediante el uso de un servidor central se almacenan los datos en la memoria del sistema embebido que opera con Raspberry Pi, PcDuino y Dragon Board, estos datos son transmitidos a la nube para que puedan ser leídos en cualquier instante de tiempo. Como resultado los datos procesados y estado de los dispositivos se muestran a través de una interfaz web, a la cual se puede acceder desde cualquier equipo con conexión a internet. [5]

1.2.1 Contextualización del Problema

En el Ecuador, el grupo poblacional de adultos mayores en 2020 alcanzó el 7,4% del total de la población ecuatoriana, esto a razón de la esperanza de vida que actualmente es de 76.6 años para hombres y 83,5 años para mujeres. Este porcentaje representa alrededor de 1.049.824 personas mayores de 65 años, lo que proyecta el incremento de este grupo poblacional en los siguientes años. Esto ha generado un incremento en los programas, planes y proyectos por parte del Ministerio de Inclusión económica y social (MIES) destinados a mejorar las condiciones de vida a cerca de 70.881 personas mayores que se benefician en centros de diversas modalidades de atención y cuya infraestructura debe cumplir con normativas técnicas orientadas al adulto mayor. [6]

En el 2021 en la provincia de Tungurahua cerca de 2.768 personas fueron beneficiarias del programa “Mis Mejores Años” promovido por el MIES, de las cuales 90 personas reciben atención paulatina en dos centros dedicados al cuidado de adultos mayores. La intención es mejorar su calidad de vida y acceso a derechos para evitar situaciones de vulnerabilidad y riesgo. [7] Para contribuir la vida del adulto mayor es necesario vincular tecnologías enfocadas al cuidado y asistencia, en combinación del personal capacitado en manejar dichas herramientas tecnológicas, y de esta forma brindar la atención adecuada para mejorar su situación social y condiciones de vida dentro de sus entornos. [8]

Según División de Protección Social y Salud del Banco Interamericano de Desarrollo (BID), determina al Ecuador como un país con creciente demanda de servicios de atención a la dependencia en materia de envejecimiento, donde la infraestructura de los sistemas de cuidado no cubre las necesidades de la población adulta mayor. De este análisis, el proyecto surge para contribuir a programas que integren tecnologías destinadas a adultos mayores y a su vez disminuir la brecha digital. Además de integrar tecnologías adaptadas a sus entornos que puede ser implementadas en las instalaciones destinadas a su cuidado. [9]

La infraestructura en centros de atención para adultos mayores, se enfoca en brindar entornos habitacionales comunales, en los cuales aún no se ha integrado tecnologías que permitan su rápida adaptación y desenvolvimiento en sus entornos. De acuerdo al análisis de necesidades, las herramientas tecnológicas para estos centros deben dar como resultado un incremento en su calidad de vida. [10]

El desarrollo del proyecto de investigación propone realizar un sistema domótico utilizando tecnología IoT junto a software y hardware libre para la gestión de los módulos electrónicos. Realiza además la ejecución de órdenes a través del reconocimiento de voz para el control de dispositivos mediante la utilización de una Raspberry Pi. El sistema incorpora una interfaz capaz de supervisar las condiciones ambientales y la gestión de notificaciones de forma automática a través de servicios de mensajería móvil.

1.3 Fundamentación Teórica

1.3.1 Centro Integral del Adulto Mayor

El Centro Integral del Adulto Mayor de la Ciudad de Ambato (CIAM) bajo administración del GAD Municipalidad de Ambato, brinda los servicios de cuidado y asistencia a personas adultas mayores de forma ambulatoria o permanente en sus instalaciones. Parte de las personas que reciben atención en el CIAM conviven en

ambientes comunales durante las 24 horas del día, mientras que un grupo reducido lo hace de forma transitoria.

1.3.2 Adulto Mayor

Los cambios que constituyen e influyen el envejecimiento son complejos. En el plano biológico y del ciclo de la vida, el envejecimiento se relaciona con la acumulación de daños moleculares y celulares en el cuerpo de una persona. Con el tiempo, estos daños reducen gradualmente las reservas fisiológicas, aumentan el riesgo de muchas enfermedades y disminuyen en general la capacidad del individuo. A la larga, sobreviene la muerte. [11]

Se considera como una persona adulta mayor al individuo que han cumplido o sobrepasa los 65 años de edad. El proceso de envejecimiento difiere de acuerdo a las condiciones sociales, educativas, culturales y económicas. Las personas de sesenta y cinco años de vida o más, según la Constitución, son un grupo de atención prioritaria. Acorde con ello deben recibir atención especializada de calidad. [12]

Discapacidad en Adultos Mayores

Con la edad aumenta el riesgo de muchos trastornos de la salud, y estos pueden tener efectos importantes en la capacidad de desenvolverse en actividades cotidianas. A medida que se envejece suelen ocurrir cambios en la anatomía o pérdida de algunas de las funciones vitales y derivar en discapacidad. Aunque existe una diversidad notable en cómo se viven estas afectaciones a nivel individual, se observan tendencias generales cuando se considera la población en su conjunto, siendo las siguientes las principales. [11]

Discapacidad Motriz

Esta discapacidad tiene relación con las deficiencias en todas las funciones de los sistemas osteoarticular y neuromuscular tendinoso, que resultan en limitaciones físicas motoras que presente el individuo al realizar una tarea o acción en su entorno. [13]

Discapacidad Sensorial

Hace referencia a las deficiencias en las funciones del sistema auditivo y los órganos de la vista, determinando las limitaciones que presente el individuo al realizar una tarea tomado como parámetro su capacidad o habilidad real, sin que sea aumentada por la tecnología o dispositivos de ayuda o terceras personas.

Discapacidad Intelectual

Es un término global que reúne la disminución o pérdida de la capacidad mental y del sistema nervioso y en las limitaciones que se presenten en el individuo al realizar una tarea. Generalmente se presentan trastornos de comportamiento, depresión o demencia. [13]

Confort y cuidado de adultos mayores

La calidad de vida del adulto mayor se relaciona al entorno que lo rodea o realiza sus actividades diarias, por lo tanto, el confort térmico se describe como el conjunto de características del ambiente que afectan el intercambio de calor entre el cuerpo humano y el medio ambiente. Al disminuirse las funciones corporales y la actividad física de los adultos mayores, resulta necesario abordar el tema de la adecuada temperatura para adultos mayores.

Para considerarse una temperatura agradable para los adultos mayores se debe tener en cuenta la sensación térmica del individuo, de manera que diversos estudios determinan que esta temperatura tiene un rango entre los 20°C - 24°C, este rango no contempla la humedad del ambiente. Un ambiente regulado asegura una calidad de vida digna, sin embargo, las normas de confort no se aplican actualmente a la población de edad avanzada. [14]

Para asegurar el bienestar y cuidado de las personas adultas mayores, se debe tomar en cuenta ciertas características que tienen relación con la dependencia, para planificar sus cuidados, motivar a que la persona cuidada sea quien realice la mayor cantidad de actividades y solo asistirle en las que no pueda realizarlas. [15]

Tecnologías digitales en adultos mayores

Los adultos mayores perciben que la utilización de tecnologías mejora la calidad de vida, incrementa su autoestima y sus habilidades personales. El uso de tecnologías con requerimientos destinados hacia personas mayores facilita la atención y asistencia, aumentando el desarrollo personal y autonomía. El impacto en el desarrollo personal favorece a una actitud más activa debido a la incorporación de las tecnologías de la información y la comunicación, y resulta en una oportunidad clave para la integración social. [16]

La domótica es la evolución lógica de la instalación eléctrica en la vivienda, constituye un medio para que el usuario pueda automatizar su casa y a su vez controlar su espacio cotidiano mediante la electricidad, la electrónica y la informática, aplicadas a la gestión técnica de viviendas. Proporciona confort, ahorro energético y seguridad, entre otras ventajas, por ello la domótica cobra un importante papel para la creación de hogares digitales o viviendas inteligentes en constante evolución. [17]

Las tareas cotidianas de un hogar como controlar la calefacción, encender/apagar luces de una o varias habitaciones, así como las tareas programadas como alarmas, detector de humo/gas, sistemas automáticos de riego, pueden cumplir su función establecida a una hora determinada quedan en manos de un sistema de control. Las casas automatizadas pueden contar con sensores, actuadores y detectores que alerten de acontecimientos o situaciones de riesgo. [18]



Figura 1. Domótica con diferentes funciones de control [17]

Los elementos básicos que componen un sistema domótico son los siguientes:

Sensores: Dispositivos encargados de adquirir información del entorno. Pueden requerir de un acondicionamiento de señal a la entrada, esto en dependencia de la magnitud medida y el tipo señal de salida que puede ser voltaje, carga o corriente.

Actuadores: Dispositivos que destinados a conectar o desconectar los dispositivos que proporcionan un servicio a la vivienda. Como por ejemplo encender o apagar una luz, cortar la llave de agua o cualquier otra acción que requiera un accionamiento de tipo mecánico.

Controladores: Dispositivos encargados de supervisar y realizar el control del sistema domótico. Procesa toda la información que recibe de los diferentes nodos o dispositivos del sistema. Pueden existir varios controladores, según la arquitectura del sistema domótico.

Interfaz de usuario: Elemento que proporciona información acerca del estado actual del sistema de forma visual o sonora. Actualmente se utilizan interfaces gráficas, páginas web o aplicaciones móviles.

Pasarela de Comunicación: Elemento encargado de brindar la comunicación entre la infraestructura de comunicaciones de la vivienda y la red de datos. Responsable de controlar los elementos que conforman el sistema. [17]

1.3.3 Tecnologías domóticas

En su primera etapa, surgieron soluciones desde empresas pequeñas basados en equipos electrónicos propietarios cuyos sistemas internos no eran publicados para evitar ser copiados por terceros. Sin embargo, empresas más grandes, conscientes del crecimiento y oportunidades que puede llegar a tener la domótica se unieron para definir y establecer protocolos comunes. [17] Las diferentes tecnologías domóticas más usadas en la actualidad se muestran en la Tabla 1.

Tabla 1. Diferentes Tecnologías Domóticas [17]

Sistema Domótico	Características
X10	De origen europeo, se basa en la transmisión de las comunicaciones por corrientes portadoras a través del cableado eléctrico de la vivienda. Sistema descentralizado. Sensible a interferencias, obsoleto desde 2013
KNX	Se basa en una conexión y comunicación en bus, es un sistema abierto e independiente del fabricante, fácilmente ampliable, soportando múltiples medios de transmisión. Puede acoplarse a otros sistemas.
Lon Works	Tecnología de automatización, basado en sistema abierto y descentralizado. Sistema robusto que permite la comunicación entre dispositivos de la misma red sin la necesidad de un controlador. Trabaja a grandes velocidades en comparación con otros sistemas
Insteon	Creado para solucionar los problemas de X10 además que es compatible con ésta, destinado a ser mucho más seguro, sencillo y de bajo costo. Combina la comunicación por corrientes portadoras con comunicación a través de radiofrecuencia.
HomeKit	Es un sistema de la marca Apple, parte de la premisa de controlar los dispositivos domóticos mediante voz y mediante el uso de smartphone, además que ha sido preparado para ser utilizado por el iPhone, iPad y Apple TV. Los diferentes dispositivos domóticos se conectan de manera inalámbrica y es compatible con diversos sistemas tales como wifi, bluetooth, Z-Wave, ZigBee e Insteon.

El nivel de despliegue o instalación de cada uno de los sistemas domóticos mostrados en la Tabla 1, depende de la arquitectura propia de cada uno de estos sistemas, por lo

que se debe analizar la infraestructura en la cual se desea implementar cada uno de estos sistemas.

1.3.4 Arquitecturas de control domótico

Según la forma en que la red une los distintos nodos, dispositivos o terminales se dispone de lo que se suele denominar como una arquitectura de control domótico. Dicha arquitectura puede ser de varios tipos: [19]

Arquitectura centralizada

Los sistemas centralizados se caracterizan por tener un único nodo de procesamiento que recibe toda la información de las entradas, que la procesa y envía a las salidas los órdenes de acción correspondientes. Están unidos a un nodo central que dispone las funciones de control y mando. Una de las ventajas de es el costo reducido o moderado combinado con una instalación sencilla.

Arquitectura Descentralizada

En esta arquitectura hay varios controladores interconectados por un bus de datos, que envía información entre ellos. La información también se envía entre actuadores e interfaces conectados a los controladores, esto en dependencia del programa, configuración e información que reciben de los sensores, sistemas interconectados o de los usuarios.

Arquitectura Distribuida

Dentro de la arquitectura distribuida, cada uno de los sensores y actuadores son también controladores, con capacidad de actuar y enviar información al sistema de acuerdo a su configuración, la información que capta por sí mismo y la que recibe de los otros dispositivos del sistema a través de un único bus de datos que interconecta a todo el sistema.

Arquitectura Híbrida (mixta)

También denominada mixta, se combinan las arquitecturas de los sistemas centralizadas, descentralizadas y distribuidas. A la vez que puede disponer de un controlador central o varios controladores descentralizados, los dispositivos de interfaces, sensores y actuadores pueden también ser controladores como en un sistema distribuido, y procesar la información según la configuración. La información que capta por sí mismo, y procesa como enviarla a otros dispositivos de la red, sin que necesariamente pasa por otro controlador. [19]

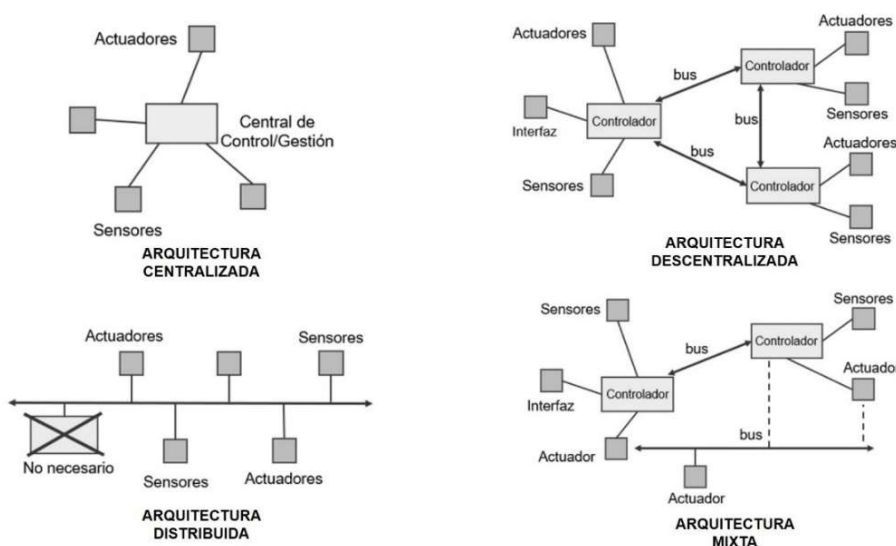


Figura 2. Arquitecturas de control domótico
Elaborado por: El investigador basado en [19]

1.3.5 Internet de las Cosas (IoT)

Percibida como una nueva revolución tecnológica, el término Internet de las Cosas (IoT) se refiere a una red cada vez más extendida, una de objetos materiales conectados a Internet, identificados y reconocidos, como todos los demás dispositivos tradicionales que se usan cotidianamente. Esta tecnología contempla la interacción entre la seguridad para conectar miles de objetos, los protocolos de comunicación, estándares y la compatibilidad para lograr una convergencia tecnológica. [20]

El Internet de las cosas además hace referencia, a una tecnología basada en la conexión de dispositivos a Internet que intercambian, agregan y procesan información sobre su entorno físico para proporcionar servicios de valor añadido a los usuarios finales. Reconoce eventos o cambios, y tales sistemas pueden reaccionar de forma autónoma y adecuada en dependencia a su configuración. Su finalidad es brindar una infraestructura que supere la barrera entre los objetos del mundo físico y su representación en los sistemas de información. [21]

La arquitectura de IoT se compone de tres componentes básicos, como dispositivos, puerta de enlace y nube. La Figura 3 muestra la arquitectura IoT.

Nube: Servidores en la nube alojados en centros de datos que consolidan, procesan y almacenan los datos con seguridad, incluyendo toma de decisiones. Varias empresas brindan servicios como cloud computing, de gran demanda para el desarrollo de sistemas IoT.

Puerta de enlace: Se encarga de proveer el servicio de transporte de datos a los diversos dispositivos, hacia la nube mediante protocolos IoT

Dispositivos: También denominados “cosas”, son dispositivos que se conectan a través de un cable o de forma inalámbrica a una red más amplia, para comunicarse con la Nube, Mayormente son sensores o actuadores. [22]



Figura 3. Arquitectura IoT

Elaborado por: El investigador basado en [22]

1.3.6 Aplicaciones del IoT

Debido al impacto que ha tenido esta tecnología en diversos sectores de la industria, el IoT ha demostrado la importancia que tendrá en el futuro dentro de diversos campos. Esto quiere decir que las empresas y los dispositivos personales serán más eficientes y competitivos para todos los negocios del mundo actual. Por ende, se relacionan las mejores aplicaciones del IoT, estas son: [23]

- Redes eléctricas inteligentes
- Salud y telemedicina
- Hogares y edificios inteligentes
- Seguridad
- Riego y Agricultura inteligente

1.3.7 Tecnologías de comunicación inalámbrica

Existe una diversa cantidad de tecnologías inalámbricas disponibles tanto para IoT y la domótica, cada una de ellas brinda ventajas y limitaciones de acuerdo al nivel de despliegue que requiera el usuario final. Entre las tecnologías que más destacan se tiene: wifi, bluetooth, zigbee, LoRa, Z-Wave; las características de estas se muestran en la Tabla 2. [24]

Tabla 2. Comparación de tecnologías inalámbricas [24]

Parámetro	WiFi	Bluetooth	Zigbee	LoRa	Z-Wave
Estándar	IEEE 802.11 a/c/b/d/g/n	IEEE 802.15.1	IEEE 802.15.4	LoRaWAN R1	IEEE 802.15.4
Frecuencia	2.4GHz 5-60 GHz	2.4 GHz	868/915 MHz, 2.4 GHz	868/900 MHz	900MHz 2.4 GHz
Velocidad de datos	1 Mb/s - 6.75 Gb/s	1 - 24 Mb/s	40 - 250 Kb/s	0.3 - 50 Kb/s	9,6-100kb/s
Alcance	20 – 80m	8 - 10 m	10 - 20m	< 15 Km	30 m
Consumo de Energía	Alto	Medio	Baja	Muy Baja	Baja

Topología	Estrella, P2P, malla	Malla, estrella, árbol	Árbol, malla estrella y P2P	Estrella	Estrella Árbol
------------------	----------------------	------------------------	-----------------------------	----------	-------------------

Elaborado por: El investigador

1.3.8 Estándar IEEE 802.11

El protocolo IEEE 802.11 ha resultado ideal para su implementación en áreas públicas y privadas en los últimos años. Los dispositivos portátiles, como los teléfonos móviles, utilizan interfaces WLAN para acceder a aplicaciones/servicios a través de Internet. El estándar IEEE 802.11 original introdujo el conjunto de servicios básicos (BSS) y la creación de redes en bandas sin licencia (de uso gratuito) de 2,4 GHz. Este estándar admite espectro ensanchado de secuencia directa (DSSS) y modulación del tipo Multiplexación por División de Frecuencias Ortogonales (OFDM) en las frecuencias de 2,4 GHz y 5GHz. [25]

Con la convergencia y avance de tecnologías inalámbricas, se desarrollaron diferentes versiones del estándar IEEE 802.11, las principales de uso comercial se muestran en la Tabla 3.

Tabla 3. Versiones del estándar IEEE 802.11 [25] [26]

Versión de IEEE 802.11	Banda de Frecuencias	Velocidad Máx.	Compatibilidad con otras versiones
802.11 ^a	5 GHz	54 MBps	ninguna
802.11b	2,4 GHz	11 MBps	ninguna
802.11b	2,4 GHz – 5 GHz	54 MBps	802.11b
802.11g	2,4 GHz	54 MBps	802.11b
802.11n	2,4 GHz – 5 GHz	600 MBps	802.11 ^a , b, g
802.11ac	2,4 GHz – 5 GHz	1,3 GBps	802.11 ^a , n

Elaborado por: El investigador

Las versiones y características mostradas en la Tabla 3, corresponden a las versiones disponibles de manera comercial, siendo las versiones 802.11g, 802.11n y 802.11ac las de mayor relevancia para el proyecto por su compatibilidad con los dispositivos a utilizarse.

1.3.9 Protocolo de comunicación MQTT

MQTT es un protocolo ligero de mensajería de publicación/suscripción, diseñado para aplicarse a la telemetría M2M (máquina a máquina) en entornos de bajo ancho de banda. Aunque comenzó como un protocolo propietario, se lanzó libre de regalías. MQTT significa MQ Telemetry Transport (Transporte de Telemetría), es importante indicar que MQTT actúa sobre el protocolo TCP/IP. [27]

Este protocolo es un conjunto de reglas que define cómo los dispositivos o cosas del IoT pueden publicar y suscribirse a datos a través de Internet. MQTT se usa para mensajería e intercambio de datos entre IoT como dispositivos integrados, sensores, actuadores, controladores, inclusive llegando a conectar PLC industriales, etc.


El protocolo se basa en eventos y a partir de estos conecta dispositivos mediante el patrón de publicación/suscripción (Pub/Sub). El remitente (Editor) y el receptor (Suscriptor) se comunican a través de Temas y están desacoplados entre sí. La conexión entre ellos es manejada por el broker MQTT. El broker MQTT filtra todos los mensajes entrantes y los distribuye correctamente a los suscriptores. [28]

Broker MQTT

El crecimiento de este protocolo de comunicación, diversas empresas han desarrollado brokers adaptados a diferentes necesidades. La página oficial de MQTT ofrece una gama de brokers a disposición de desarrolladores, los principales se muestran a continuación en la Tabla 4.

Tabla 4. Brokers MQTT

Broker MQTT	Descripción y Características
<p style="text-align: center;">EMQX</p> 	<p>Broker de mensajería MQTT distribuida de código abierto, diseñado para ser escalable y altamente disponible para aplicaciones en M2M, móviles e IoT. Puede manejar decenas de millones de clientes simultáneos ya que la interfaz de usuario y configuración permite gestionarlos de manera eficiente. Presenta compatibilidad con mensajes QoS0, QoS1, QoS2.</p>
<p style="text-align: center;">Mosquitto</p> 	<p>Es un broker Eclipse Mosquitto MQTT disponible públicamente para distribuciones como Linux, mac o Windows. MQTT es un protocolo muy ligero que hace uso de un modelo de publicación/suscripción entre los dispositivos. Soporta las más recientes versiones de MQTT v3.1 v3.1.1 y en v5.0 algunas características no se utilizan directamente</p>
<p style="text-align: center;">Hive MQ</p> 	<p>HiveMQ creó desde cero, tomando en cuenta la máxima escalabilidad y la seguridad empresarial. Viene con soporte de socket web nativo y un complemento SDK de código abierto para ampliar su operatividad o a su vez integrarlo con otros componentes del iot. También está disponible un servidor de prueba público.</p>
<p style="text-align: center;">Cassandana</p>	<p>Cassandana es un broker de mensajes MQTT de código abierto, desarrollado completamente en Java. Este proyecto comenzó su vida como una rama de desarrollo de Moquette, y luego se depuró para su optimización y adición de funciones adicionales. Ahora está listo para</p>

	funcionar como un intermediario de mensajes en ámbitos empresariales e industriales.
<p>Mosca</p> 	<p>Agente MQTT de amplio despliegue en proyectos, este broker fue diseñado para ser altamente compatible con Node.js y gestionar de manera eficiente los dispositivos. Cuenta con la versión MQTT 3.1 y 3.1.1.</p> <p>Agrega además QoS0 y QoS1.</p>

Elaborado por: El investigador basado en [29] [30]

Funcionamiento de MQTT

La comunicación en MQTT se basa en temas (topics), y para que un cliente MQTT tenga acceso a información debe estar suscrito al tema. Un cliente puede publicar mensajes, y los nodos que deseen recibir estos mensajes deben estar suscritos al tema específico. La comunicación puede ser de uno a uno, o de uno a muchos. Un topic se representa mediante una cadena de texto y tiene una estructura jerárquica separada con “/”. La forma básica del funcionamiento se muestra en la Figura 4. [31]

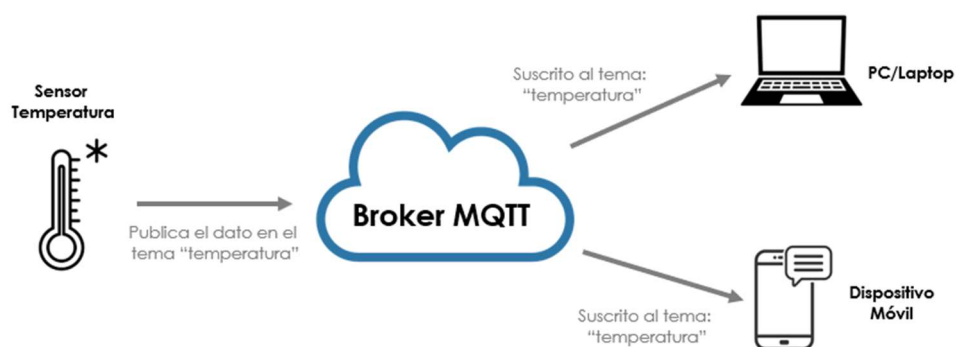


Figura 4. Funcionamiento de MQTT

Elaborado por: El investigador basado en [31]

Calidad de Servicio

Para brindar la seguridad adecuada en la comunicación del protocolo MQTT, se integra el nivel de calidad de servicio (QoS), es un acuerdo entre el remitente de un

mensaje y el receptor que define la garantía de entrega de un mensaje específico. Hay 3 niveles de QoS en MQTT. [32]

- **QoS 0:** Una vez el mensaje se ha enviado, el destinatario no acusa el recibo del mensaje y el remitente no almacena ni retransmite el mensaje.
- **QoS 1:** Garantiza que el mensaje se entregue al menos una vez al receptor. El mensaje se puede enviar o entregar varias veces.
- **QoS 2:** este nivel de servicio garantiza que los destinatarios previstos reciban cada mensaje solo una vez. QoS 2 es el nivel de servicio más seguro y a la vez el más lento.

1.3.10 Servidor LAMP

Un servidor LAMP es un conjunto de servicios que se pueden emplear para desarrollar y configurar un servidor web completamente funcional, además que permite la creación de diversos tipos de interfaz de usuario. LAMP es el acrónimo de:

- Linux es el sistema operativo, libre, gratuito y con diversas distribuciones en acorde a la necesidad del usuario.
- Apache es un servidor de código abierto HTTP. Es un servidor escalable, eficiente y seguro que permite almacenar páginas web.
- MySQL es un gestor de base de datos de código abierto en el que se almacena y administra la información de manera eficiente y fácil.
- PHP y a su vez lenguajes como Perl o Python, son utilizados para desarrollar o ejecutar scripts de manera fácil y orientados a objetos, eficientes e independientes de la arquitectura en la que se aplique. [33]

1.3.11 Node Red

Node Red es una herramienta que proporciona un editor basado en node.js, facilita la conexión de flujos (flows) mediante una amplia gama de nodos (nodes) disponibles en la paleta que se pueden implementar en su tiempo de ejecución con un solo clic. Permite conectar dispositivos de hardware, API y servicios en línea de formas nuevas

e interesantes, haciendo uso de variados protocolos comunicación, a través del puerto 1880. [34]

Al ser una opción para crear plataformas de procesamiento, control y monitoreo, resulta ser la herramienta ideal de programación basada en el navegador web para proyectos de IoT. Si es instalado dentro de una instancia remota puede trabajar en la nube o de forma local como microcontroladores como Raspberry, éste último en sus actualizaciones recientes viene preinstalado y listo para su uso y configuración. [35]

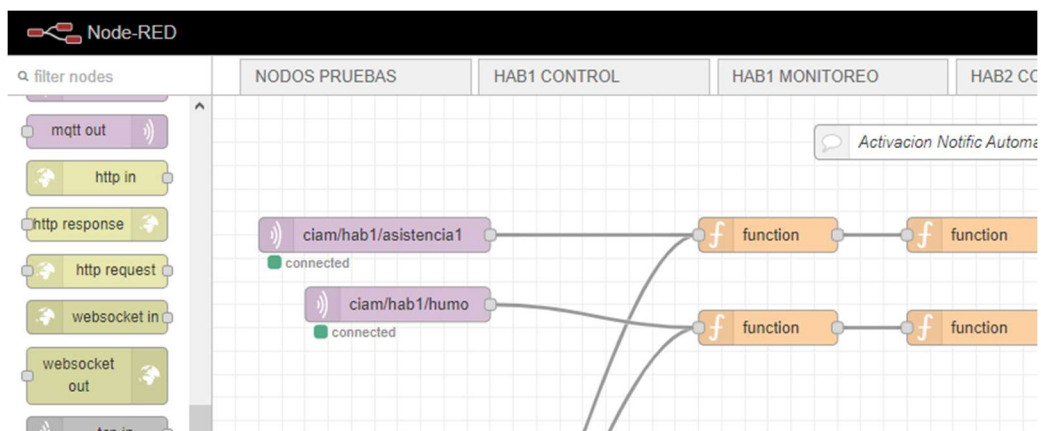


Figura 5. Programación a través nodos y flujos en Node Red

Elaborado por: El Investigador

1.3.12 Servicios de mensajería móvil

Los servicios de mensajería móvil o instantánea se basan en la arquitectura cliente-servidor para enviar y recibir mensajes en tiempo real. El cliente está conectado en una computadora o smartphone, la cual viene a ser la interfaz mediante una aplicación que utiliza para comunicarse con uno o varios usuarios, siendo el servidor es el que maneja toda la comunicación cliente/servidor de forma bidireccional, todo ello mantenido por un proveedor de dicho servicio. En este modelo, el servidor no solo tiene la responsabilidad del envío de los mensajes, sino también de la autenticación de los usuarios. [36]

Algunos de los servicios de mensajería instantánea poseen características específicas, que los hacen más atractivos para los usuarios, sin embargo, la preferencia de éstos

también se relaciona con la cantidad de usuarios que disponen de este servicio. Algunos de los más importantes se detallan con sus características en la Tabla 5.

Tabla 5. Características de los servicios de mensajería móvil [37]

Parámetro	Whatsapp	Telegram	Messenger Meta	Hangouts
Envío de Ficheros	Si	Si	Si	Si
Chatbots	Si	Si	Si	No
Canales	No	Si	No	No
Tipo de Registro	Número celular	Número celular	Correo Electrónico o número celular	Correo Electrónico
Notas de Audio	Si	Si	Si	Si

Elaborado por: El investigador

1.3.13 Reconocimiento de Voz

El reconocimiento de voz hace referencia al proceso de convertir una señal de voz en una secuencia de palabras por medio de un algoritmo implementado como un programa de computadora. Tiene el potencial de ser un modo importante de interacción entre humanos y máquinas con distintas finalidades como enviar órdenes a sistemas, control de elementos o asignación de tareas. [38]

El proceso de reconocimiento de voz engloba varios componentes, inicialmente requiere del habla de la persona, el sistema realiza un análisis espectral de la información recibida para determinar un modelo acústico y la clasificación de patrones, finalmente establece una puntuación de confianza, que verifica las palabras dichas por el hablante. Este proceso se aprecia en la Figura 6. [39]

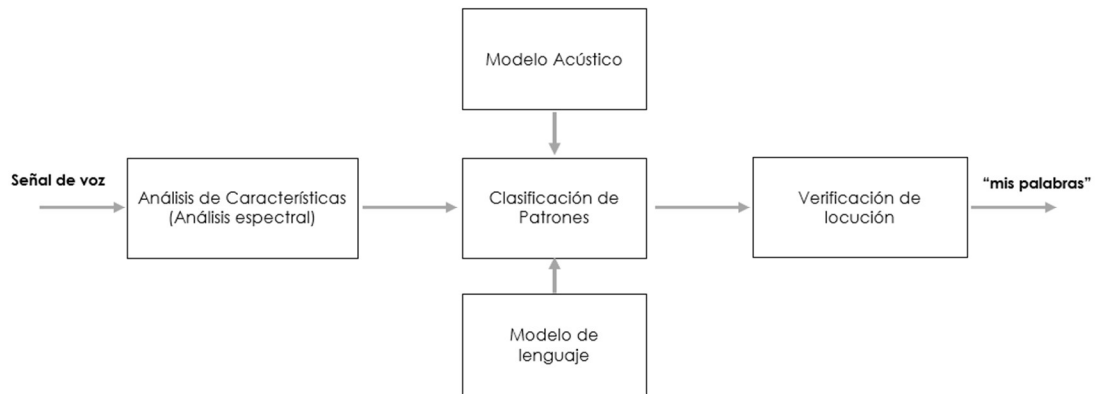


Figura 6. Sistema de reconocimiento de voz
Elaborado por: El investigador Basado en [39]

Sistemas de Dictado

El sistema de dictado es una de las opciones más recurrentes en la tecnología del reconocimiento de voz. Se utiliza una estructura de palabras más comunes para la precisión de este tipo de sistemas. Sus aplicaciones se han extendido en varios campos, a fin de ayudar a y facilitar tareas específicas. Hacen uso de diversos métodos de reconocimiento de la voz, los más usados a día de hoy son los modelos ocultos de Markov (HMM) y redes neuronales. [40]

Google Speech to Text




Con “Speech-to-Text”, se puede realizar una integración sencilla de las tecnologías de reconocimiento de voz de Google en las aplicaciones de los desarrolladores. Envía audios y recibe una transcripción de texto del servicio de la API de Speech-to-Text. La API está disponible en todas las regiones y permite su uso a partir de la configuración de elementos de entrada del sistema como la selección del idioma o nivel de interferencia. [41]

Para el proyecto se utilizará este servicio debido a que se puede implementar a partir de la descarga del paquete “Speech Recognition”. en las librerías de python.

1.3.14 Ordenadores de placa reducida

Una computadora u ordenador de placa única se define como una unidad de procesamiento de propósito general construida en una placa única con uno o varios microprocesadores, memorias, puertos de entrada/salida (E/S) y otras características propias de una computadora funcional. Los factores decisivos para el rendimiento general del sistema en estos ordenadores se pueden resumir en lo siguiente: software, tamaño de placa y tiempo de comercialización. [42]

Tabla 6. Ordenadores de placas reducidas

Características	Odroid-C2	Jaguar One	Raspberry Pi 4
Dispositivo			
Dimensiones	85mm x 56mm	101.9mm x 73.8mm	85mm x 56 mm
Procesador	Amlogic S905	Intel Atom Z3735G	Broadcom BCM2711
Memoria RAM	2GB DDR3	1GB DDR3	2GB, 4GB, 8GB
Capacidad máx. de memoria	16GB	16GB	32GB
Conectividad	Ethernet, WiFi	Ethernet	Ethernet, Wifi, Bluetooth
Pines GPIO	40 pines	4 pines + I2C	40 pines
SO compatible	Linux, Android	Linux, Android, Windows	Linux, Windows, otros

Elaborado por: El investigador basado en [43] [44] [45]

1.3.15 Microcontrolador ESP32

El microcontrolador ESP32 producido por “Espressif”, es conocido por sus diversas tarjetas electrónicas comercialmente conocidas como módulos de desarrollo, contienen la conectividad Wifi y Bluetooth en la misma placa. Los módulos que

contienen la ESP32 se enfocan a una amplia variedad de aplicaciones relacionadas con el internet de las cosas, que van desde redes de sensores de baja potencia, transferencia de información o procesamiento de datos provenientes de sensores o aplicaciones IoT generalmente en ambientes no industriales. [46]

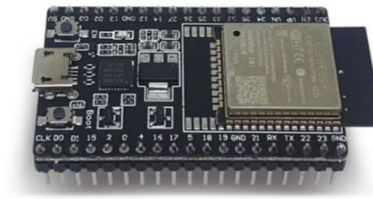


Figura 7. Microcontrolador ESP32 devkitc v4 [47]

Las principales prestaciones del microcontrolador ESP32 son las siguientes:

- Conectividad WiFi 802.11 b/g/n
- Conectividad Bluetooth y BLE(blueetooth low energy)
- Memoria ROM 448 Kbytes
- Memoria SRAM 520 Kbytes
- 24 pines de propósito General (Total 38 pines)
- Frecuencia de reloj hasta 240MHz
- Memoria QSPI Flash/SRAM 4 Mbytes
- Volteje de alimentación 5v DC
- Voltaje de entrada/salida 3.3v DC
- Puerto USB micro B
- Botones de BOOT y ENABLE [46]

Para los microcontroladores ESP32 existen diversas placas con la misma cantidad de pines, pero no mantienen las dimensiones de la imagen mostrada en le Figura 7, algunas versiones requieren la conexión física de una antena, mientras que otras integran su antena por sobre la misma placa de desarrollo.




1.3.16 Sensores

Sensores de Temperatura y Humedad

Este tipo de sensores, se conforman de dos etapas en las cuales, la primera se encarga de medir la temperatura ambiente basándose en la variación de la resistencia de materiales como el Platino o Paladio. Mientras que para medir la humedad relativa del ambiente utiliza un elemento capacitivo, el cual varía sus características eléctricas respondiendo a variaciones de humedad, de tal manera que al absorber humedad disminuye su capacidad. Estos cambios son detectados, linealizados y amplificados por un circuito electrónico para su posterior transmisión de datos. [48]

En la siguiente tabla se muestran diversos sensores para la adquisición de las magnitudes de temperatura y humedad relativa del aire:

Tabla 7. Comparación entre sensores de Temperatura y Humedad

Parámetro	SHT-31	DHT22	BME280
Dispositivo			
Dimensiones	18mm x 16mm	22mm x 28mm	17mm x 18mm
Número de Pines	7 pines (Vin, GND, SCL, SDA, ADR, RST, ALR)	3 pines (GND, VCC y DAT)	7 pines (Vin, 3Vo, GND, SCK, SDO, SDI, CS)
Alimentación	3.3 -5 VDC	3.3 – 6VDC	3 – 5VDC
Rango de Temperatura	-40°C a 90°C 0 – 100% HR	-40°C a 80°C 0%HR a 100%HR	-40 a 85°C 0 – 100% HR 300 – 1100 hPa
Comunicación	I2C	Single Bus Data	SPI-I2C

Elaborado por: El investigador basado en [49] [50] [51]


La tabla de sensores de temperatura y humedad mostrados anteriormente pueden ser integrados al proyecto y presentan compatibilidad de funcionamiento para la adquisición, procesamiento y comunicación de datos con el microcontrolador ESP32.

Sensores de humo

Los medidores de humo pueden utilizar diferentes tecnologías, habitualmente utilizan la tecnología de infrarrojos. Esto se debe a que los sensores de infrarrojos son más estables que los químicos y, además tienen un mayor ciclo de vida y soportan altos niveles de humedad, polvo, suciedad y otros agentes adversos del ambiente. El sistema más usado por los fabricantes y el recomendado, es el denominado NDIR (Infrarrojo No Dispersivo) está basado en el principio de absorción de energía de los compuestos a una determinada longitud de onda, normalmente en el infrarrojo. [52]

En la Tabla 8 se muestra distintos sensores para medir o detectar humo, además de otros gases.

Tabla 8. Comparación entre sensores de Co2

Parámetro	DYPME-0010	MQ-135	MG8011
Dispositivo			
Dimensiones	70mm x 50mm	32mm x 22mm	32mm x 4 mm
Número de pines	5 pines (GND, 9V, NO, NC, Common Relay)	4 pines (Vcc, GND, Digital out, Analog out)	4 Pines (Vcc, GND, Analog & digital out)
Alimentación	9VDC	5VDC	3.7 – 5VDC
Detección	CO2	CO2, NH3, alcohol, NOx	CO2, CO, alcohol
Sensibilidad Ajustable	Si	Si	No

Temperatura de trabajo	-5°C a 50°C.	-10°C a 45°C	20°C a 50°C
Señal de salida	Relevador NC - NO	0- 5V Digital	2.7 – 4.1V Digital

Elaborado por: El investigador basado en [53] [54] [55]

Sensores de Movimiento

En los sensores de movimiento, el sensor PIR consta en realidad de 2 elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso de las series HC, la señal generada por el sensor ingresa al circuito de detección, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales. Así mismo pueden poseer el ajuste de parámetros para modificar la sensibilidad y distancia de detección. [56]

Tabla 9. Comparación entre sensores movimiento

Parámetro	HC SR501	HC SR505
Tipo	PIR	PIR
Dimensiones	32mm x 24mm x 18mm	10mm x 23mm x 10mm
Número de pines	3 pines (Vcc, GND, Digital out)	3 pines (Vcc, Digital output, GND)
Alimentación	5VDC	5VDC
Distancia de detección	3 – 7 metros	7 metros
Sensibilidad ajustable	Tiempo de disparo y distancia	No
Ángulo de detección	110° de abertura	100° de abertura
Señal de salida	0 – 3,3VDC digital	0 – 3,3VDC digital
Temperatura de trabajo	-15°C a 70°C	-20°C a 80°C

Elaborado por: El Investigador basado en [56] [57]

1.3.17 Sistemas de Control

Un sistema de control se conforma de varios componentes eléctricos, electrónicos y de software que permiten realizar acciones controlados por parámetros ingresados en una unidad de procesamiento que se encarga de la toma de decisiones en base a un valor consigna (set point). Toda la información que proviene de los elementos que conforman el sistema se analizan y almacenan con la finalidad de tomar el control de actuadores que responden ante variaciones de variables que utiliza el sistema para su funcionamiento. [58]

Sistemas de lazo abierto

El sistema de lazo abierto consiste en el que la acción de control es en cierto modo, independiente de la salida. Este tipo de sistemas generalmente utilizan un regulador o actuador mecánico o eléctrico, con la finalidad de obtener la respuesta deseada a la salida del sistema. La capacidad que tales sistemas tienen para ejecutar una acción con exactitud depende de su calibración. Su funcionamiento se muestra a partir de diagramas de bloques en la Figura 8. [59]



Figura 8. Diagrama de bloques del Sistema de lazo abierto

Elaborado por: El Investigador basado en [59]

Sistemas de lazo cerrado

La acción de control depende del estado o valor de salida del sistema. El sistema de lazo abierto utiliza un elemento de medición (sensor) que detecta la respuesta real para compararla, con una señal de referencia en la entrada. Por esta razón, los sistemas de lazo cerrado se denominan sistemas retroalimentados. La importancia de estos sistemas es que la medición de salida del sistema se usa para regular la entrada del

sistema, de manera de conseguir que la variable controlada se mantenga próxima a un valor de referencia o “setpoint”. [60]

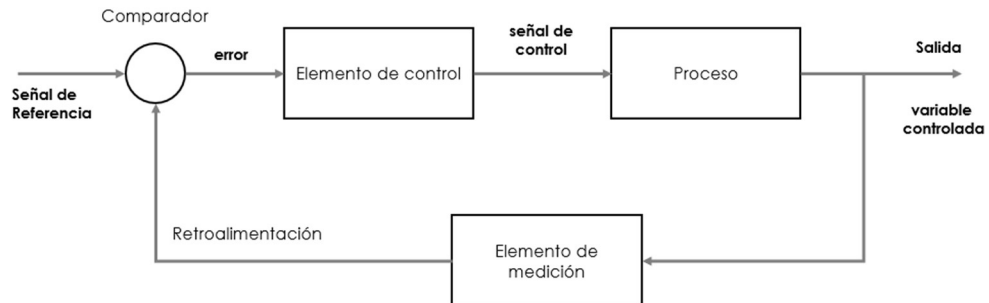


Figura 9. Diagrama de bloques del Sistema de lazo cerrado
Elaborado por: El Investigador basado en [60]

1.4 Objetivos

1.4.1 Objetivo General

Implementar un sistema domótico basado en IoT en el Centro Integral del Adulto Mayor de la Ciudad de Ambato.

1.4.2 Objetivos específicos

- Analizar el funcionamiento de los sistemas domóticos actuales, así como tecnologías de comunicación.
- Desarrollar el sistema IoT de control de condiciones ambientales y generación de notificaciones automáticas.
- Implementar los módulos de control de mando del sistema de asistencia de adultos mayores en el Centro Integral del Adulto Mayor.

CAPÍTULO II

METODOLOGÍA

2.1 Materiales

Para la elaboración del presente proyecto de investigación se requirió la utilización de diversos materiales como: información de artículos, tesis, libros, revistas, y documentación web relacionada a la temática del proyecto. Se requiere además de dispositivos electrónicos como actuadores, sensores, microcontroladores y un ordenador de placa reducida. Además de contar con las instalaciones designadas por el Municipio de Ambato para la instalación de los dispositivos.

2.2 Métodos

2.2.1 Modalidad de investigación

Investigación aplicada puesto que se aplicó conocimientos ya existentes, relacionados al análisis y solución de problemas asociados a la gestión técnica de asistencia en entornos para personas adultas mayores.

Investigación bibliográfica documentada que proporcionó la información teórica que ayudó al diseño del sistema domótico basado en IoT, dicha información fue recopilada de libros, artículos científicos, trabajos desarrollados a nivel nacional o internacional relacionados al tema de investigación.

Investigación experimental debido a que se evaluó la funcionalidad mediante una serie de pruebas los dispositivos electrónicos y de comunicación, así como los diversos componentes del sistema domótico.

Investigación de campo ya que la recolección de información y la determinación de los requerimientos técnicos se realizaron en el Centro Integral del Adulto Mayor, de esta forma se propone un proyecto que cumple con los objetivos planteados.

2.2.2 Recolección de información

Para la recolección de información se usaron medios físicos y digitales, libros, revistas, artículos científicos, proyectos similares desarrollados, manuales y guías técnicas orientados al diseño de sistemas domóticos e Internet de las cosas, para un adecuado desarrollo y ejecución del proyecto.

2.2.3 Procesamiento y análisis de datos

Para el procesamiento y análisis de datos se realizaron los siguientes pasos:

1. Revisión de la información recopilada.
2. Identificación de datos consistentes para aplicarlos en el desarrollo de la investigación.
3. Análisis de las características del sistema para realizar la implementación.
4. Determinación de la propuesta de solución más óptima.
5. Planteamiento de la propuesta de solución.

2.2.4 Desarrollo del proyecto

A continuación, se detallan las actividades llevadas a cabo para el desarrollo del proyecto:

1. Análisis y evaluación de los sistemas domóticos existentes en la actualidad.
2. Comparación de las tecnologías de comunicación inalámbrica y arquitecturas utilizadas en domótica.
3. Determinación de los requerimientos técnicos del sistema precisando las necesidades de adultos mayores.
4. Identificación de los recursos tecnológicos de red disponibles para el sistema IoT en el Centro Integral del Adulto Mayor (CIAM).
5. Identificación de las variables ambientales que pueden ser medidas y controladas dentro del sistema domótico.

6. Determinación de los sensores de medición y actuadores necesarios para el sistema IoT.
7. Elaboración y programación de las placas electrónicas de comunicación destinadas para el control o sonorización.
8. Selección del módulo electrónico para el reconocimiento de voz.
9. Selección del servidor en la nube y programación del software para la gestión de notificaciones a través de servicios de mensajería móvil.
10. Diseño de la interfaz gráfica para el control y monitoreo del sistema.
11. Implementación del sistema en las instalaciones del Centro Integral del Adulto Mayor para verificar su rendimiento.
12. Realización de pruebas para la detección y corrección de errores del sistema.
13. Validación del sistema domótico basado en IoT mediante pruebas por parte del personal a cargo del centro.
14. Elaboración del informe final del proyecto.

CAPITULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

La implementación de un sistema domótico basado en IoT ofrece la oportunidad de incorporar nuevas tecnologías destinadas al bienestar y asistencia de adultos mayores, con un sistema capaz de conectar dispositivos con un servidor en la nube, mismo que están destinados a mantener un ambiente óptimo para su cuidado. El sistema cuenta con dispositivos enlazados inalámbricamente a una arquitectura de red destinada a monitorear diversas variables del ambiente. También tiene la capacidad de reconocimiento de voz para el control de las instalaciones o mediante una interfaz gráfica, en conjunto con la generación de notificaciones automáticas ante situaciones adversas que registre el sistema.

3.2 Desarrollo de la propuesta

3.2.1 Beneficios de la Domótica

La domótica al combinar la electricidad, la electrónica y la informática destinados a la gestión técnica de espacios habitacionales tanto en casas o edificios, porta una serie de beneficios sobre los siguientes aspectos:

Confort: El hecho de que una serie de tareas repetitivas se puedan automatizar para que se realicen por sí solas proporciona un gran confort. Por ejemplo, encender las luces cuando es de noche.

Seguridad: Si se detecta gases químicos, el sistema domótico puede enviar notificaciones o, cuando detecta que un intruso ha accedido a la vivienda, el sistema domótico puede activar una alarma.

Ahorro energético. Cuando no hay nadie en la habitación, puedes ahorrar energía apagando las luces y apagando o reduciendo la calefacción.

Comunicaciones: Se puede acceder a los sistemas domóticos desde el exterior de la vivienda, y al integrar redes inalámbricas se pueden establecer conexiones entre diversos nodos. Por ejemplo, se puede ver a una persona llamando a la puerta de un apartamento en la televisión. [17]

3.2.2 Asistencia de adultos mayores

De acuerdo a las directrices del personal a cargo del centro y la revisión del marco teórico, los principales requerimientos para mantener una adecuada asistencia de las personas adultas mayores son las siguientes:

Sistema de Seguridad: La instalación de sensores que permitan conocer el estado de las condiciones ambientales, resulta de mucha importancia para mantenerse al tanto de la situación dentro del entorno habitacional de los adultos mayores. Así mismo se integra botones de asistencia en caso de requerir ayuda a causa de un accidente o la realización de una tarea específica. La generación de notificaciones para el sistema de seguridad tiene un papel importante ante escenarios de riesgo.

Control de Iluminación: Es necesario realizar la activación como desactivación de las luminarias instaladas dentro del área habitacional, tanto desde el mismo dispositivo o de forma remota, tanto de adultos mayores como personal encargado de su cuidado.

Control de Calefacción: Debido a que el centro se encuentra en la región andina y la mayor parte del año se presentan temperaturas bajas, se requiere un control para el encendido y apagado de equipos de calefacción, cuando la temperatura ambiente sea menor a 16°C, por tanto, está 4° menos al rango especificado por recomendación del personal del centro, relacionado a la sensación térmica.

Control de Ventilación: Para evitar la sofocación y mantener una temperatura fresca en el área habitacional en días que se presenten altas temperaturas, es conveniente un control para el encendido y apagado de equipos de ventilación cuando se supere los 28°C en el ambiente, este valor está 4° por encima del rango especificado, por recomendación del personal del centro.

A partir de estos estos lineamientos se desarrolla el diseño del sistema domótico basado en IoT para el Centro Integral del Adulto Mayor, en beneficio de las personas adultas mayores, bajo supervisión del personal que labora en las instalaciones.

3.2.3 Requerimientos del sistema domótico basado en IoT

Para el desarrollo del sistema domótico basado en IoT, que permita controlar las instalaciones y monitorizar variables del ambiente de adultos mayores, se plantearon los siguientes requerimientos:

- Comunicar cada uno de los dispositivos de control o sensorización del sistema con servidor en la nube.
- Generar las notificaciones automáticas a través de servicios de mensajería móvil de acuerdo a las condiciones ambientales o asistencia requerida.
- Almacenar los registros históricos de las alertas generadas por el sistema en servidor en la nube.
- Visualizar es estado de los dispositivos de control y las variables medidas mediante de los sensores en la interfaz web.

3.2.4 Identificación de variables a ser monitorizadas por el sistema

A partir de los requerimientos para la asistencia de adultos mayores, las variables más importantes a ser monitorizadas por el sistema dentro de su área habitacional son las siguientes:

Temperatura: Relacionada a la sensación térmica dentro del ambiente habitacional es importante mantener monitorizada esta variable para ejercer cambios en otros dispositivos.

Humedad: Medida como una unidad relativa, se monitorizará únicamente para determinar el porcentaje de humedad en el ambiente.

Humo: Mediante la medida de la calidad del aire, si existe algún elemento que ha comenzado quemarse y represente un riesgo para la vida de las personas.

Presencia o Movimiento: A través de la perturbación infrarroja, permite conocer si existe la presencia o no de personas dentro del área habitacional.

Estado del pulsador de asistencia: Al cambiar de estado, indica que se requiere la asistencia del personal que cuida de los adultos mayores ante la ocurrencia de algún incidente.

3.2.5 Identificación de variables a ser controladas por el sistema.

Iluminación: Controla el actuador para realizar el encendido y apagado de luces del área habitacional.

Calefactor: Para realizar el encendido y apagado de un equipo calefactor en dependencia de temperaturas bajas en el área habitacional.

Ventilador: Para realizar el encendido y apagado de un equipo ventilador, frente a temperaturas altas dentro de área habitacional.

3.2.6 Selección de sensores para el sistema domótico

Selección del Sensor de temperatura y humedad

Para poder adquirir datos físicos del ambiente relacionados con el ambiente, se ha seleccionado el sensor DHT-22 que se muestra en la Tabla 7, debido a la facilidad de adquirir este sensor en el mercado, además de brindar la estabilidad requerida para el sistema domótico que utiliza tecnología de recolección de señales digitales, asegurando su confiabilidad y estabilidad. Este sensor tiene un modelo de compensación de temperatura y está calibrado en una cámara de calibración que opera cuando el sensor está detectando las variaciones en la magnitud de humedad y temperatura.

Selección del sensor de humo (Co2)

Al tener en cuenta la seguridad de las instalaciones, se precisa agregar al sistema un sensor de humo o CO₂ (dióxido de Carbono), que pueda detectar las perturbaciones en el aire del ambiente en caso de presentarse una situación de riesgo relacionada a un incendio.

Para el desarrollo del proyecto se ha seleccionado el sensor MQ-135 mostrado en la Tabla 8, que resulta de fácil adquisición en el mercado y es de un precio relativamente bajo. Además, que cumple con el tipo de ajuste de sensibilidad y la salida de tipo digital

que requiere el sistema al detectar alguna perturbación en la calidad del aire o detección de humo.

Selección del sensor movimiento

Pese a que los sensores presentados en la Tabla 9 son muy parecidos en cuanto sus características, estos difieren en cuanto a la densidad ajustable, por lo que para el proyecto se selecciona el Sensor HC SR501. Este sensor es de un precio bajo por lo que es accesible su uso en el proyecto. La funcionalidad de este sensor se ve incrementada al disponer el ajuste de sensibilidad del sensor.

Pulsador de Asistencia

El pulsador o botón de asistencia es un elemento de acción momentánea, enviará la información acerca de su estado durante un tiempo determinado hacia el sistema. Al existir diversos modelos de pulsadores en el mercado, no se tomará en cuenta sus características, puesto que su única función es la de permitir o cortar el flujo de corriente en un sistema electrónico. El modelo que actualmente se puede encontrar en el mercado es el que se muestra en la Figura 10.



Figura 10. Pulsador de asistencia [61]

3.2.7 Actuador para el sistema domótico

Dentro del sistema se ha determinado tener el control de dispositivos eléctricos como luminarias, calefactor y ventilador, cuya alimentación de tensión es de 120V-AC. Se realiza la selección del actuador de tipo relé ya que sus características corresponden a los requerimientos del sistema. Su funcionamiento se complementa con la etapa de

potencia que se debe realizar en la utilización de este dispositivo electrónico. Sus principales características son:

- Voltaje de Alimentación 5V
- 2 pines de control
- Pines de Potencia 3 (COM, NO, NC)
- Dimensiones 19mm x15mm x15mm
- Rangos máximos 10Amp – 125V



Figura 11. Actuador tipo relé de 5V [62]

Las características de corriente y voltaje se toman en cuenta para el diseño de los módulos de control, por tanto, al conectar una carga que supere estos valores puede resultar en la avería inmediata de este dispositivo.

3.2.8 Diagrama de bloques del sistema

El diagrama de bloques del sistema domótico basado en IoT, está conformado por tres etapas las cuales permiten su óptimo funcionamiento. La primera etapa hace referencia al servidor en la nube, el cual se encarga de brindar la interfaz de usuario, almacenamiento de datos, la gestión de notificaciones y alojar un broker MQTT para la comunicación con los dispositivos. La segunda etapa tiene relación a los dispositivos IoT (nodos) que son los actuadores para el control de iluminación, ventilador o calefactor, así como los sensores para conocer las condiciones ambientales. La tercera etapa tiene relación con el control por voz a través del procesamiento en una Raspberry pi 4 que ejecuta comandos enviados hacia el servidor en la nube.

El sistema en conjunto tiene la capacidad de ser accesible desde cualquier computador o smartphone que cuente con una conexión a internet. Se han agregado credenciales de acceso a cada uno de los componentes del sistema para evitar alteraciones en la configuración y asegurar la comunicación entre los dispositivos. En la Figura 12 se visualiza el diagrama de bloques del sistema domótico basado IoT.

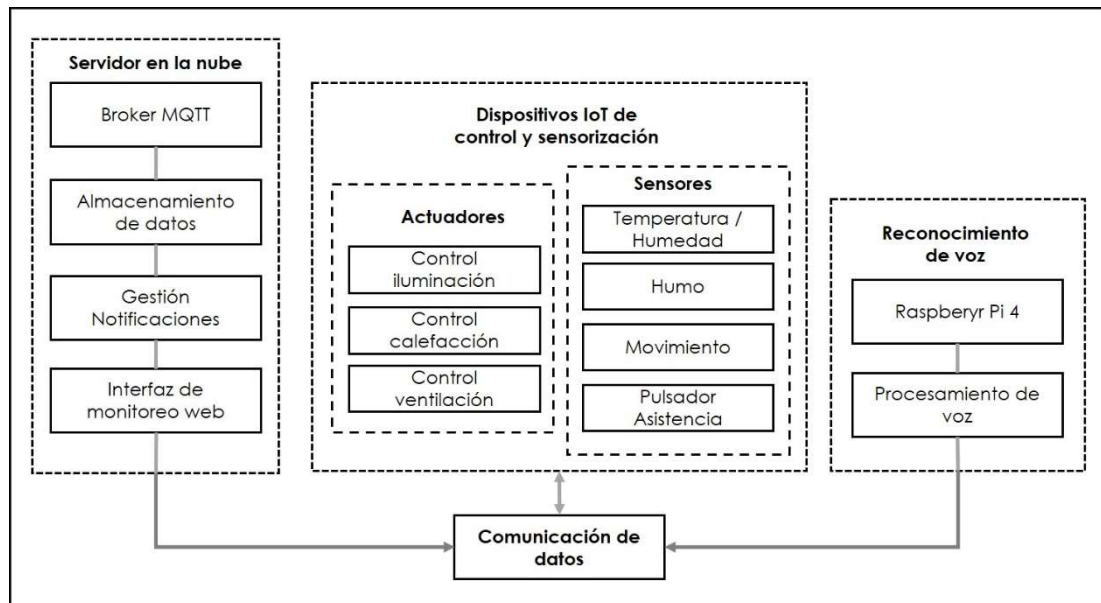


Figura 12. Diagrama de bloques del sistema domótico basado en IoT




Elaborado por: El Investigador

La arquitectura centralizada utilizada para el sistema domótico es la que mejor se adapta para su desarrollo, permite conectar los diferentes módulos directamente con el servidor en la nube al igual que el nodo de reconocimiento de voz. Brinda además la capacidad de implementar nuevos dispositivos a futuro en otras locaciones del centro.

3.2.9 Servidor en la nube

Alojar un servidor en la nube requiere de modelo tecnológico que permita el acceso bajo demanda a un conjunto compartido de recursos de computación (por ejemplo: redes, servidores, equipos de almacenamiento, servicios y aplicaciones), que pueden ser desplegados rápidamente y conseguir el acceso bajo una interacción mínima con el proveedor del servicio. [63]

Tabla 10. Empresas proveedoras de servicios en la nube [63] [64]

Proveedores	Microsoft Azure 	Amazon Web Services (AWS) 	Google Cloud  Google Cloud
Servicios líderes	Máquinas virtuales y SQL dedicado, Kubernetes, Contenedores, aplicaciones de machine Learning.	Computación elástica en la nube (EC2), Contenedores, Bases de datos y aplicaciones web	Compute Engine, analítica de datos, redes neuronales, IA y aprendizaje automático. API's

Elaborado por: El Investigador

De acuerdo a las áreas en las que son líderes cada una de estas empresas, se ha elegido hacer uso de los servicios de la empresa AWS, por el soporte existente de cada uso de sus servicios, especialmente en la computación elástica (EC2). Además de contar con la capa gratuita con una ampliación reciente a 12 meses de uso.

3.2.9.1 Implementación del servidor en la nube

Para poder crear el servidor en Amazon Web Services (AWS), es necesario crear una cuenta y registrar una tarjeta de débito o crédito, una vez validados estos datos es posible el uso de la plataforma Elastic Compute Cloud (EC2) donde se crea (genera) el servidor en la nube para el sistema domótico. Permitirá la comunicación de cada uno de los componentes del sistema, el monitoreo y control de dispositivos a través de una interfaz web, así como las notificaciones automáticas generadas por el sistema y registros de eventos en la base de datos.

Creación del Servidor en la nube - AWS EC2

Dentro de EC2 de AWS existen diversos Sistemas operativos y distribuciones de Linux, para ser virtualizados en la nube, denominados instancia. Para el proyecto se ha seleccionado una instancia de Ubuntu Server 22.04 LTS, en relación que esta distribución de Linux en la capa gratuita ofrece mayores prestaciones y el menor

consumo de operaciones de procesamiento. Además de contar con información actualizada acerca del manejo de este sistema operativo, y una amplia gama de repositorios de paquetes actualizados.

Se selecciona la instancia con mayores prestaciones dentro de la capa gratuita de AWS con 1Gb de memoria y 8Gb de almacenamiento. Al hacer uso de esta instancia, también se determina valor dado en dólares (USD) por concepto de uso cuando sobrepase la cuota de transferencia de datos o culmine la etapa de uso gratuito de 12 meses. Este dato será importante a futuro para un posterior análisis.

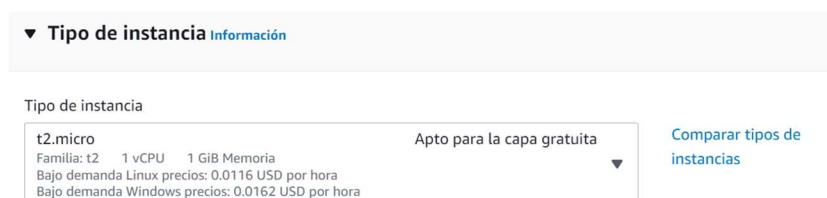


Figura 13. Detalles de la instancia creada en la capa gratuita de AWS.

Elaborado por: El Investigador

Asignación de IP pública elástica (estática)

Dentro del panel EC2 de AWS se procede a asignar una IP pública estática, debido a que los componentes del sistema requieren de una IP fija para no perder la conexión, además de evitar la actualización constante de la dirección IP si esta no posee dicha configuración.



Figura 14. Asignación de IP elástica al servidor en la nube.

Elaborado por: El Investigador

Una vez seleccionada la instancia dentro de la plataforma de AWS-console, se procede a asociar la dirección elástica, ésta se asigna de forma aleatoria por lo que ahora la dirección de la instancia es 54.148.93.152, a través de la cual se puede acceder desde internet.

Name	ID de la in...	Estado de la instancia	Tipo de inst...	Comprobación ...	E..	DNS de IPv4 púb...	Dirección IPv4 pública	IP elástica
server	i-013ce47...	En ejecución	t2.micro	2/2 comprobador	+	ec2-54-148-93-152...	54.148.93.152	54.148.93.152

Figura 15. Dirección IP publica y elástica del servidor en la nube

Elaborado por: El Investigador

Configuración de reglas de red para el servidor

Con la IP estática de la instancia, se procede a crear un grupo de seguridad con los parámetros correspondientes al tipo de conexión que se pretende tener y el listado de puertos que utiliza, para una comunicación óptima entre los diversos componentes del sistema. A continuación, se muestra las reglas establecidas para los puertos correspondientes.

ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
sgr-0fd6509e63b1572c4	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
sgr-084f1b7c58499165d	IPv4	TCP personalizado	TCP	1883	0.0.0.0/0	mqtt
sgr-00bd1fa9b14136af3	IPv4	TCP personalizado	TCP	21	0.0.0.0/0	ftp
sgr-070349aeb98fe0ad2	IPv4	TCP personalizado	TCP	1880	0.0.0.0/0	node-red
sgr-013b350d671112...	IPv4	SSH	TCP	22	0.0.0.0/0	-
sgr-0946e3ae1734e3d...	IPv4	TCP personalizado	TCP	12000 - 12100	0.0.0.0/0	aux-ports ftp
sgr-0d4962e761940a...	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Figura 16. Reglas de tráfico de datos del servidor en la nube

Elaborado por: El Investigador

Configuración de conexión remota SSH al servidor en la nube

Para establecer una conexión segura a través de SSH con el servidor en la nube, es necesario que dentro de las opciones de AWS se genere el par de claves, para el proyecto se selecciona la encriptación del tipo RSA, del cual se obtendrá el archivo de salida en formato “.pem”; este archivo nos permite acceder a la instancia vía ssh a través del puerto 22.

Nombre del par de claves

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves

RSA
Par de claves públicas y privadas cifradas por RSA

ED25519
Los pares de claves privadas y públicas cifradas ED25519 (no se admite para instancias de Windows)

Formato de archivo de clave privada

.pem
Para usar con OpenSSH

.ppk
Para usar con PuTTY

Cancelar **Crear par de claves**

Figura 17. Generación del par de claves para el servidor en la nube
Elaborado por: El Investigador

Con el archivo generado “llave.pem”, se procede a ubicarlo en el directorio y establecer la conexión a través de PowerShell con el comando “ssh -i "llave.pem" ubuntu@ec2-54-148-93-152.us-west-2.compute.amazonaws.com” tal como se muestra en la Figura 18:

```
PS C:\Users\Jonan\Desktop\llavessh> ssh -i "llave.pem" ubuntu@ec2-54-148-93-152.us-west-2.compute.amazonaws.com
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1013-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

System information as of Fri Mar 18 21:12:15 UTC 2022

System load:  0.0          Processes:    101
Usage of /:   39.3% of 7.58GB Users logged in: 0
Memory usage: 32%        IPv4 address for eth0: 172.31.13.89
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Jul 11 20:00:41 2022 from 186.71.213.102
ubuntu@ip-172-31-13-89:~$
```

Figura 18. Conexión con el servidor en la nube a través de SSH.
Elaborado por: El Investigador

Cada vez que se ingrese a la configuración del sistema Ubuntu, será necesario disponer del archivo “llave.pem”. Al tener el par de claves, se evita el acceso público desde cualquier otra máquina.

3.2.9.2 Instalación de LAMP en el servidor

Para almacenar los datos en la nube de registros históricos del sistema domótico, se realiza a la instalación de un servidor LAMP dentro de la instancia. Antes de iniciar la instalación se debe acceder a la terminal mediante SSH y ejecutar la actualización de repositorios y paquetes, mediante el siguiente comando:

```
sudo apt update && sudo apt upgrade -y
```

Una vez la actualización y comprobación de paquetes hayan finalizado, se procede a instalar el servidor LAMP, siguiendo los pasos que se detalla a continuación:

1. Instalar el Servicio Apache2 ejecutando el siguiente comando:

```
#apt install apache2 -y
```

Para comprobar correcto funcionamiento de Apache2 se accede desde cualquier navegador a la dirección del servidor: <http://54148.93.1527>.

2. Luego instalar el servidor MySQL y se asegura su instalación, ejecutando los siguientes comandos:

```
#apt install mysql-server mysql-client libmysqlclient-dev
```

```
#mysql_secure_installation
```

Se solicitará ingresar una contraseña para root y los atributos de ejecución.

3. Instalar PHP y sus librerías ejecutando el siguiente comando:

```
#apt install php libapache2-mod-php php-mysql
```

4. Instalar el gestor gráfico de base de datos phpMyAdmin mediante el siguiente comando:

```
#apt install phpmyadmin php-mbstring
```

5. Habilitar la extensión entre php y MySQLi ejecutando el comando:

```
#phpenmod mysqli
```

6. Se reinicia el servicio de Apache2:

```
#service apache2 restart
```

7. Acceder a la interfaz de phpMyAdmin, con la dirección ip del servidor en la nube **http://54148.93.1527/phpmyadmin** y comprobar la correcta instalación de la base de datos MySQL y funcionamiento.

Es necesario indicar que el directorio en la cual servidor en la nube, almacena los archivos referentes al diseño web, es “./var/www/html”, siendo la carpeta html a la cual se le han asignado los permisos de ejecución, lectura y escritura para ejecutarlos desde el cualquier navegador.

3.2.9.3 Broker Mosquitto en el servidor

El Servidor Mosquitto nos proporciona un intermediario entra las comunicaciones que se realizan bajo el protocolo MQTT, inicialmente se debe agregar los repositorios necesarios para su instalación mediante el siguiente comando:

```
#apt-add-repository ppa:mosquitto-dev/mosquitto-ppa -y
```

Una vez agregados los repositorios, se procede con la instalación de mosquitto y mosquitto clientes ejecutando el siguiente comando:

```
#apt install mosquitto mosquitto-clients -y
```

Se edita el archivo mosquitto.conf, bajo el directorio /etc/mosquitto/ y se agrega las siguientes líneas al final del archivo:

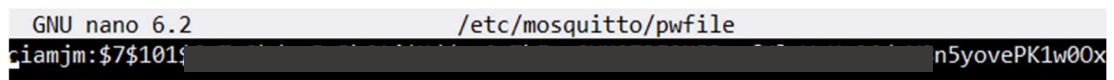
listener 1883

allow_anonymous false

Debido a que el broker está en una instancia accesible desde internet, resulta necesario establecer que toda conexión requiera autenticación con el broker Mosquitto, para ello, se ejecuta el siguiente comando, añadiendo al final un nombre de usuario:

```
#mosquitto_passwd -c /etc/mosquitto/pwfile jonathan
```

Luego de esta acción se solicita ingresar una contraseña, la cual se utiliza para establecer comunicación con el broker Mosquitto. Seguidamente, se revisa el contenido del archivo “pwfile”, que debe contener el nombre de usuario seguido de un hash generado a partir de la contraseña ingresada.



```
GNU nano 6.2 /etc/mosquitto/pwfile
ciamjm:$7$101$ n5yovePK1w00x
```

Figura 19. Autenticación al broker Mosquitto

Elaborado por: El Investigador

Luego de esta ejecución, es necesario reiniciar el servicio mosquitto y establecer que cada vez que se inicie el servidor, este servicio se active de forma automática, se lo realiza a través de los siguientes comandos:

```
#systemctl restart mosquitto
```

```
#systemctl enable mosquitto
```

3.2.9.4 Instalación de Node Red en el servidor

Node-Red es una interfaz basada en node.js, que proporciona los elementos necesarios para la gestión de sistemas IoT, a través de herramientas y paquetes que se añaden directamente desde su repositorio. Antes de comenzar con su instalación es necesario agregar el repositorio mediante el siguiente comando:

```
#curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

Una vez completada esta acción, se procede a la instalación de node red ejecutando el siguiente comando:

```
#apt-get install -y nodejs build-essential
```

Para que node red funcione de forma estable se asigna la configuración a pm2 que es el gestor de procesos de node.js a través del siguiente comando:

```
#npm install -g --unsafe-perm node-red pm2
```

Mediante pm2 se configura node red para que se active de forma automática cada vez que inicie el servidor en la nube, para ello, se ejecutan ordenadamente las siguientes líneas de código:

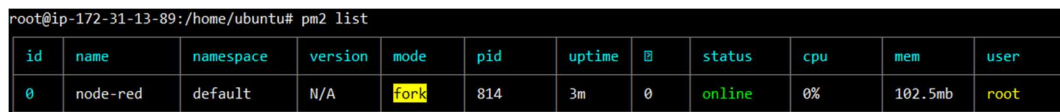
```
#pm2 start `which node-red` -- -v
```

```
#pm2 save
```

```
#pm2 startup
```

Mediante la consola se comprueba el estado de node red, visualizando la tabla pm2, ejecutando el siguiente comando:

```
#pm2 list
```



```
root@ip-172-31-13-89:/home/ubuntu# pm2 list
```

id	name	namespace	version	mode	pid	uptime	B	status	cpu	mem	user
0	node-red	default	N/A	fork	814	3m	0	online	0%	102.5mb	root

Figura 20. Node Red instalado en el servidor en la nube.

Elaborado por: El Investigador

Acceso con credenciales a Node Red

Al tener instalado Node-red en el servidor en la nube, es necesario agregar las seguridades correspondientes como usuario y contraseña para evitar daños en la configuración de los nodos. Para generar las claves cifradas en node red, se lo realiza a través del siguiente comando:

```
#node-red admin hash-pw
```

```
usuario y clave para NodeRed
root@ip-172-31-13-89:/home/ubuntu# node-red admin hash-pw
Password:
$2b$08$[redacted]vQftSEYjm
root@ip-172-31-13-89:/home/ubuntu#
```

Figura 21. Generación de contraseña cifrada para node red.

Elaborado por: El Investigador

Se asigna un usuario y contraseña para acceder a panel inicial de node red, se debe modificar los parámetros del archivo “settings.js” dentro del directorio `./node-red/` y sustituir la nueva contraseña cifrada:

```
GNU nano 6.2 .node-red/settings.js *
adminAuth: {
  type: "credentials",
  users: [{
    username: "jonathanfisei",
    password: "$2b$08$[redacted]vQftSEYjm",
    permissions: "*"
  }],
},
```

Figura 22. Credenciales de acceso al panel principal de node red.

Elaborado por: El Investigador

Dentro del mismo archivo, se modifica y se asigna un usuario y contraseña para acceder a cualquier página web generada mediante node red, tal como se muestra a continuación:

```
GNU nano 6.2 .node-red/settings.js
*/
httpNodeAuth: {user:"gadma",pass:"$2b$08$[redacted]3h7j6"},
//httpStaticAuth: {user:"user",pass:"$2a$08$zZwTXtja0fB1pzD4sHCMYOCMYz2Z6dNbM6t18sJogENOMcxkV9DN."},
```

Figura 23. Credenciales de acceso a páginas web de node red.

Elaborado por: El Investigador

3.2.10 Elaboración de los módulos electrónicos

Las placas electrónicas tanto para sensorizar las variables del ambiente como para el control, se definen como módulos dentro del sistema domótico, además que a partir de este punto pueden ser llamados dispositivos IoT. El diseño de cada una de los prototipos se realiza a partir de los requerimientos del sistema, además se utilizan los sensores y actuadores seleccionados en los apartados 3.2.6 y 3.2.7 respectivamente. Se utiliza el software Proteus 8.8 para el diseño de los circuitos, distribución de componentes y diagramas de las placas PCB.

Cada módulo debe contener su armazón o caja de protección (case), dentro del cual se ubica la placa electrónica con los diferentes elementos que la componen. Se ha optado por utilizar la plataforma de desarrollo gratuita de Autodesk TinkerCad, con la finalidad de diseñar los modelos de case 3D para cada uno de los módulos.

Todas las placas electrónicas hacen uso del transformador HiLink modelo HLK-PM01 que a su entrada se conecta 120VAC y a la salida brinda 5VDC – 0.6Amp, necesario para el funcionamiento de los componentes del circuito. La salida del HLK-PM01 se conecta inicialmente en paralelo a un capacitor electrolítico de 200uF, evitando de esta forma sobretensiones en el circuito. Luego pasa a energizar completamente al circuito, y para separar las fuentes al momento de cargar un programa a la ESP-32 se instala un Dip Switch de dos vías, anterior a los terminales de 5V y GND respectivos. Esta especificación se realiza para evitar redundancia en la descripción de cada uno de las placas electrónicas elaboradas.

3.2.10.1 Módulo de temperatura y humedad

Diseño del Circuito

El diseño del circuito encargado de tomar los datos relacionados a la temperatura y humedad del ambiente, se compone de la fuente de alimentación, el microcontrolador y 3 espadines tipo hembra a los cuales se conectará el sensor. El microcontrolador

ESP-32 se encarga a través del pin GPIO 12 de recibir de los datos provenientes el sensor DHT-22. El circuito esquemático se muestra en la figura 24.

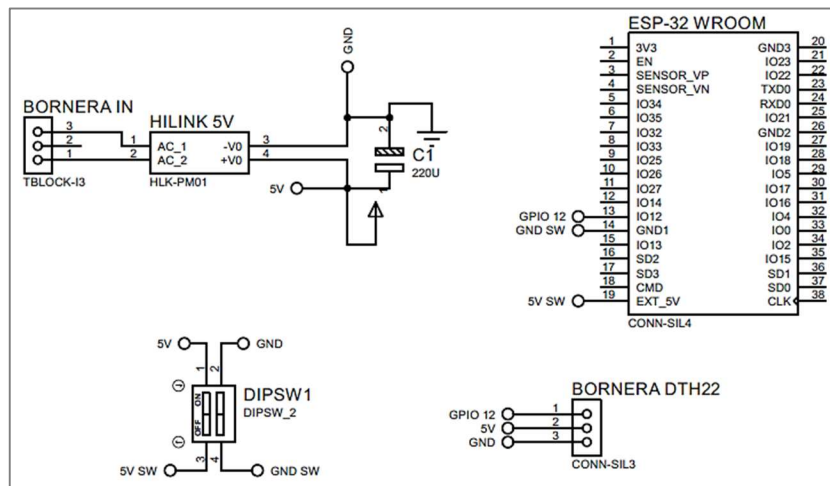


Figura 24. Circuito esquemático del módulo de temperatura y humedad.

Elaborado por: El Investigador

Distribución de los componentes electrónicos

A partir del circuito esquemático mostrado en la Figura 24, se procede a la ubicación de los componentes electrónicos en la placa, siendo la medida de 10cm x 5cm, el tamaño final de la placa. A continuación, se muestra la distribución de componentes y el trazado de pistas en el diagrama PCB.

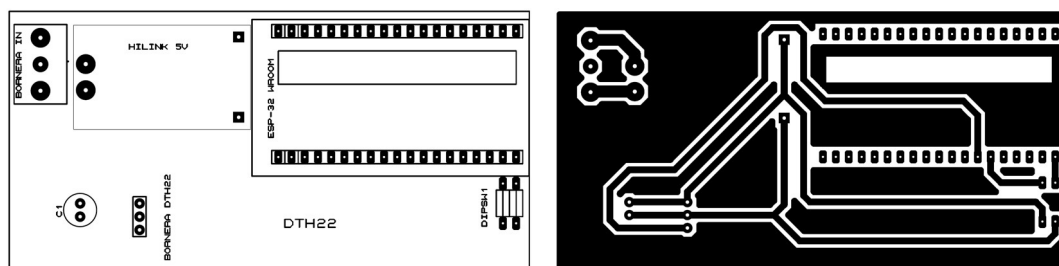


Figura 25. Distribución de elementos y diagrama PCB del módulo de temperatura y humedad

Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB que se ha realizado y las pistas que tiene el circuito, se lo traspasa hacia una baquelita de capa de cobre en fibra de vidrio para un menor desgaste en el proceso de elaboración. Se procede a perforar de acuerdo a la disposición de pines de los componentes y se sueldan respectivamente.

El sensor DHT22 se mantiene conectado a los espadines hembra de forma cableada debido a que posteriormente se lo ubicará dentro de una case. La Figura 26 muestra el resultado final de la elaboración del módulo de temperatura y humedad.

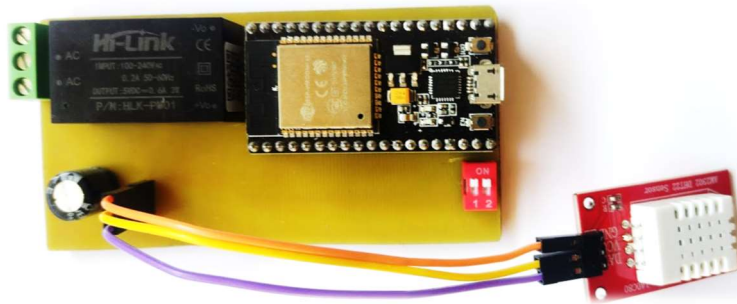


Figura 26. Montaje de componentes del módulo de temperatura y humedad
Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa y del sensor DHT-22 se procede con el diseño 3D de la caja que los contiene, la Figura 27 muestra el diseño del armazón para el módulo de temperatura y humedad. Las dimensiones reales se detallan en el Anexo A.

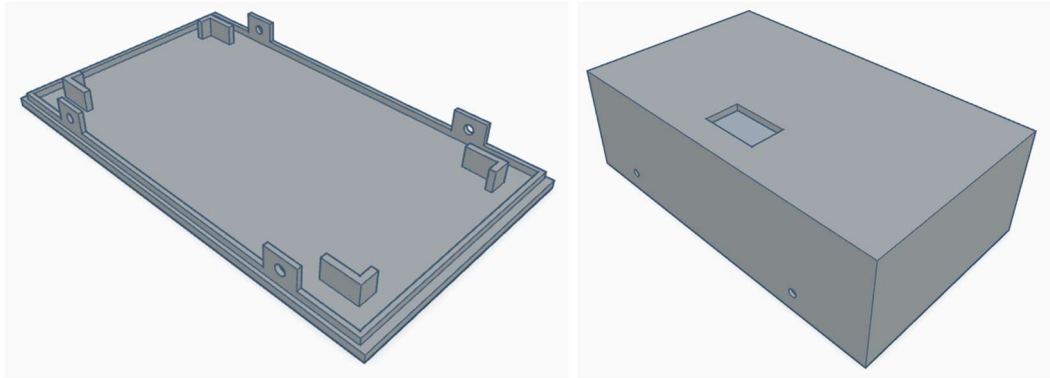


Figura 27. Caja contenedora del módulo de temperatura y humedad
Elaborado por: El Investigador

3.2.10.2 Módulo detector de humo

Diseño del Circuito

El diseño del circuito encargado de tomar los datos relacionados a la detección de humo (CO₂), se compone de la fuente de alimentación, el microcontrolador y 4 espadines tipo hembra a los cuales se conectará el sensor MQ-135. El microcontrolador ESP-32 se encarga a través del pin GPIO 14 de recibir de la señal de voltaje digital que indique la presencia de humo en el ambiente. El circuito esquemático se muestra en la Figura 28.

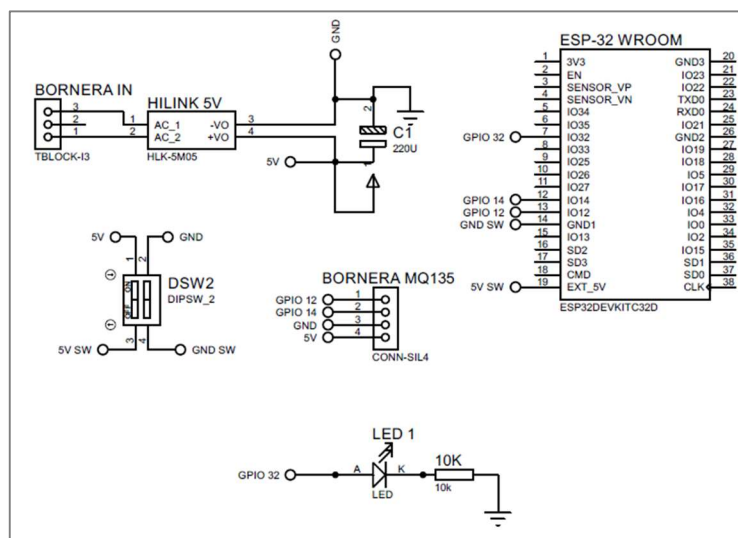


Figura 28. Circuito esquemático del módulo de detección de humo (CO₂).
Elaborado por: El Investigador

Distribución de los componentes electrónicos

A partir del circuito esquemático mostrado en la Figura 28, se procede a la colocación de los componentes electrónicos en la placa, siendo la medida de 10cm x 5cm, el tamaño final de la placa. A continuación, se muestra la distribución de componentes y el trazado de pistas en el diagrama PCB.

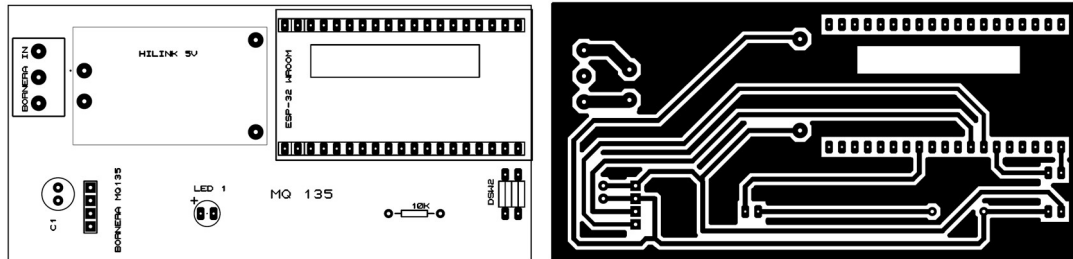


Figura 29. Distribución de elementos y diagrama PCB del módulo de detección de humo (CO₂).

Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB que se ha diseñado y las pistas que tiene el circuito, se lo plasma hacia la baquelita. Se procede a realizar las perforaciones de acuerdo a la disposición de pines de los componentes y se sueldan respectivamente. El sensor MQ-135 se mantiene conectado a los 4 espadines hembra de forma cableada debido a que posteriormente se lo ubicará dentro de una case. La Figura 33 muestra el resultado final de la elaboración del módulo de detección de CO₂.

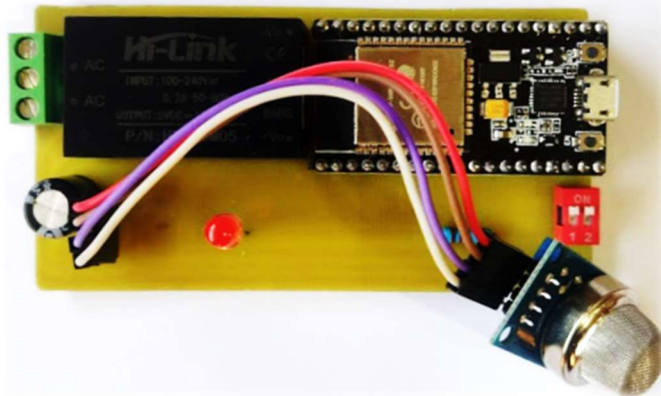
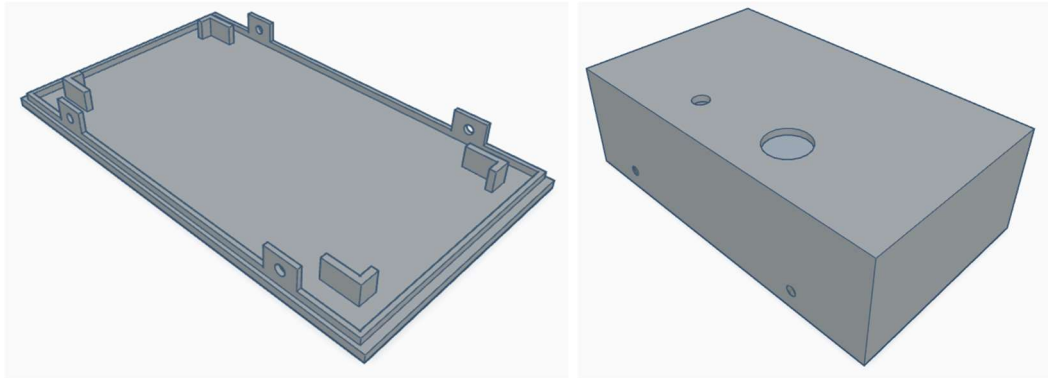


Figura 30. Montaje de componentes del módulo de detección de humo.

Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa y del sensor MQ135, se procede con el diseño 3D de la caja que los contiene, la Figura 31 muestra el diseño del armazón para el módulo detector de humo. Las dimensiones reales se detallan en el Anexo B.



**Figura 31. Caja contenedora del módulo de detección de humo
Elaborado por el Investigador**

3.2.10.3 Módulo detector de movimiento

Diseño del Circuito

El diseño del circuito destinado a la detección de movimiento o presencia, se compone de la fuente de alimentación, el microcontrolador y 4 espadines tipo hembra a los cuales se conecta el sensor HC SR501. El microcontrolador ESP-32 se encarga a través del pin GPIO 34 de recibir de la señal de voltaje digital que proviene desde el sensor e indique el movimiento o presencia de personas, se añade un indicador led cada vez que se dispare el sensor. El circuito esquemático se muestra en la figura 35.

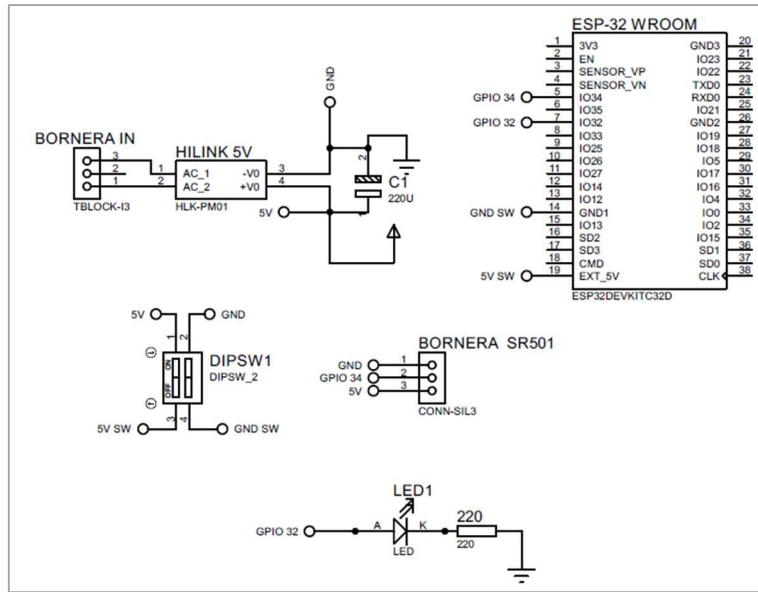


Figura 32. Circuito esquemático del módulo de detección de movimiento.
Elaborado por: El Investigador

Distribución de los componentes electrónicos

A partir del circuito esquemático mostrado en la Figura 35, se colocan respectivamente todos los componentes electrónicos en la placa, siendo la medida de 10cm x 5cm, el tamaño final de la placa para este módulo. A continuación, se muestra la distribución de componentes y el trazado de pistas en el diagrama PCB.

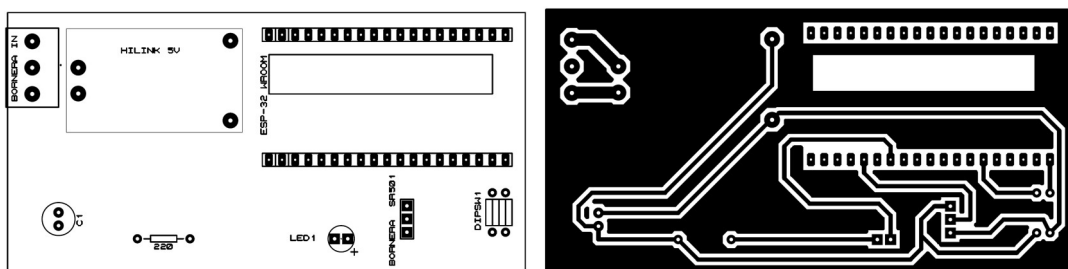


Figura 33. Distribución de elementos y diagrama PCB del módulo detector de movimiento.
Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB que se ha realizado y las pistas que tiene el circuito mostrado en la Figura 33, se procede a realizar las perforaciones de acuerdo a la disposición de pines de los componentes y se sueldan respectivamente cada uno de éstos.

El sensor HC SR501 se mantiene conectado a los 3 espadines hembra de forma cableada debido a que posteriormente se ubicará dentro de una case de acuerdo a sus medidas. La Figura 34 muestra el resultado de la elaboración del módulo de detección de movimiento.



Figura 34. Montaje de componentes del módulo detector de movimiento.

Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa y del sensor MQ135, se procede con el diseño 3D de la caja que los contiene, la Figura 38 muestra el diseño de la caja rectangular, para el módulo detector de humo. Las dimensiones reales se detallan en el Anexo C.

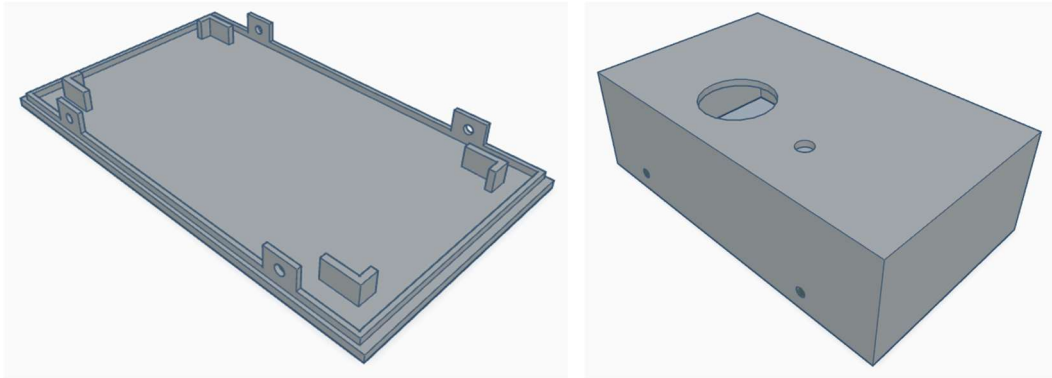


Figura 35. Caja contenedora del módulo de detección de humo
Elaborado por: el Investigador

3.2.10.4 Módulo de Asistencia

Diseño del circuito

El diseño del circuito que posee el pulsador o botón de asistencia, se compone de la fuente de alimentación, el microcontrolador, 1 bornera doble y leds indicadores. El pulsador posee 3 terminales que son: común (COM), normalmente abierto (NO) y normalmente cerrado (NC), de los cuales en la bornera se conectan los terminales COM y NO del pulsador, para que el microcontrolador detecte el voltaje suministrado al pin GPIO 34, cuando éste sea presionado. El circuito esquemático de este módulo se muestra en la Figura 36.

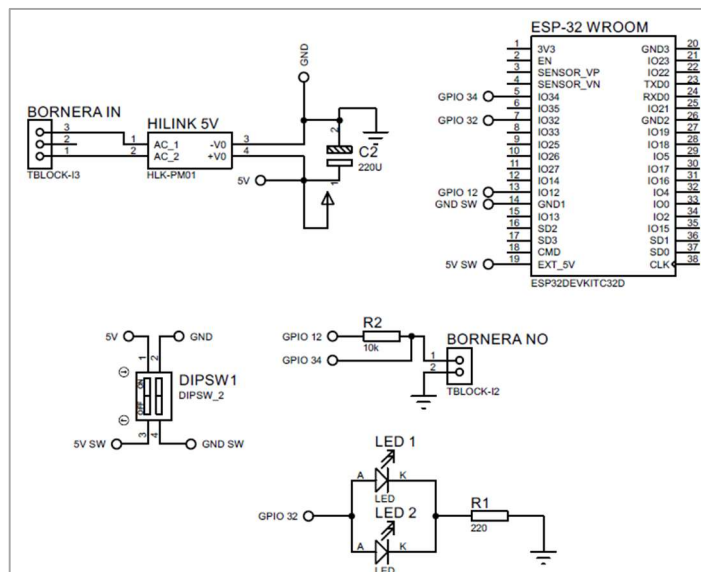


Figura 36. Circuito esquemático del módulo de asistencia
Elaborado por: El Investigador

Distribución de los componentes electrónicos

Conforme al circuito de la Figura 36, se procede a la ubicación óptima de los componentes electrónicos en la placa, resultando ser la medida de la placa de 11cm x 5cm. A continuación, se muestra la distribución de los componentes y el trazado de pistas en el diagrama PCB.

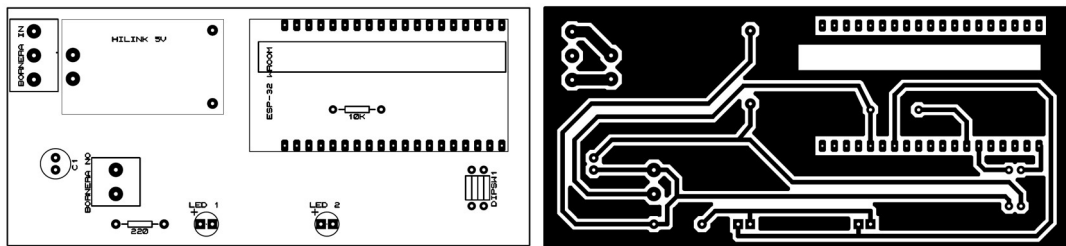


Figura 37. Distribución de elementos y diagrama PCB del módulo de asistencia

Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB que se ha realizado y las pistas del circuito, se procede a perforar en las ubicaciones correspondientes a cada componente, posteriormente se suelda cada uno de los pines en la baquelita. La Figura 38 muestra el resultado final de la elaboración del módulo de asistencia.

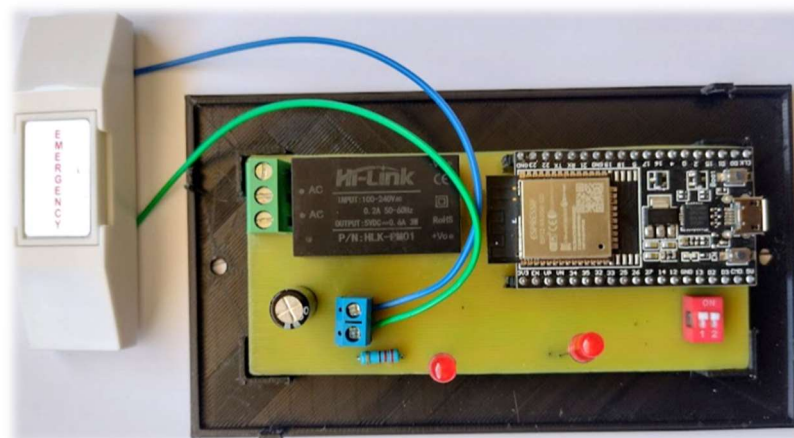
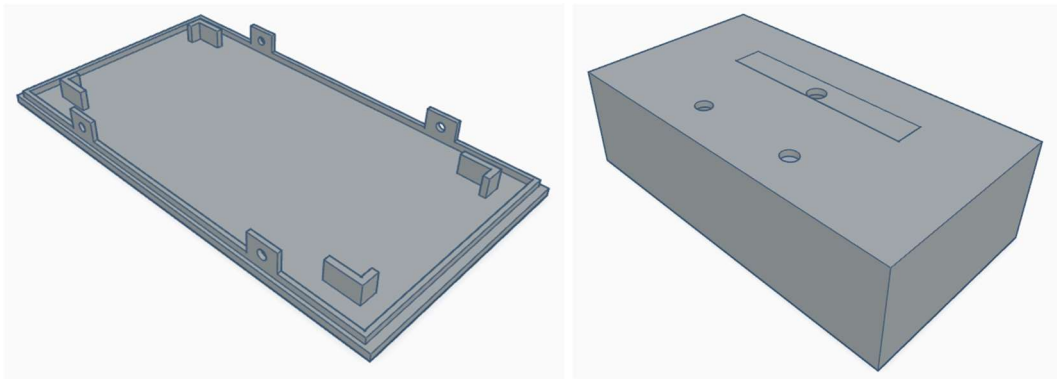


Figura 38. Montaje de componentes del módulo detector de movimiento.

Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa y del sensor DHT-22 se procede con el diseño 3D de la caja que los contiene, la Figura 42 muestra el diseño del armazón para el módulo de asistencia del cual sobresale un pulsador. Las dimensiones reales se detallan en el Anexo D.



**Figura 39. Caja contenedora del módulo de asistencia
Elaborado por el Investigador**

3.2.10.5 Módulo de control de iluminación

Diseño del circuito

El diseño del circuito para el control de encendido y apagado de luces, se compone de la fuente de alimentación, el microcontrolador y circuitos de potencia que incluyen 3 relés NO y NC, botones y circuitos integrados tipo latch. El circuito debe responder ante las acciones manuales que se realicen, así como la ejecución de funciones de forma remota. Su diseño se muestra en la Figura 40.

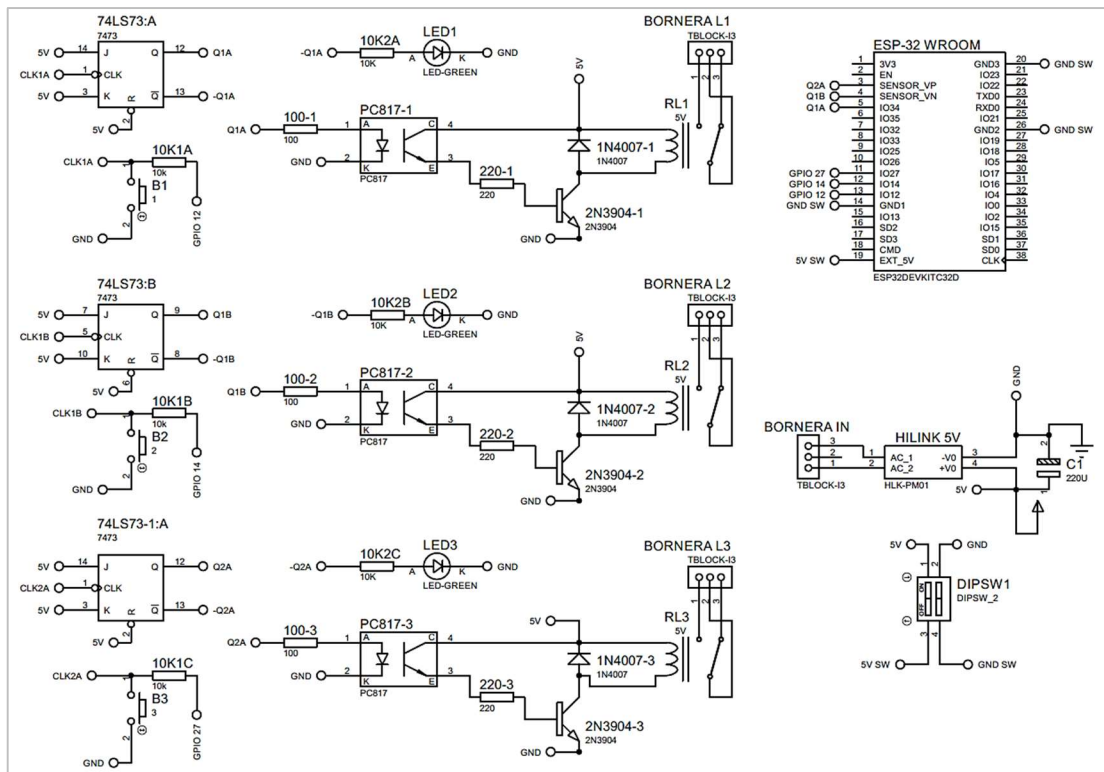


Figura 40. Circuito esquemático para control de iluminación

Elaborado por: El Investigador

El circuito de control de iluminación, consta de botones para el control manual tanto del encendido o apagado de luces, agrega elementos biestables o latch de tipo J-K (CI-74LS73) que mantienen el estado a la salida en 0 o 1. Una vez se presione el botón, éste genera un pulso en la señal de reloj que cambiará el estado de la salida. Los pines GPIO 12, 14, 2; pertenecientes a la ESP-32 también pueden ejercer un cambio en la señal de reloj de los biestables.

Puesto que se requiere mantener una lectura constante del estado de las salidas de los biestables, se implementa un optoacoplador PC817 previo al circuito de potencia, de esta forma se evita una caída de tensión que impide el cambio a la salida de los biestables y se puede realizar la lectura mediante su conexión hacia los pines GPIO 34, 39 y 36.

Para el circuito de potencia, éste consta de transistores 2N3904 de tipo NPN, empleados para activar las bobinas de los relés de 5V. Su operación en esta sección del circuito se basa en utilizar el nivel de voltaje suministrado por el optoacoplador,

de esta manera supera la tensión de umbral de la base del transistor que es de 0.6 V, permitiendo el paso de corriente entre las terminales de la bobina del relé, activándolo y conmutando sus terminales de COM-NO a COM-NC. Para evitar picos de tensión que afecten a los componentes electrónicos del circuito, se conecta a la bobina del relé un diodo 1N4007 en su configuración inversamente polarizado que ayuda a controlar sobre picos de voltaje.

Distribución de los componentes electrónicos

Conforme al circuito de la Figura 40, se procede a la ubicación óptima de los componentes electrónicos en la placa, con el fin de priorizar un reducido tamaño frente al número de conexiones que tiene este circuito. La medida de la placa es 10cm x 10cm. A continuación, La distribución de los componentes y el trazado de pistas en el diagrama PCB se muestran en la Figura 41 y 42 respectivamente.

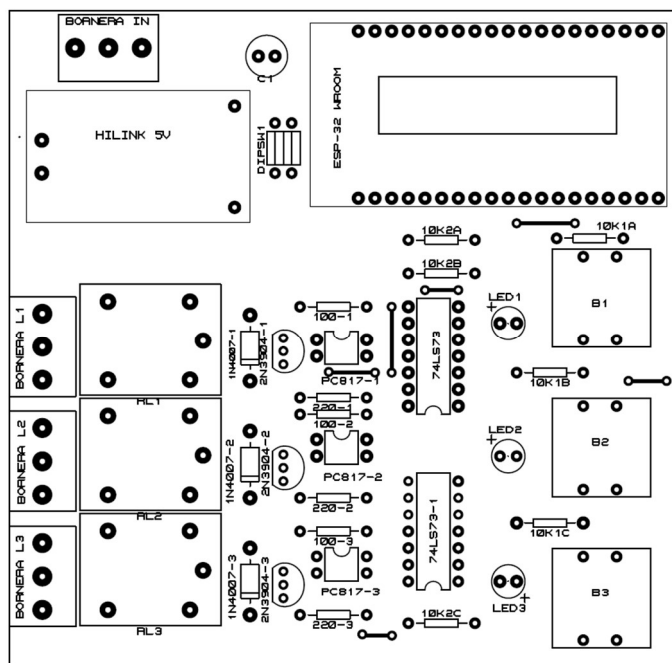


Figura 41. Distribución de componentes del módulo de control de iluminación

Elaborado por: El Investigador

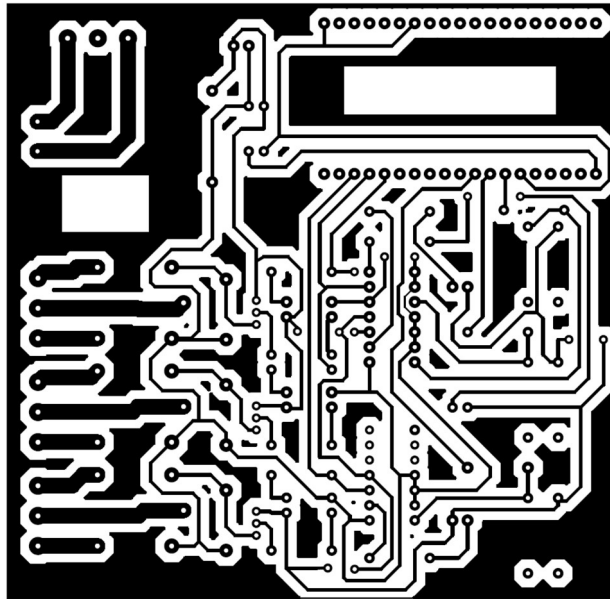


Figura 42. Diagrama PCB del módulo de control de iluminación

Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB de la Figura 42 transferido a la baquelita, se procede a perforar en las ubicaciones correspondientes de cada componente electrónico, posteriormente se suelda cada uno de los pines con su elemento. La Figura 43 muestra el resultado final de la elaboración del módulo de control de iluminación.

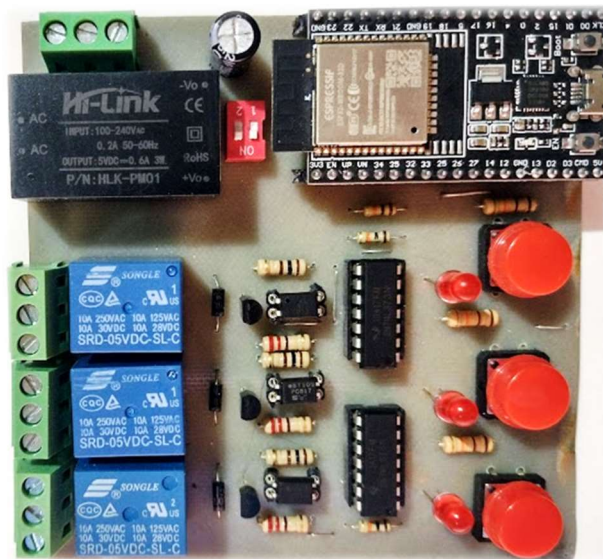


Figura 43. Montaje de componentes del módulo de control de iluminación.

Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa junto con todos los elementos presentados en la Figura 43, se procede con el diseño 3D de la caja, esta posee una geometría irregular por cuanto la altura de los botones y leds no se encuentran al mismo nivel que el resto de componentes., la Figura 44 muestra el diseño de la caja para el módulo de control de iluminación. Las dimensiones reales se detallan en el Anexo E.

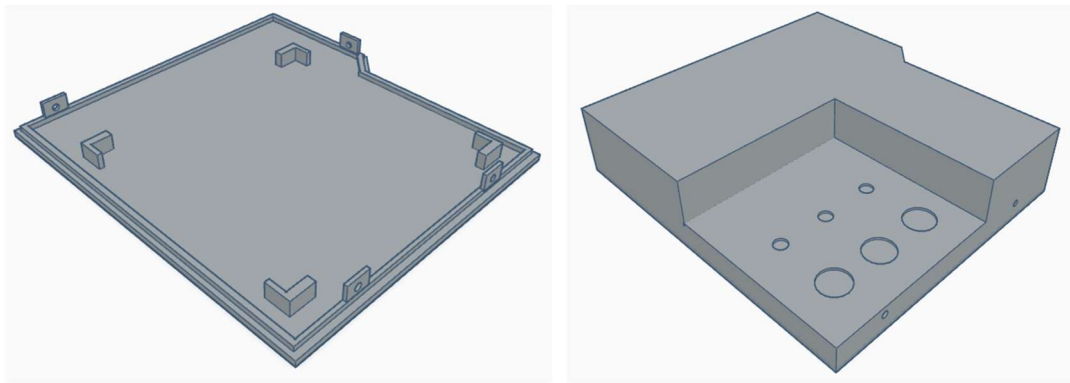


Figura 44. Caja contenedora del módulo de control de iluminación
Elaborado por: el Investigador

3.2.10.6 Módulos de control del calefactor y ventilador

Diseño del circuito

El diseño del circuito para el control de encendido y apagado del calefactor, se compone de la fuente de alimentación, el microcontrolador ESP-32 y circuitos de potencia que incluyen cada uno un relé NO y NC, un botón y un circuito integrado tipo latch. El circuito debe responder ante las acciones manuales que se realicen, así como la ejecución de funciones de forma remota. Su diseño es muy similar al Módulo de control de iluminación del punto 3.2.10.5. El circuito esquemático se muestra en la Figura 45.

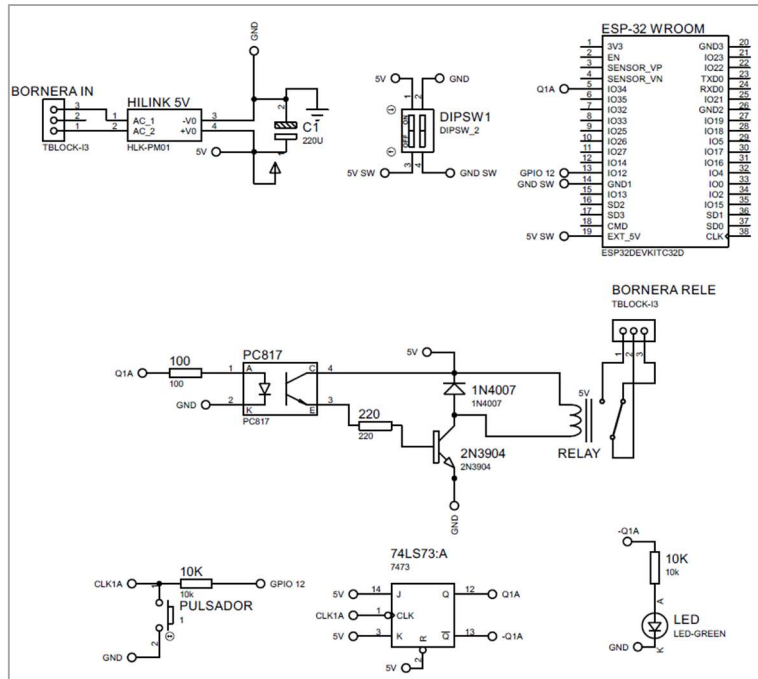


Figura 45. Circuito esquemático para Módulo de control de calefactor y ventilador
Elaborado por: El Investigador

Al analizar el circuito de la Figura 45, posee las características del botón para el envío de la señal de reloj hacia el biestable, o desde el GPIO 12. Para conocer el estado de la salida del relé se lo hace a través del optoacoplador PC817 con conexión al GPIO 43 del microcontrolador ESP32 respectivamente. Comparte la misma característica del circuito de potencia para el control de la activación o desactivación del relé revisados para el módulo de control de iluminación para evitar sobre picos de voltaje.

Distribución de los componentes electrónicos

Basado en el circuito esquemático de la Figura 45, se procede a la ubicación óptima de los componentes electrónicos, alcanzando un reducido tamaño frente al número de conexiones del circuito. La medida de la placa es 10cm x 6cm. La distribución de los componentes y el trazado de pistas en el diagrama PCB se muestran en la Figura 46.

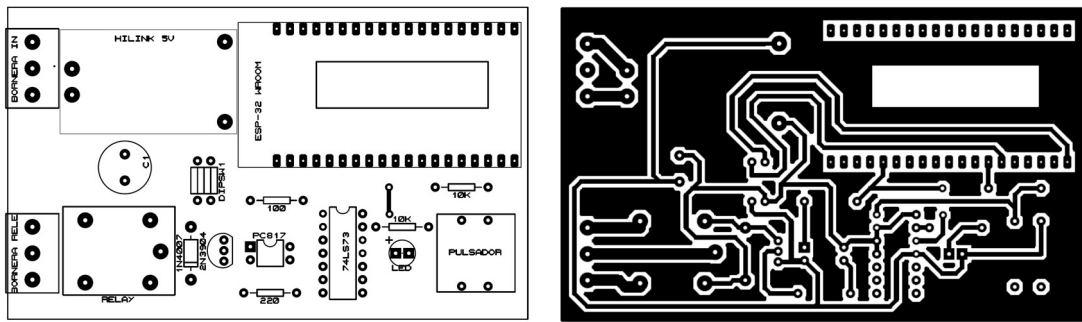


Figura 46. Distribución de elementos y diagrama PCB

Elaborado por: El Investigador

Elaboración de la placa

Con el diagrama PCB que se ha realizado y las pistas que tiene el circuito mostrado en la Figura 46, se procede a realizar perforaciones en dependencia de la disposición de los pines de los componentes y se sueldan respectivamente cada uno de ellos.

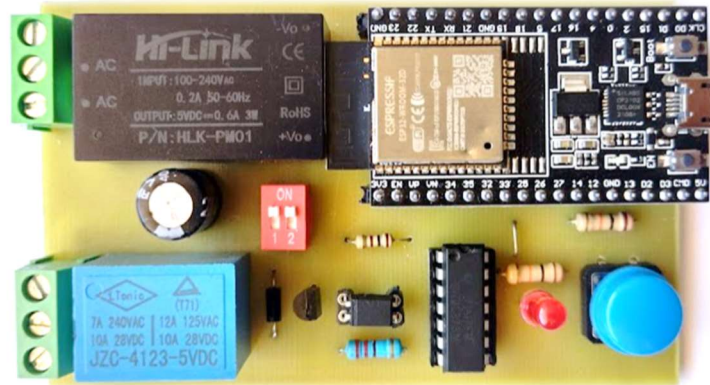


Figura 47. Montaje de componentes del módulo detector de movimiento.

Elaborado por: El Investigador

Diseño de la caja

Para la colocación de la placa presentada en la Figura 47, se procede con el diseño 3D de la forma de la caja, esta posee una geometría irregular por cuanto la altura de los botones y leds no se encuentran al mismo nivel que el resto de componentes, la Figura 48 muestra su diseño. Las dimensiones reales se detallan en el Anexo F.

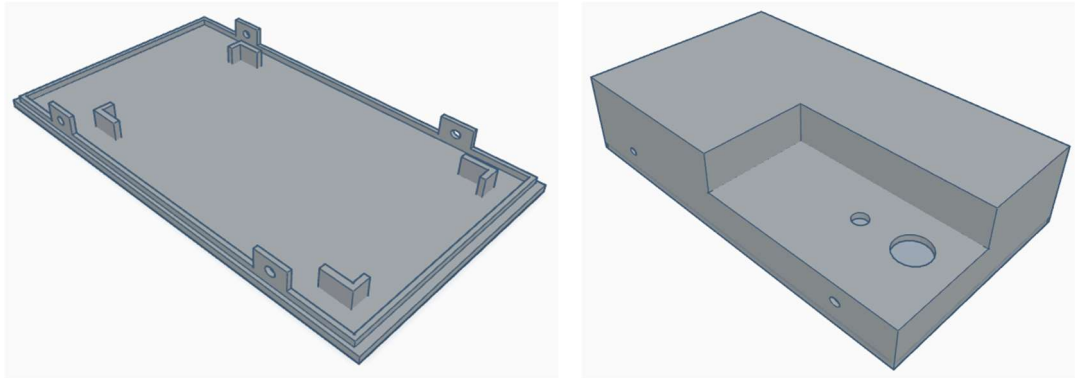


Figura 48. Caja contenedora del módulo de control de calefactor y ventilador
Elaborado por: el Investigador

3.2.11 Programación de los módulos

La programación de cada uno de los módulos, se ha llevado a cabo mediante el uso de software de uso libre Arduino IDE, incluyendo las librerías necesarias para la configuración de los microcontroladores ESP-32 y los sensores seleccionados.

El uso la librería “PubSubClient.h”, permite establecer la comunicación con el broker MQTT instalado en el servidor en la nube, sin embargo, tiene la limitante de publicar los mensajes en el nivel más bajo de QoS0. La configuración de SSID y contraseña es la misma para todos los dispositivos al trabajar dentro de una misma red wifi.

Módulo de temperatura y humedad

La programación del módulo establece la adquisición del valor de la temperatura y humedad del sensor DHT-22, mismos que se leen en el pin GPIO 12, realizando esta acción cada 5 segundos. A partir de estas dos medidas, el módulo publica hacia el broker MQTT en los topics correspondientes a “ciam/hab1/temperatura” y “ciam/hab1/humedad”. El código completo de este módulo se encuentra en el Anexo G, a continuación, se muestra el diagrama de flujo.

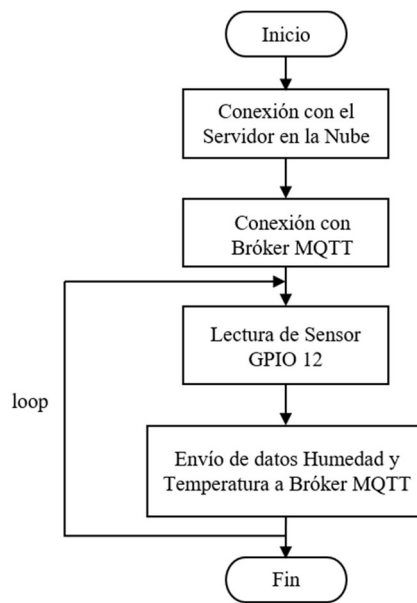


Figura 49. Diagrama de flujo del módulo de temperatura y humedad

Elaborado por: El Investigador

Módulo detector de humo

La programación del módulo detector de humo, cumple la función de realizar la lectura permanente del estado del pin D0 del sensor MQ135 en el pin GPIO 14. Si la lectura arroja un flanco bajo, el módulo activa un led indicador de forma intermitente durante 1 minuto y publica hacia el broker el valor “1”, caso contrario envía “0” en el topic correspondiente a “ciam/hab1/humo”. El código completo de este módulo se encuentra en el Anexo H, a continuación, se muestra el diagrama de flujo.

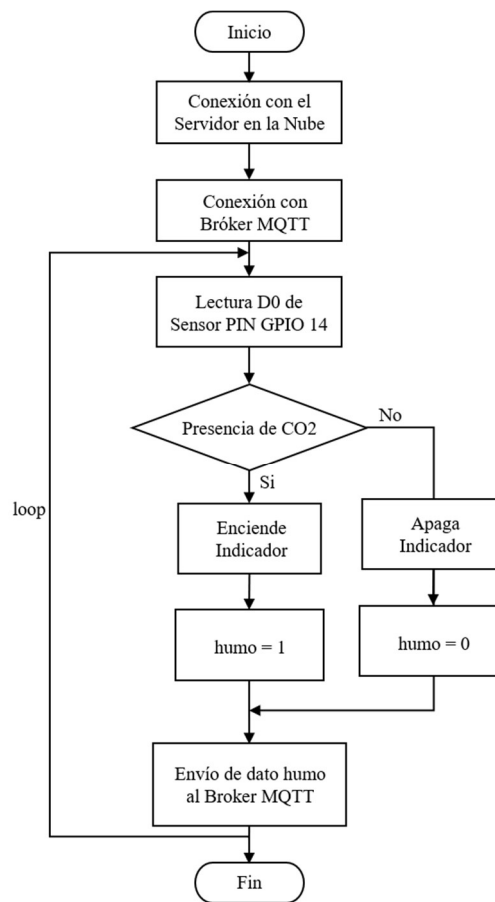


Figura 50. Diagrama de flujo del módulo detector de humo
Elaborado por: El Investigador

Módulo detector de movimiento

La programación de este módulo realiza la lectura de un flanco en alto proveniente del sensor HC SR501, al detectar el valor 1, el módulo activa un led indicador intermitente durante 10 segundos y publica hacia el broker el valor “1”, caso contrario envía “0” en el topic “ciam/hab1/movimiento”. El código completo de este módulo se encuentra en el Anexo I, a continuación, se muestra el diagrama de flujo.

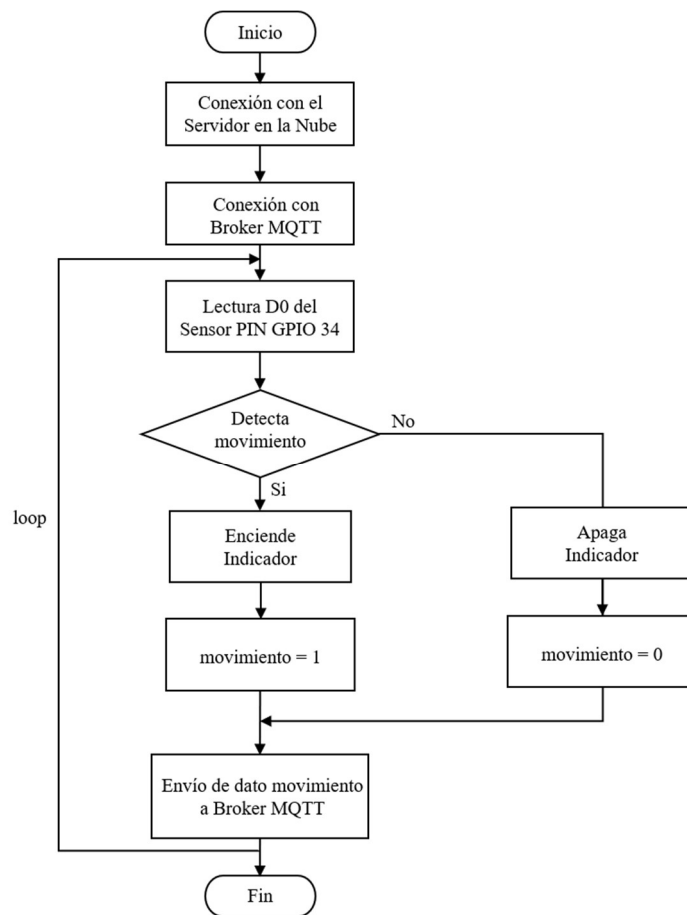


Figura 51. Diagrama de flujo del módulo detector de movimiento.

Elaborado por: El Investigador

Módulo de Asistencia

La programación del módulo de asistencia, detecta el cambio del valor de 5V a 0V. Este cambio indica que se ha presionado el pulsador, seguidamente se publica en el topic “ciam/hab1/asistencia1” el valor “1” y los leds se encienden intermitentemente durante 60 segundos. De no existir cambios se enviará “0”. El código completo de este módulo se encuentra en el Anexo J, a continuación, se muestra el diagrama de flujo.

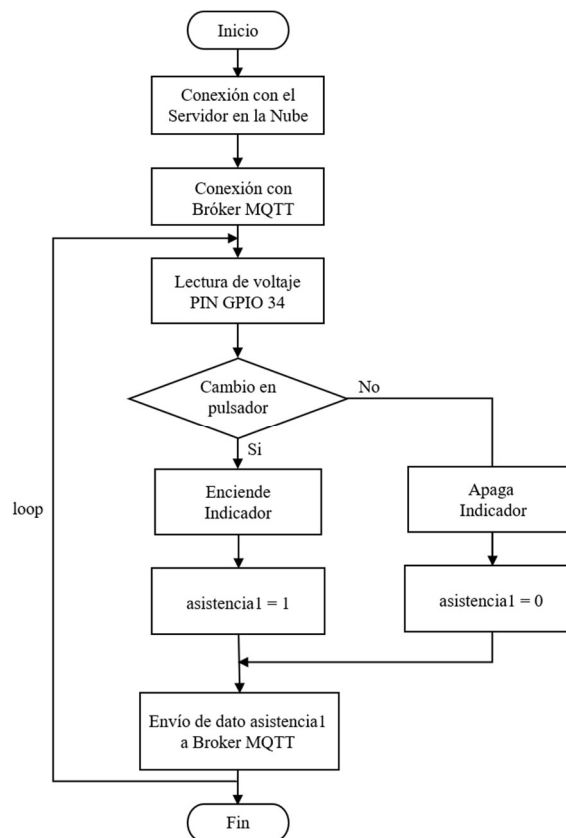


Figura 52. Diagrama de flujo del módulo de asistencia
Elaborado por: El Investigador

Módulo de control de iluminación

Se toma en cuenta los parámetros de diseño mencionados en el apartado 3.2.10.5 para la programación de este módulo. El módulo debe publicar el estado de las salidas de relé y además suscribirse para recibir y ejecutar cambios en el estado de los relés. Se integra una función de callback (petición de nuevo) para ejecutar cambios a partir de valores que contengan “sw” que lleguen desde los topics de control. Para ello se suscribe al topic “ciam/hab1/iluminación/#” que gestiona la llegada de mensajes en “ciam/hab1/iluminación/ret2”, “ciam/hab1/iluminación/ret2” y “ciam/hab1/iluminación/ret3”.

El envío respectivo del estado de cada una de las salidas de los relés, se hace con la publicación al broker MQTT de los siguientes topics “ciam/hab1/stret1”, “ciam/hab1/stret2” y “ciam/hab1/stret3”, los valores correspondientes a “1” y “0”. El código de programación de este módulo se encuentra en el Anexo K.

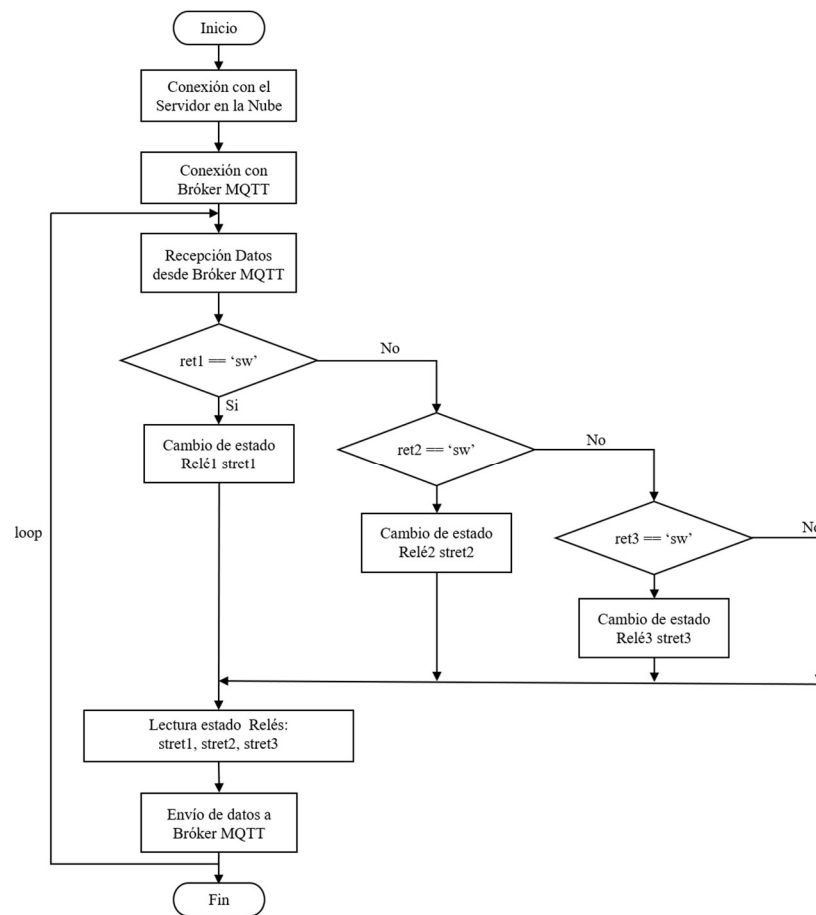


Figura 53. Diagrama de flujo del módulo de control de iluminación.

Elaborado por: El Investigador

Módulo de control de calefactor y ventilador

Para la programación tanto para el módulo de control de calefactor y ventilador, se toman consideraciones del diseño mencionados en el apartado 3.2.10.6. Debido a que ambos módulos funcionan electrónicamente de forma similar, se presentan diferencias notables en su código de programación. En ambos casos se debe publicar el estado de las salidas de relé y además suscribirse para recibir y ejecutar cambios en el estado de los relés. Se integra una función de callback para ejecutar cambios a partir de valores que lleguen desde los topics de control.

Para el control de calefactor se suscribe al topic “ciam/hab1/ambiente/relecal”, donde se reciben los mensajes de activación o desactivación al reconocer “sw” proveniente del broker MQTT. El envío del estado del relé, se hace con la publicación al topic

“ciam/hab1/stcalefactor”, los valores correspondientes a “1” y “0”. El código de programación de este módulo se encuentra en el Anexo L.

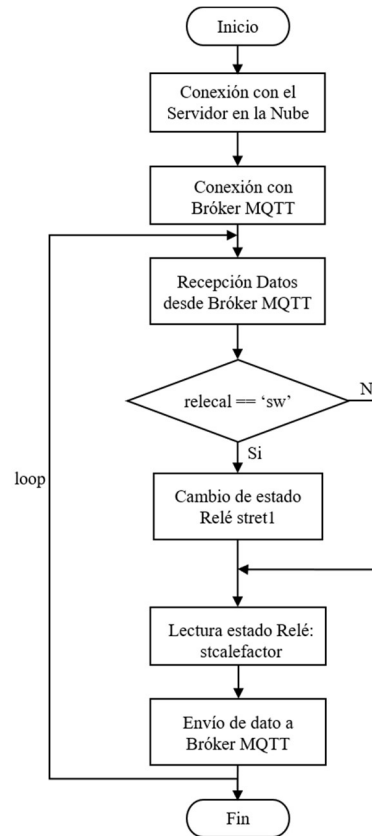


Figura 54. Diagrama de flujo del módulo de control del calefactor.

Elaborado por: El Investigador

Para el control de ventilador se suscribe al topic “ciam/hab1/ambiente/releven”, se reciben los mensajes de activación o desactivación al reconocer “sw” proveniente del broker MQTT. El envío del estado del relé, se hace con la publicación al topic “ciam/hab1/stventilador”, los valores correspondientes a “1” y “0”. El código de programación de este módulo se encuentra en el Anexo M.

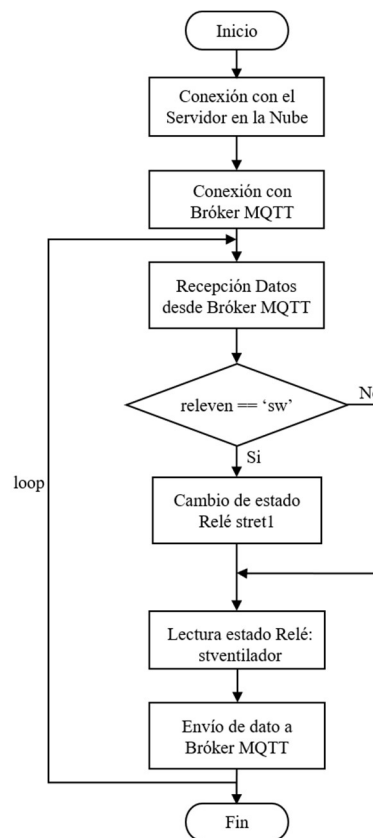


Figura 55. Diagrama de flujo del módulo de control del ventilador.

Elaborado por: El Investigador

3.2.12 Creación de la Base de datos

De acuerdo a los requerimientos del sistema, se desarrolla el reporte de alertas o eventos del sistema que muestra el registro histórico de mensajes generados por el sistema, este reporte funciona a partir de una base de datos, a la cual arriban las notificaciones en dependencia de la configuración del sistema.

Interfaz phpMyAdmin

Mediante la utilización de phpMyAdmin a través de la siguiente dirección <http://54148.93.1527/phpmyadmin>, se procede a la creación de la base de datos que almacena los diferentes mensajes que emite el sistema a través de su configuración en Node-Red.

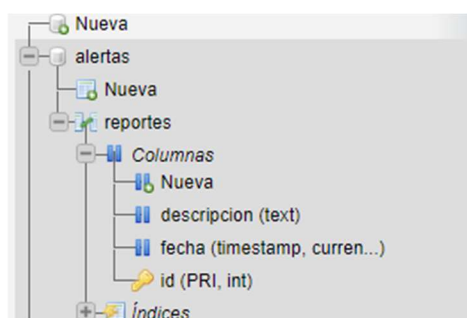


Figura 56. Base de datos para el reporte de alertas

Elaborado por: El Investigador

Desde la misma interfaz de phpMyAdmin, se realizan pruebas para constatar el correcto funcionamiento, con los campos de id, descripción y fecha completos.

				id	descripcion	fecha
<input type="checkbox"/>				1	Enviado desde phpMyAdmin alerta 1	2022-06-02 11:34:18
<input type="checkbox"/>				2	Mensaje de prueba alerta 2	2022-06-02 11:34:57
<input type="checkbox"/>				3	Notificación de prueba alerta 3	2022-06-02 11:35:26

Figura 57. Funcionamiento de la base de datos con los campos de la tabla

Elaborado por: El Investigador

3.2.13 Configuración de Node Red

El uso de la interfaz de Node Red, permite al investigador utilizar diversas librerías para lograr la convergencia entre protocolos de comunicación para los dispositivos, base de datos, servicios de mensajería móvil y la interfaz gráfica. Para acceder a la interfaz de configuración se debe acceder a la siguiente dirección web: **http://54.148.93.152:1880**, la página solicitará el usuario y contraseña, mismos que fueron establecidos en el apartado 3.2.9.4.

3.2.13.1 Conexión con la base de datos del servidor en la nube

El nodo “mysql” y la configuración mostrada en la Figura 61, permite conectar Node Red con la base de datos del servidor, el nodo requiere de los siguientes parámetros: host, puerto, usuario de mysql, contraseña y nombre de la base de datos. Una vez se hayan llenado estos campos, aparecerá el estado conectado (connected).

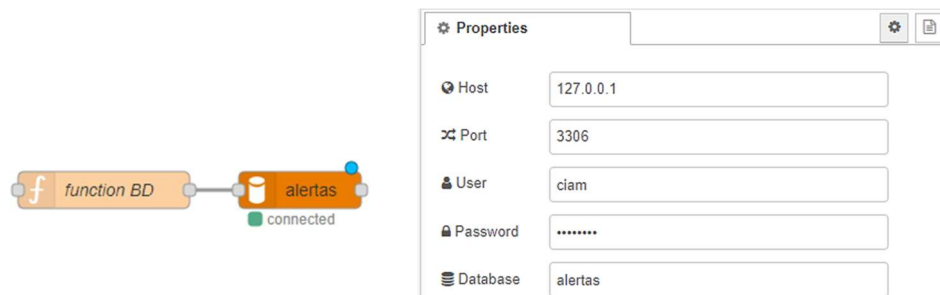


Figura 58. Configuración del nodo MySQL

Elaborado por: El Investigador

Toda la información que se envía a la base de datos del servidor en la nube, tiene su programación dentro del nodo “function”. Para enviar la información se utiliza principalmente la siguiente línea de código:

```
msg.topic = "INSERT INTO reportes(descripcion) VALUES ('"+texto+"");
```

3.2.13.2 Conexión con el broker MQTT del servidor en la nube

El nodo “mqtt” de Node Red es el utilizado para establecer la comunicación con el broker y determina su funcionamiento como suscripción o publicación.



Figura 59. Nodos “mqtt out” y “mqtt in”, para publicación y suscripción.

Elaborado por: El Investigador

Al configurar este nodo, se despliega una ventana con diversos parámetros a configurar. En la pestaña de Conexión, inicialmente se requiere especificar un nombre, la dirección del servidor la cual es “localhost” al tener el broker de mosquitto en la misma instancia EC2 y el puerto por defecto que es el 1880. En la pestaña de Seguridad, se ingresa las credenciales para la conexión configuradas al momento de la instalación del broker.

The screenshot shows the 'Properties' window for an MQTT broker named 'BrokerCIAM'. The 'Connection' tab is active, showing the following settings:

- Name:** BrokerCIAM
- Server:** localhost
- Port:** 1883
- Connect automatically:**
- Use TLS:**
- Protocol:** MQTT V3.1.1

Figura 60. Campos de configuración de conexión con el broker MQTT

Elaborado por: El Investigador

The screenshot shows the 'Security' tab of the MQTT broker configuration for 'Broker_MQTT_AWS'. The following security settings are visible:

- Name:** Broker_MQTT_AWS
- Username:** ciamjm
- Password:** [Redacted]

Figura 61. Campos de configuración de seguridad del broker MQTT.

Elaborado por: El Investigador

3.2.13.3 Conexión con bot de Telegram

Para la gestión y envío de las notificaciones que se envía a través de servicios de mensajería móvil, dentro de Node Red se integra un nodo de conexión hacia los servicios de Telegram que permite activar bots para la transferencia de mensajes (respuestas automáticas) sin la necesidad de registrar un número celular. Para este caso el Chat Bot seleccionado es “BotFther”, cuyo funcionamiento es compatible con el “nodo telegram” y por esta razón se ha eslegido este servicio de mensajería móvil.

Para la creación del bot, se localiza el servicio de chat BotFather en Telegram y se procede con los siguientes pasos:



Figura 62. Chatbot automatizado BotFather en Telegram

Elaborado por: El Investigador

1. Envío de texto “/start”, seguidamente desplegará las opciones, de las cuales se escoge la opción “/newbot”.
2. Solicitará ingresar el nombre del bot, es cual se introduce como “CIAMchat_bot”
3. Solicitará ingresar el nombre de usuario visible del bot, es cual se escribe como “Sistema-CIAM”, a continuación se recibe un mensaje indicando que la creación del bot ha sido exitosa.
4. Se revisa las propiedades del bot creado y con el envío del texto “/mybots”, y se elije la opción de API-Token. Este código sirve para realizar el envío de desde el sistema hacia un solo usuario de Telegram. Se agrega a demás una imagen para identificar el chatbot.



Figura 63. Token generado del chatbot para el usuario “Sistema-CIAM”
Elaborado por: El Investigador

5. Se configura el nodo “telegram receiver” con el nombre del bot creado “CIAMchat_bot” y el código del token mostrado en la figura 63.

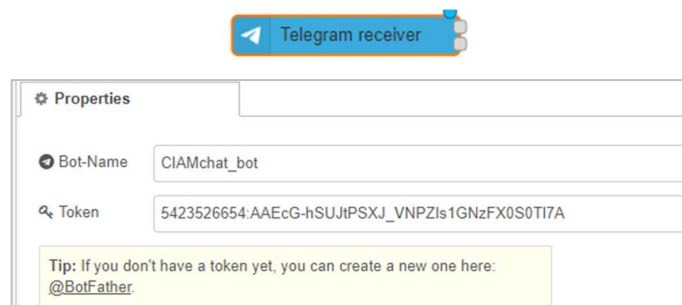


Figura 64. Configuración del nodo telegram.
Elaborado por: El Investigador

6. Debido a que el sistema debe enviar las notificaciones a varias cuentas de usuario a la vez, se crea un grupo con el nombre “Grupo Ciam”. Se busca generar el chatID que direcciona los mensajes hacia el grupo creado. Me diante la utilización del “nodo receiver”, se captura un mensaje enviado desde el grupo.

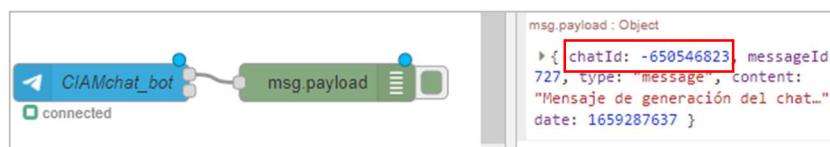


Figura 65. Nodo “receiver” y ChatID del grupo de telegram.
Elaborado por: El Investigador

7. A partir del código del chatID, se puede realizar el direccionamiento de los mensajes hacia el grupo. Para una prueba, se inyecta un texto “Mensaje de prueba para el grupo”, hacia el nodo “function”, con su respectiva programación, tal como se muestra a continuación.

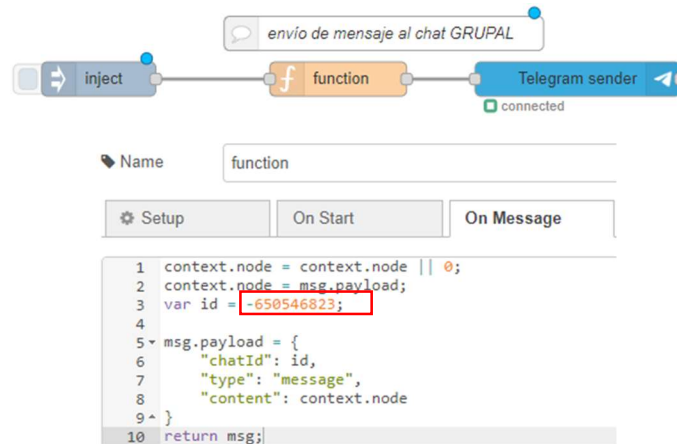


Figura 66. Programación del nodo “function” para el envío de mensaje al grupo de Telegram
Elaborado por: El Investigador



Figura 67. Mensaje desde el chatbot al grupo de telegram
Elaborado por: El Investigador

3.2.13.4 Configuración de la interfaz del tablero de monitoreo

Las siguientes configuraciones de nodos, se agrupan para procesar los datos MQTT publicados desde los diferentes dispositivos IoT encargados de brindar información de las condiciones del entorno. Su configuración completa se aprecia en el Anexo N.

Temperatura

De forma general se muestran los nodos configurados para la monitorización de temperatura. A través del nodo “mqtt in”, se suscribe a los datos que lleguen desde el topic ciam/hab1/temperatura para verse reflejados en un indicador (gauge).

Adicionalmente, se derivan otros nodos para el envío de mensajes hacia la base de datos al detectar temperaturas bajas menores a 16°C o superiores a los 28°C.

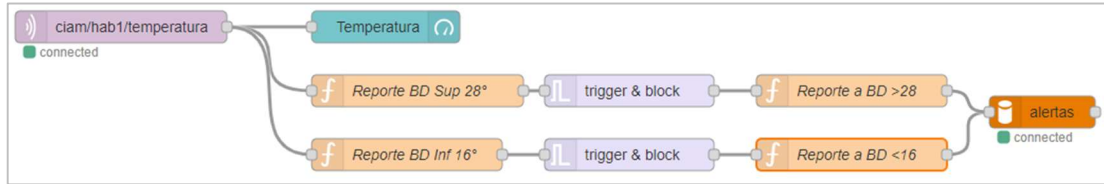


Figura 68. Nodos para el monitoreo de temperatura

Elaborado por: El Investigador

A través del nodo funcion “Reporte BD Sup 28”, se configura el envío de “1” o “0” hacia el nodo trigger, que permite enviar el mensaje “*Temperatura en Habitación 1 Supera los 28°C*” hacia la base de datos, de esta forma se evita duplicados cada vez que se reciba valores superiores a 28°C en el topic ciam/hab1/temperatura. Se reinicia el nodo trigger una vez que la temperatura es inferior a los 28°.

Al igual que en el anterior nodo funcion “Reporte BD Sup 16”, se configura el envío de “1” o “0” hacia el nodo trigger, envía el mensaje “*Temperatura en Habitación 1 es Inferior los 16°C*” hacia la base de datos evitando duplicados cuando se reciba valores inferiores a 16°C en el topic ciam/hab1/temperatura. Se reinicia el nodo trigger una vez que la temperatura es superior a los 16°C.

Humedad

Mediante el nodo “mqtt in” suscrito al topic ciam/hab1/humedad, se enlaza directamente hacia un indicador (gauge) para viasualizar los valores de la humedad relativa.

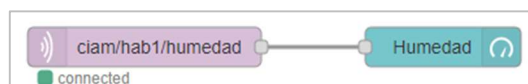


Figura 69. Nodos para el monitoreo de humedad

Elaborado por: El Investigador

Humo

El monitoreo de esta variable se relaciona con la seguridad de las personas, por lo tanto se establece un indicador tipo led, que responde al dato que arriva al topic `ciam/hab1/humo`, que determina la presencia de humo (CO₂) en el ambiente y enviar un mensaje a la base de datos del servidor. Se agregan funciones adicionales como el nodo “play audio”, que reproduce en forma de voz el mensaje, así mismo el nodo “show dialog” para mostrar como un notificación en la interfaz el mensaje a enviar a la base de datos.

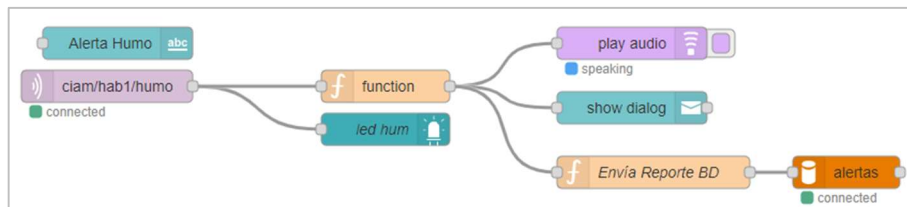


Figura 70. Nodos para el monitoreo de humo (co2)

Elaborado por: El Investigador

Movimiento

Se recoge los datos en el nodo “mqtt in” suscrito al topic `ciam/hab/movimiento`, en dependencia del dato que se reciba activa el indicador led de la interfaz. No se ha configurado nodos adicionales, debido a que solo quiere conocer la presencia de alguna persona en el área habitacional.

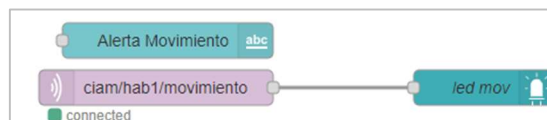


Figura 71. Nodos para el monitoreo de movimiento

Elaborado por: El Investigador

Estado de Asistencia

El monitoreo del valor que arroja el módulo de asistencia, lo recibe el nodo “mqtt in” suscrito al topic `ciam/hab1/asistencia1`, que enciende o apaga un indicador led en la

interfaz de usuario. Los nodos que conforman esta configuración envían el mensaje “Asistencia Requerida en Habitación 1” hacia la base de datos. Además se agregan los elementos de alerta como “show dialog” y “play audio”.

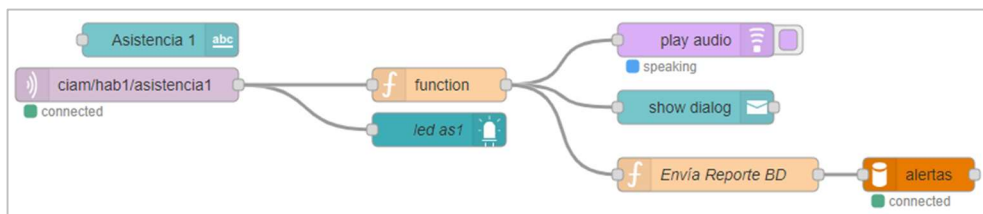


Figura 72. Nodos para asistencia

Elaborado por: El Investigador

La interfaz generada a través del dashboard que proporciona Node Red, se edita para ubicar los elementos de forma intuitiva para la interpretación de cada uno de los valores o indicadores. La Figura 73 muestra el grupo de elementos para el monitoreo.

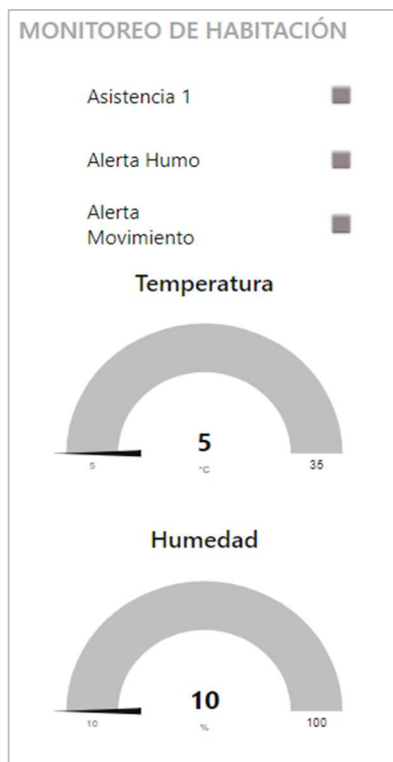


Figura 73. Interfaz con elementos para el monitoreo de variables.

Elaborado por: El Investigador

3.2.13.5 Configuración de la interfaz de tablero de control

Control manual y automático del módulo de calefactor

En la Figura 74, se muestra los nodos referentes al control manual de este módulo, a través del botón Calefactor, se carga el dato “sw”, que publica a través del nodo “mqtt-out”, bajo el siguiente topic “ciam/hab1/ambiente/relecal”.



Figura 74. Nodos de control manual del módulo de control del calefactor
Elaborado por: El Investigador

El control automático de este módulo requiere de parámetros adicionales, como el estado del switch de la interfaz, el valor de temperatura y estado actual del dispositivo. El nodo function “Calefacción Automática” es el encargado de tomar la decisión de cambiar el estado del módulo en base a los datos adquiridos, su programación en JavaScript se detalla en el Anexo O.

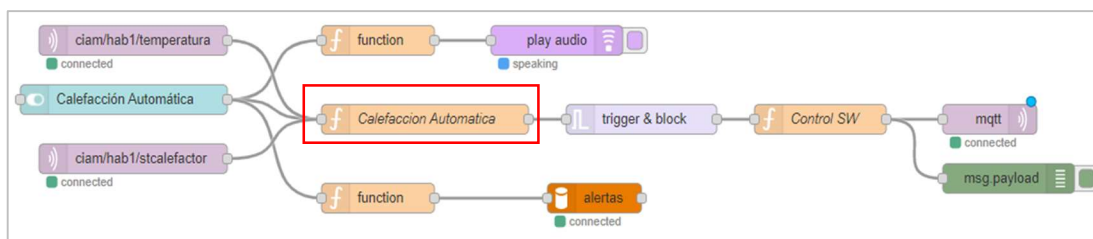


Figura 75. Nodos del control automático del módulo del calefactor
Elaborado por: El Investigador

Control manual y automático del módulo del ventilador

La configuración del control manual resulta similar al del módulo anterior, en el cual, se carga el dato “sw”, para publicarlo a través del nodo “mqtt out”, bajo el siguiente topic “ciam/hab1/ambiente/releven”.



Figura 76. Nodos de control manual del módulo de control del ventilador
Elaborado por: El Investigador

De igual manera, el control automático de este módulo requiere de parámetros adicionales, como el estado del switch de la interfaz, el valor de temperatura y estado actual del dispositivo. El nodo function “Ventilación Automática” se encarga de tomar la decisión para cambiar el estado del módulo, su programación en JavaScript se detalla en el Anexo P.

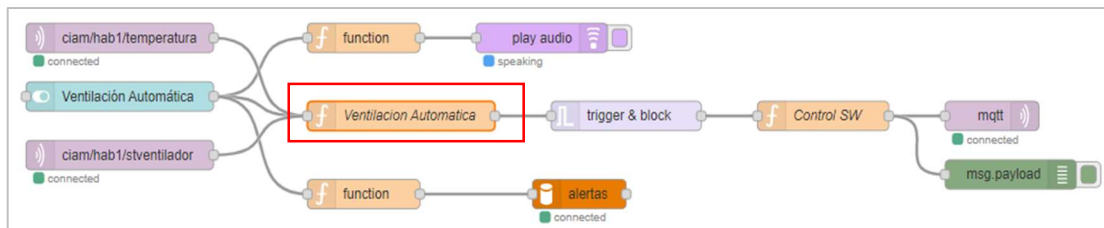


Figura 77. Nodos del control automático del módulo del ventilador

Elaborado por: El Investigador

Control de Iluminación

En Node Red se configuran botones, cada uno con payload que contiene el dato “sw”, que realiza el cambio de estado de cualquiera de los 3 relés del módulo de control de iluminación. Se publican respectivamente hacia los topics ciam/hab1/iluminación/ret1, ciam/hab1/iluminación/ret2, ciam/hab1/iluminación/ret3.

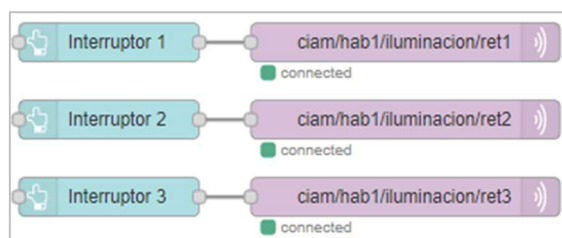


Figura 78. Nodos de control para iluminación

Elaborado por: El Investigador

La interfaz generada a través del dashboard que proporciona Node Red, se edita para ubicar los elementos destinados al control. La Figura 79 muestra el grupo de elementos para el control. La configuración general se muestra en el Anexo Q.



Figura 79. Interfaz con elementos del control y monitoreo de dispositivos
Elaborado por: El Investigador

3.2.13.6 Generación de Notificaciones Automáticas

Las notificaciones automáticas se emiten a través de Telegram en dependencia del valor que reciben los nodos “mqtt in”. Activar o desactivar esta opción almacena un mensaje en la base de datos. La siguiente tabla describe cuándo se genera cada una de ellas y la alerta que se envía.

Tabla 11. Condiciones para el envío de Notificaciones

Topic	Valor	Mensaje a enviar
ciam/hab1/asistencia1	1	Se requiere Asistencia en Habitación 1
ciam/hab1/humo	1	Alerta de Humo en Habitación 1
ciam/hab1/temperatura	>28	Temperatura superior a 28°C en Habitación 1
ciam/hab1/temperatura	<16	Temperatura inferior a 16°C en Habitación 1

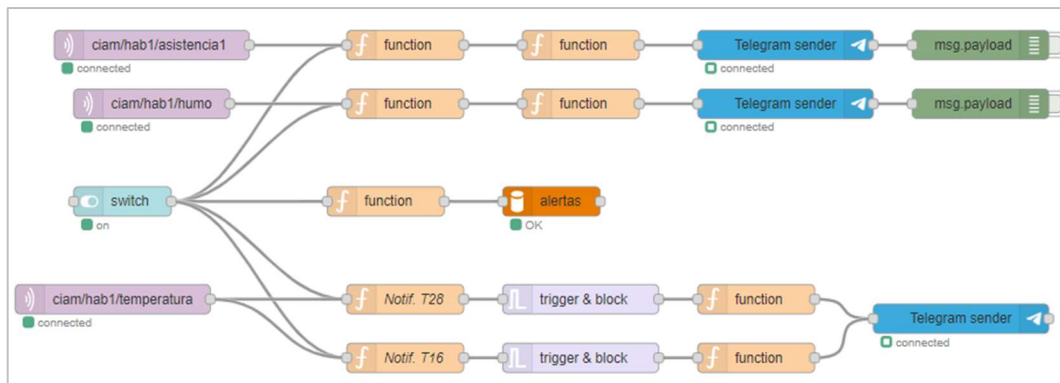


Figura 80. Nodos de configuración para las notificaciones vía Telegram
Elaborado por: El Investigador

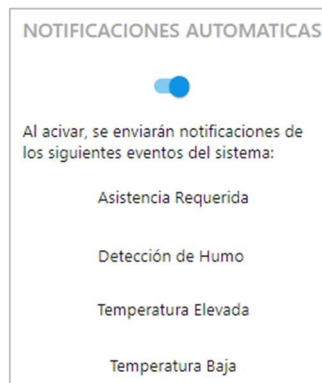


Figura 81. Control de activación/desactivación de las notificaciones automáticas.
Elaborado por: El Investigador

3.2.14 Reporte de alertas

Para mostrar de forma ordenada cada una de las alertas enviadas a la base de datos y por rango de fechas, se realiza la programación de un archivo escrito en lenguaje php, el archivo “enlazar.php”, conecta con la base de datos del servidor en la nube y extrae de forma ordenada los datos de la tabla. Luego, a través del archivo “index.php”, se integran los controles para la visualización de la información extraída desde el archivo “enlazar.php”, El resultado es una interfaz con la capacidad de mostrar las alertas, filtrar a través de la búsqueda o rango de fechas y un botón para vaciar la tabla.

54.148.93.152/index.php

Reporte de Alertas del Sistema Domótico

Mostrar 10 registros por página. Búsqueda

Orden ID	Descripción de la Alerta	Fecha
3	Alerta desde phpmyadmin 4	10/06/2022 11:10:53
2	Alerta desde reporte.php 2	10/06/2022 11:10:02
1	Alerta desde phpmyadmin 1	10/06/2022 11:09:16

Mostrando página 1 de 1. Anterior 1 Siguiente

Figura 82. Interfaz del reporte de alertas del Sistema domótico

Elaborado por: El Investigador

3.2.15 Reconocimiento de Voz

Para realizar el control por voz de los módulos de control de iluminación, calefactor y ventilador, se plantea el siguiente esquema de funcionamiento.

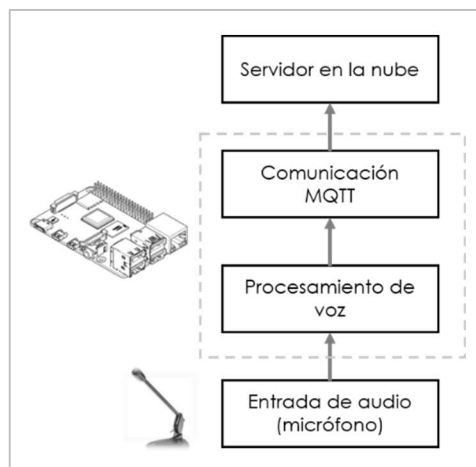


Figura 83. Diagrama de bloques para el control por voz

Elaborado por: El Investigador

Para la adquisición de audio de la voz, se ha optado por utilizar el Micrófono AE-MICU04, cuya interfaz de comunicación es USB de bajo consumo, permite conectarlo hacia los puertos USB de la Raspberry. Cuenta con botón de encendido y apagado. Posee con alta sensibilidad a una distancia efectiva de 2m. [65] Se selecciona este micrófono debido a la facilidad de encontrarlo en el mercado a un precio bajo.



Figura 84. Micrófono AE-MICU04 para el reconocimiento de voz [65]

Elaborado por: El Investigador

3.2.15.1 Instalación del sistema operativo en la Raspberry Pi.

Para la realización del reconocimiento de voz se utiliza la Raspberry Pi 4. Se procede con la instalación del Sistema operativo en una Memory card SD de 32GB. A través del Software “Raspberry Pi Imager”.



Figura 85. Instalación del sistema operativo RaspiOS con Raspberry Pi Imager

Elaborado por: El Investigador

Para facilitar la configuración de la Raspberry Pi, se habilita en su configuración el servicio SSH para el acceso remoto a la terminal de comandos y VNC para acceder a la interfaz de la Raspberry de forma remota.

Antes de pasar al proceso de instalación de paquetes en la Raspberry, es necesario actualizar el sistema, se lo realiza ejecutando los siguientes comandos.

```
# apt update
```

```
# apt full-upgrade
```

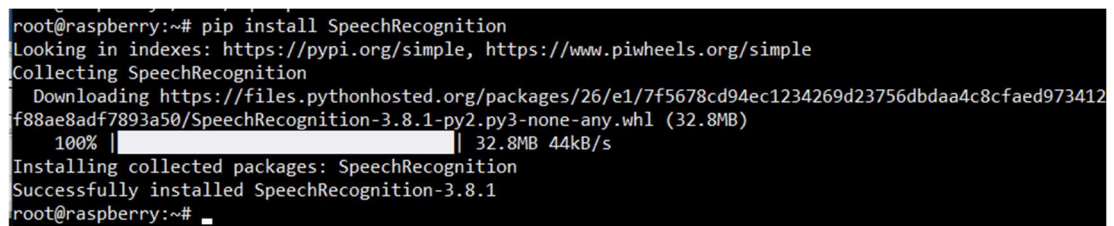
Por defecto el sistema operativo de la Raspberry tiene instalado el paquete de Python, por tanto se debe actualizar a la versión más reciente y estable con la que se trabaja en la actualidad, ejecutando la siguiente línea de comando.

```
# update-alternatives --install /usr/bin/Python Python /usr/bin/python3
```

3.2.15.2 Speech Recognition

Se instala el paquete de python que contiene la librería Speech Recognition en la versión 3.8.1 para python, que hace uso de la API de Google Cloud Speech, para el análisis de audio y retorno de la transcripción en forma de texto. Para instalar la librería se utiliza el siguiente comando en la consola.

```
#pip install SpeechRecognition
```



```
root@raspberrypi:~# pip install SpeechRecognition
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting SpeechRecognition
  Downloading https://files.pythonhosted.org/packages/26/e1/7f5678cd94ec1234269d23756dbdaa4c8cfaed973412f88ae8adf7893a50/SpeechRecognition-3.8.1-py2.py3-none-any.whl (32.8MB)
    100% |#####| 32.8MB 44kB/s
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.8.1
root@raspberrypi:~#
```

Figura 86. Instalación de la librería SpeechRecognition en la Raspberry

Elaborado por: El Investigador

Para la adquisición de audio a través de la entrada micrófono (plug o USB), se instala el paquete Python3-pyaudio y Python-pyaudio para sostener dos versiones compatibles con la versión Python que se encuentra instalada en la Raspberry, se lo realiza a través de los siguientes comandos:

```
#apt-get install python3-pyaudio -y
```

```
#apt-get install python-pyaudio
```

Para establecer un codec que almacene el audio de la voz, se instala FLAC, que realiza una compresión específicamente para audio. Para un pronto análisis de la voz, la API

de Google soporta el procesamiento de FLAC. [66] Se ejecuta el siguiente comando para su instalación:

```
#apt-get install flac -y
```

3.2.15.3 PAHO-MQTT

Debido a que se requiere una comunicación con el broker MQTT instalado en el servidor en la nube, se descarga e instala la librería PAHO que habilita la comunicación MQTT, con el siguiente comando.

```
# pip 3 install paho-mqtt
```

```
root@raspberrypi:/home/pi# pip3 install paho-mqtt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: paho-mqtt in /usr/local/lib/python3.7/dist-packages (1.6.1)
root@raspberrypi:/home/pi#
```

Figura 87. Instalación de la librería PAHO-MQTT en python.

Elaborado por: El Investigador

Se verifica la conexión hacia el servidor en la nube ejecutando el archivo “pahomqtt.py” en la consola.

```
GNU nano 3.2                                pahomqtt.py

import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("ciam")

def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

client = mqtt.Client(client_id=" ", clean_session=True, userdata=None, protocol=mqtt.MQTTv311, transport="tcp")
client.on_connect = on_connect
client.on_message = on_message

client.username_pw_set(username="ciamjm", password="proyecto22")
print("Connecting...")
client.connect("192.168.0.8", 1883, 60)
client.loop_forever()
```

Figura 88. Archivo para suscripción al broker MQTT del servidor en la nube

Elaborado por: El Investigador

```
root@raspberrypi:/home/mqtt-pruebas# python pahomqtt.py
Connecting...
Connected with result code 0
ciam b'textomqtt'
ciam b'1234567890'
```

Figura 89. Recibo de datos del topic “ciam” desde el servidor en la nube

Elaborado por: El Investigador

3.2.15.4 Tkinter

Para hacer uso del reconocimiento de voz, se debe utilizar una interfaz gráfica amigable para el usuario, evitando la utilización de comandos a través de la consola, bajo este requerimiento, se procede a instalar el paquete Tkinter con la finalidad de diseñar la interfaz gráfica en la Raspberry Pi, por tanto, se ejecuta las siguientes líneas de comando:

```
#apt install python3-tk -y
```

```
#pip install tkterminal
```

3.2.15.5 Interfaz para el reconocimiento de voz y conexión con MQTT

Se debe tener en consideración que la interfaz debe cumplir óptimamente con el reconocimiento de voz, a partir del análisis de la voz y las transcripciones en texto que se generen, enviar los mensajes en los topics correspondientes.

El diseño se ha realizado haciendo uso de la librería Tkinter, en la cual se añaden controles para la interfaz en el archivo “01Reconocim_voz.py” cuya programación se puede revisar en el Anexo T. En la figura 90 se muestra la interfaz final para el reconocimiento de voz.

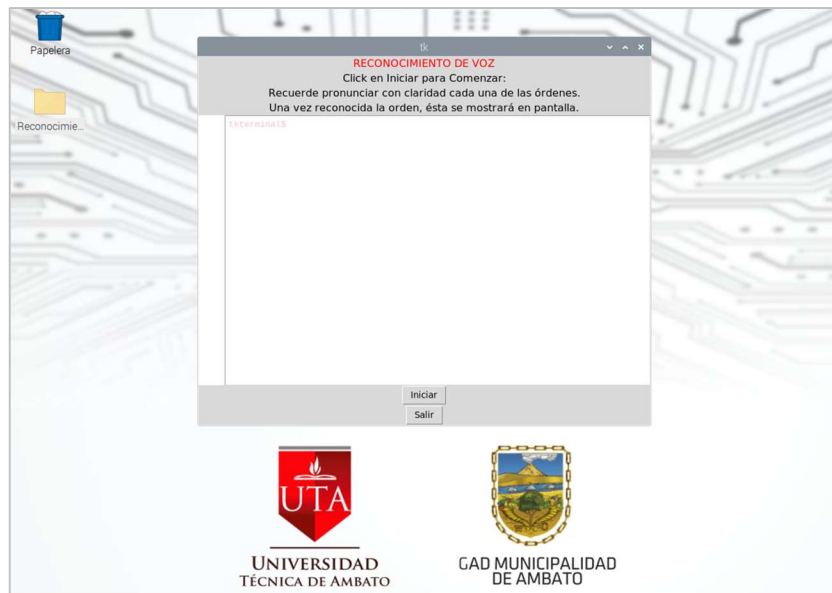


Figura 90. Interfaz gráfica para el reconocimiento de voz
Elaborado por: El Investigador

Al dar clic en el botón iniciar, se realiza el llamado del archivo “Ordenes_por_voz.py”, que ejecuta primero la conexión con el broker MQTT en la nube, una vez ha establecido la comunicación, procede con el reconocimiento de voz en idioma Español Latinoamérica México [MX]. La figura 91 muestra la inicialización del reconocimiento de voz.



Figura 91. Inicialización del reconocimiento de voz
Elaborado por: El Investigador

El archivo “Ordenes_por_voz.py”, contiene las frases o comandos de voz a ser reconocidos, junto con los topics y mensajes a publicarse en el broker MQTT del servidor en la nube. La siguiente tabla explica estas relaciones para la ejecución de órdenes.

Tabla 12. Órdenes por voz

Comando/Frase	Topic de publicación	Mensaje
“cambia el estado de habitación 1”	ciam/hab1/iluminación/ret1	‘sw’
	ciam/hab1/iluminación/ret2	‘sw’
	ciam/hab1/iluminación/ret3	‘sw’
“cambia el estado del calefactor 1”	ciam/hab1/ambiente/relecal	‘sw’
“cambia el estado del ventilador 1”	ciam/hab1/ambiente/releven	‘sw’

Elaborado por: El Investigador

3.2.16 Implementación del sistema

3.2.16.1 Infraestructura del área habitacional.

El lugar en donde se va a instalar los diversos componentes del sistema, se encuentra ubicada en la provincia de Tungurahua en el cantón Ambato, en el Centro Integral del Adulto Mayor, dentro de sus instalaciones se ha designado para el proyecto una habitación la cual se precisa su distribución interna y medidas, tal como se muestra en la Figura 93.

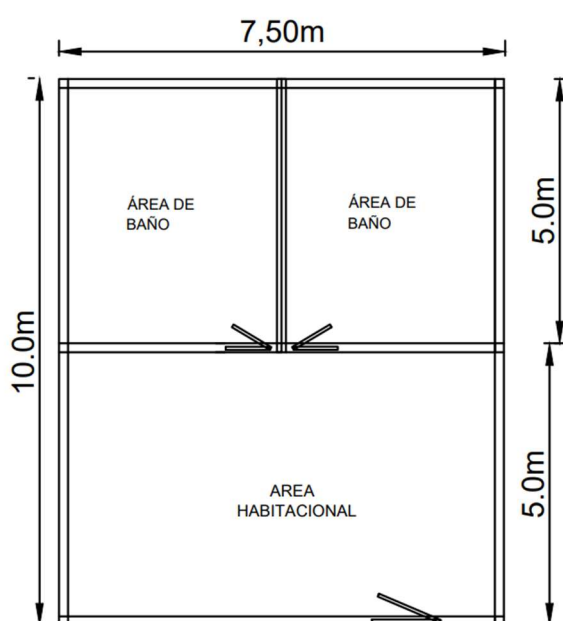


Figura 92. Área habitacional designada para el proyecto

Elaborado por: El Investigador

3.2.16.2 Recursos tecnológicos de red disponibles en el centro

El CIAM cuenta con rack de comunicaciones con conexión a internet administrada por el departamento de Tecnologías de la información perteneciente al GAD Municipalidad de Ambato. La red que posee el centro es gestionada por un router MikroTik que mantiene conexión cableada hacia una sola computadora del centro. Para proceder con el desarrollo del proyecto se asignó una dirección IP para la conexión y acceso a la navegación en Internet.



Figura 93. Rack de comunicaciones
Elaborado por: El Investigador

La configuración de red asignada para acceder a la conexión a internet a través del rack de comunicaciones se muestra en la siguiente tabla.

Tabla 13. Configuración de dirección IP

Dirección IP	10.10.67.123
Máscara de subred	255.255.255.0
Puerta de enlace predeterminada	10.10.67.1
DNS Primario	10.10.0.8
DNS Secundario	10.10.43.22

3.2.16.3 Despliegue de la Red Wifi

Al no contar con equipos que proporcionen conectividad Wifi para los diversos dispositivos a instalarse en el centro. Se plantea el despliegue de la red Wifi en la habitación designada para la implementación del proyecto. Por lo que se utiliza un equipo que pueda configurarse a través de las direcciones IP asignadas:

Selección y Configuración del Equipo Wifi

Se selecciona el equipo TP-Link Archer C24 por sus características de funcionamiento, ya que puede trabajar en 3 modos distintos: Router, Punto de Acceso y Extensor de

señal en las bandas de 2.4GHz y 5GHz. Para el proyecto se trabaja con el modo Router, debido a que requiere la configuración IP en su puerto WAN.



Figura 94. Router TP-Link Archer C24 [67]

Elaborado por: El Investigador

Primero se configura la Red Wifi del equipo en la que se deja activa solo la banda de 2.4GHz, en la cual se establece el SSID: Red_Proyecto; y como contraseña: proyectodom22. En la configuración de internet del equipo, se completan los campos con los datos de la Tabla 13.

The screenshot shows the 'Internet' configuration page in the router's web interface. The 'Internet' tab is selected in the top navigation bar. The page title is 'Internet' and the subtitle is 'Configurar una conexión a Internet con la información de servicio proporcionada por su ISP (Proveedor de servicio de internet)'. The configuration fields are as follows:

Tipo de Conexión de Internet:	IP Estático	Seleccione este tipo si su ISP le proporciona parámetros IP específicas.
Dirección IP:	10.10.67.123	
Máscara de Subred:	255.255.255.0	
Puerta de Enlace Predeterminada:	10.10.67.1	
DNS Primario:	10.10.0.8	
DNS Secundario:	10.10.43.22	(Opcional)

Figura 95. Configuración de IP estática en el router Wifi

Elaborado por: El Investigador

Finalmente, se establece la conexión por cable tup-cat5 entre el puerto ETH07 del router MikroTik hasta el área habitacional donde se encuentra el router wifi. El equipo se instala en un lugar pueda brindar la máxima cobertura de señal.



Figura 96. Instalación del Router Wifi en el área habitacional
Elaborado por: El Investigador

3.2.16.4 Instalación de los módulos de monitoreo

Para proceder a instalar los módulos y determinar su mejor ubicación, se han tomado consideraciones tanto del personal que labora en el centro como del autor. En dependencia de la infraestructura, la distribución de los dispositivos es la siguiente:

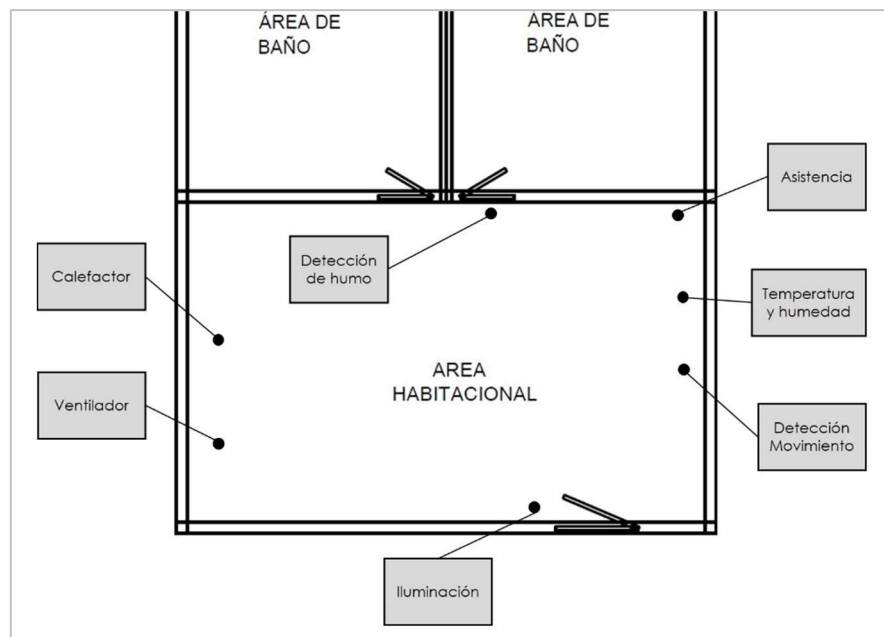


Figura 97. Distribución de los dispositivos
Elaborado por: El Investigador

Una vez se ha ubicado en el plano los dispositivos, se procede con la instalación física dentro de área habitacional, tratando de ser lo menos invasiva, con utilización de canaleta para evitar cables de conexión a la vista.

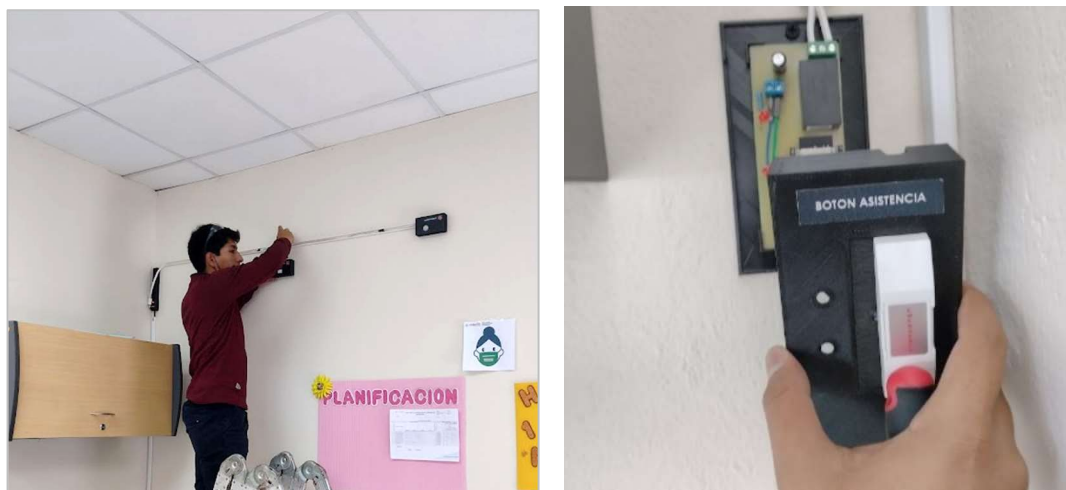


Figura 98. Proceso de instalación de los módulos de temperatura, detección de movimiento y asistencia.

Elaborado por: El Investigador

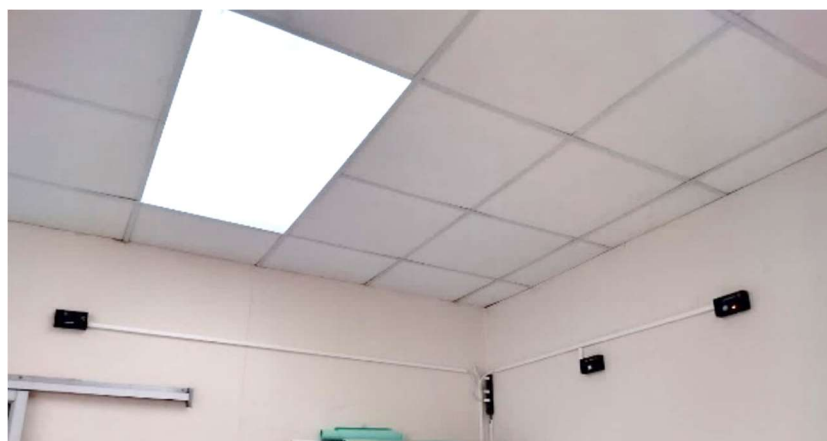


Figura 99. Módulos de monitoreo instalados

Elaborado por: El Investigador



Figura 100. Módulos de Asistencia y detección de humo instalados
Elaborado por: El Investigador

3.2.16.5 Instalación de los módulos de control

En el centro se cuenta con un pequeño calefactor de 400W-120V y un ventilador que funciona con un cargador de 5V. Se realiza el cálculo de corriente para el calefactor

$$Corriente = \frac{potencia}{voltaje} = \frac{400W}{127V} = 3.14[Amperios]$$

La corriente que circula por el calefactor, es soportada por el módulo de control del calefactor cuya capacidad máxima es de 10Amp. Su instalación se lleva a cabo cerca a la conexión del tomacorriente de 120V, puesto que este dispositivo cambiará de ubicación dentro del área habitacional, al no tener un lugar fijo donde ubicarlo.



Figura 101. Conexión del módulo de control del calefactor
Elaborado por: El Investigador

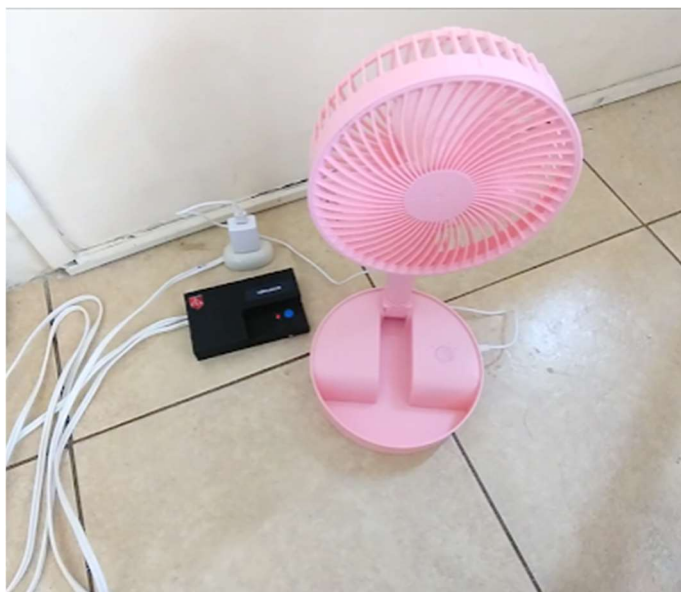


Figura 102. Conexión del módulo de control del ventilador
Elaborado por: El Investigador

Para la instalación del módulo de control de iluminación, también se realizan las medidas relacionadas a la corriente. Al tener instalados tubos fluorescentes, es necesario realizar la medición con la pinza amperimétrica.

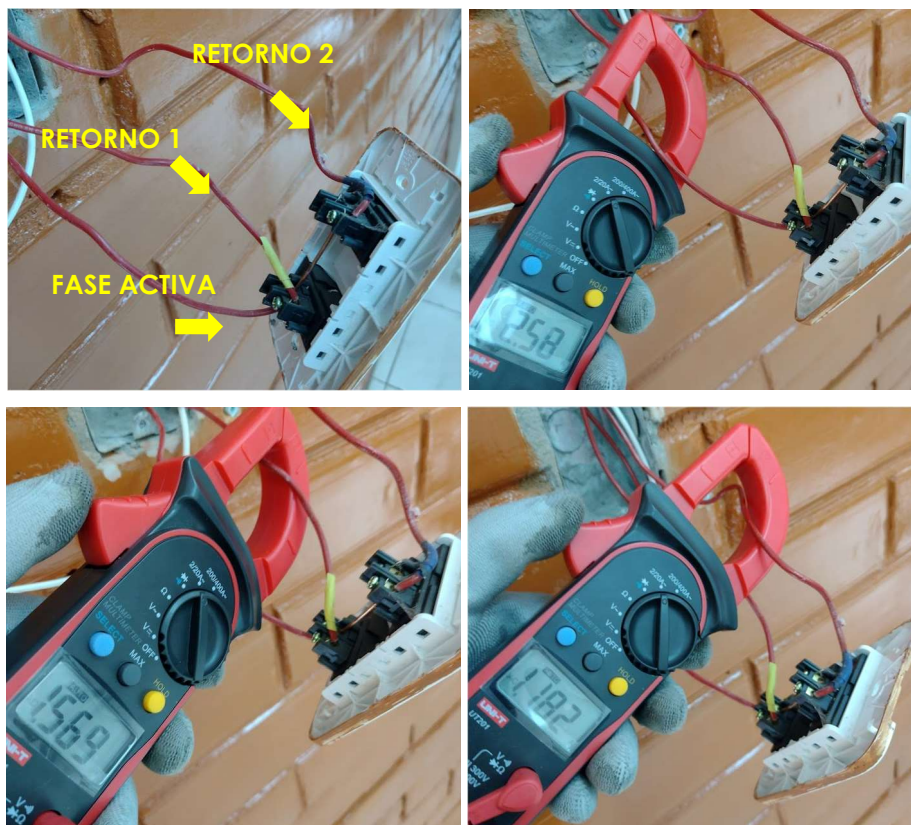


Figura 103. Mediciones de corriente en los interruptores para luces previa instalación del módulo de control de iluminación.

Elaborado por: El Investigador

Al ser la corriente total que circula por el interruptor manual de 2.58 Amperios, se procede a sustituir el interruptor actual por el módulo diseñado, el cual puede controlar 3 retornos y soporta 10 Amperios.

Previo a su instalación se ha realizado una conexión al neutro, ya que el dispositivo funciona a 120V. Su instalación es la siguiente.

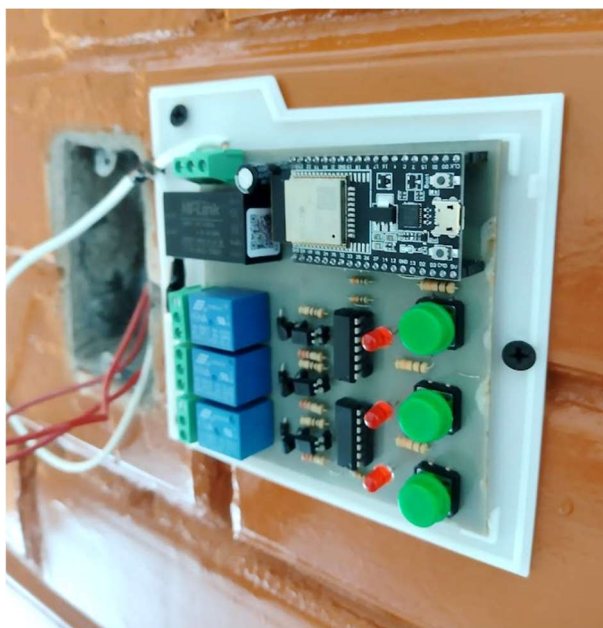


Figura 104. Proceso de instalación del módulo de control de iluminación
Elaborado por: El Investigador

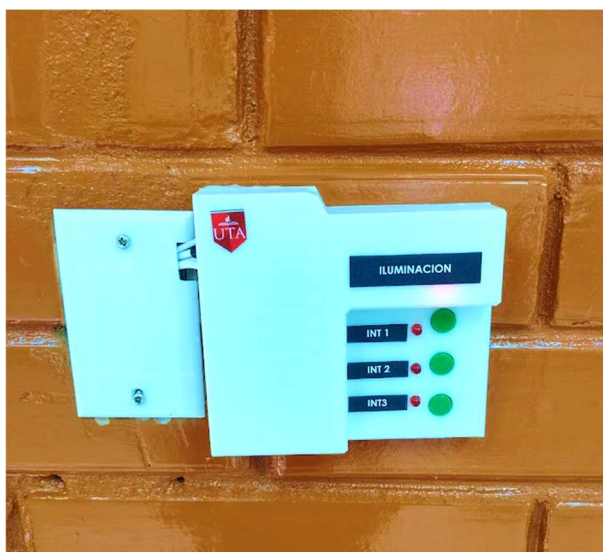


Figura 105. Módulo instalado para el control de luminarias
Elaborado por: El Investigador

3.2.16.6 Estación de monitoreo y control por voz

Dentro de la oficina departamental del centro, se ha asignado un espacio para instalar y conectar la Raspberry pi 4 con sus periféricos computacionales y Monitor, para el control por Voz.

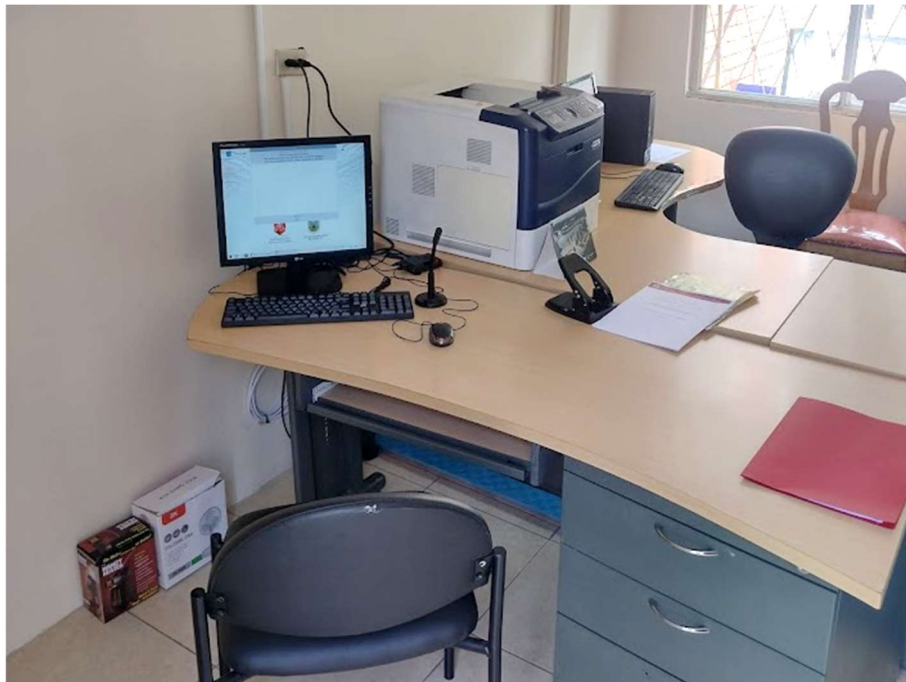


Figura 106. Oficinas departamentales del Centro donde se ubica la Raspberry Pi

Elaborado por: El Investigador

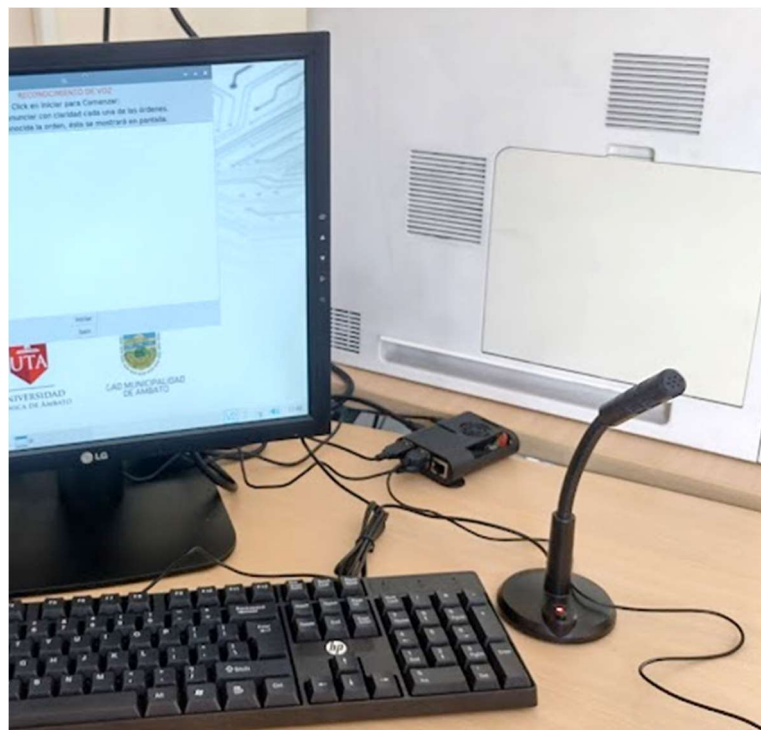


Figura 107. Raspberry pi y micrófono para el reconocimiento de voz

Elaborado por: El Investigador

3.2.16.7 Pruebas de Funcionamiento

Pruebas de conexión de los dispositivos con el Broker MQTT

Para realizar esta prueba ha sido necesario utilizar el programa MQTT-Explorer, cuya interfaz permite visualizar los topics en los cuales se están publicando mensajes, al mismo tiempo permite detectar qué dispositivos envían datos hacia el servidor.



Figura 108. Topics de los dispositivos IoT del sistema

Elaborado por: El Investigador

Interfaz del sistema

La interfaz creada para el sistema, debe resultar fácil de manejar, puesto que desde los controles agregados al Dashboard se monitorizan las variables del ambiente en el que conviven las personas adultas y el control de actuadores para encender y apagar las luces, un calefactor y un ventilador. Se han generado 3 interfaces. La primera contiene conexión con todos los dispositivos IoT instalados. La segunda interfaz activa las notificaciones automáticas vía Telegram. La tercera brinda la información necesaria de los registros del sistema.

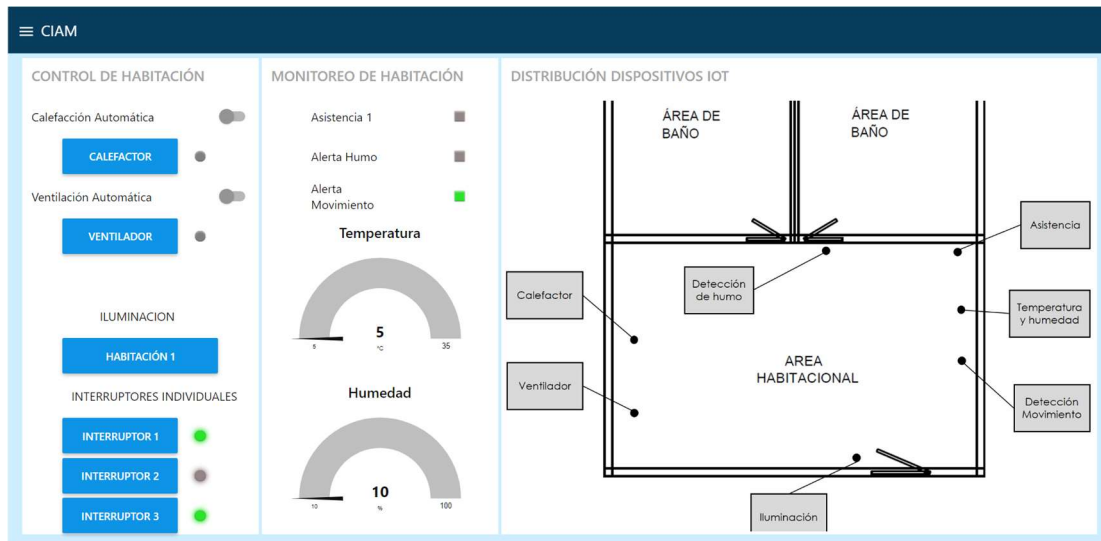


Figura 109. Interfaz principal del sistema domótico

Elaborado por: El Investigador

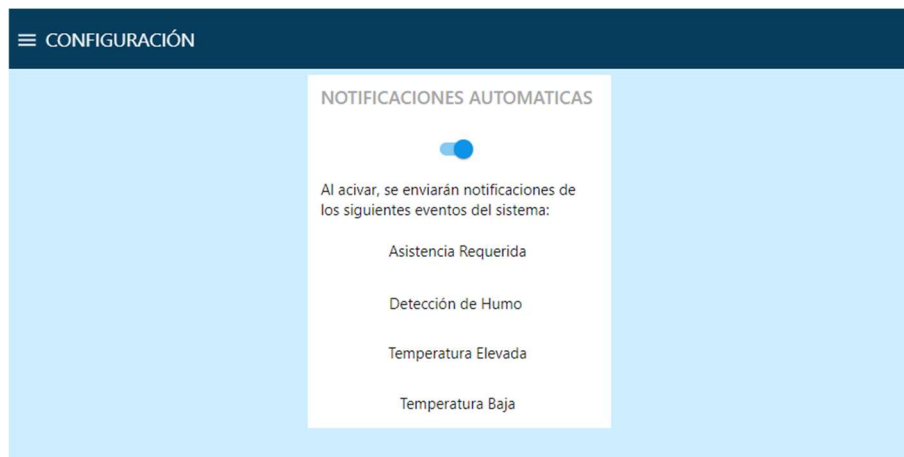


Figura 110. Interfaz de activación de notificaciones automáticas vía Telegram

Elaborado por: El Investigador

Para las pruebas se mantiene activado el switch de las notificaciones, de esta forma se asegura el envío de mensajes a través de Telegram.

Sensorización de las variables del ambiente.

Debido a que el sistema debe trabajar en conjunto, se activan las notificaciones automáticas vía Telegram, así mismo los mensajes llegarán de forma automática a la base de datos. Además, se revisa la Tabla 11 concerniente a las condiciones para el

envío de Notificaciones vía Telegram, el resto de nodos de conexión a la base de datos están configurados en el grupo de nodos correspondiente.

Monitoreo y alertas de Temperatura

Para genera las notificaciones de temperatura elevada, el módulo de temperatura y humedad se lo expone de un foco incandescente para elevar la temperatura por encima de los 28°C, mientras que para obtener una temperatura baja se simula a través de mensajes publicados a través de MQTT.



Figura 111. Notificación vía telegram, Temperatura superior a 28°C

Elaborado por: El Investigador

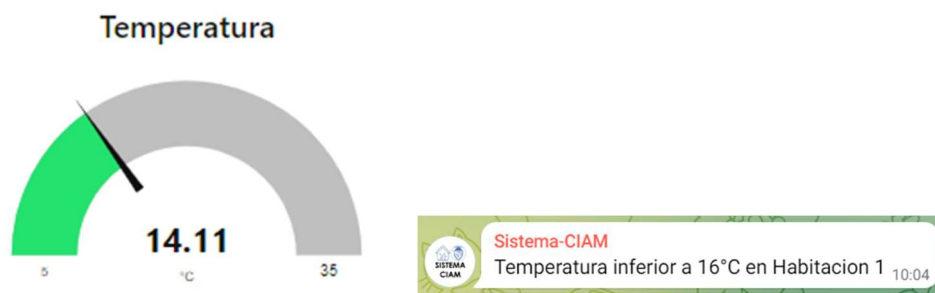


Figura 112. Notificación vía telegram, Temperatura superior a 160°C

Elaborado por: El Investigador

El reporte de alertas, con conexión a la base de datos también se actualiza y muestra las nuevas entradas de forma ordenada, tal como se muestra en la siguiente figura.

4.148.93.152/index.php

Reporte de Alertas del Sistema Domótico

Buscar Vaciar

Mostrar 10 registros por página. Búsqueda

Orden ID	Descripción de la Alerta	Fecha
2	Temperatura en Habitación 1 es inferior a 16°C	18/07/2022 10:04:23
1	Temperatura en Habitación 1 Supera los 28°C	18/07/2022 10:02:06

Mostrando página 1 de 1 Anterior 1 Siguiente

Figura 113. Mensajes de los nodos de Temperatura a la base de datos
Elaborado por: El Investigador

Monitoreo de Humedad.

Debido a que el módulo contiene el sensor DHT-22, este trabaja en conjunto. No se han configurado alertas o notificaciones vía Telegram para esta variable por consideraciones del personal del centro.

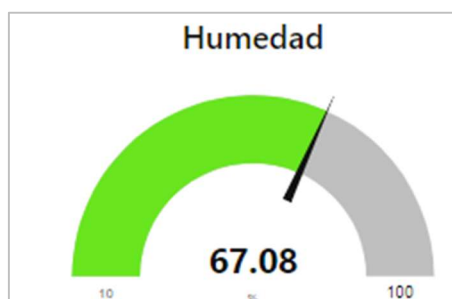


Figura 114. Medición de humedad relativa
Elaborado por: El Investigador

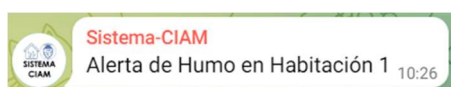
Detección y alertas de Humo

Para que el módulo pueda reconocer una perturbación en la calidad del aire a través del sensor MQ-135, ajustado manualmente a su mínimo rango de sensorización, se procede a acercar un frasco de vidrio con algodón en llamas, sin representar peligro alguno para el área habitacional, con fines probatorios.



Figura 115. Notificación en la interfaz (show dialog) e indicador de Alerta de humo

Elaborado por: El Investigador



Orden ID	Descripción de la Alerta	Fecha
3	Alerta de Humo en Habitación 1	18/07/2022 10:26:02

Figura 116. Notificación vía Telegram y Actualización del reporte de alertas

Elaborado por: El Investigador

Detección de Movimiento

La configuración de este nodo únicamente incluye el indicador led en la interfaz. La finalidad de este indicador, es conocer la presencia de alguien en el área habitacional, sin requerir de notificaciones vía Telegram, ni enviar mensajes a la base de datos. El sensor activa y desactiva de forma intermitente el led del módulo durante 10 segundos, una vez haya detectado el movimiento.

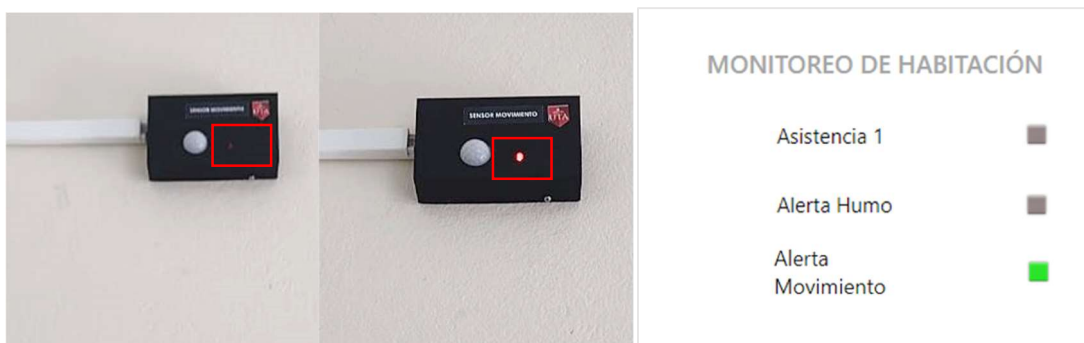


Figura 117. Funcionamiento del módulo detector de movimiento

Elaborado por: El Investigador

Asistencia

Al presionar el pulsador de asistencia, debe de enviar una alerta hacia la base de datos, la notificación automática vía Telegram y la notificación en la interfaz web.

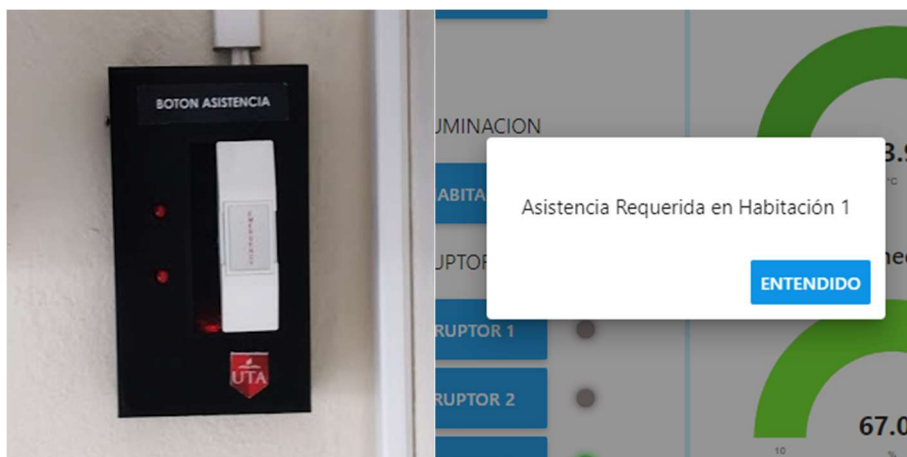
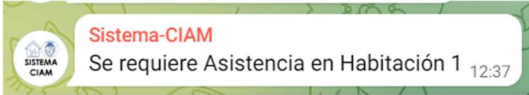


Figura 118. Notificación de asistencia en la interfaz web

Elaborado por: El Investigador



Orden ID	Descripción de la Alerta	Fecha
4	Asistencia Requerida en Habitación 1	18/07/2022 12:37:06

Figura 119. Notificación automática vía Telegram de asistencia y envío de alerta a la base de datos

Elaborado por: El Investigador

Calefacción automática

Para realizar la prueba de funcionamiento de este grupo de nodos, se interviene los datos del valor de temperatura. A fin de obtener el funcionamiento requerido.

El módulo del calefactor se activa cuando se ha activado la “Calefacción automática”, la temperatura está por debajo del 16°C y se recibe el estado actual del módulo., Si alguna de estas condiciones no se cumple, el módulo cambiará el estado a apagado.



Figura 120. Funcionamiento de Calefacción automática

Elaborado por: El Investigador



Figura 121. Cambio de estado (apagado a encendido) con “Calefacción automática”

Elaborado por: El Investigador

Orden ID	Descripción de la Alerta	Fecha
3	Temperatura en Habitación 1 es Inferior a 16°C	18/07/2022 12:50:51
2	Activada la calefacción automática en Habitación 1	18/07/2022 12:49:47

Figura 122. Envío del mensaje de activación de calefacción automática la base de datos

Elaborado por: El Investigador

Cuando se activa esta opción, si se intenta cambiar desde el botón del dispositivo, este volverá al estado que requiere el sistema. Se impide el cambio de forma manual al activar este control de tipo switch.

Ventilación automática

Se interviene los datos del valor de temperatura. A fin de obtener el funcionamiento requerido al activar esta opción. El módulo del ventilador se activa cuando se ha superado los 28°C. Al activar la “Ventilación automática”, se recibe el estado actual del módulo. Si alguna de estas condiciones no se cumple, el módulo cambiará el estado a apagado.



Figura 123. Funcionamiento de Calefacción automática

Elaborado por: El Investigador



Cambio de estado (apagado a encendido) con “ventilación automática”

Elaborado por: El Investigador

6	Activada la ventilación automática en Habitación 1	18/07/2022 13:05:33
5	Temperatura en Habitación 1 Supera los 28°C	18/07/2022 13:05:26

Figura 124. Envío del mensaje de activación de la ventilación automática a la base de datos
Elaborado por: El Investigador

Iluminación

La configuración de estos nodos, tiene la función de controlar el módulo de control de iluminación, por lo cual se muestra a continuación el estado del dispositivo, junto con el estado de ellos leds indicadores.

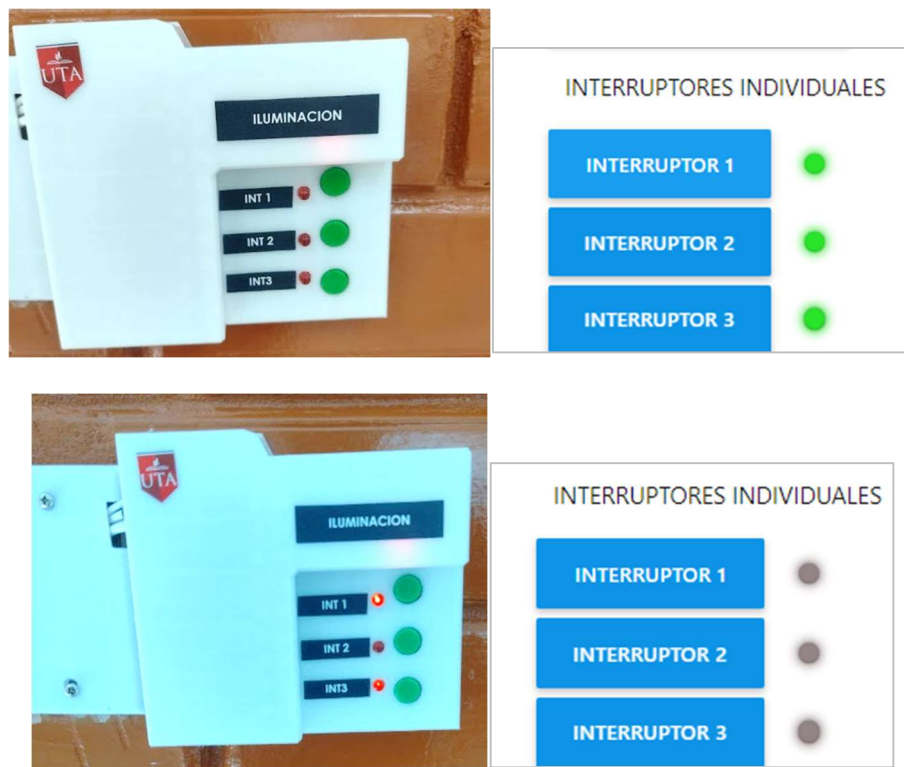


Figura 125. Estados del módulo de control de iluminación en la interfaz web
Elaborado por: El Investigador



Figura 126. Botón de encendido/apagado general
Elaborado por: El Investigador

Mediante este último botón, se apagan o encienden todos los relés del módulo de iluminación

Reconocimiento de voz

A través de la interfaz creada con la librería Tkinter, han realizado pruebas pertinentes al control por voz de los dispositivos de control.

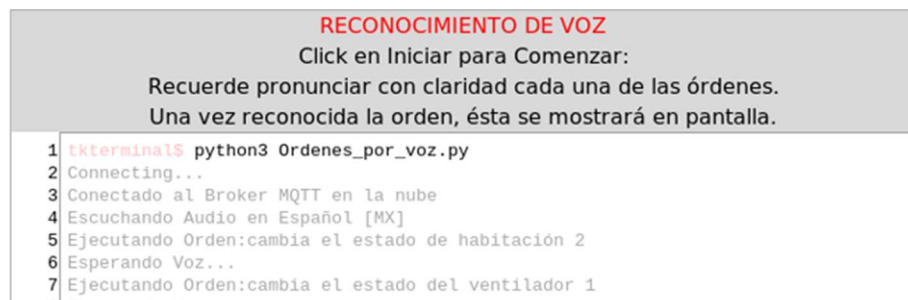


Figura 127. Ejecución de órdenes a través del reconocimiento de voz
Elaborado por: El Investigador

Para comprobar el nivel de desempeño del sistema de reconocimiento de voz se realizó las siguientes pruebas con el personal que labora en el centro, en dos diferentes horarios, el primero entre las 11H a 12H y el segundo entre las 14H y 15H, esto durante 2 días. Dentro de esos rangos de tiempo, personal del centro, puede acercarse y decir el comando respectivo como máximo 3 ocasiones en un orden indistinto. Se marca un 1 cuando reconoce el comando, mientras que se marca 0 cuando no se reconoce el comando.

Tabla 14. Evaluación del reconocimiento de voz en Raspberry pi.

Fecha	Comando	ocasión		
		1ra	2da	3ra
16/07/2022	“cambia el estado de habitación 1”	1	1	1
Entre las 11H y 12H	“cambia el estado del calefactor 1”	1	1	0
	“cambia el estado del ventilador 1”	1	1	1
16/07/2022	“cambia el estado de habitación 1”	1	1	0
Entre las 14H y 15H	“cambia el estado del calefactor 1”	1	1	1
	“cambia el estado del ventilador 1”	0	1	1
17/07/2022	“cambia el estado de habitación 1”	1	1	1
Entre las 11H y 12H	“cambia el estado del calefactor 1”	1	1	0
	“cambia el estado del ventilador 1”	1	1	1
17/07/2022	“cambia el estado de habitación 1”	1	0	1
Entre las 14H y 15H	“cambia el estado del calefactor 1”	1	1	1
	“cambia el estado del ventilador 1”	1	1	1
18/07/2022	“cambia el estado de habitación 1”	1	1	1
Entre las 11H y 12H	“cambia el estado del calefactor 1”	1	1	1
	“cambia el estado del ventilador 1”	1	1	1
18/07/2022	“cambia el estado de habitación 1”	1	1	1
Entre las 14H y 15H	“cambia el estado del calefactor 1”	0	1	1
	“cambia el estado del ventilador 1”	1	1	0
	Errores de ejecución de comandos	3	1	4

Elaborado por: El Investigador

En la siguiente tabla se muestra el resultado global del reconocimiento de voz.

Tabla 15. Porcentaje de fiabilidad del control por voz

Comandos de Voz	Cantidad	Porcentaje
Reconocidos	47	87,03%
No reconocidos.	7	12,97%
Total	54	100,00%

Elaborado por: El Investigador

En los resultados de la Tabla 15, indican que los comandos reconocidos alcanzan el 87,03%, y los no reconocidos o no ejecutados llegan al 12,97%, De acuerdo a estos datos se determina una alta fiabilidad del sistema para ser controlado por voz. Factores como el uso de mascarilla al momento de pronunciar los comandos, ruido exterior o características propias del micrófono influyen en reconocimiento de voz óptimo. reconocimiento de voz.

3.2.16.8 Presupuesto del proyecto

El presupuesto del proyecto se ha dividido en dos valores, el primero hace referencia al presupuesto de materiales del proyecto, mientras que el segundo se refiere al presupuesto del diseño y ejecución. Los valores correspondientes a los materiales del proyecto del sistema se detallan en la siguiente Tabla:

Tabla 16. Presupuesto de materiales del proyecto

Presupuesto				
ítem	Detalle	Cantidad	Valor U.	Total
1	ESP-32 Wroom 39 pins	7	11,50	80,50
2	Baquelitas Fibra de vidrio	3	4,50	13,50
3	Relés 05V – 5 pines	5	1,10	5,50
4	Pack Componentes electrónicos Varios	1	35,00	35,00
5	Rasberry Pi 4 – GB	1	180,00	180,00
6	Monitor LG 19”	1	80,00	80,00
7	Periféricos de computación	1	35,00	35,00
8	Cable AWG 14 (m)	19	0,65	12,35
9	Regletas de conexión 120V	3	5,80	17,40
10	Canaleta 20*20mm	5	3,90	19,50
11	Fuente Hilink 5V Pm01	7	6,00	42,00
12	Cajas impresas en 3D	7	11,00	77,00
13	Instancia AWS mes de Julio	1	2,61	2,61
			TOTAL	\$600.36

Elaborado por: El Investigador

Para obtener el presupuesto de diseño y ejecución del proyecto, se toma en cuenta el salario mínimo de acuerdo a la tabla de “Salarios mínimos sectoriales 2022”. El salario mínimo para un profesional con formación en electrónica es de \$457,52 mensuales [67], tomando la cantidad de 21 días al mes se procede con el desglose del cálculo del sueldo por hora a través de las siguientes ecuaciones:

$$Sueldo_{día} = \frac{Salario\ mínimo}{días\ laborables}$$

$$Sueldo_{día} = \frac{457,52}{21} = \mathbf{21,78\ USD}$$

$$Sueldo_{hora} = \frac{Sueldo_{día}}{horas\ laborables}$$

$$Sueldo_{hor} = \frac{21,78\ USD}{8} = \mathbf{2,72\ USD}$$

Tal como se calcula anteriormente el valor de cada hora corresponde a 2,72 dólares. Por lo que el investigador determina la inversión de 280 horas para la investigación, diseño y ejecución del proyecto, dicho valor se especifica en la siguiente tabla:

Tabla 17. Presupuesto Diseño y Ejecución del proyecto

Detalle	Cantidad	Valor U.	Total
Hora investigación, diseño y ejecución	280	2.72	761,60

Elaborado por: El Investigador

Para obtener el Presupuesto total del Proyecto se procede a sumar los valores totales obtenidos en la Tablas 16 y la Tabla 17. Dicho resultado se muestra a continuación:

Tabla 18. Presupuesto total del Proyecto

ítem	Detalle	Total
1	Presupuesto de materiales del proyecto	597,75
2	Presupuesto Diseño y Ejecución del proyecto	761,60
	TOTAL	\$1359.35

Elaborado por: El Investigador

Presupuesto referencial para todas las áreas habitacionales del CIAM

El Centro Integral del Adulto Mayor actualmente destina tres áreas habitacionales, de similares características mostradas en la Figura 93, por lo tanto, cada una de estas áreas mantiene la misma cantidad de dispositivos mostradas en la Figura 97. Tomando en cuenta que solo se mantiene una única estación de monitoreo y control por voz en la oficina departamental, se genera la siguiente tabla de valores referenciales relacionadas únicamente a los materiales.

Tabla 19. Presupuesto referencial para tres áreas habitacionales del CIAM

Presupuesto referencial				
Item	Detalle	Cantidad	Valor U.	Total
1	ESP-32 Wroom 39 pins	21	11,50	241,50
2	Baquelitas Fibra de vidrio	9	4,50	40,50
3	Relés 05V – 5 pines	15	1,10	16,50
4	Pack Componentes electrónicos Varios	1	120,00	120,00
5	Raspberry Pi 4 – 8 GB	1	180,00	180,00
6	Monitor LG 19”	1	80,00	80,00
7	Periféricos de computación	1	35,00	35,00
8	Cable AWG 14 (m)	60	0,65	39,00
9	Regletas de conexión 120V	7	5,80	40,60
10	Canaleta 20*20mm	15	3,90	58,50
11	Fuente Hilink 5V Pm01	21	6,00	126,00
12	Cajas impresas en 3D	21	11,00	231,00
			TOTAL	\$1208,60

Elaborado por: El Investigador

El valor total obtenido de 1208.60 dólares en la Tabla 19, tiene la finalidad de indicar el valor aproximado de la inversión necesaria para desarrollar los dispositivos IoT, así como implementar también el reconocimiento de voz.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Una vez culminado el trabajo de investigación se determinaron las siguientes conclusiones:

- El uso de la domótica en centros destinados al cuidado de adultos mayores, incrementa el nivel de cuidado y asistencia que estos pueden recibir, a su vez integra a este grupo poblacional a interactuar con nuevas tecnologías.
- A través de la comunicación de los dispositivos IoT mediante el protocolo MQTT diseñados para el proyecto, el servidor en la nube puede ser capaz de incrementar sus prestaciones, resultando en un sistema de mayor escalabilidad que soporte la comunicación entre más dispositivos, suscritos a más temas.
- Se realizó el control de condiciones ambientales a partir de módulos de sensorización que, en dependencia de los valores adquiridos, efectúan la publicación de mensajes hacia el servidor en la nube, con el fin de establecer un ambiente óptimo para adultos mayores de acuerdo a personal del centro.
- El control por voz que se emplea, resulta ser fiable en cuanto al descifrado, y al combinarlo con la librería PAHO-MQTT se logra una convergencia de servicios, ya que desde los comandos de voz se pueden enviar mensajes hacia el broker en la nube, efectuando el control de los módulos de iluminación, calefacción y ventilación.
- El módulo referente a la asistencia de adultos mayores, tienen gran utilidad en caso de presentarse escenarios en los que personas adultas requieran de ayuda o se encuentren en escenarios de riesgo. Al combinar servicios de mensajería móvil, se envían las notificaciones de forma automática con el mensaje correspondiente a la alerta generada.

4.2 Recomendaciones

- La utilización de componentes electrónicos de mejor calidad de los que se encuentran en el mercado, disminuye las perturbaciones e interferencias generadas en comunicación wifi del microcontrolador ESP-32, a la vez que reduce el tamaño de las placas de los módulos.
- La configuración de las reglas de seguridad del servidor en la nube, deben ser acorde a los servicios que se instalan dentro de la instancia, la apertura de puertos que no se utilicen puede significar a futuro una posible vulneración a la configuración del servidor en la nube.
- Se puede optar por otras API's para el reconocimiento de voz, ya que a través de los servicios de la API Google Speech to Text, debido a las constantes actualizaciones puede dar lugar a respuestas erróneas en la transcripción del comando ejecutado.
- Se puede implementar un servidor de Backup en la Raspberry Pi 4, que centralice la comunicación MQTT entre dispositivos de forma local, para luego enviar esta información al servidor en la Nube. De esta forma se conserva el control y monitoreo del sistema ante fallos de conexión al internet.
- El sistema puede tener compatibilidad con otros servicios en la nube, como enlazar el servidor hacia contenedores que gestionan de manera más eficaz la comunicación MQTT, pudiendo agregar la Calidad de Servicio Qos2 sin generar retardos perceptibles entre los dispositivos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Darío Herrera, «Diseño e implementación de un prototipo de seguridad para control domótico basado en IoT bajo ambientes de dispositivos móviles con Android,» Quito, 2020.
- [2] M. V. Wilks, «Diseño y desarrollo de un sistema domótico de bajo costo,» Madrid, 2019.
- [3] J. B. Brun, «Smart Home usando IoT y Chatbots,» UCM-ED, Madrid, 2018.
- [4] I. S. Ruiz, «Diseño e implementación de una aplicación móvil para el control de un sistema domótico,» Madrid, 2017.
- [5] M. F. Díaz, «Caracterización, implantación y evaluación de dispositivos domóticos para adultos mayores mediante sistemas embebidos e internet de las cosas,» Bogotá, 2017.
- [6] Ministerio de Inclusión Económica y Social, «Dirección Población Adulta Mayor,» MIES, 2018. [En línea]. Available: <https://www.inclusion.gob.ec/direccion-poblacion-adulta-mayor/>.
- [7] Ministerio de Inclusión Económica y Social, «Programa “MIS MEJORES AÑOS” Tungurahua,» MIES, 2021. [En línea]. Available: <https://www.inclusion.gob.ec/programa-mis-mejores-anos-atiende-a-mas-de-2-700-adultos-mayores-de-tungurahua/>.
- [8] B, Quinde. M. Mosquera. A. Vázquez, «Brecha Digital en Adultos Mayores: Accesibilidad Tecnológica,» 2020. [En línea]. Available: <https://www.gigapp.org/ewp/index.php/GIGAPP-EWP/article/view/220/228>.
- [9] Paula Forttes Valdivia, «Envejecimiento y atención a la dependencia en ECUADOR,» BID, 2020. [En línea]. Available: <https://publications.iadb.org/publications/spanish/document/Envejecimiento-y-atencion-a-la-dependencia-en-Ecuador.pdf>.
- [10] R. Teneda, «Uso de las nuevas tecnologías de la información (TIC) en el Adulto Mayor,» UCE, Quito, 2015.
- [11] Organización Mundial de la Salud, «El envejecimiento y la Salud,» OMS, 2015. [En línea]. Available: https://apps.who.int/iris/bitstream/handle/10665/186466/9789240694873_spa.pdf.
- [12] Consejo Nacional para la Igualdad Intergeneracional, «Estado de situación de las personas adultas mayores,» 2017. [En línea]. Available: <https://www.igualdad.gob.ec/estado-de-situacion-de-las-personas-adultas-mayores/#>.
- [13] Ciapat ORG, «Discapacidad,» Ciapat Cono sur, 2018. [En línea]. Available: <http://www.ciapat.org/biblioteca/pdf/1004-Discapacidad.pdf>.

- [14] N. Forcada, A. Mendez, «Análisis del confort térmico en residencias de ancianos,» CENIE, 2019. [En línea]. Available: <https://cenie.eu/es/blogs/conterma/analisis-del-confort-termico-en-residencias-de-ancianos-en-el-espacio-de-cooperacion#:~:text=La%20temperatura%20%EF%BF%B3ptima%20ser%20a%20alrededor.>
- [15] R. Lara, A. Velarde,, «Manual para cuidadores de la persona adulta mayor dependiente,» MSP, Quito, 2011.
- [16] A. Rivor, M. Morales, A. Casamayou,, «Usos y percepciones de las tecnologías digitales en personas mayores. Limitaciones y beneficios para su calidad de vida,» Revista Austral, 2019.
- [17] L. Cerdá Filiu, M. Bueno, «Instalaciones Domóticas,» Paraninfo ED, Madrid, 2020.
- [18] Cano-Lara, M. Armenta, E. Cabal, H. Juárez, «Diseño y simulación de un sistema de control Smart Home,» *Revista de Aplicación Científica y Técnica*, vol. 2, nº 4, 2016.
- [19] J. Martín, E. Mancilla, «Domótica,» UCM3, Madrid, 2015.
- [20] N. Bouhai, I. Saleh, J. Whiley, «Internet of Things : Evolutions and Innovations,» 2017. [En línea]. Available: <https://0110c2o0h-y-https-ebookcentral-proquest-com.uta.metaproxy.org/lib/uta-ebooks/reader.action?docID=5122089&query=Internet+of+Things+%3A+Evolutions+and+Innovations..>
- [21] M. J. Bermejo, «Internet de las cosas (2a. ed.),» 2020. [En línea]. Available: https://0110o2o0i-y-https-elibro-net.uta.metaproxy.org/es/lc/uta/titulos/185096?as_all=Internet__de__las__cosas.
- [22] RS, «Qué es el Internet de las Cosas,» Radiospare, 2020. [En línea]. Available: <https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things.>
- [23] W. López, J. Cárdenas, «Internet of Things (IoT) technology and the big data,» 2019. [En línea]. Available: <https://0110o2o0i-y-https-elibro-net.uta.metaproxy.org/es/ereader/uta/127847.>
- [24] P. P. Ray, «A survey on Internet of Things architectures,» 2016. [En línea]. Available: <https://core.ac.uk/download/pdf/82196957.pdf>.
- [25] Emad Felemban, ««A Comparative Study of IEEE 802.11 family Protocols,» *International Journal of Computer Science and Network Security*, vol. 20, nº 7, 2020.
- [26] IEEE Computer Society, «Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,» 2020. [En línea]. Available: <https://thewifiofthings.com/wp-content/uploads/2021/08/802.11-2020-Preview.pdf>.
- [27] S. Cope, «Beginners Guide To The MQTT Protocol,» Steve's internet Guide, 2021. [En línea]. Available: <http://www.steves-internet-guide.com/mqtt/>.

- [28] H. M. Team,, «MQTT Essentials,» Hive MQ, 2015. [En línea]. Available: <https://www.hivemq.com/mqtt-essentials/>.
- [29] MQTT Organization, «MQTT Software,» MQTT ORG, 2022. [En línea]. Available: <https://mqtt.org/software/>.
- [30] Eclipse Foundation, «Mosquitto & MQTT,» MQTT ORG, 2022. [En línea]. Available: <https://test.mosquitto.org>.
- [31] Firtec, «Que es MQTT,» 2019. [En línea]. Available: <https://www.firtec.com.ar/cms/53-que-es-mqtt>.
- [32] Hive MQ Team, «Quality of Service (QoS) 0,1, & 2 MQTT Essentials,» Hive MQ, 2019. [En línea]. Available: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/#:~:text=QoS%20%20is%20the%20highest>.
- [33] SNP Technologies, Microsoft, «MODERNIZE OPEN-SOURCEWEB APPLICATIONS WITH AZURE,» 2019. [En línea]. Available: <https://www.snp.com/sites/default/files/2020-08/SNP-eBook-Modernize-Open-Source-Web-Applications-with-Microsoft-Azure.pdf>.
- [34] Open JS Foundation, «About Node RED,» The OpenJS Foundation, 2020. [En línea]. Available: <https://nodered.org/about/#history>.
- [35] I. Takaya, N. Kazuki,, «ntroduction of the IoT Platform Node-RED and Hitachi Activities,» 2018. [En línea]. Available: https://events19.linuxfoundation.org/wp-content/uploads/2017/11/Introduction-of-the-IoT-Platform-Node-RED-and-Hitachi_s-Activities-OSS-Takaya-Ide_-Kazuki-Nakanishi.pdf.
- [36] R. Josefina, A. Cerón, M. Peñaloza, J. Figueroa, «LA MENSAJERÍA INSTANTÁNEA COMO ECOSISTEMA MEDIÁTICO BAJO EL MODELO INNIS. UN ANÁLISIS DE SU IMPACTO SOCIAL Y TECNOLÓGICO,» *FACE*, vol. 21, n° 3, 2022.
- [37] E. Santos, «Comparativa de Whatsapp, Telegram, Hangouts, Skype y Facebook Messenger,» Webdia, 2021. [En línea]. Available: <https://www.genbeta.com/comparativa/comparativa-de-whatsapp-telegram-hangouts-y-facebook-messenger-cual-es-el-mejor-desde-el-pc>.
- [38] S. Naziya, R. R. Deshmukh, «Speech Recognition System – A Review,» *Journal of Computer Engineering (IOSR-JCE)*, vol. 18, n° 4, 2021.
- [39] I. Calle, «Reconocimiento de voz basado en características DNN Bottleneck,» Madrid, 2019.
- [40] D. Tomalá, «Sistema domótico controlado por voz para personas con discapacidad en extremidades superiores, utilizando tarjeta Raspberry,» 2018. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/15141/1/UPS-GT002060.pdf>.
- [41] Google ORG, «Documentación de Speech-to-Text,» Google Cloud Found., 2022. [En línea]. Available: <https://cloud.google.com/speech-to-text/docs>.

- [42] P. Javanovic, M. Mileusnic, B.Pavic, B. Miskovic, «Applications of the Single Board Computers in the Software Defined Radio Systems,» Sinteza, Belgrad, 2014.
- [43] O. UK, «Odroid C2 - 64-bit quad-core Single Board Computer,» Odroid UK, 2017. [En línea]. Available: <https://www.odroid.co.uk/hardkernel-odroid-c2-board>.
- [44] J. E. H.K., «Jaguarboard: Jaguar one,» JG-HK, 2016. [En línea]. Available: http://www.jaguarboard.org/index.php/com_virtuemart_menu_configuration/products/buy/jaguarboard/207/jaguarboard-detail.html#specification..
- [45] Raspberry Pi Trading Ltd, «Raspberry Pi 4 Computer - Model B,» 2019. [En línea]. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>.
- [46] E. Systems, «ESP-WROOM-32D/ESP32-WROOM-32U Datasheet,» 2017. [En línea]. Available: https://empa.com/empaiot/esp-wroom-32d_esp32-wroom-32u_datasheet_en.pdf.
- [47] Components101, «ESP32 - DevKitC,» C-101 Corp, 2019. [En línea]. Available: <https://components101.com/microcontrollers/esp32-devkitc>.
- [48] SiAR, «Sensor de temperatura y humedad relativa del aire,» 2017. [En línea]. Available: https://www.mapa.gob.es/es/desarrollo-rural/temas/gestion-sostenible-regadios/Sondas%20de%20T-H12_tcm30-82944.pdf.
- [49] Adafruit Learning ORG, «Adafruit SHT31-D Temperature & Humidity,» 2021. [En línea]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-sht31-d-temperature-and-humidity-sensor-breakout.pdf>.
- [50] Aosong Electronics Co. Ltd, «Digital-output relative humidity & temperature DHT22 sensor/module,» 2017. [En línea]. Available: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.
- [51] Adafruit Learning ORG, «Adafruit BME280 Humidity + Barometric Pressure + Temperature,» 2018. [En línea]. Available: <https://datasheet.octopart.com/2652-Adafruit-Industries-datasheet-129166738.pdf>.
- [52] N. Osasun, «Medidores de Dióxido de carbono (CO2),» 2021. [En línea]. Available: <https://www.navarra.es/NR/rdonlyres/B4807750-8314-45A0-9768-64D22181DBD7/471579/MEDIDORESDECO2YVENTILACIONV3.pdf>.
- [53] Hetpro Electronics, «ME0010-A SMOKE DETECTOR,» 2014. [En línea]. Available: <https://hetpro-store.com/PDFs/DYP-ME0010.pdf>.
- [54] Componentes101, «MQ-135 - Gas Sensor for Air Quality,» C101 Corp, 2018. [En línea]. Available: <https://components101.com/sensors/mq135-gas-sensor-for-air-quality>.
- [55] DF Robots, «CO2 Gas Sensor For Arduino (MG-811 Sensor),» 2016. [En línea]. Available: <https://www.dfrobot.com/product-1023.html>.
- [56] PF-SA, «Sensor de movimiento HC-SR501,» 2017, 2017. [En línea]. Available: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>.

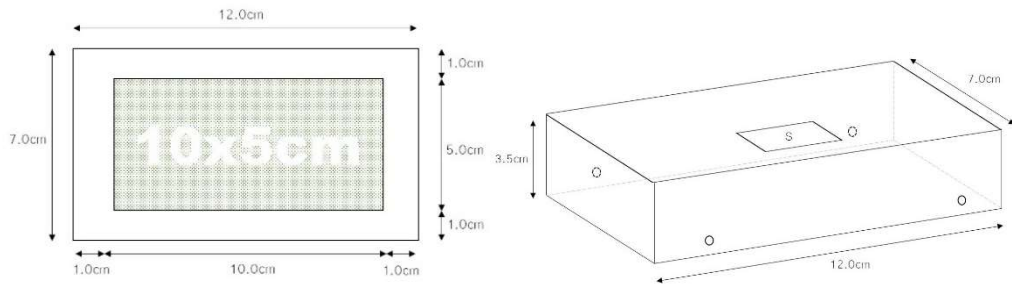
- [57] Tru Sensors, «PIR Motion Module HC-SR505,» 2016. [En línea]. Available: https://static.rapidonline.com/pdf/78-4110_v1.pdf.
- [58] A. Merabet, «Advanced Control Systems for electric drives,» DMPI Electronics, Halifax, 2020.
- [59] R. Hernandez, «Introducción a los sistemas de control,» Prentice Hall, México, 2010.
- [60] D. Cegelski, E. Katogui, A. Stoisa, o J. Corrado, «SISTEMAS DE CONTROL, La automatización como el motor del desarrollo,» Posadas, 2019.
- [61] SistemSeguridad, «Botón de emergencia o Salida sku pv2,» 2021. [En línea]. Available: <https://www.sistemseguridad.com/producto/boton-de-emergencia-o-salida/>.
- [62] S.-C. LTD., «Songle 5V Relay Datasheet,» 2021. [En línea]. Available: <https://pdfcoffee.com/5v-relay-datasheetpdf-pdf-free.html>.
- [63] S. Sánchez, «CLOUD COMPUTING: FUNDAMENTOS Y DESPLIEGE DE UN SERVICIO EN LA NUBE,» Madrid, 2021.
- [64] G. Corporation, «Descripción general de Google Cloud,» Google Cloud, 2021. [En línea]. Available: <https://cloud.google.com/docs/overview?hl=es-419>.
- [65] A. Electronics, «Micrófono USB AV 2,» 2021. [En línea]. Available: <https://avelectronics.cc/producto/microfono-usb/>.
- [66] J. Coalsen, «What is FLAC,» Xiph.Org Foundation, 2022. [En línea]. Available: <https://xiph.org/flac/>.
- [67] TP-Link Corporation Limited, «Router Wi-Fi de doble banda AC750 Archer C24,» TP-Link, 2020. [En línea]. Available: <https://www.tp-link.com/mx/home-networking/wifi-router/archer-c24/>.

ANEXOS

ANEXO A

Dimensiones de la caja del módulo de temperatura y humedad

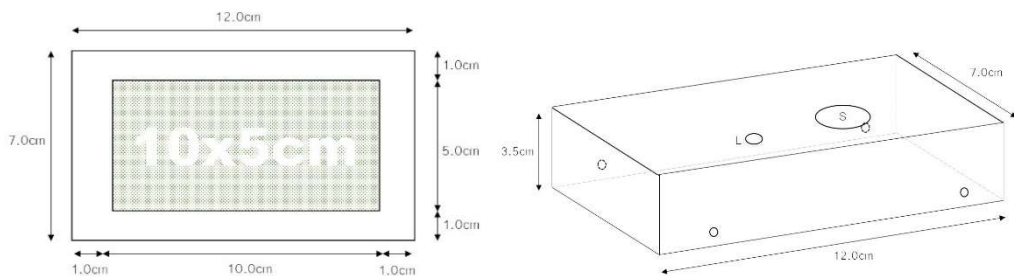
TEMPERATURA/HUMEDAD



ANEXO B

Dimensiones de la caja del módulo de detección de humo

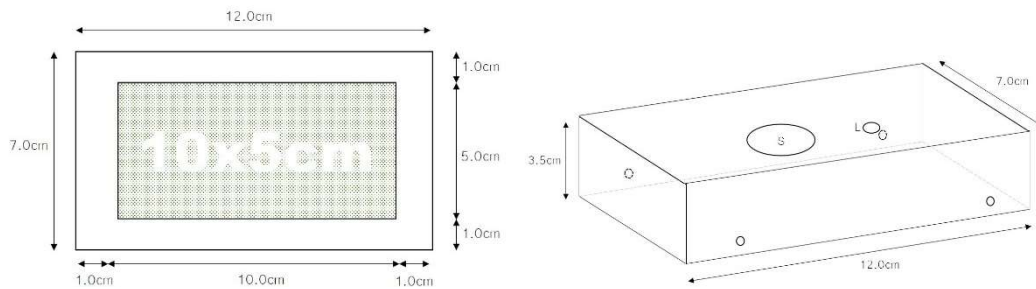
HUMO - CO2



ANEXO C

Dimensiones de la caja del módulo de detección de movimiento

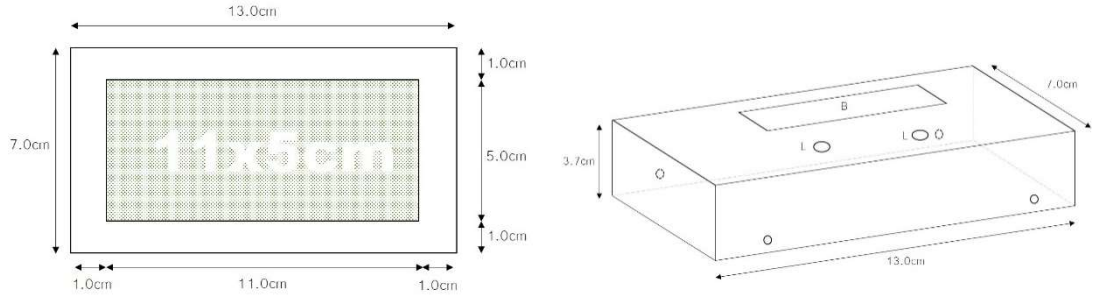
MOVIMIENTO



ANEXO D

Dimensiones de la caja del módulo de asistencia

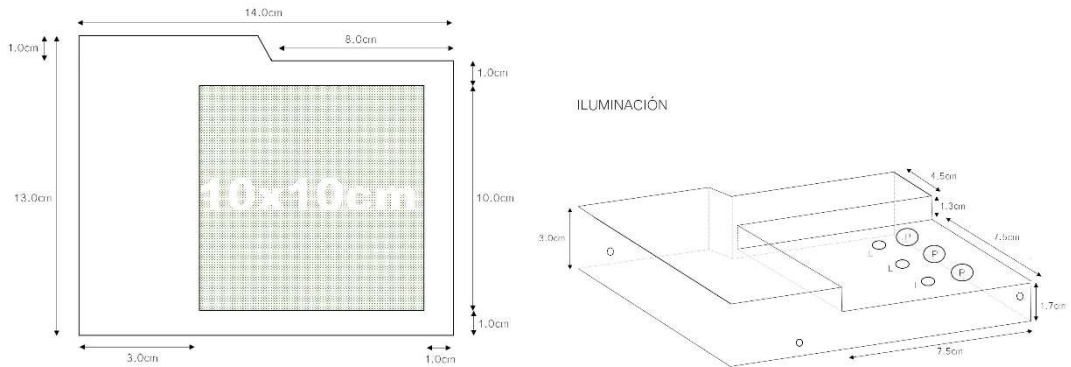
ASISTENCIA



ANEXO E

Dimensiones de la caja del módulo de control de iluminación

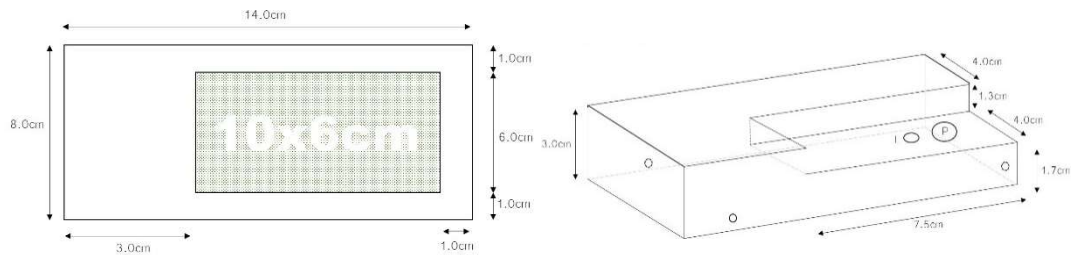
ILUMINACIÓN



ANEXO F

Dimensiones de la caja del módulo de control de calefactor / ventilador

CONTROL RELE



ANEXO G

Código de Programación del módulo de temperatura y humedad

```
//Incluimos librerias
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h> //Libreria del Sensor DHT22

#define DHT_SENSOR_PIN 12 // GPIO12 conectado al sensor PINDATA
#define DHT_SENSOR_TYPE DHT22
DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152"; //dirección de nuestro servidor cloud
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm"; //usuario creado
const char* mqttPassword = "proyecto22"; //contraseña

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);

  setup_wifi(); //Metodo de parametros de conexion WiFi

  client.setServer(mqtt_server, mqtt_port);
  dht_sensor.begin(); //Inicializamos el Sensor
}

void setup_wifi() {
  delay(10);
  //Iniciamos la Conexión a la RED WIFI
```



```

Serial.println();
Serial.print("Conectando con: ");
Serial.println(ssid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("Direccion IP: "); //indica la ip del dispositivo
Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando Broker MQTT... ");
    //Intento de Conexión
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX); //id-cliente
    if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
      Serial.println("CONECTADO con ID: "+clientId);
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente...");
      delay(5000);
    }
  }
}

void loop() {
  //Siempre verifica que mantenga la conexion con el Broker
  if (!client.connected()) {
    reconnect();
  }

  client.loop();
}

```

```

//Acondicionamiento de Datos del Sensor
float hum = dht_sensor.readHumidity(); //Lectura Datos de Humedad
float tempC = dht_sensor.readTemperature(); //Lectura Datos Temp Centigrados
if ( isnan(tempC) || isnan(hum)) { //Si la Lectura es errónea
  Serial.println("Error en Lectura del Sensor!!!");
}else{
  Serial.print("Humedad Relativa: ");
  Serial.print(hum);
  Serial.print("%");
  Serial.print(" | ");
  Serial.print("Temperatura: ");
  Serial.print(tempC);
  Serial.println("°C ");
}

//Convertimos el valor bruto a Vector - TEMPERATURA(temp)
char tempString[8];
dtostrf(tempC, 1, 2, tempString);
client.publish("ciam/hab1/temperatura", tempString);

//Convertimos el valor bruto a Vector - HUMEDAD(hum)
char humString[8];
dtostrf(hum, 1, 2, humString);
client.publish("ciam/hab1/humedad", humString);

delay(5000);

}

```

ANEXO H

Código de Programación del módulo de detección de humo

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

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
////////Declaracion de Pines y Variables de Lectura
int pinaviso = 32; // GPIO32 de la ESP32 (LED o BUZZER)
int gasA0 = 12; //Pin Lectura analoga GPIO12 de la ESP32
int gasD0 = 14; //Pin Lectura digital GPIO14 de la ESP32

void setup() {
Serial.begin(115200);
pinMode(pinaviso, OUTPUT); //indicador
pinMode(gasA0, INPUT); //lectura analoga del sensor
pinMode(gasD0, INPUT); //lectura digital del sensor

setup_wifi();
client.setServer(mqtt_server, mqtt_port);
}

void setup_wifi() {
delay(10);
//Iniciamos la Conexión a la RED WIFI
Serial.println();
Serial.print("Conectando con: ");
Serial.println(ssid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
}
```

```

Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("Direccion IP: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando Broker MQTT... ");
    //Intento de Conexión
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
      Serial.println("CONECTADO con ID: "+clientId);
      // Subscribe no requerido
      //client.subscribe("ciam/hab1/humo");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente en 5seg");
      delay(5000);
    }
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  client.loop();

  int gassensorDigital = digitalRead(gasD0);
  Serial.print("Presencia de HUMO: ");
  Serial.print("\t");

  if(gassensorDigital != 1) {
    Serial.println("Si, aviso a sistema");
    client.publish("ciam/hab1/humo", "1");
    for(int i = 0; i<=60; i++){ //parametro duracion de aviso
      digitalWrite (pinaviso, HIGH) ;
      delay(500);
      digitalWrite (pinaviso, LOW) ;
      delay(500);
    }
  }
}

```

```

    Serial.println(i);
  }
  client.publish("ciam/hab1/humo","0");
} else {
  Serial.println(" No ");
  client.publish("ciam/hab1/humo","0");
}
}
delay(2000);
}

```

ANEXO I

Código de Programación del módulo de detección de movimiento

```

//Incluimos librerias
#include <WiFi.h>
#include <PubSubClient.h>

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

const int pinSensor = 34; //PIN GPIO43 Lectura D0 del sensor
int estadoD0 = 0;
int pinaviso = 32;

void setup() {
  Serial.begin(115200);
  pinMode(pinaviso, OUTPUT);
  pinMode(pinSensor, INPUT);

  setup_wifi(); //Metodo de parametros de conexion WiFi

  client.setServer(mqtt_server, mqtt_port);
}

```

```

void setup_wifi() {
  delay(10);
  //Iniciamos la Conexión a la RED WIFI
  Serial.println();
  Serial.print("Conectando con: ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi Conectado");
  Serial.println("Direccion IP: ");
  Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando Broker MQTT... ");
    //Intento de Conexión
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
      Serial.println("CONECTADO con ID: "+clientId);
      // Subscribe no requerido
      //client.subscribe("ciam/hab1/movimiento");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente en 5seg");
      delay(5000);
    }
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  client.loop();
}

```

```
estadoD0 = digitalRead(pinSensor);
Serial.print(" Movimiento Detectado: " +estadoD0);
if (estadoD0 == 1) { //Si Detecta movimiento
  client.publish("ciam/hab1/movimiento","1");
  for(int i =0; i <=10; i++){ //duracion de aviso
    digitalWrite (pinaviso, HIGH) ;
    delay(500);
    digitalWrite (pinaviso, LOW) ;
    delay(500);
    Serial.println(i);
  }
} else{
  client.publish("ciam/hab1/movimiento","0");
}
delay(1000);
}
```

ANEXO J

Código de Programación del módulo de asistencia

```
//Incluimos librerias
#include <WiFi.h>
#include <PubSubClient.h>

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

const int pinST = 12; //PIN GPIO12 estado inicial
const int pinPulsador = 34; //PIN GPIO34 Lectura Pulsador
int estadoP = 0;
int pinaviso = 32; //PIN GPIO12 Indicador

void setup() {
  Serial.begin(115200);
  pinMode(pinaviso, OUTPUT);
  pinMode(pinST, OUTPUT);
  digitalWrite(pinST, HIGH);
  pinMode(pinPulsador, INPUT);

  setup_wifi(); //Metodo de parametros de conexion WiFi

  client.setServer(mqtt_server, mqtt_port);
}

void setup_wifi() {
  delay(10);
  //Iniciamos la Conexión a la RED WIFI
  Serial.println();
  Serial.print("Conectando con: ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```



```

delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("Direccion IP: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
while (!client.connected()) {
Serial.print("Conectando Broker MQTT... ");
//Intento de Conexión
String clientId = "ESP32Client-";
clientId += String(random(0xffff), HEX);
if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
Serial.println("CONECTADO con ID: "+clientId);
// Subscribe no requerido
//client.subscribe("ciam/hab1/pulsador2");
} else {
Serial.print("Fallo, rc=");
Serial.print(client.state());
Serial.println(" intentando nuevamente en 5seg");
delay(5000);
}
}
}

void loop() {
if (!client.connected()) {
reconnect();
}

client.loop();

estadoP = digitalRead(pinPulsador);
Serial.print(" Estado del Pulsador de Asistencia: ");
Serial.println(estadoP);

if (estadoP == LOW) { //Si se ha presionado, detecta 0
client.publish("ciam/hab1/asistencia1","1");
Serial.println("Enviando Aviso al Sistema");
for(int i =0; i <=60; i++){ //Duracion de Aviso
digitalWrite (pinaviso, HIGH) ;
delay(500);
}
}
}

```

```

digitalWrite (pinaviso, LOW) ;
delay(500);
Serial.println(i);
}
client.publish("ciam/hab1/asistencia1","0");
} else {
client.publish("ciam/hab1/asistencia1","0");
}
delay(1500);
}

```

ANEXO K

Código de Programación del módulo de control de iluminación

```

//Incluimos librerias
#include <WiFi.h>
#include <PubSubClient.h>

////////Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

////////Declaracion de Pines y Variables de Lectura
const int pinCTRL1 = 12; //PIN que envía pulso CLK al FF1
const int pinQ1 = 34; //PIN Lectura Q del FF para conocer estado ON/OFF del
RELE
int estadoQ1 = 0;
const int pinCTRL2 = 14; //PIN que envía pulso CLK al FF2
const int pinQ2 = 39; //PIN Lectura Q del FF para conocer estado ON/OFF del
RELE
int estadoQ2 = 0;
const int pinCTRL3 = 27; //PIN que envía pulso CLK al FF3

```

```

const int pinQ3 = 36;    //PIN Lectura Q del FF para conocer estado ON/OFF del
RELE
int estadoQ3 = 0;

void setup() {
  Serial.begin(115200);
  pinMode(pinCTRL1, OUTPUT); //GPIO 12 como salida, inicia en alto
  digitalWrite(pinCTRL1, HIGH);
  pinMode(pinQ1, INPUT);    //GPIO 34 como entrada/lectura
  pinMode(pinCTRL2, OUTPUT); //GPIO 14 como salida, inicia en alto
  digitalWrite(pinCTRL2, HIGH);
  pinMode(pinQ2, INPUT);    //GPIO 39 como entrada/lectura
  pinMode(pinCTRL3, OUTPUT); //GPIO 17 como salida, inicia en alto
  digitalWrite(pinCTRL3, HIGH);
  pinMode(pinQ3, INPUT);    //GPIO 36 como entrada/lectura

  setup_wifi();            //Metodo de parametros de conexion WiFi

  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}

void setup_wifi() {
  delay(10);
  //Iniciamos la Conexión a la RED WIFI
  Serial.println();
  Serial.print("Conectando con: ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi Conectado");
  Serial.println("Direccion IP: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Message arrived on topic: ");
  Serial.print(topic);
  Serial.print(". Message: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {

```

```

Serial.print((char)message[i]);
messageTemp += (char)message[i];
}
Serial.println();

if(String(topic) == "ciam/hab1/iluminacion/ret1") {
  Serial.print("Realizando cambio de estado R1 con: ");
  if(messageTemp == "sw"){
    Serial.println("sw");
    digitalWrite(pinCTRL1, HIGH);
    delay(500);
    digitalWrite(pinCTRL1, LOW);
    delay(500);
    digitalWrite(pinCTRL1, HIGH);
  }else if(messageTemp != "sw"){
    Serial.println("sin cambio Rele 1");
  }
}

if(String(topic) == "ciam/hab1/iluminacion/ret2") {
  Serial.print("Realizando cambio de estado R2 con: ");
  if(messageTemp == "sw"){
    Serial.println("sw");
    digitalWrite(pinCTRL2, HIGH);
    delay(500);
    digitalWrite(pinCTRL2, LOW);
    delay(500);
    digitalWrite(pinCTRL2, HIGH);
  }else if(messageTemp != "sw"){
    Serial.println("sin cambio Rele 2");
  }
}

if(String(topic) == "ciam/hab1/iluminacion/ret3") {
  Serial.print("Realizando cambio de estado R3 con: ");
  if(messageTemp == "sw"){
    Serial.println("sw");
    digitalWrite(pinCTRL3, HIGH);
    delay(500);
    digitalWrite(pinCTRL3, LOW);
    delay(500);
    digitalWrite(pinCTRL3, HIGH);
  }else if(messageTemp != "sw"){
    Serial.println("sin cambio Rele 3");
  }
}

void reconnect() {

```

```

while (!client.connected()) {
  Serial.print("Conectando Broker MQTT... ");
  //Intento de Conexión
  String clientId = "ESP32Client-";
  clientId += String(random(0xffff), HEX);
  if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
    Serial.println("CONECTADO con ID: "+clientId);
    // Subscribe
    client.subscribe("ciam/hab1/iluminacion/#");
  } else {
    Serial.print("Fallo, rc=");
    Serial.print(client.state());
    Serial.println(" intentando nuevamente en 5seg");
    delay(5000);
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  client.loop();

  estadoQ1 = digitalRead(pinQ1);
  Serial.println(estadoQ1);
  //Revisa el Estado del Relé, si esta encendido o apagado
  if (estadoQ1 == HIGH){
    //1:encendido
    client.publish("ciam/hab1/stret1","1");
  } else {
    //0:apagado
    client.publish("ciam/hab1/stret1","0");
  }

  estadoQ2 = digitalRead(pinQ2);
  Serial.println(estadoQ2);
  //Revisa el Estado del Relé, si esta encendido o apagado
  if (estadoQ2 == HIGH){
    //1:encendido
    client.publish("ciam/hab1/stret2","1");
  } else {
    //0:apagado
    client.publish("ciam/hab1/stret2","0");
  }

  estadoQ3 = digitalRead(pinQ3);
  Serial.println(estadoQ3);

```

```

//Revisa el Estado del Relé, si esta encendido o apagado
if (estadoQ3 == HIGH){
  //1:encendido
  client.publish("ciam/hab1/stret3","1");
} else {
  //0:apagado
  client.publish("ciam/hab1/stret3","0");
}

delay(1000);

}

```

ANEXO L

Código de Programación del módulo del calefactor

```

#include <WiFi.h>
#include <PubSubClient.h>

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
////////Declaracion de Pines y Variables de Lectura
const int pinCTRL1 = 12; //PIN GPIO12 pulso para cambio de estado
const int pinQ1 = 34; //PIN GPIO34 Lectura Q conocer estado ON/OFF del
RELE
int estadoQ1 = 0;

void setup() {
  Serial.begin(115200);
  pinMode(pinCTRL1, OUTPUT); //GPIO 12 como salida, inicia en flanco alto FF
  digitalWrite(pinCTRL1, HIGH); //pasa a flanco bajo FF
  pinMode(pinQ1, INPUT); //GPIO 34 como entrada/lectura
}

```

```

    setup_wifi();
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);

}

void setup_wifi() {
    delay(10);
    //Iniciamos la Conexión a la RED WIFI
    Serial.println();
    Serial.print("Conectando con: ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi Conectado");
    Serial.println("Direccion IP: ");
    Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Dato recibido desde el topic: ");
    Serial.print(topic);
    Serial.print(" - Dato: ");
    String messageTemp;

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.println();
    //***** Lectura del Dato Recibido
    if(String(topic) == "ciam/hab1/ambiente/relecal") {
        Serial.print("Realizando cambio de estado con ");
        if(messageTemp == "sw"){
            Serial.println("sw");
            digitalWrite(pinCTRL1, HIGH);
            delay(500);
            digitalWrite(pinCTRL1, LOW);
            delay(500);
            digitalWrite(pinCTRL1, HIGH);
        }else if(messageTemp != "sw"){
            Serial.println("sin cambio");
        }
    }
}

```

```

    }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando Broker MQTT... ");
    //Intento de Conexión
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
      Serial.println("CONECTADO con ID: "+clientId);
      // Subscribe
      client.subscribe("ciam/hab1/ambiente/relecal");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente en 5seg");
      delay(5000);
    }
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  client.loop();

  estadoQ1 = digitalRead(pinQ1);
  Serial.println(estadoQ1);
  //Revisa el Estado del Relé, si esta encendido o apagado
  if (estadoQ1 == HIGH){
    //1:encendido
    client.publish("ciam/hab1/stcalefactor","1");
  } else {
    //0:apagado
    client.publish("ciam/hab1/stcalefactor","0");
  }

  delay(1000);
}

```


ANEXO M

Código de programación del módulo de control del ventilador

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

//Declaramos Credenciales de la RED a la que nos conectaremos
const char* ssid = "Red_Proyecto";
const char* password = "proyectodom22";

//Agregamos la configuracion del Broker MQTT
const char* mqtt_server = "54.148.93.152";
const int mqtt_port = 1883;
const char* mqttUser = "ciamjm";
const char* mqttPassword = "proyecto22";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
////////Declaracion de Pines y Variables de Lectura
const int pinCTRL1 = 12; //PIN GPIO12 pulso para cambio de estado
const int pinQ1 = 34; //PIN GPIO34 Lectura Q conocer estado ON/OFF del
RELE
int estadoQ1 = 0;

void setup() {
  Serial.begin(115200);
  pinMode(pinCTRL1, OUTPUT); //GPIO 12 como salida, inicia en flanco alto FF
  digitalWrite(pinCTRL1, HIGH); //pasa a flanco bajo FF
  pinMode(pinQ1, INPUT); //GPIO 34 como entrada/lectura

  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}

void setup_wifi() {
  delay(10);
  //Iniciamos la Conexión a la RED WIFI
```

```

Serial.println();
Serial.print("Conectando con: ");
Serial.println(ssid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("Direccion IP: ");
Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Dato recibido desde el topic: ");
  Serial.print(topic);
  Serial.print(" - Dato: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
  Serial.println();
  //***** Lectura del Dato Recibido
  if(String(topic) == "ciam/hab1/ambiente/releven") {
    Serial.print("Realizando cambio de estado con ");
    if(messageTemp == "sw"){
      Serial.println("sw");
      digitalWrite(pinCTRL1, HIGH);
      delay(500);
      digitalWrite(pinCTRL1, LOW);
      delay(500);
      digitalWrite(pinCTRL1, HIGH);
    }else if(messageTemp != "sw"){
      Serial.println("sin cambio");
    }
  }
}
}

```

```

void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando Broker MQTT... ");
    //Intento de Conexión
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str(), mqttUser, mqttPassword)) { //WiFiClient
      Serial.println("CONECTADO con ID: "+clientId);
      // Subscribe
      client.subscribe("ciam/hab1/ambiente/releven");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente en 5seg");
      delay(5000);
    }
  }
}

```

```

void loop() {
  if (!client.connected()) {
    reconnect();
  }

```

```

  client.loop();

```

```

estadoQ1 = digitalRead(pinQ1);
Serial.println(estadoQ1);
//Revisa el Estado del Relé, si esta encendido o apagado
if (estadoQ1 == HIGH){
  //1:encendido
  client.publish("ciam/hab1/stventilador","1");
} else {
  //0:apagado
  client.publish("ciam/hab1/stventilador","0");
}

```

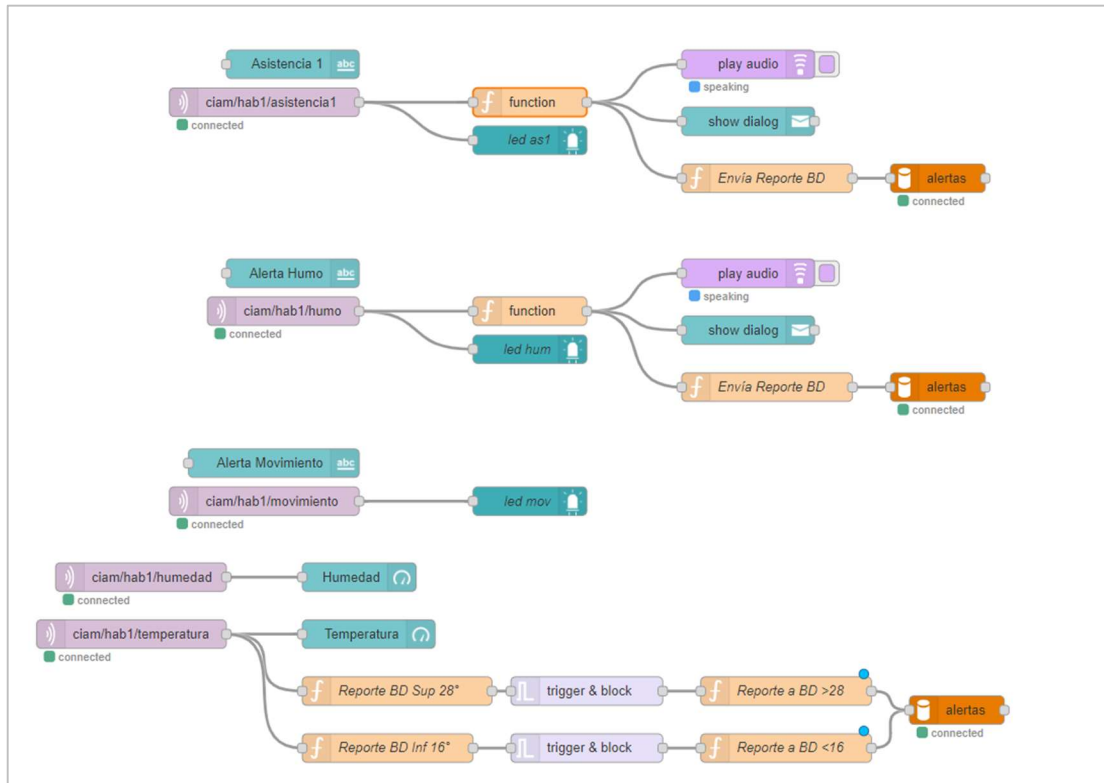
```

delay(1000);
}

```

ANEXO N

Configuración general de la interfaz de monitoreo



ANEXO O

Programación del nodo function “Calefacción Automática”

```
context.data = context.data || {};  
  
switch (msg.topic){  
  case "ciam/hab1/temperatura":  
    context.data.temperatura = msg.payload;  
    msg = null;  
    break;  
  case "calefaccion":  
    context.data.automatico = msg.payload;  
    msg = null;  
    break;  
  case "ciam/hab1/scalefactor":  
    context.data.estado = msg.payload;  
    msg = null;  
    break;  
  default:  
    msg = null;  
    break;  
}  
  
var temp = parseFloat(context.data.temperatura);  
var swcal = context.data.automatico;  
var est = context.data.estado;  
var control = "  
  
//Cuando la temperatura es menor a 16 grados  
if(temp <= 16 && swcal == '1'){  
  if(est == '1'){          //Si ya está encendido: SIN CAMBIOS  
    control = '0';        //Envía 0 para reset Trigger  
    return {payload: control};  
  }else if(est == '0'){   //Si esta apagado: HAY CAMBIOS(ENCIENDE)  
    control = '1';        //Activa Trigger  
    return {payload: control};  
  }  
}  
}  
  
///  
//Cuando la temperatura es mayor a 16 grados  
if(temp > 16 && swcal == '1'){  
  if(est == '1'){          //Si esta encendido: HAY CAMBIOS(APAGA)
```

```

        control = '1';          //Activa Trigger
        return {payload: control};
    }else if(est == '0'){      //Si ya está encendido: SIN CAMBIOS
        control = '0';        //Envía 0 para reset Trigger
        return {payload: control};
    }
}
}

```

ANEXO P

Programación del nodo function “Ventilación Automática”

```

context.data = context.data || {};

switch (msg.topic){
    case "ciam/hab1/temperatura":
        context.data.temperatura = msg.payload;
        msg = null;
        break;
    case "ventilacion":
        context.data.automatico = msg.payload;
        msg = null;
        break;
    case "ciam/hab1/stventilador":
        context.data.estado = msg.payload;
        msg = null;
        break;
    default:
        msg = null;
        break;
}

var temp = parseFloat(context.data.temperatura);
var swven = context.data.automatico;
var est = context.data.estado;
var control = "";

```

```

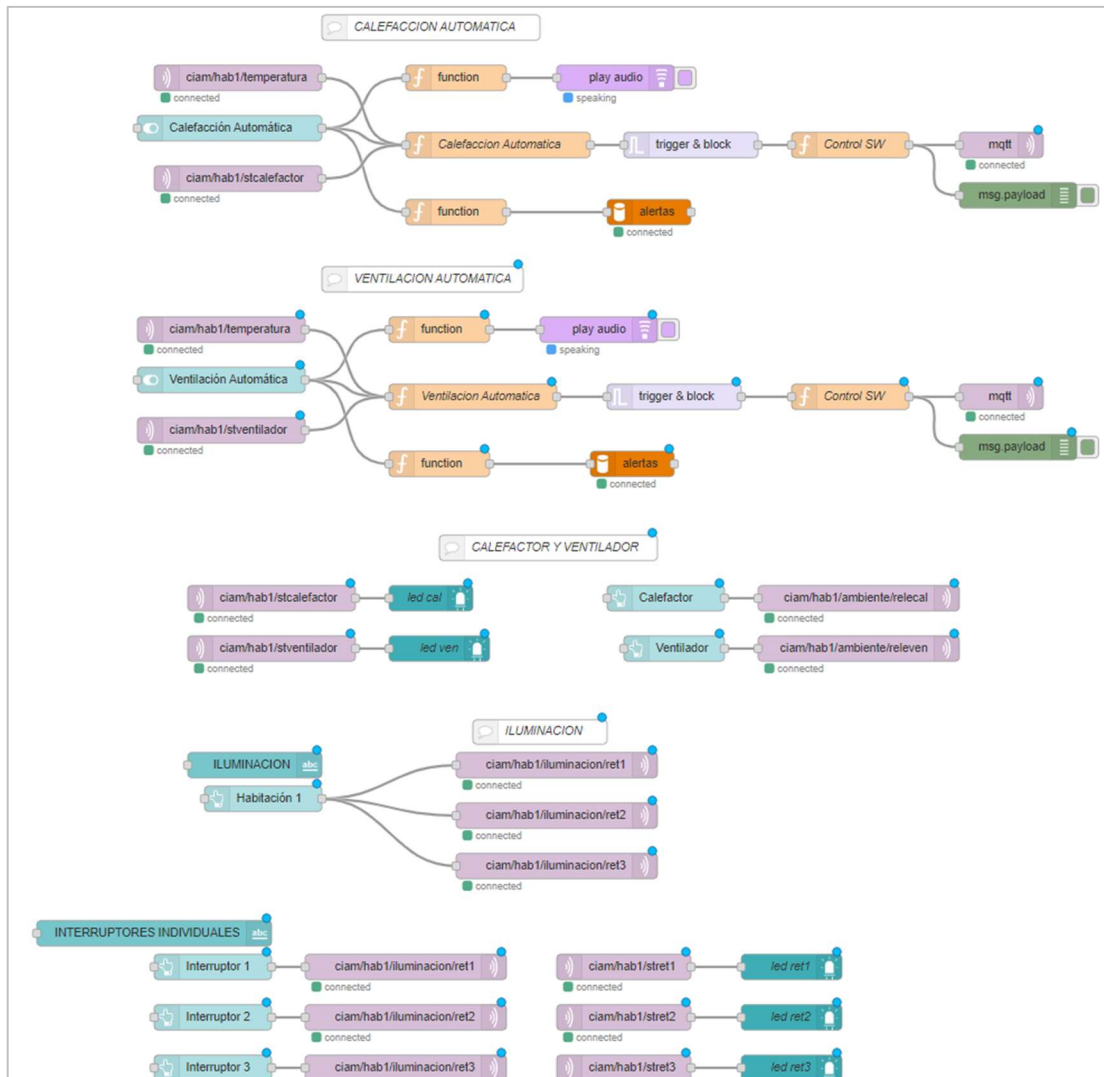
//Cuando la temperatura es mayor a 28 grados
if(temp >= 28 && swven == '1'){
    if(est == '1'){          //Si ya está encendido: SIN CAMBIOS
        control = '0';      //Envía 0 para reset Trigger
        return {payload: control};
    }else if(est == '0'){   //Si esta apagado: HAY CAMBIOS(ENCIENDE)
        control = '1';      //Activa Trigger
        return {payload: control};
    }
}

////Cuando la temperatura es inferior a 28 grados
if(temp < 28 && swven == '1'){
    if(est == '1'){          //Si esta encendido: HAY CAMBIOS(APAGA)
        control = '1';      //Activa Trigger
        return {payload: control};
    }else if(est == '0'){   //Si ya está encendido: SIN CAMBIOS
        control = '0';      //Envía 0 para reset Trigger
        return {payload: control};
    }
}

```

ANEXO Q

Configuración general de la interfaz de control de módulos



ANEXO R

Código de programación del archivo “enlazar.php”

```
<?php
$connect = mysqli_connect("localhost", "ciam", "proyecto22", "alertas");//Configurar los
datos de conexion
$columns = array('id', 'descripcion', 'fecha');
$query = "SELECT * FROM reportes WHERE ";

if($_POST["is_date_search"] == "yes")
{
    $query .= 'fecha BETWEEN "' . $_POST["start_date"] . '" AND "' . $_POST["end_date"] . '"
AND ';
}

if(isset($_POST["search"]["value"]))
{
    $query .= '
(id LIKE "%' . $_POST["search"]["value"] . '%"
OR descripcion LIKE "%' . $_POST["search"]["value"] . '%"
';
}

if(isset($_POST["order"]))
{
    $query .= 'ORDER BY ' . $columns[$_POST["order"][0]['column']] . '
' . $_POST["order"][0]['dir'] . '
';
}
else
{
    $query .= 'ORDER BY id DESC ';
}

$query1 = ";

if($_POST["length"] != -1)
{
    $query1 = 'LIMIT ' . $_POST["start"] . ', ' . $_POST["length"];
}

$number_filter_row = mysqli_num_rows(mysqli_query($connect, $query));

$result = mysqli_query($connect, $query . $query1);

$data = array();
```

```

while($row = mysqli_fetch_array($result))
{
    $fecha=date("d/m/Y H:i:s", strtotime($row["fecha"]));
    $sub_array = array();
    $sub_array[] = $row["id"];
    $sub_array[] = $row["descripcion"];
    $sub_array[] = $fecha;

    $data[] = $sub_array;
}

function get_all_data($connect)
{
    $query = "SELECT * FROM reportes";
    $result = mysqli_query($connect, $query);
    return mysqli_num_rows($result);
}

$output = array(
    "draw" => intval($_POST["draw"]),
    "recordsTotal" => get_all_data($connect),
    "recordsFiltered" => $number_filter_row,
    "data" => $data
);

echo json_encode($output);

?>

```

ANEXO S

Archivo “Reconocim_voz.py” – Interfaz gráfica

```

import tkinter as tk
import os, sys
from tkterminal import Terminal

root = tk.Tk()
#terminal = Terminal(width = 5, height = 10)

terminal = Terminal(pady=5, padx=5)

encabezado = tk.Label(root, text = "RECONOCIMIENTO DE VOZ", font =
30,fg='#f00')
encabezado.pack()

parrfo1 = tk.Label(root, text = "Click en Iniciar para Comenzar:", font = 5)
parrfo1.pack()

```

```

indicaciones0 = tk.Label(root, text = "Recuerde pronunciar con claridad cada una de
las órdenes.", font = 5)
indicaciones0.pack()

indicaciones1 = tk.Label(root, text = "Una vez reconocida la orden, ésta se mostrará
en pantalla.", font = 5)
indicaciones1.pack()

terminal.shell = True
terminal.linebar = True
terminal.pack(expand=True, fill='both')
b1 = tk.Button(
    root, text="Iniciar", fg="Black",
    command=lambda: terminal.run_command('python3 Ordenes_por_voz.py'))
b1.pack()
b2 = tk.Button(
    root, text="Salir", fg="Black",
    command=root.destroy)
b2.pack()

root.mainloop()

```

ANEXO T

Archivo “Ordenez_por_voz.py” – Reconocimiento de voz y conexión con broker MQTT del servidor en la nube

```

import speech_recognition as sr
import paho.mqtt.client as mqtt

from datetime import date
from time import sleep

#Definimos cada una de las frases para ser reconocidas
hab1_sw = "cambia el estado de habitación 1"
cal1_sw = "cambia el estado del calefactor 1"
ven1_sw = "cambia el estado del ventilador 1"

client = mqtt.Client();
client.username_pw_set(username="ciamjm",password="proyecto22")
print("Connecting...")
client.connect("54.148.93.152", 1883, 60);
sleep(1)
print("Conectado al Broker MQTT en la nube")

```

```

#obtener el audio del microfono USB
r = sr.Recognizer()
mic = sr.Microphone()

print("Escuchando Audio en Español [MX]")
while True:
    with mic as source:
        #r.adjust_for_ambient_noise(source, duration=1) #configuracion de ruido no
necesario
        audio = r.adjust_for_ambient_noise(source)
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        words = r.recognize_google(audio, language = 'es-MX')
        #print(words)

        #COMIENZA RECONOCIMIENTO DE FRASES
        if hab1_sw in words:
            client.publish("ciam/hab1/iluminacion/ret1", "sw")
            client.publish("ciam/hab1/iluminacion/ret2", "sw")
            client.publish("ciam/hab1/iluminacion/ret3", "sw")
            print("Ejecutando Orden:"+words)

        if call_sw in words:
            client.publish("ciam/hab1/ambiente/relecal", "sw")
            print("Ejecutando Orden:"+words)

        if ven1_sw in words:
            client.publish("ciam/hab1/ambiente/releven", "sw")
            print("Ejecutando Orden:"+words)

        if words == "fecha":
            print(date.today())

        if words == "prueba":
            print("has dicho prueba")

        if words == "adios":
            print("...")
            sleep(1)
            print("...")
            sleep(1)
            print("...")
            sleep(1)
            print("Goodbye")
            break

    except sr.UnknownValueError:

```

```
print("Esperando Voz...")
except sr.RequestError as e:
    print("API sin respuesta desde Google; {0}".format(e))
```