



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE INGENIERÍA ELECTRÓNICA

*Implementación De Una Interfaz Gráfica De Control Domótico Usando
HMI y ESP32.*

**Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica**

AUTORES: Rodríguez Vines Jerry Alberto

Suarez Arreaga Jordán David

TUTOR: Ing. Vicente Avelino Peñaranda Idrovo MSc.

Guayaquil – Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL
TRABAJO DE TITULACIÓN**

Nosotros, Rodríguez Vínces Jerry Alberto con documento de identificación N° 0950175422 y Suarez Arreaga Jordán David con documento de identificación N° 0919349720; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 25 de febrero de 2025

Atentamente,



Rodríguez Vínces Jerry Alberto
0950175422



Suarez Arreaga Jordán David
0919349720

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN
A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Rodríguez Vinces Jerry Alberto con documento de identificación N° 0950175422 y Suarez Arreaga Jordán David con documento de identificación N° 0919349720, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del proyecto de investigación: "Implementación De Una Interfaz Gráfica De Control Domótico Usando HMI y ESP32.", el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 25 de febrero de 2025

Atentamente,



Rodríguez Vinces Jerry Alberto
0950175422



Suarez Arreaga Jordán David
0919349720

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Vicente Avelino Peñaranda Idrovo con documento de identificación N° 0916113426, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UNA INTERFAZ GRÁFICA DE CONTROL DOMÓTICO USANDO HMI Y ESP32. realizado por Jerry Alberto Rodríguez Vinces con documento de identificación N° 0950175422 y Jordán David Suarez Arreaga con documento de identificación N° 0919349720, obteniendo como resultado final el trabajo de titulación bajo la opción tesis que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 25 de febrero de 2025

Atentamente,



Ing. Vicente Avelino Peñaranda Idrovo MSc.

CI: 0916113426

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de esta tesis. En primer lugar, agradezco al Ing. Vicente Avelino Peñaranda Idrovo MSc, mi tutor, por su orientación, apoyo y paciencia durante todo el proceso. Sus valiosos consejos y conocimientos han sido fundamentales para el desarrollo de este trabajo. Agradezco profundamente a mi compañero de investigación, Jordán David Suárez Arreaga, quien me acompañó durante todo el proyecto, brindándome su apoyo en la recolección de datos, análisis y desarrollo de ideas clave. Su colaboración ha sido esencial para alcanzar los objetivos propuestos.

Quiero hacer un reconocimiento especial a mi madre, Ana Maiden Vincés Pérez, por su amor incondicional, su apoyo constante y por ser mi principal fuente de fortaleza. Su comprensión y motivación me han impulsado a lo largo de todo este proceso, y sin ella, este logro no hubiera sido posible.

También quiero agradecer a mi familia, por su amor y apoyo constante, y a mis amigos y compañeros de trabajo, quienes me han ofrecido su apoyo, ideas y ánimo en todo momento. Finalmente, agradezco a todas las personas que, con su colaboración y apoyo, hicieron posible el desarrollo de esta investigación. A cada uno de los que brindaron su tiempo, conocimientos y recursos, contribuyendo significativamente a la realización de este trabajo. Sin su ayuda, este proyecto no habría alcanzado los resultados esperados.

RODRÍGUEZ VINCÉS JERRY ALBERTO

AGRADECIMIENTO

Este trabajo no habría sido posible sin la colaboración de muchas personas a quienes me gustaría expresar mi más sincero reconocimiento.

A Dios por darme salud y cuidarme hasta ahora en este proceso, y así poder concluir un objetivo más en mi vida personal, que es ser un profesional.

A familia, por ser mi mayor sostén. Su amor, comprensión y respaldo han sido esenciales para que pudiera enfrentar los desafíos de este proyecto. Sin ellos, nada de esto habría sido posible.

A mis amigos y compañeros, por su apoyo constante, su ánimo y por ser una fuente de inspiración durante este proceso. Las conversaciones, los intercambios de ideas, las risas y el simple hecho de estar ahí, hicieron más llevadero este camino.

A mi tutor de tesis, Ing. Vicente Avelino Peñaranda Idrovo MSc., por su invaluable ayuda, por su paciencia y por siempre brindarme los recursos y consejos necesarios para realizar este trabajo. Su presencia ha sido clave en este proceso.

Finalmente, agradezco a todas aquellas personas que, aunque no mencionadas directamente, han sido parte importante de mi vida académica y personal, ayudándome a seguir adelante y creer en mí.

SUAREZ ARREAGA JORDÁN DAVID

RESUMEN DEL PROYECTO

Año	Alumnos	Tutor de Proyecto de titulación	Proyecto de titulación
2024	Rodríguez Vínces Jerry Alberto y Suarez Arreaga Jordán David	Ing. Vicente Peñaranda MSc.	Implementación De Una Interfaz Gráfica De Control Domótico Usando HMI y ESP32.

El propósito fundamental de este proyecto de titulación es crear e implementar una interfaz gráfica para el control domótico, empleando HMI (Interfaz Hombre-Máquina) y ESP32. La elaboración de este tipo de prototipos facilita el desarrollo de futuras investigaciones que pueden ser llevadas a cabo por estudiantes, docentes e investigadores en el campo de la ingeniería electrónica, permitiendo la creación de aplicaciones domóticas tanto para el sector industrial como para el ámbito doméstico.

Se desarrolló un prototipo domótico aplicable al hogar donde se controla diferentes sensores, como sensor de temperatura, sensor de movimiento, sensor de gas, sensor de humedad relativa, sensor de humedad de suelo, los parámetros obtenidos por estos sensores se monitorizan y se visualizan en una interfaz HMI para el control y monitoreo dentro del hogar.

Para el desarrollo de este proyecto se utilizó una interfaz HMI (Interfaz maquina humano) del fabricante Elecrow , pantalla de 7 pulgadas con ESP32 integrado, una pcb socket para un ESP32 y varios sensores como el DHT22 para el monitoreo de temperatura y humedad, sensor MQ-2 para la detección de gases, sensor detección de movimiento HC-SR501, sensor de humedad de suelo y varios relés mecánicos para el accionamiento de luces de dormitorios, cocina, sala, estudio, una alarma de control de intrusos y puerta eléctrica para el accionamiento remoto del ingreso al domicilio. El ESP32 se encarga de procesar todos los datos que se obtienen de los sensores, dichos datos son enviados a la pantalla HMI para la visualización de la información.

ABSTRACT

Year	Students	Degree Project Tutor	Technical Degree Project
2024	Rodríguez Vinces Jerry Alberto y Suarez Arreaga Jordán David	Ing. Vicente Peñaranda MSc.	Implementation of a Graphical Home Automation Control Interface Using HMI and ESP32

The fundamental purpose of this graduation project is to create and implement a graphical interface for home automation control, using HMI (Human-Machine Interface) and ESP32. The development of this type of prototype facilitates future research that can be carried out by students, teachers, and researchers in the field of electronic engineering, allowing the creation of home automation applications for both the industrial sector and domestic environments.

A home automation prototype was developed where different sensors are controlled, such as a temperature sensor, motion sensor, gas sensor, relative humidity sensor, and soil moisture sensor. The parameters obtained from these sensors are monitored and displayed on an HMI interface for control and monitoring within the home.

For the development of this project, an HMI (Human-Machine Interface) from the manufacturer Elecrow was used, featuring a 7-inch screen with an integrated ESP32, a PCB socket for an ESP32, and several sensors such as the DHT22 for monitoring temperature and humidity, the MQ-2 sensor for gas detection, the HC-SR501 motion detection sensor, a soil moisture sensor, and several mechanical relays for controlling lights in the bedrooms, kitchen, living room, study, an intruder alarm, and an electric door for remote access to the home. The ESP32 is responsible for processing all the data obtained from the sensors, and this data is sent to the HMI screen for information visualization.

ÍNDICE GENERAL

INTRODUCCIÓN	1
1 El problema	2
1.1 Descripción del problema	2
1.2 Antecedentes	2
1.3 Importancia y alcance	3
1.4 Delimitación	3
1.4.1 Delimitación temporal	3
1.4.2 Delimitación espacial	3
1.4.3 Delimitación académica	3
1.5 Beneficiarios de la propuesta	4
1.6 Propuesta de solución	4
1.7 Innovación e impacto del proyecto	5
1.8 Objetivos	6
1.8.1 Objetivo general	6
1.8.2 Objetivo específico	6
2 Fundamentos teóricos	7
2.1 Domótica	7
2.2 Tipo de sensores para domótica	8
2.2.1 Sensor de temperatura y humedad relativa DHT22	8
2.2.2 Sensor de gas GLP MQ2	10
2.2.3 Sensor de humedad de suelo capacitivo	11
2.2.4 Sensor de movimiento PIR HC-SR501	11
2.3 Elecrow ESP32 HMI LCD	13
2.3.1 Características de pantalla HMI	15
2.3.2 Especificación técnica de pantalla HMI	15
2.3.3 Admisión de múltiples entornos de desarrollo	18
2.3.4 Admisión de biblioteca de gráficos de código abierto-LVGL	19
2.3.5 Comparación de pantallas CrowPanel ESP32	21
2.4 ESP32	22
2.4.1 Características del ESP32:	22
2.4.2 ESP32-S3-WROOM-1-N4	23

2.5	Módulo Relé de 4 canales	25
3	Marco metodológico	26
3.1	Tipo de investigación	26
3.2	Diseño de investigación	26
3.3	Enfoque de la investigación	26
3.4	Metodología de investigación	27
3.5	Desarrollo de proyecto de investigación	27
3.6	Diseño de maqueta de vivienda domótica	31
3.7	Montaje de elementos en maqueta domótica	34
4	Resultados	42
5	Conclusiones	48
6	Recomendaciones	49
7	Bibliografía	50
8	Anexos	52
8.1	Código ESP32 pantalla HMI	52
8.2	Código de MicroPython	62
8.3	Esquemático del ESP32-S3-WROOM-1	69
8.4	Boom de materiales	70

ÍNDICE DE FIGURAS

Figura 1 Diagrama de bloque de un panel de control domótico Usando HMI y ESP32.....	4
Figura 2 Domótica casa inteligente	8
Figura 3 Sensor DHT22.....	9
Figura 4 Sensor MQ-2.....	10
Figura 5 Sensor de humedad de suelo capacitivo	11
Figura 6 Sensor de movimiento PIR HC-SR501	12
Figura 7 Elecrow HMI 7 pulgadas.....	14
Figura 8 HMI con ESP32 integrado	14
Figura 9 HMI vista interna	16
Figura 10 Vista trasera de la pantalla HMI.....	16
Figura 11 Esquemático de pantalla HMI.....	17
Figura 12 Escenarios de aplicaciones de pantalla HMI.....	18
Figura 13 Pantalla HMI con soporte multiplataforma	19
Figura 14 Soporte de código abierto LVGL.....	20
Figura 15 ESP32.....	23
Figura 16 Módulo relé de 4 canales.....	25
Figura 17 Pruebas de laboratorio de pantalla HMI.....	28
Figura 18 Placa PCB de prototipo	29
Figura 19 Diseño 3D de placa PCB.....	29
Figura 20 Diseño de pistas de placa PCB del prototipo	30
Figura 21 Maqueta de vivienda domótica	31
Figura 22 Electrónica y pantalla HMI	32
Figura 23 Vista general de la maqueta	33
Figura 24 Montaje de fuentes de poder	34
Figura 25 Montaje puerta garaje.....	35
Figura 26 Montaje motor en puerta garaje.....	35
Figura 27 Montaje de piscina en maqueta.....	36
Figura 28 Montaje de tanque de piscina.....	36
Figura 29 Montaje de resistencias de potencia.....	37
Figura 30 Montaje de relés.....	37

Figura 31 Montaje de cableado de módulo de relés	38
Figura 32 Montaje pantalla HMI.....	38
Figura 33 Sensor de humedad de suelo.....	39
Figura 34 Sensor de movimiento.....	39
Figura 35 Montaje de sensores y buzzer.....	40
Figura 36 Módulos relés de 4 canales.....	41
Figura 37 Divisiones con las luces leds encendidas.....	42
Figura 38 Relé activos.....	43
Figura 39 Módulo regulador de voltaje.....	44
Figura 40 Cerradura eléctrica activada desde la pantalla HMI.....	44
Figura 41 Sección piscina.....	45
Figura 42 Módulo servo SG90.....	45
Figura 43 Encendido de pantalla HMI.....	46
Figura 44 Disposición de botones en pantalla HMI.....	47
Figura 45 Botón de WIFI en pantalla HMI.....	47
Figura 52 Esquemático del ESP32-S3-WROOM-1.....	69
Figura 53 Esquemático de periféricos del ESP32-S3-WROOM-1.....	69
Figura 54 Dimensiones físicas del ESP32-S3-WROOM1.....	70

ÍNDICE DE TABLAS

Tabla 1 Descripción de conectores y botones de pantalla HMI.....	17
Tabla 2 Comparativas de diferentes tamaños de pantalla	21
Tabla 3 Características del ESP32-S3-WROOM-1-N4	23
Tabla 4 Boom de materiales del prototipo	70

INTRODUCCIÓN

En términos generales la domótica, implica el control y monitorización de dispositivos domésticos, buscando sustituir la interacción humana en varias tareas diarias (como encender o apagar los electrodomésticos) y ahorrar tiempo a las personas. La domótica proporciona ahorro energético, seguridad, confort, entretenimiento y tranquilidad, permite la monitorización remota, amplía la eficiencia y es flexible para nuevas aplicaciones y dispositivos (Murphy, 2018) (Miori & Russo, 2014)

Detrás de las tecnologías que hacen viable la domótica está el Internet de las Cosas (IoT). IoT se refiere a la conexión en red de objetos que, a través de Internet, pueden enviar y recibir datos automáticamente. Esta tecnología se ha posicionado como una industria emergente y estratégica a nivel mundial. Introduce un cambio radical en la calidad de vida de las personas ya que sus aplicaciones son ilimitadas y se pueden adaptar a diversos campos, como la educación, la salud, la seguridad, la agricultura y el medio ambiente. La implementación de la domótica implica sensores y actuadores inteligentes que automatizan los sistemas del hogar, como seguridad, cerraduras, alarmas, etc. (Chen et al., 2014) (Rodríguez-Serra et al., 2022)

Este proyecto de investigación propone un prototipo domótico para el hogar que se encarga de la automatización a través de sensores y elementos de bajo costo como el ESP32 y la implementación de una pantalla táctil HMI (Interfaz hombre máquina), lo que permite una interacción desde consola hacia los sensores y elementos instalados dentro del hogar.

Se proyecta que este prototipo sea utilizado como objeto de práctica y puesta en marcha de experimentación para los futuros ingenieros de la carrera de ingeniería en electrónica de la Universidad Politécnica Salesiana sede Guayaquil, con la finalidad de realizar prácticas enfocadas en la domótica y la automatización.

1 El problema

1.1 Descripción del problema

Los costos elevados de sistemas domóticos para la automatización de los hogares en donde lo único que se busca es la eficiencia a bajo costo, la comodidad y seguridad que brinda agregar esta característica al día a día, asegurar una interacción intuitiva y efectiva con los usuarios, teniendo en cuenta la diversidad de dispositivos y perfiles de usuario en un entorno doméstico aplicando las medidas de seguridad para proteger la red domótica contra posibles amenazas cibernéticas y garantizar la privacidad de los datos de los usuarios contribuyendo a la eficiencia energética en el hogar al controlar y optimizar el uso de dispositivos electrónicos desarrollando un sistema que puede adaptarse y escalar para satisfacer las necesidades cambiantes de un hogar o edificio a medida que se agregan más dispositivos y funcionalidades explorando cómo integrar tecnologías emergentes, como la inteligencia artificial o el aprendizaje automático, para mejorar la automatización y la toma de decisiones en el sistema domótico.

En este contexto se ha encontrado un problema de estudio que proporciona una base sólida para la investigación y desarrollo de un proyecto centrado en la implementación y mejora de un sistema de control domótico utilizando HMI y ESP32, abordando aspectos esenciales relacionados con la eficiencia, la seguridad y la usabilidad de manera local.

1.2 Antecedentes

En virtud de la creciente demanda de soluciones inteligentes y eficientes para el hogar, que aborden la búsqueda de comodidad y eficiencia en la vida cotidiana. Esta demanda está respaldada por la necesidad de mejorar la eficiencia energética y la sostenibilidad, así como de aumentar la seguridad y la conveniencia en los hogares. El uso de tecnologías emergentes, como la interfaz HMI y el microcontrolador ESP32, proporciona la oportunidad de integrar soluciones más avanzadas y adaptativas en sistemas domóticos, lo que tiene un impacto significativo en la calidad de vida de las personas y contribuye al avance de la investigación y el desarrollo tecnológico en el campo de la domótica. En última instancia, este proyecto busca satisfacer las necesidades cambiantes de los usuarios, brindando soluciones que sean más eficientes, seguras y personalizadas, al tiempo que promueven la eficiencia energética y la sostenibilidad en los entornos residenciales.

1.3 Importancia y alcance

El trabajo de titulación propuesto es muy importante debido a que se busca implementar la domotización residencial mediante tecnologías IoT y de automatización, así como también utilizando plataformas locales, y elementos de bajo costo como ESP32, e interfaces HMI (Interfaz hombre máquina), sensores y actuadores.

1.4 Delimitación

1.4.1 Delimitación temporal

El tiempo estimado para el diseño, implementación y pruebas de funcionamiento del prototipo es de 4 meses, desde septiembre 2024 a diciembre del 2024.

1.4.2 Delimitación espacial

Las validaciones del funcionamiento y las demostraciones del prototipo se realizaron en el domicilio de los autores del proyecto. Este prototipo se presentó como prototipo de pruebas para la carrera de ingeniería electrónica de la Universidad Politécnica Salesiana sede Guayaquil campus centenario.

1.4.3 Delimitación académica

El diseño e implementación del prototipo domótico es muy importante para desarrollar las capacidades y destrezas de los futuros ingenieros electrónicos de la Universidad Politécnica Salesiana sede Guayaquil ya que sirve como guía para prácticas en laboratorio y para la investigación.

1.5 Beneficiarios de la propuesta

Este proyecto beneficia a múltiples grupos de personas y entidades, incluyendo residentes en viviendas que disfrutan de mayor comodidad y eficiencia energética, personas con discapacidades y personas mayores que encuentran en la accesibilidad y la seguridad mejoradas un apoyo vital, padres y cuidadores que pueden supervisar de manera efectiva a sus seres queridos, empresas de seguridad y proveedores de servicios IoT que ven oportunidades comerciales, profesionales en la construcción y diseño de interiores que pueden agregar valor a sus proyectos, gobiernos y autoridades locales que pueden impulsar la sostenibilidad y la eficiencia energética, y el medio ambiente que se beneficia de la conservación de recursos y la reducción del consumo de energía.

1.6 Propuesta de solución

A continuación, en la figura 1 se observa un diagrama de bloques simplificado que representa la conexión de un HMI (Interfaz Hombre-Máquina) a un ESP32, sensores y la nube:

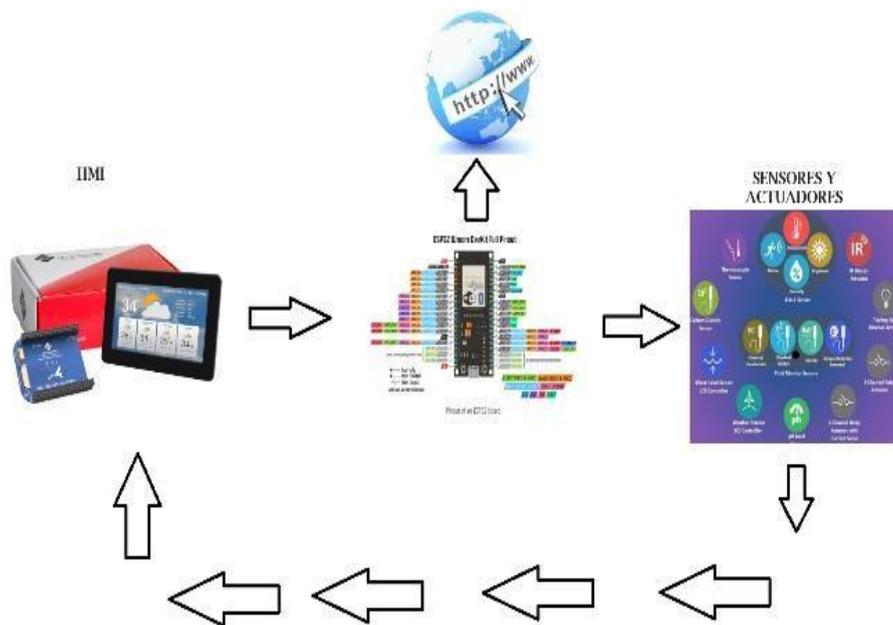


Figura 1 Diagrama de bloque de un panel de control domótico Usando HMI y ESP32

Este diagrama de bloques muestra la interacción general entre los componentes de un sistema domótico que utiliza una interfaz HMI para controlar dispositivos y monitorear datos a través de un ESP32.

"Interfaz Hombre-Máquina" (HMI) representa el panel de control o la pantalla táctil desde la cual los usuarios interactúan con el sistema. Puede ser una aplicación móvil, una pantalla táctil, un sitio web, etc.

"ESP32 (Controlador)" es el microcontrolador que actúa como el cerebro del sistema. Se encarga de la comunicación entre el HMI, los sensores y la nube. Realiza el procesamiento de datos y controla los dispositivos según las instrucciones del HMI y las lecturas de los sensores.

"Sensores" son los dispositivos que capturan datos del entorno, como temperatura, humedad, movimiento, etc. Estos datos son enviados al ESP32 para su procesamiento.

"Conexión a la Nube" representa la conexión del ESP32 a servicios en la nube (por ejemplo, servicios de almacenamiento de datos o plataformas de IoT). Los datos procesados por el ESP32 pueden ser enviados a la nube para su almacenamiento, análisis y acceso remoto.

1.7 Innovación e impacto del proyecto

El avance de este proyecto de investigación es innovador, ya que busca modernizar hogares e instituciones mediante la domotización para optimizar el ahorro energético, mejorar la seguridad y facilitar el monitoreo a distancia. Este tipo de proyectos tiene un gran impacto tecnológico al integrar tecnologías innovadoras, servidores en la nube e internet.

1.8 Objetivos

1.8.1 Objetivo general

Desarrollar un sistema de control domótico que utilice una Interfaz Hombre Máquina (HMI) para el hogar.

1.8.2 Objetivo específico

- Diseñar un prototipo para la implementación de un sistema domótico dentro de una casa.
- Implementar una interfaz HMI, para el control y monitoreo de dispositivos.
- Implementar un histórico de accionamiento de las alarmas.

2 Fundamentos teóricos

2.1 Domótica

La domótica, al automatizar y controlar diversos aspectos de un hogar a través de la tecnología, puede resolver una serie de problemas y mejorar la calidad de vida de las personas. Algunos de los problemas que la domótica puede solucionar o mitigar incluyen:

Ahorro de energía: Los sistemas de domótica pueden optimizar el consumo de energía al regular la iluminación, la calefacción, la refrigeración y otros dispositivos eléctricos. Esto puede resultar en una reducción significativa de los costos de energía y en un menor impacto ambiental

Mayor comodidad: La automatización permite a los residentes controlar fácilmente diferentes aspectos de su hogar, como la temperatura, la iluminación, las persianas y la música, mediante simples comandos de voz o a través de una aplicación móvil. Esto crea un entorno más cómodo y personalizado.

Seguridad: Los sistemas de seguridad domótica incluyen cámaras de vigilancia, sensores de movimiento y detectores de humo que pueden proporcionar una mayor seguridad y protección en el hogar. Los residentes pueden recibir alertas en tiempo real y tomar medidas en caso de intrusiones o emergencias.

Gestión del hogar a distancia: La domótica permite a las personas controlar y supervisar su hogar incluso cuando no están presentes físicamente. Pueden ajustar la temperatura, encender y apagar luces o revisar cámaras de seguridad desde cualquier lugar con acceso a internet. En la figura 2 se observa una gráfica de casa inteligente.



Figura 2 Domótica casa inteligente
(Casadomo.com, 2024)

2.2 Tipo de sensores para domótica

Existen muchos tipos de sensores domóticos que detectan y emiten diferentes señales en el hogar. Funcionan por contacto directo, sin contacto y pueden ser iónicos, ópticos, mecánicos o eléctricos. La mayoría de estas señales se transmiten al teléfono para alertar si sucede algo inusual. Para este proyecto de investigación se utilizaron los siguientes sensores que se detallan a continuación.

2.2.1 Sensor de temperatura y humedad relativa DHT22

El DHT22 (AM2302) es un sensor digital que mide temperatura y humedad relativa, destacando por su eficiencia y bajo costo. Integra un sensor capacitivo para la humedad y un termistor para la temperatura, transmitiendo datos mediante una señal digital. Es común en control automático de temperatura, climatización y monitoreo ambiental. Su integración con Arduino, Raspberry Pi y NodeMCU es sencilla, con bibliotecas compatibles para su uso. Se conecta mediante tres pines: VCC a 3-5V, GND a tierra y un pin digital para datos. Cada sensor necesita su propio pin de datos y la actualización se limita a cada 2 segundos. Los sensores se calibran en fábrica, asegurando estabilidad a largo plazo. La comunicación se realiza con un solo cable. Se sugiere protegerlo de la luz solar. Comparado con el DHT11, el DHT22 ofrece mejor resolución, precisión y un diseño más robusto.

2.2.1.1 Especificaciones técnicas del sensor DHT22:

- Voltaje de funcionamiento: 3V - 6V DC
- Intervalo de medición de temperatura: -40°C a 80°C
- Precisión en la medición de temperatura: $\leq \pm 0.5^\circ\text{C}$
- Resolución de temperatura: 0.1°C
- Rango de medición de humedad: 0% a 100% RH
- Precisión en la medición de humedad: 2% RH
- Resolución de humedad: 0.1% RH
- Tiempo de respuesta: 2 segundos
- Interfaz digital: Single-bus (bidireccional)
- Modelo: AM2302
- Dimensiones: 20 x 15 x 8 mm
- Peso: 3 gramos
- Carcasa de plástico blanco

En la figura 3 se observa el sensor DHT22 y su distribución de pines.

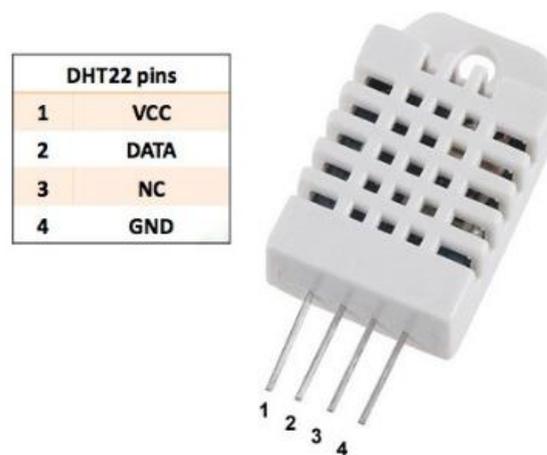


Figura 3 Sensor DHT22
(Naylampmechatronics.com, 2024c)

2.2.2 Sensor de gas GLP MQ2

El Módulo Sensor MQ-2 detecta concentraciones de gas GLP y GNV en el ambiente. Es sensible a varios gases, incluidos GLP, i-butano, propano, metano, alcohol, hidrógeno (H₂) y humo.

2.2.2.1 Especificaciones técnicas del sensor MQ-2:

- Voltaje de operación: 5V DC
- Rango de detección: 300 a 10,000 ppm
- Gas de referencia: 1000 ppm de isobutano
- Resistencia de sensado: 1K Ω a 50 ppm de tolueno hasta 20K Ω
- Tiempo de respuesta: \leq 10 segundos
- Tiempo de recuperación: \leq 30 segundos
- Temperatura de operación: -20°C a +55°C
- Humedad: \leq 95% RH
- Contenido de oxígeno ambiental: 21%
- Consumo: menos de 150 mA a 5V

En la figura 4 se observa el sensor de gas GLP MQ-2.



Figura 4 Sensor MQ-2
(Naylampmechatronics.com, 2024a)

2.2.3 Sensor de humedad de suelo capacitivo

El funcionamiento del sensor es bastante sencillo. Mide la capacitancia entre dos electrodos que se insertan en el suelo. A medida que la humedad cambia, también lo hace la capacitancia: un suelo muy húmedo tiene una capacitancia baja, mientras que un suelo seco presenta una capacitancia alta. La señal que se genera se procesa a través de una placa de acondicionamiento, que produce una salida analógica. Esta salida (AO) oscila entre 0V cuando el suelo está húmedo y 5V cuando está seco. Para conectarlo a un **Arduino**, solo necesitas vincular la salida analógica del sensor a una entrada analógica del microcontrolador. Recuerda que debes introducir en la tierra únicamente la parte de los electrodos, manteniendo los circuitos protegidos de la humedad.

En la figura 5 se observa el sensor capacitivo de humedad de suelo.



Figura 5 Sensor de humedad de suelo capacitivo
(Naylampmechatronics.com, 2024b)

2.2.4 Sensor de movimiento PIR HC-SR501

Los sensores PIR se utilizan para detectar movimiento, especialmente el de personas dentro de su rango de acción. Estos dispositivos son económicos, compactos, de bajo consumo y fáciles de usar, además de tener una larga vida útil sin desgaste. Son comunes en electrodomésticos y dispositivos de oficina y hogar, y se conocen como "Sensores Infrarrojos" o "Sensores de movimiento".

El módulo HC-SR501 incluye un sensor PIR RE200B, una lente Fresnel PIR 8002-2B, y un chip controlador PIR BISS0001, junto con la electrónica de acondicionamiento necesaria. El sensor piroeléctrico RE200B detecta cambios en la radiación infrarroja. Todo objeto o cuerpo humano emite radiación, que aumenta con la temperatura. El sensor de movimiento está dividido en dos partes para detectar diferencias de radiación y no promedios; estas mitades están conectadas de manera que se anulan entre sí, y cuando una mitad recibe más o menos radiación, la salida cambia entre niveles alto y bajo. La lente Fresnel extiende el rango de detección a unos 10 metros y ofrece un ángulo de detección de aproximadamente 100°.

Los sensores PIR son ideales para detectar cuando alguien entra o sale de un espacio, permitiendo la programación de acciones automáticas como encender o apagar luces, activar sonidos, alarmas, mensajes de texto o llamadas. Aunque pueden detectar la presencia de personas, no pueden indicar cuántas hay ni la dirección de su movimiento, y en algunos casos pueden activarse por mascotas. Ajustar la sensibilidad es clave para optimizar su funcionamiento.

En la figura 6 se observa el sensor PIR HC-SR501.



Figura 6 Sensor de movimiento PIR HC-SR501

2.3 Elecrow ESP32 HMI LCD

La pantalla Elecrow ESP32 de 7,0 pulgadas es una potente pantalla táctil HMI con una resolución LCD de 800*480. Utiliza el módulo ESP32-S3-WROOM-1-N4R2 como procesador de control principal, con un microprocesador LX6 de 32 bits de doble núcleo, funciones inalámbricas WIFI y Bluetooth integradas, una frecuencia principal de hasta 240 MHz, proporcionando un rendimiento potente y versátil. Aplicaciones, adecuadas para dispositivos de aplicaciones IoT y otras escenas.

El módulo incluye una pantalla LCD de 7,0 pulgadas y una placa de controlador. La pantalla utiliza una tecnología táctil resistiva y viene con un lápiz táctil resistivo, lo que hace que el uso de la pantalla sea más flexible. Además, la placa ha reservado una ranura para tarjeta TF, múltiples interfaces periféricas, interfaz USB, interfaz de altavoz, interfaz de batería, etc., brindando más posibilidades de expansión. Admite entornos de desarrollo como Arduino IDE, Espressif IDF, Lua RTOS, Micro Python y es compatible con la biblioteca de gráficos LVGL. Esto permite a los desarrolladores no sólo personalizar sus propias interfaces UI sino también crear proyectos interesantes de forma rápida y sencilla, acortando enormemente el ciclo de desarrollo.

La pantalla ESP32 de 7,0 pulgadas es adecuada para una amplia gama de escenas, como HMI automotriz, equipos médicos, control industrial, energía, electrónica civil, automatización, GPS, nuevas energías y dispositivos de aplicaciones de IoT. Sus diversas interfaces y funciones de expansión le permiten satisfacer las necesidades de diferentes campos, brindando a los usuarios una solución más completa. En las figuras 7 y 8 se observa la pantalla HMI. (elecrow.com, 2024).



Figura 7 Elecrow HMI 7 pulgadas
(elecrow.com, 2024)



Figura 8 HMI con ESP32 integrado
(elecrow.com, 2024)

2.3.1 Características de pantalla HMI

A continuación, se detallan las características principales de la interfaz HMI

- Módulo ESP32-S3-WROOM-1-N4R2 integrado, comunicación inalámbrica integrada Wi-Fi de 2,4 GHz (802,11 b/g/n) y Bluetooth 5,0;
- Admite entorno de desarrollo Arduino IDE, Espressif IDF, Lua RTOS, Micro Python y compatible con la biblioteca de gráficos LVGL;
- Interfaz de demostración LVGL integrada y ejemplo de Arduino, plug and play;
- LCD 800*480 TFT-LCD de 7,0 pulgadas con controlador IC EK9716BD3/EK73002ACGB; Las interfaces periféricas y las funciones de expansión le permiten satisfacer las necesidades de diferentes campos (elecrow.com, 2024)

2.3.2 Especificación técnica de pantalla HMI

A continuación, se detallan las especificaciones técnicas de la interfaz HMI.

- Modelo: pantalla ESP32 de 7,0 pulgadas
- Procesador principal: ESP32-S3-WROOM-1-N4R2
- Resolución: 800*480
- Profundidad de color: 16M
- Tipo táctil: pantalla táctil resistiva
- Panel táctil: Panel TN
- Pantalla: Pantalla TFT-LCD
- Controlador de pantalla: EK9716BD3/EK73002ACGB
- Fuente de alimentación externa: DC5V-2A
- Interfaz: 1*Ranura para tarjeta TF, 1*GPIO, 1*Speak, 1*UART1, 1*UART0, 2*I2C (elecrow.com, 2024).

En la figura 9 se observa el esquemático de la pantalla HMI en una vista interna.

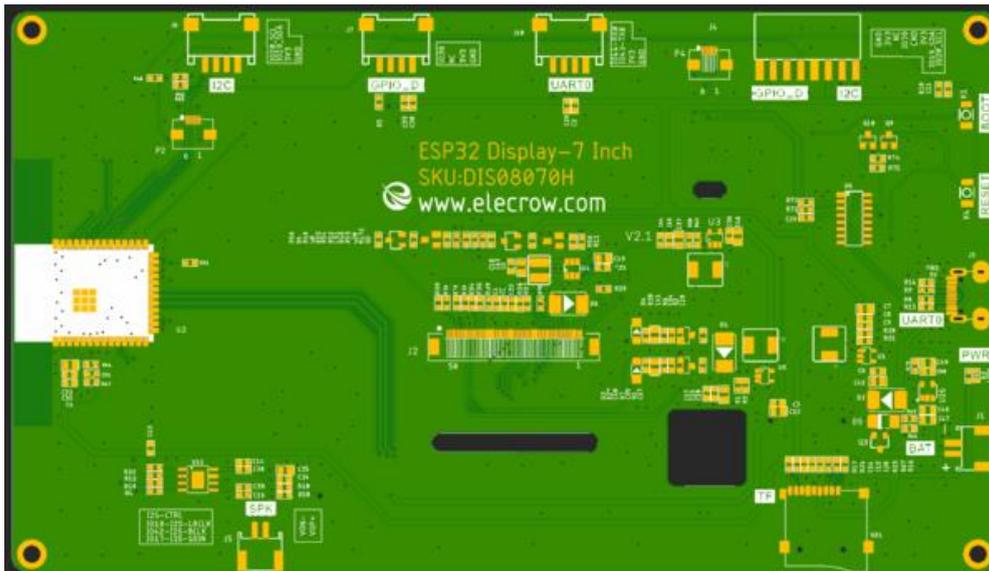


Figura 9 HMI vista interna
(elecrow.com, 2024)

En la figura 10 se observa la parte trasera del case del HMI donde se indican las distancias de los orificios que sostienen la pantalla, así como también los puertos de conexiones.

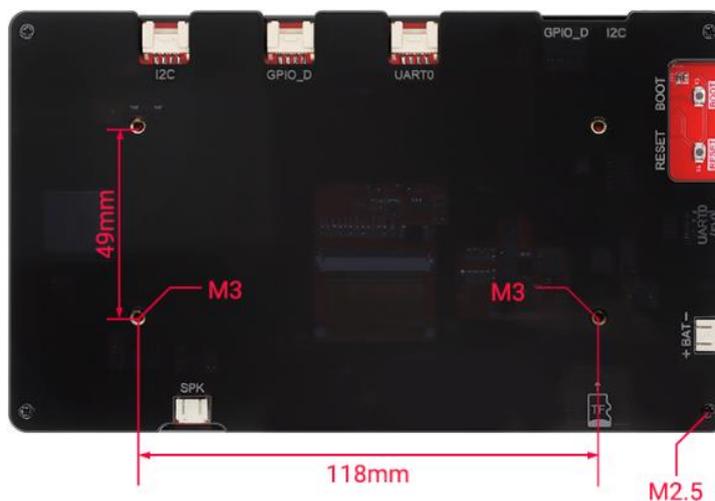


Figura 10 Vista trasera de la pantalla HMI
(elecrow.com, 2024)

En la figura 11 se observa la descripción de cada conector y botón de la pantalla HMI.

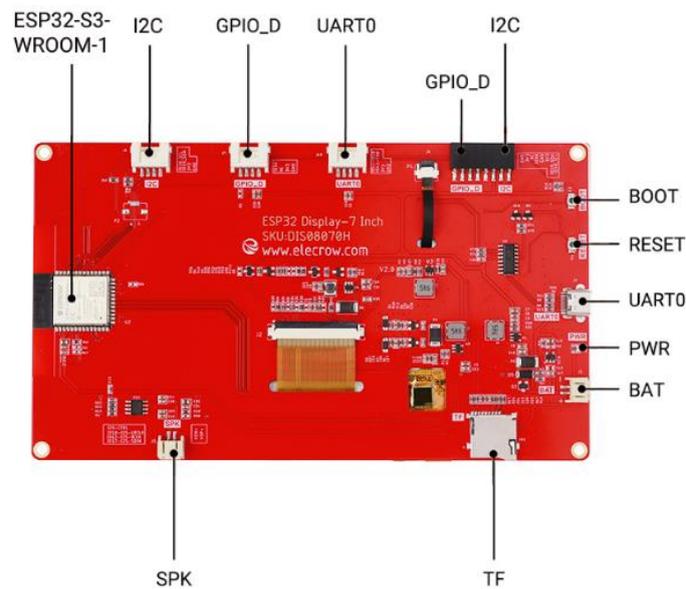


Figura 11 Esquemático de pantalla HMI
(elecrow.com, 2024)

En la tabla 1 se detalla la descripción de los conectores y botones de la pantalla HMI.

Tabla 1 Descripción de conectores y botones de pantalla HMI

SPK	Señal de audio de salida, conectada con altavoces. La placa base viene con un circuito de chip amplificador de potencia.
PWR	LED de encendido.
RST	Botón de reinicio. Presiónelo para reiniciar el sistema.
BOOT	
GPIO_D	Interfaz de E/S digitales y artificiales.
TF	Proporciona almacenamiento fuera de línea y espacio de almacenamiento adicional.
UART0	Comunicación entre los módulos lógicos, incluido el módulo de comunicación en serie y el módulo de impresión (no se puede usar el UART0 (Tipo-c) como entrada al mismo tiempo).
BATERÍA	Conector para la batería de litio. Se puede cargar la batería.
UART0(USB-C)	Voltaje de suministro (transforma USB a UART0), proporciona comunicación en serie e impresión de información en serie.

(elecrow.com, 2024)

En la figura 12 se observan los diferentes escenarios de aplicaciones para la pantalla HMI.



Figura 12 Escenarios de aplicaciones de pantalla HMI
(elecrow.com, 2024)

2.3.3 Admisión de múltiples entornos de desarrollo.

La interfaz HMI admite una variedad de entornos de desarrollo como Arduino IDE, Espressif IDF, PlatformIO y MicroPython, para satisfacer las necesidades de diferentes desarrolladores y hacer posible cada proyecto. En la figura 13 se observan las diferentes plataformas de soporte. Para este proyecto de investigación se usó el entorno de desarrollo MicroPython.

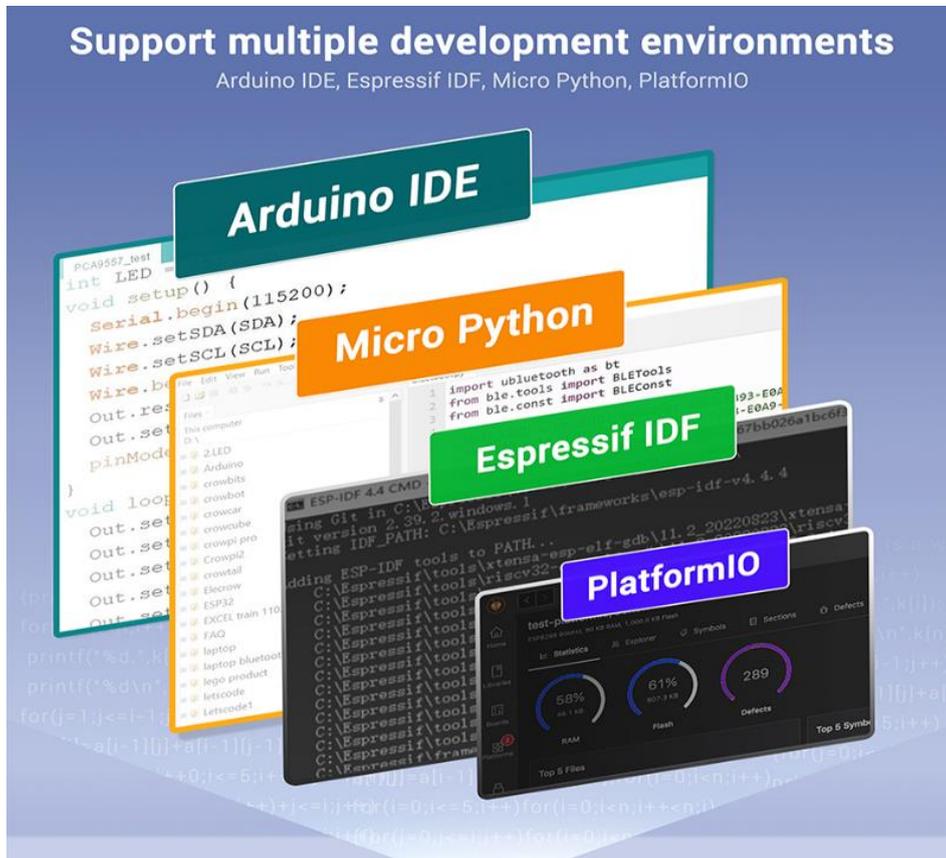


Figura 13 Pantalla HMI con soporte multiplataforma (elecrow.com, 2024)

2.3.4 Admisión de biblioteca de gráficos de código abierto-LVGL

La interfaz HMI es compatible con la biblioteca de gráficos LVGL. Puede utilizar el editor de interfaz de usuario de arrastrar y soltar para diseñar e implementar su propia interfaz de usuario. (La siguiente imagen es el software de diseño de interfaz de usuario oficial de LVGL, Squareline Studio). La figura 14 se observa el soporte de la pantalla para código abierto LVGL.

Support open-source Graphics Library-LVGL

LVGL is the most popular free and open-source embedded graphics library to create beautiful UIs for any MCU, MPU and display type. User can use SquareLine Studio, to drag and drop UI editor to simplify the development, design and implement your next UI. Simple drag and drop components will make our HMI screens easier to use, greatly reduce your HMI project development effort, and develop projects quickly in a cost-effective manner.



Figura 14 Soporte de código abierto LVGL
(elecrow.com, 2024)

2.3.5 Comparación de pantallas CrowPanel ESP32

En la tabla 2 se detalla una comparativa de las pantallas Elecrow de diferentes tamaños.

Tabla 2 Comparativas de diferentes tamaños de pantalla

Tamaño	2,4"	2,8"	3,5"	4,3"	5,0"	7,0"
Resolución	320*240	320*240	480*320	480*272	800*480	800*480
Tipo táctil	Toque resistivo	Toque resistivo	Toque resistivo	Toque resistivo	Toque capacitivo	Toque capacitivo
Procesador principal	ESP32-WROOM-32-N4	ESP32-WROOM-32-N4	ESP32-WROVER-B	ESP32-S3-WROOM-1-N4R2	ESP32-S3-WROOM-1-N4R8	ESP32-S3-WROOM-1-N4R8
Frecuencia	240MHz	240MHz	240MHz	240MHz	240MHz	240MHz
Destello	4MB	4MB	4MB	4MB	4MB	4MB
SRAM	520KB	520KB	520KB	512KB	512KB	512KB
ROM	448KB	448KB	448KB	384KB	384KB	384KB
PSRAM	/	/	8MB	2MB	8MB	8MB
Controlador de pantalla	ILI9341V	ILI9341V	ILI9488	NV3047	ILI6122 y ILI5960	EK9716BD3 y EK73002ACGB
Tipo de pantalla	TFT	TFT	TFT	TFT	TFT	TFT
Interfaz	1*UART0,1*UART1,1*I2C,	1*UART0,1*UART1,1*I2C,	2*UART0,1*I2C,	1*UART0,2*UART1,	2*UART0,2*GPIO,2*I2C,	2*UART0,2*GPIO,2*I2C,1*Batería
	1 * GPIO, 1 * batería	1 * GPIO, 1 * batería	1 * GPIO, 1 * batería	2 * GPIO, 1 * batería	1 * batería	
Conector de altavoz	Sí	Sí	Sí	Sí	Sí	Sí
Ranura para tarjetas TF	Sí	Sí	Sí	Sí	Sí	Sí
Área activa	36,72*48,96 mm (ancho x alto)	43,2*57,6 mm (ancho x alto)	48,96*73,44 mm (ancho x alto)	95,04*53,86 mm (ancho x alto)	108*64,8 mm (ancho x alto)	153,84*85,63 mm (ancho x alto)

(elecrow.com, 2024)

2.4 ESP32

La plataforma ESP32 ofrece Wi-Fi, Bluetooth y BLE a un costo accesible, con una CPU de doble núcleo que alcanza hasta 240 MHz y control independiente. Incluye periféricos como un sensor táctil, sensor de efecto Hall, amplificador de bajo ruido, y soporte para tarjetas SD, Ethernet, y varios protocolos de comunicación.

Su extensa comunidad en línea brinda apoyo y desarrolla nuevas herramientas constantemente. Para el desarrollo, hay una amplia gama de software y lenguajes, como Espidf, Arduino, Simba, sistemas operativos en tiempo real, MicroPython, LUA, Javascript y Basic.

2.4.1 Características del ESP32:

A continuación, se presentan las características técnicas del ESP32.

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- CPU principal: Tensilica Xtensa 32-bit LX6
- Frecuencia de Reloj: hasta 240Mhz
- Desempeño: Hasta 600 DMIPS
- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth:v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- Memoria:
- 448 Kbyte ROM
- 520 Kbyte SRAM
- 16 Kbyte SRAM in RTC
- QSPI Flash/SRAM, 4 Mbyte
- Pines Digitales GPIO: 24 (Algunos pines solo como entrada)
- Conversor Analógico Digital: Dos ADC de 12bits tipo SAR, soporta mediciones en hasta 18 canales, algunos pines soportan un amplificador con ganancia programable
- Chip USB-Serial: CP2102

- Antena en PCB (espressif.com, 2024).

En la figura 15 se observa la imagen frontal del ESP32.

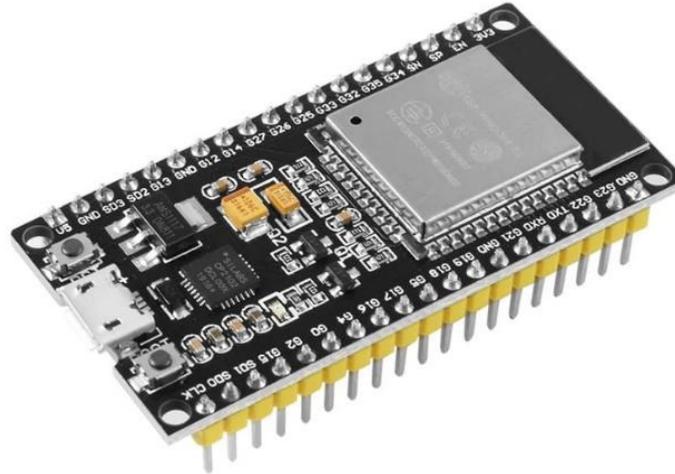


Figura 15 ESP32
(espressif.com, 2024)

2.4.2 ESP32-S3-WROOM-1-N4

En la tabla 3 se observan las características principales del ESP32-S3-WROOM-1-N4.

Tabla 3 Características del ESP32-S3-WROOM-1-N4

Atributo del producto	Valor de atributo
Fabricante:	Espressif
Categoría de producto:	Módulos multiprotocolo
Restricciones de envío:	Este producto puede necesitar documentación adicional para ser exportado desde los Estados Unidos.
RoHS:	<u>Detalles</u>

Frecuencia:	2.4 GHz
Potencia de salida:	20.5 dBm
Tipo de interfaz:	I2C, I2S, SPI, PWM, UART, USB
Voltaje de alimentación - Mín.:	3 V
Voltaje de alimentación - Máx.:	3.6 V
Temperatura de trabajo mínima:	- 40 C
Temperatura de trabajo máxima:	+ 85 C
Conector de tipo antena:	PCB
Dimensiones:	25.5 mm x 18 mm x 3.1 mm
Protocolo: Bluetooth, BLE - 802.15.1:	Bluetooth 5.0
Protocolo: WiFi - 802.11:	WiFi
Empaquetado:	Bulk
Marca:	Espressif Systems
Velocidad de transmisión de datos:	150 Mb/s
Sensibles a la humedad:	Yes
Voltaje de alimentación operativo:	3 V to 3.6 V
Tipo de producto:	Multiprotocol Modules
Protocolo admitido:	802.11 b/g/n, Bluetooth 5.0
Cantidad de empaque de fábrica:	650
Subcategoría:	Wireless & RF Modules

(mouser.ec, 2024)

2.5 Módulo Relé de 4 canales

El módulo relé de 4 canales es un dispositivo electrónico que permite controlar cargas eléctricas de alta potencia mediante señales de bajo voltaje, generalmente provenientes de microcontroladores o sistemas de automatización. Cada canal del módulo actúa como un interruptor que puede ser activado o desactivado de forma independiente, lo que lo hace ideal para aplicaciones en domótica, automatización industrial y proyectos de robótica. Su diseño incluye optoaisladores para proteger los circuitos de control y facilitar la conexión con diversas fuentes de alimentación. Este módulo es fundamental para el manejo seguro y eficiente de dispositivos eléctricos, permitiendo la integración de sistemas de control en proyectos tecnológicos.

En la figura 16 se observa un módulo relé mecánico de 4 canales.



Figura 16 Módulo relé de 4 canales
(naylampmechatronics.com, 2024)

3 Marco metodológico

3.1 Tipo de investigación

Este proyecto explora y desarrolla una solución accesible y práctica para controlar de manera inteligente un hogar, utilizando una interfaz gráfica que permita a los usuarios manejar distintos dispositivos y sensores a través de la tecnología HMI y el microcontrolador ESP32. A través de esta investigación exploratoria, se pretende entender mejor los retos y beneficios de implementar un sistema domótico que pueda ser fácil de usar y, al mismo tiempo, eficiente en cuanto a ahorro energético y seguridad.

La investigación exploratoria es ideal aquí porque, en el área de control domótico, muchas soluciones actuales aún no han abordado completamente cómo lograr una interacción simple y efectiva con los dispositivos en el hogar. Este proyecto intenta llenar ese vacío, probando y documentando cómo esta tecnología puede integrarse de manera funcional y accesible, tanto para hogares como para personas que deseen gestionar su espacio con mayor comodidad y control. Así, este enfoque permite no solo imaginar sino también construir una base para futuras mejoras y aplicaciones en el día a día de quienes buscan una vida más conectada y segura en sus hogares.

3.2 Diseño de investigación

El diseño de investigación es experimental porque se establece una relación entre causa y efecto en la pregunta. Por ejemplo, se observan y analizan los datos obtenidos por los sensores y la interfaz de la pantalla táctil HMI.

3.3 Enfoque de la investigación

En el proyecto se empleó una combinación de métodos cualitativos y cuantitativos para darle profundidad y precisión al desarrollo del sistema. Primero, se investigó a fondo sobre domótica, revisando libros, artículos científicos, tesis y otras fuentes confiables. Esta exploración ayudó a entender cómo funciona la tecnología de control para hogares inteligentes y a definir los componentes clave del sistema.

Luego, en la parte práctica, se realizaron pruebas con varios sensores, como los de temperatura, movimiento y humedad, recopilando datos para asegurar de que funcionen de manera óptima. Este análisis cuantitativo permitió ajustar el sistema para que cada sensor respondiera correctamente y mejorara la precisión del control domótico. Al combinar ambos enfoques, se logra construir una solución bien fundamentada y efectiva para el control inteligente del hogar.

3.4 Metodología de investigación

Ejecución: se realizaron las siguientes actividades:

- Se analizó la cantidad y el tipo de dispositivos a controlar.
- Se recopiló información relevante de un domicilio en maqueta donde se llevaron a cabo las pruebas y la implementación.
- Se evaluó el equipamiento tecnológico necesario.
- Se estructuró y direccionó la red de sensores, así como la red eléctrica de los LEDs.
- Se implementó y probaron las funciones de la maqueta domótica para garantizar su correcto funcionamiento.

3.5 Desarrollo de proyecto de investigación

En esta sección se describen los elementos que se utilizaron en el proyecto, así como también se detalla las configuraciones realizadas en todos sus elementos y como se preparó la maqueta para obtener los resultados del siguiente capítulo.

En la figura 17 se observan las pruebas que se realizaron en laboratorio para la configuración de la pantalla HMI.

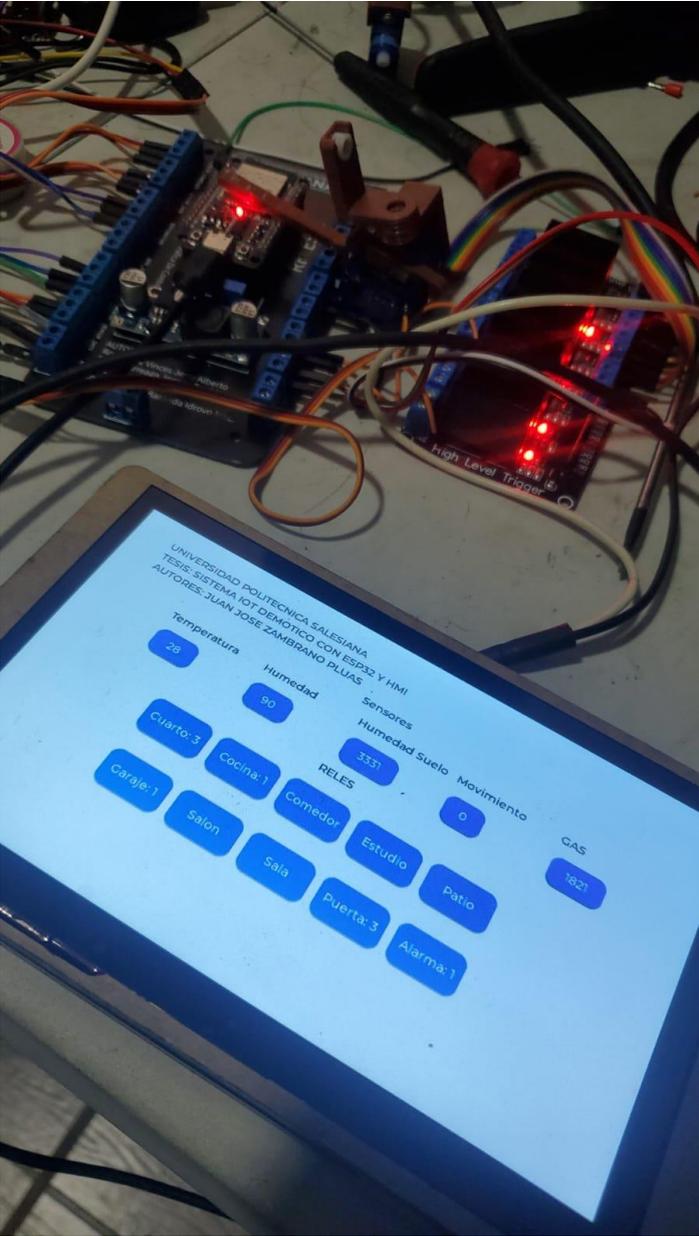


Figura 17 Pruebas de laboratorio de pantalla HMI

Para la electrónica se realizó un diseño de placa PCB para montar el ESP32 y las borneras que conectan a los sensores. Tal como se observa en la figura 18.



Figura 18 Placa PCB de prototipo

En la figura 19 se observa el diseño en 3D de la placa PCB.

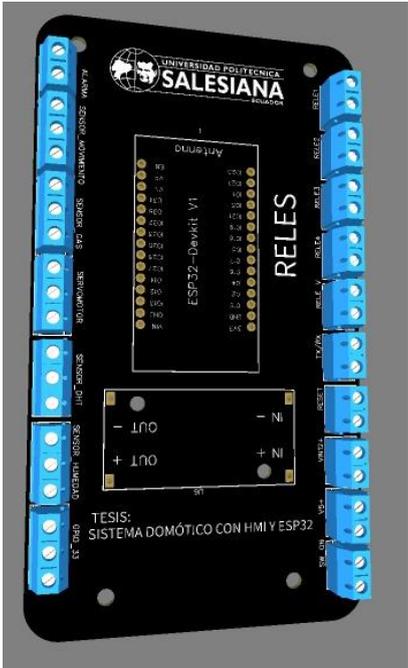


Figura 19 Diseño 3D de placa PCB

En la figura 20 se observa el diseño de las pistas de la tarjeta PCB.

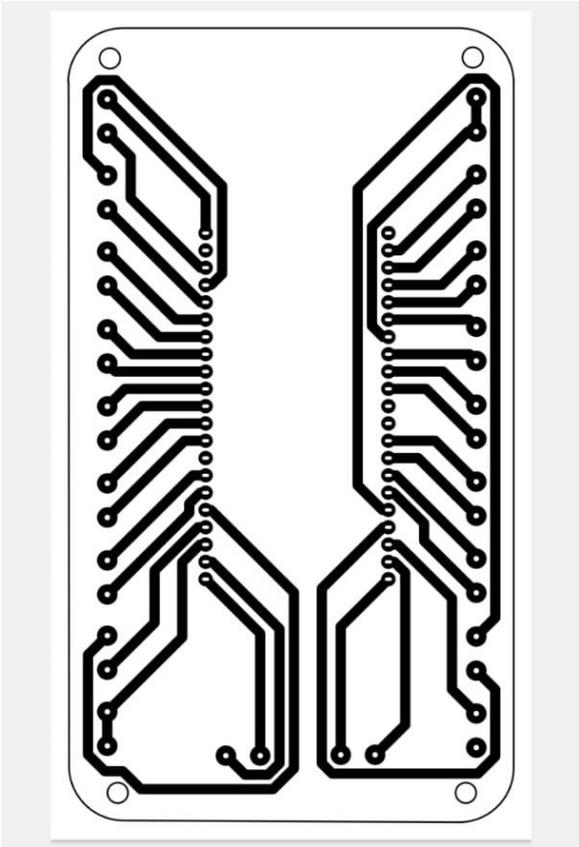


Figura 20 Diseño de pistas de placa PCB del prototipo

3.6 Diseño de maqueta de vivienda domótica

Se realizó un diseño de maqueta de vivienda domótica con divisiones para dormitorios, sala, cuarto de estudio, garaje, patio, cocina y baños, de material MDF, plywood y acrílico, a una escala de 1:50, tal como se muestra en la figura 21.



Figura 21 Maqueta de vivienda domótica

La electrónica, así como las fuentes de poder, la pantalla HMI y los relés se sitúan a un lado de la maqueta, tal como se observa en la figura 22.

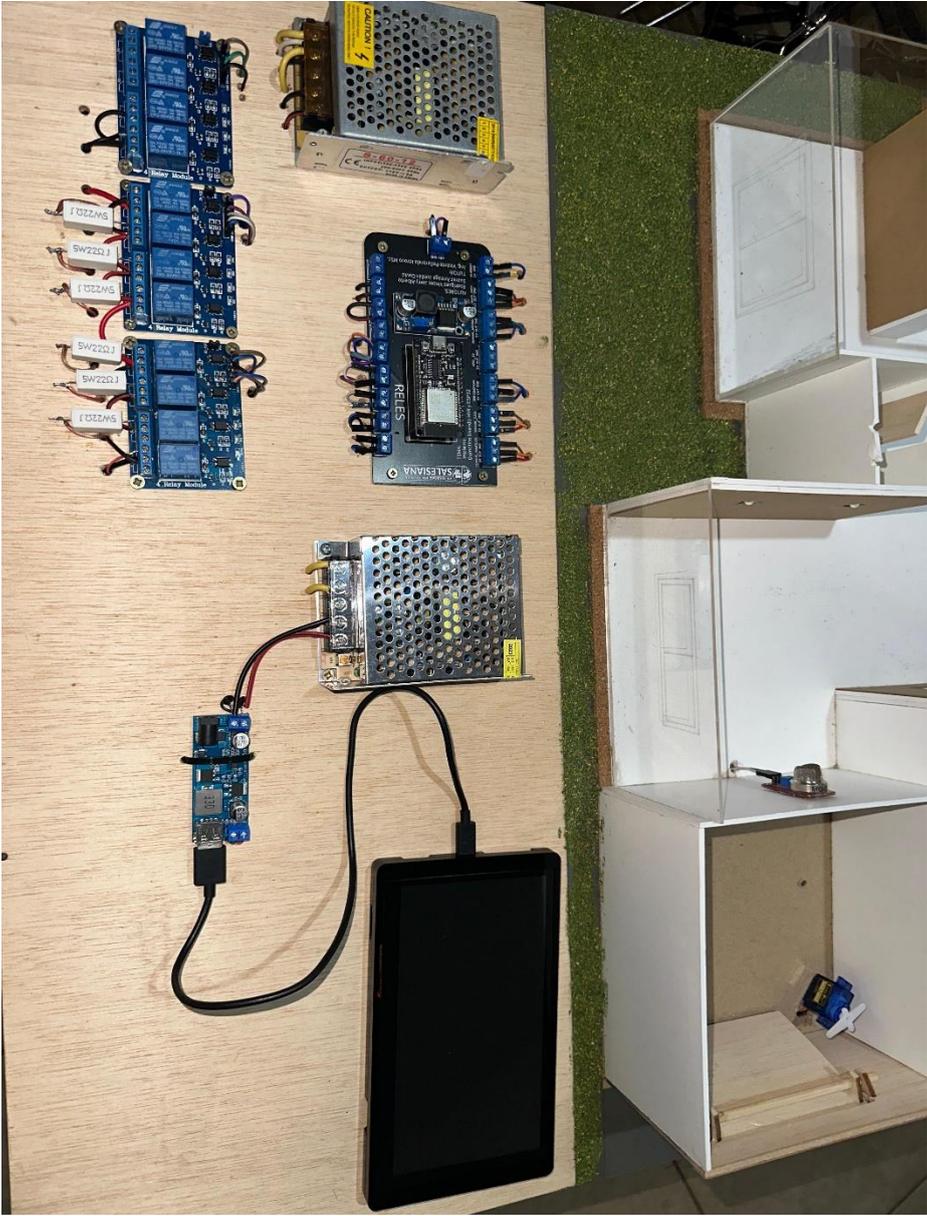


Figura 22 Electrónica y pantalla HMI

En la figura 23 se observa en forma general toda la estructura de la maqueta junto a la electrónica del prototipo.



Figura 23 Vista general de la maqueta

3.7 Montaje de elementos en maqueta domótica

En esta sección se observan las imágenes del montaje de elementos como sensores, actuadores y demás elementos para el prototipo domótico.

En la figura 24 se observa el montaje de las fuentes de poder de 12VDC 5A.



Figura 24 Montaje de fuentes de poder

En las figuras 25 y 26 se observa el montaje de la puerta de garaje.



Figura 25 Montaje puerta garaje



Figura 26 Montaje motor en puerta garaje

En las figuras 27 y 28 se observa el montaje de la piscina en la maqueta.



Figura 27 Montaje de piscina en maqueta



Figura 28 Montaje de tanque de piscina

En las figuras 29, 30 y 31 se observa el montaje de relé con sus respectivas resistencias de potencia 22 ohmios 5 watts.

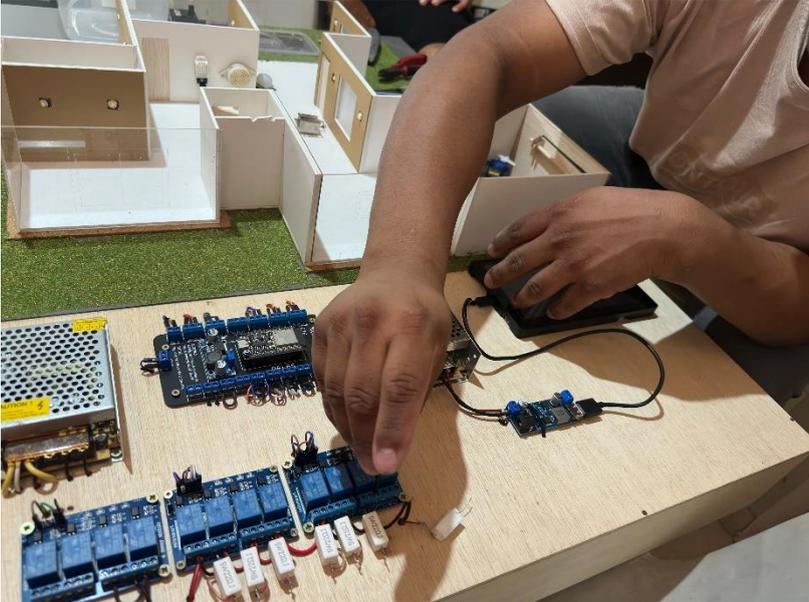


Figura 29 Montaje de resistencias de potencia



Figura 30 Montaje de relés

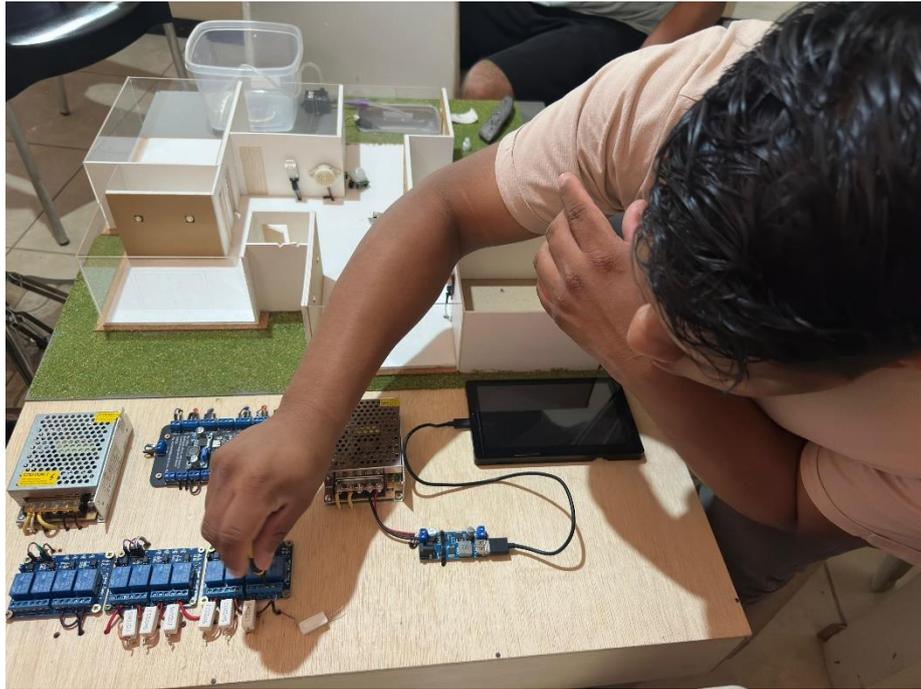


Figura 31 Montaje de cableado de módulo de relés.

En la figura 32 se observa el montaje de la pantalla HMI

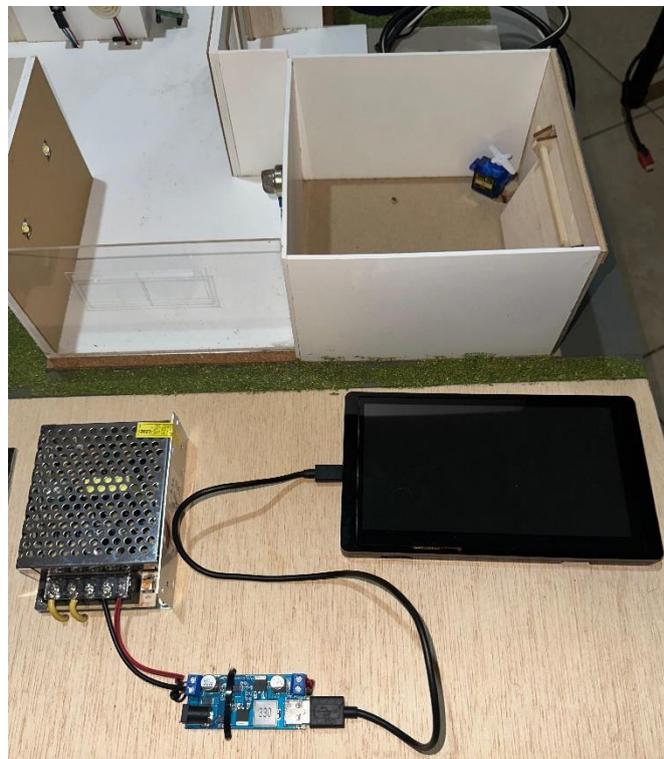


Figura 32 Montaje pantalla HMI

En la figura 33 se observa el montaje del sensor de humedad de suelo que es utilizado para medir el nivel de agua en la piscina.



Figura 33 Sensor de humedad de suelo

En la figura 34 se observa el montaje del sensor de movimiento PIR



Figura 34 Sensor de movimiento

En la figura 35 se observa el montaje de los sensores de temperatura y humedad relativa, el buzzer, el sensor de gas y el sensor de movimiento.



Figura 35 Montaje de sensores y buzzer

En la figura 36 se observan los módulos relés con las resistencias de alta potencia, las mismas que sirven para bajar la corriente que llegan a los leds de alta potencia.

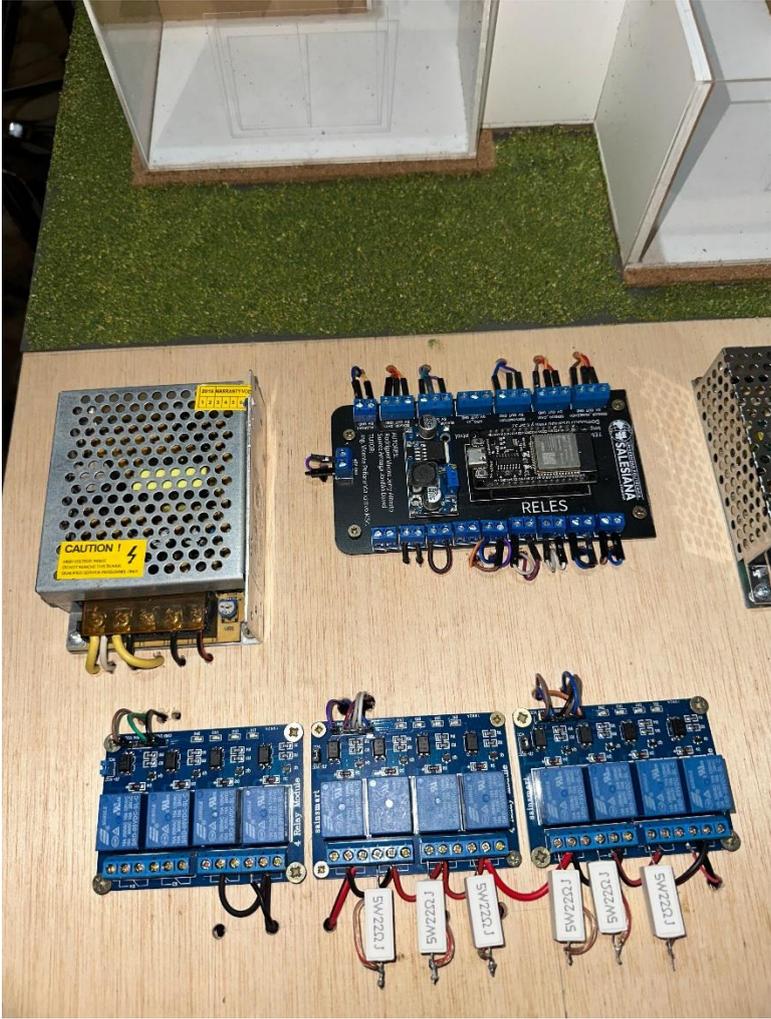


Figura 36 Módulos relés de 4 canales

4 Resultados

En esta sección de resultados se encuentra las pruebas finales realizadas con el prototipo domótico de interfaz HMI y ESP32.

Como se observa en las siguientes imágenes se detallan los elementos conectados e instalados en la maqueta.

En la figura 37 se observa la maqueta con sus divisiones que representan dormitorios, sala, cocina y espacios típicos dentro de una vivienda, cada división tiene sus respectivas luces leds de alta potencia.



Figura 37 Divisiones con las luces leds encendidas

En la figura 38 se observa la ubicación de las fuentes de poder de 12V 5A que alimentan a la pantalla HMI y a la placa de desarrollo con el ESP32. Una vez energizado la maqueta se puede observar las luces de cada relé encendida, lo que significa que activan las salidas de luces y sensores.

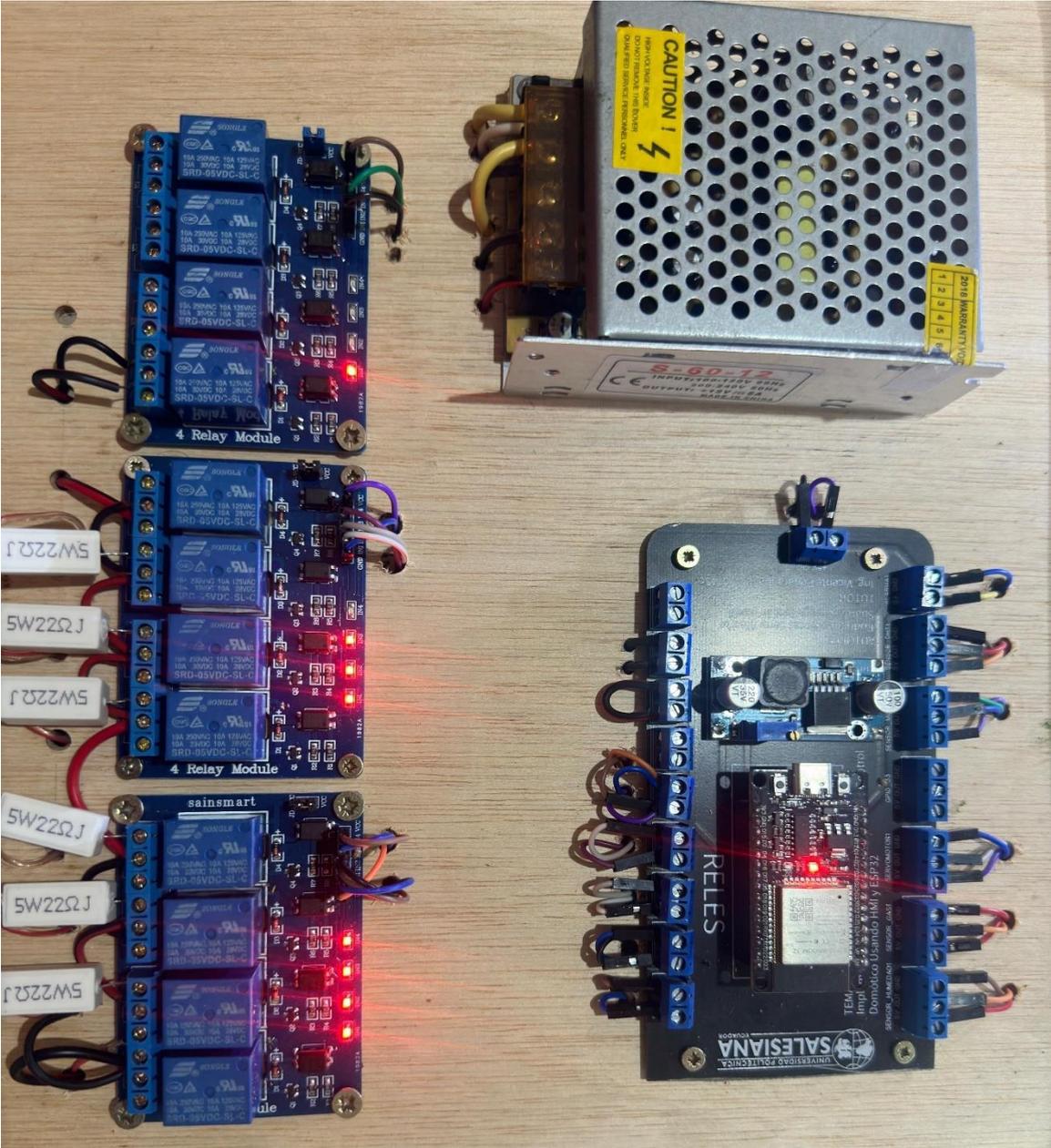


Figura 38 Relé activos

En la figura 39 se observa el módulo regulador de voltaje de 12 VDC a 5 VDC con salida USB, el mismo que es utilizado para alimentar las luces LEDs y la pantalla HMI.

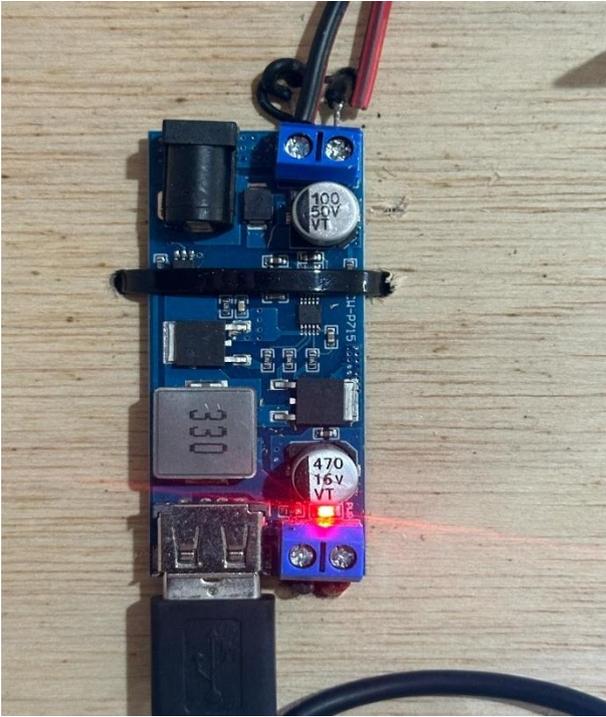


Figura 39 Módulo regulador de voltaje

En la figura 40 se observa la cerradura de la puerta principal, la cual es activada desde el panel HMI.



Figura 40 Cerradura eléctrica activada desde la pantalla HMI

En la figura 41 se observa el sensor de humedad de suelo instalado junto a la piscina, también se puede observar la bomba de agua de 12 VDC que se activa mediante el panel HMI.

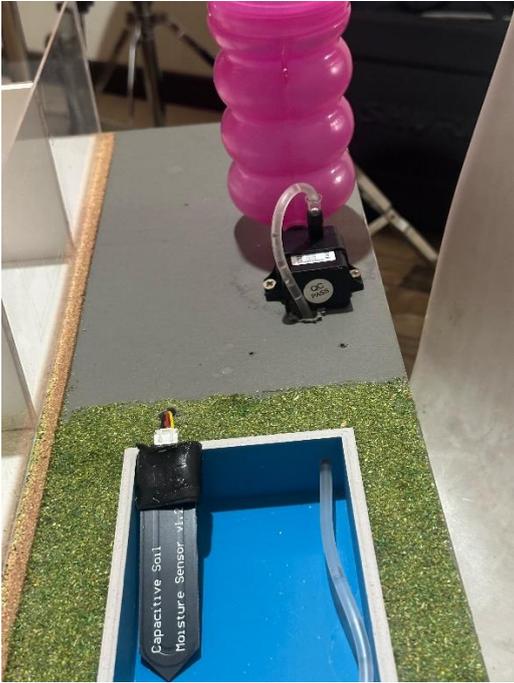


Figura 41 Sección piscina

En la figura 42 se observa el servo SG90, el cual simula el levantamiento de la puerta de garaje.

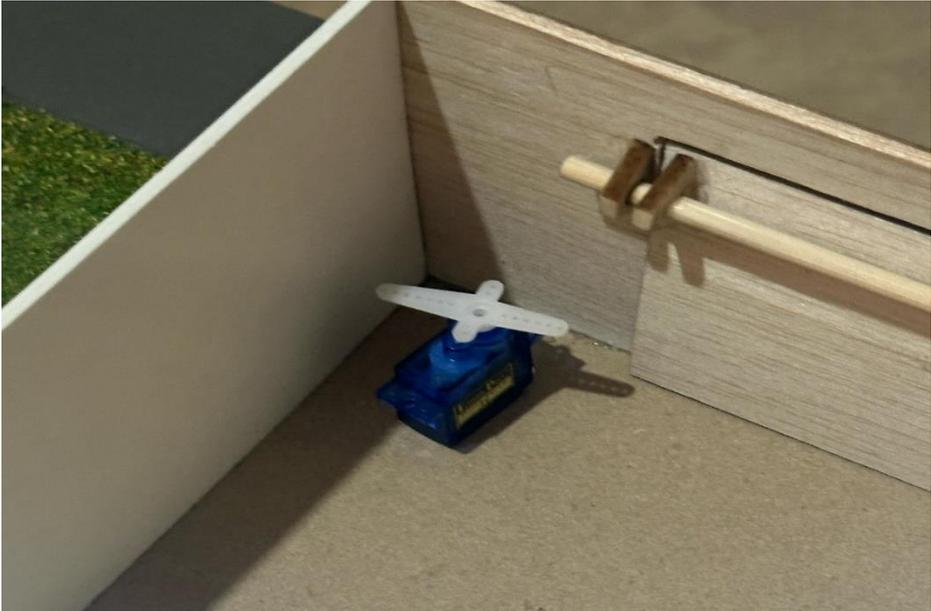


Figura 42 Módulo servo SG90

En la figura 43 se observa el encendido de la pantalla HMI

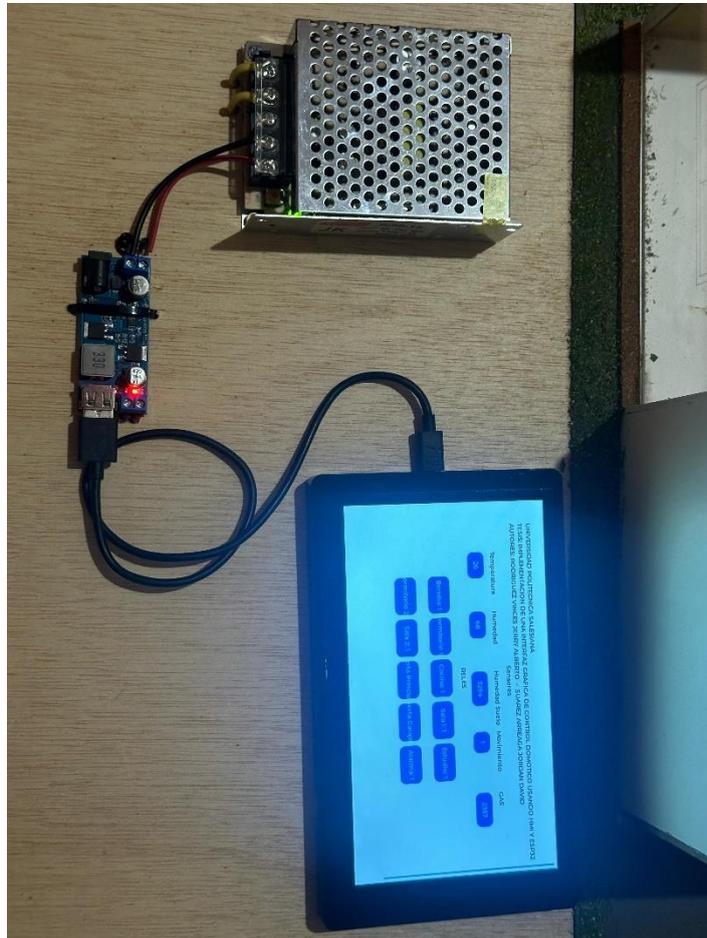


Figura 43 Encendido de pantalla HMI

En las figuras 44 y 45 se observan las configuraciones de la pantalla HMI, con sus respectivas botoneras de control de luces, alarma, puerta principal, motor de agua, motor de puerta de garaje, etc.

La pantalla HMI del vendor Elecrow es previamente configurada desde el IDE de Arduino y el programa Thonny el cual se carga los códigos para la configuración de los botones táctiles de encendido de luces, activación de alarma, cerradura, datos de la temperatura, humedad relativa, humedad de suelo, movimiento, etc.

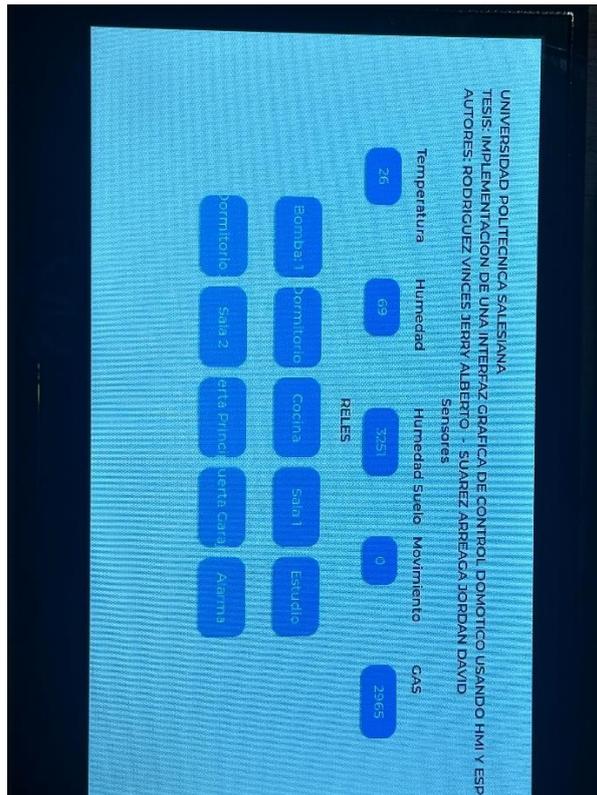


Figura 44 Disposición de botones en pantalla HMI

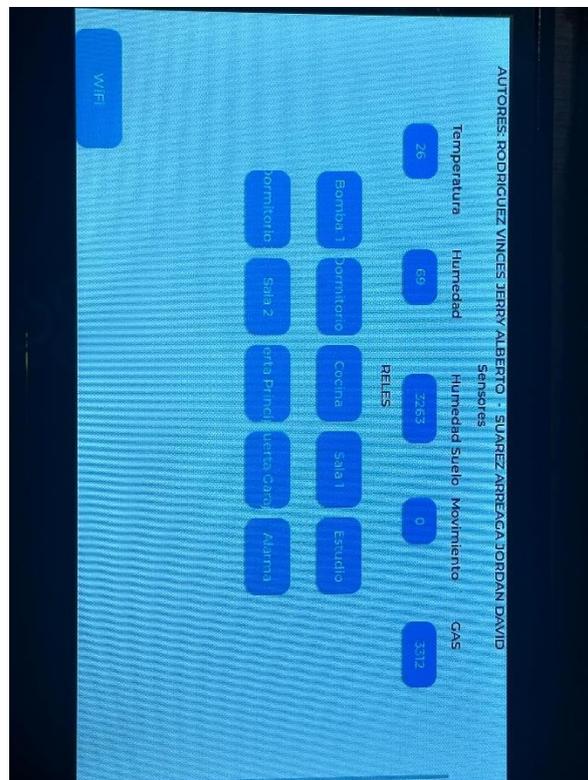


Figura 45 Botón de WIFI en pantalla HMI

5 Conclusiones

Las conclusiones del proyecto de investigación se detallan a continuación:

- Se realizó un sistema de control domótico que utiliza una Interfaz Hombre Máquina (HMI) para el hogar. Para el cumplimiento de este objetivo principal se utilizó una interfaz HMI del vendor Elecrow, que, junto con el ESP32, sensores y actuadores instalados sobre una maqueta domótica, las pruebas fueron exitosas completando lo requerido al inicio de la investigación.
- Se realizó el diseño de un prototipo para la implementación de un sistema domótico dentro de una casa, para el cumplimiento de este objetivo se utilizó una maqueta en escala de una vivienda, con divisiones para dormitorios, sala, cocina, baños, etc. Dentro de la maqueta se instalaron los sensores, actuadores y elementos que complementan el sistema domótico para una vivienda.
- Se realizó la implementación de una interfaz HMI, para el control y monitoreo de dispositivos. Para el cumplimiento de este objetivo se utilizó una interfaz HMI táctil que tiene el ESP32 integrado como su controlador principal, la configuración de la HMI se la realizó con MicroPyhton.
- Se realizó la implementación de un histórico de accionamiento de las alarmas, esto se observa en la pantalla de manera incremental, en cada botón del sistema HMI.

6 Recomendaciones

Las recomendaciones del proyecto de investigación se detallan a continuación:

- Se recomienda implementar este sistema en un domicilio real para observar el funcionamiento, durabilidad y practicidad del proyecto. Con esta información se puede realizar una solución llave en mano para sistemas de domotización local a bajo costo implementado para urbanizaciones y casas inteligentes.
- Se recomienda implementar más sensores y actuadores que puedan sumarse al sistema domótico, la ventaja de trabajar con elementos como el ESP32 es que son elementos escalables que permiten incrementar más entradas o conexiones paralelas WIFI para conectar más elementos como sensores y actuadores.
- Se recomienda investigar si el prototipo acepta más tipos de pantallas HMI, ya que actualmente en el mercado se encuentran varios vendedores que desarrollan este tipo de elementos para prototipos.
- Se recomienda conectar el sistema domótico a la nube e implementar una APP para el control remoto desde un dispositivo móvil, este proyecto puede utilizarse para encaminar otros tipos de investigación y desarrollo de proyectos futuros.

7 Bibliografía

- Casadomo.com. (2024). *El mercado de la Domótica e Inmótica continúa creciendo, según el último estudio de CEDOM* • CASADOMO. 2024. <https://www.casadomo.com/2018/02/01/mercado-domotica-inmotica-continua-creciendo-ultimo-estudio-cedom>
- Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet of Things Journal*, 1(4), 349–359. <https://doi.org/10.1109/JIOT.2014.2337336>
- elecrow.com. (2024). *ESP32 display-7.0 Inch HMI Display 800x480 RGB TFT LCD Touch Screen Compatible with Arduino/LVGL*. <https://www.elecrow.com/esp32-display-7-inch-hmi-display-rgb-tft-lcd-touch-screen-support-lvgl.html>
- espressif.com. (2024). *ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems*. 2024. <https://www.espressif.com/en/products/socs/esp32>
- Miori, V., & Russo, D. (2014). Domotic Evolution towards the IoT. *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, 809–814. <https://doi.org/10.1109/WAINA.2014.128>
- mouser.ec. (2024). *ESP32-S3-WROOM-1-N4 Espressif Systems | Mouser Ecuador*. <https://www.mouser.ec/ProductDetail/Espressif-Systems/ESP32-S3-WROOM-1-N4?q=Li%252BoUPsLEnsO%2FkiJOUK0FQ%3D%3D>
- Murphy, R. R. (2018). Smart houses and domotics. *Science Robotics*, 3(24), eaav6015. <https://doi.org/10.1126/scirobotics.aav6015>
- naylampmechatronics.com. (2024). *Módulo Relay, 4 canales 5VDC*. <https://naylampmechatronics.com/drivers/152-modulo-relay-4-canales-5vdc.html>
- Naylampmechatronics.com. (2024a). *Sensor de Gas MQ2*.

<https://naylampmechatronics.com/sensores-gas/71-sensor-mq-2-gas-glp-gnv.html>

Naylampmechatronics.com. (2024b). *Sensor de Humedad de Suelo Capacitivo v1.2*.

<https://naylampmechatronics.com/sensores-temperatura-y-humedad/538-sensor-de-humedad-de-suelo-capacitivo-v1.html>

Naylampmechatronics.com. (2024c). *Sensor de temperatura y humedad relativa DHT22 (AM2302)*.

<https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>

pentadom.com. (2024). ► *Tipos de Sensores para Domótica | Usos y aplicaciones -*

Pentadom. 2024. <https://pentadom.com/tipos-de-sensores-para-domotica/>

Rodríguez-Serra, M., García-Vásquez, P., & Libaque-Saenz, C. F. (2022). Enhancing student learning with domotics projects. *2022 Congreso de Tecnología, Aprendizaje y Enseñanza de La Electrónica (XV Technologies Applied to Electronics Teaching Conference)*, 1–7.

<https://doi.org/10.1109/TAEE54169.2022.9840653>

8 Anexos

8.1 Código ESP32 pantalla HMI

Definición de pines y librerías

En esta sección, se definen las librerías necesarias y los pines a los que se conectarán los sensores y actuadores.

```
#include <WiFi.h>
#include <WebServer.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>
#include "DHT.h"
#include <ThingerESP32.h>
#include <mi_wifi.h>

// Definiciones de pines
#define AOUT_PIN 34 // Sensor de humedad del suelo
#define AOUT_PIN2 35 // Sensor de gas
#define DHTPIN 4 // Pin digital conectado al sensor DHT
#define DHTTYPE DHT22 // Tipo de sensor DHT
#define SERVO_PIN 32
#define ALARMA_PIN 12
#define MOTION_SENSOR_PIN 27
```

Configuración de credenciales y WiFi

Aquí se definen las credenciales para conectar el dispositivo a Thinger.io y la red WiFi.

```
// Configura tus credenciales de Thinger.io
#define THINGER_USERNAME "jerryh_1311"
#define THINGER_DEVICE_ID "esp32"
#define THINGER_DEVICE_CREDENTIAL "mQRG%z!OCZSVa29c"

// Configuración de WiFi
const char* ssid = "ESP32_Server";
const char* password = "Clave123456";
ThingESP32      thing(THINGER_USERNAME,      THINGER_DEVICE_ID,
THINGER_DEVICE_CREDENTIAL);
```

Declaración de objetos y variables

Se declaran las variables y objetos necesarios para el control de relés, sensores, y el servo motor.

```
// Declaración de objetos y variables
DHT dht(DHTPIN, DHTTYPE);

Servo servoMotor;

WebServer server(80);

// Definiciones para control de relés y sensores
const int numReles = 8;
const int relePins[numReles] = {23, 22, 21, 19, 18, 5, 2, 15};
bool releStates[numReles] = {false};
```

Control del servo motor

En este bloque se gestiona el movimiento del servo utilizando una máquina de estados.

```
// Variables de control para el servo

enum ServoState { IDLE, MOVING_UP, WAITING, MOVING_DOWN };

ServoState servoState = IDLE;

unsigned long lastTime = 0;

const int delayTime = 3000; // Tiempo de espera en milisegundos

int servoPosition = 0;

bool moveRequested = false;
```

Control de la alarma

Este bloque maneja el estado de la alarma, que puede ser activada tanto manualmente como en respuesta a eventos como la detección de intrusión o fuga de gas.

```
// Variables para la alarma

bool alarmaActivada = false;

unsigned long alarmaStart = 0;

bool alarmaSonando = false;

unsigned long alarmaInterval = 2000; // Tiempo de encendido de alarma en ms

unsigned long alarmaPausa = 2000; // Tiempo de pausa en ms
```

Página HTML

En esta parte se define la estructura y estilo de la página web que se servirá al usuario para controlar los dispositivos.

```
// Página HTML mejorada
```

```
const char index_html[] PROGMEM = R"rawliteral(
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>CONTROL DOMÓTICO</title>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <style>
```

```
    body {
```

```
      font-family: Arial, sans-serif;
```

```
      background-color: #002147; /* Azul oscuro */
```

```
      margin: 0;
```

```
      padding: 20px;
```

```
    }
```

```
    .center-content {
```

```
      text-align: center;
```

```
      margin-bottom: 20px;
```

```
      color: #000;
```

```
    }
```

```
    .button, .sensor {
```

```
      border-radius: 5px;
```

```
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
margin: 10px auto;
transition: transform 0.2s;
}
.button {
  display: inline-block;
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
  border: none;
  color: #FFD700;
  background-color: #007bff;
  transition: background-color 0.3s, transform 0.2s;
  width: 100%;
  max-width: 250px;
  box-sizing: border-box;
}
.button:hover {
  background-color: #0056b3;
}
.footer {
  text-align: center;
  margin-top: 20px;
  font-size: 14px;
  background-color: #002147;
  padding: 10px;
  color: #fff;
  font-weight: bold;
```

```

    }
</style>
</head>
<body>
  <div class="container">
    <h2>CONTROL DOMÓTICO</h2>
    <div class="button-container" id="relays">
      <button class="button" id="rele1Button" onclick="toggleRele(1)">Bomba
(OFF)</button>
      <button class="button" id="rele2Button" onclick="toggleRele(2)">Dormitorio 1
(OFF)</button>
      <button class="button" id="rele3Button" onclick="toggleRele(3)">Cocina
(OFF)</button>
      <button class="button" id="rele4Button" onclick="toggleRele(4)">Sala 1
(OFF)</button>
      <button class="button" id="rele5Button" onclick="toggleRele(5)">Estudio
(OFF)</button>
      <button class="button" id="rele6Button" onclick="toggleRele(6)">Dormitorio 2
(OFF)</button>
      <button class="button" id="rele7Button" onclick="toggleRele(7)">Sala 2
(OFF)</button>
      <button class="button" id="rele8Button" onclick="toggleRele(8)">Puerta
Principal (OFF)</button>
    </div>

    <h2>SENSORES</h2>
    <div class="sensor" id="sensor1">Temperatura Ambiente:</div>

```

```

<div class="sensor" id="sensor2">Humedad Relativa:</div>
<div class="sensor" id="sensor3">Nivel de Agua Piscina:</div>
<div class="sensor" id="sensor4">Detección de Movimiento:</div>
<div class="sensor" id="sensor5">Fuga de Gas:</div>
</div>

<div class="footer">
  <p>PROYECTO DE TITULACIÓN</p>
</div>
</body>
</html>
)rawliteral";

```

Manejo de los relés

Este bloque permite activar o desactivar los relés, los cuales controlan dispositivos como luces, bombas y otros actuadores.

// Función para manejar los relés del 1 al 10

```

void handleRele(int index) {
  if (index >= 0 && index < numReles) {
    releStates[index] = !releStates[index];
    digitalWrite(relePins[index], releStates[index] ? LOW : HIGH); // Lógica invertida
    server.send(200, "text/plain", releStates[index] ? "ON" : "OFF");
    Serial.printf("Estado del relé %d: %s\n", index + 1, releStates[index] ? "activado" :
"desactivado");
  } else {
    server.send(404, "text/plain", "Relé no encontrado");
  }
}

```

```
}  
}
```

Manejo del servo motor (puerta del garaje)

Este bloque controla el movimiento del servo motor (utilizado para abrir y cerrar la puerta del garaje).

```
// Función para abrir/cerrar la puerta del garaje
```

```
void handleRele9() {  
    if (servoState == IDLE) {  
        moveRequested = true; // Marca que se solicitó el movimiento  
        server.send(200, "text/plain", "Movimiento del servo iniciado");  
    } else {  
        server.send(200, "text/plain", "El servo ya está en movimiento");  
    }  
}
```

```
// Controla el movimiento del servo con la máquina de estados
```

```
void updateServo() {  
    unsigned long currentTime = millis();  
  
    switch (servoState) {  
        case IDLE:  
            if (moveRequested) {  
                moveRequested = false;  
                servoState = MOVING_UP;  
                lastTime = currentTime;  
            }  
    }  
}
```

```

    break;
case MOVING_UP:
    if (servoPosition < 180) {
        servoPosition += 1;
        servoMotor.write(servoPosition);
        lastTime = currentTime;
    } else {
        servoMotor.detach(); // Desactiva el servo al llegar a 180 grados
        servoState = WAITING;
        lastTime = currentTime;
    }
    break;

case WAITING:
    if (currentTime - lastTime >= delayTime) {
        servoMotor.attach(SERVO_PIN); // Vuelve a activar el servo
        servoState = MOVING_DOWN;
    }
    break;

case MOVING_DOWN:
    if (servoPosition > 0) {
        servoPosition -= 1;
        servoMotor.write(servoPosition);
        lastTime = currentTime;
    } else {
        servoState = IDLE; // Retorna a estado inactivo

```

```
    }  
    break;  
  }  
}
```

Alarma y detección de intrusos y gas

Este bloque maneja las detecciones de gas y movimiento, activando la alarma en caso necesario.

```
// Función de detección de movimiento o gas
```

```
void handleAlarma() {
```

```
    if (digitalRead(MOTION_SENSOR_PIN) == HIGH || analogRead(AOUT_PIN2) >  
    umbralGas) {
```

```
        if (!alarmaActivada) {
```

```
            alarmaActivada = true;
```

```
            alarmaStart = millis();
```

```
            digitalWrite(ALARMA_PIN, HIGH); // Activa la alarma
```

```
        }
```

```
    } else {
```

```
        if (alarmaActivada) {
```

```
            alarmaActivada = false;
```

```
            digitalWrite(ALARMA_PIN, LOW); // Desactiva la alarma
```

```
        }
```

```
    }
```

```
}
```

8.2 Código de MicroPython

```
import lvgl as lv
import lv_utils
import tft_config
import gt911
from machine import Pin, I2C, reset
import time
import network
import socket
import json
import urequests as requests

WIDTH = 800
HEIGHT = 480

# Agregar un delay de 10 segundos para que se inicien los módulos
time.sleep(2)
print("Comienzo.")
# tft driver
tft = tft_config.config()

# touch driver
i2c = I2C(1, scl=Pin(20), sda=Pin(19), freq=400000)
tp = gt911.GT911(i2c, width=800, height=480)
tp.set_rotation(tp.ROTATION_INVERTED)

lv.init()

if not lv_utils.event_loop.is_running():
    event_loop = lv_utils.event_loop()
    print(event_loop.is_running())

# create a display 0 buffer
disp_buf0 = lv.disp_draw_buf_t()
```

```

buf1_0 = bytearray(WIDTH * HEIGHT)
disp_buf0.init(buf1_0, None, len(buf1_0) // lv.color_t.__SIZE__)

# register display driver
disp_drv = lv.disp_drv_t()
disp_drv.init()
disp_drv.draw_buf = disp_buf0
disp_drv.flush_cb = tft.flush
disp_drv.hor_res = WIDTH
disp_drv.ver_res = HEIGHT
disp0 = disp_drv.register()
lv.disp_t.set_default(disp0)

# touch driver init
indev_drv = lv.indev_drv_t()
indev_drv.init()
indev_drv.disp = disp0
indev_drv.type = lv.INDEV_TYPE.POINTER
indev_drv.read_cb = tp.lvgl_read
indev = indev_drv.register()

scr = lv.obj()

class CounterBtn():
    def __init__(self, scr):
        nombres_lugares = ["Bomba", "Dormitorio 1", "Cocina", "Sala 1", "Estudio", "Dormitorio
2", "Sala 2", "Puerta Principal", "Puerta Garaje", "Alarma"]
        self.counters = [0] * 10 # Inicializa una lista de contadores para cada botón
        self.buttons = []
        for i in range(10):
            btn = lv.btn(scr)
            btn.set_size(80, 50)
            x_offset = -200 + (i % 5) * 90 # Ajuste horizontal, cinco botones por fila
            y_offset = 20 if i < 5 else 100 # Ajuste vertical, primera fila y segunda fila
            btn.align(lv.ALIGN.CENTER, x_offset, y_offset)

```

```

btn.add_event_cb(self.btn_event_cb, lv.EVENT.CLICKED, None)
label = lv.label(btn)
label.set_text("{}".format(nombres_lugares[i]))
label.center()
self.buttons.append(btn)

def btn_event_cb(self, evt):
    nombres_lugares = ["Bomba", "Dormitorio 1", "Cocina", "Sala 1", "Estudio", "Dormitorio
2", "Sala 2", "Puerta Principal", "Puerta Garaje", "Alarma"]
    btn = evt.get_target()
    index = self.buttons.index(btn)
    self.counters[index] += 1
    print("{}: {}".format(nombres_lugares[index], self.counters[index]))

if wifi.isconnected():
    url = "http://{}rele{}".format(SERVER_IP, index + 1)
    try:
        response = requests.get(url)
        if response.status_code == 200:
            label = btn.get_child(0)
            label.set_text("{}: {}".format(nombres_lugares[index], self.counters[index]))
            print("Comando para cambiar el estado de {} enviado con
éxito".format(nombres_lugares[index]))
        else:
            print("Error: Código de estado", response.status_code)
            response.close()
    except Exception as e:
        print("Error al enviar la solicitud:", e)
    else:
        print("WiFi no conectado")

def connect_wifi(evt):
    ssid = "ESP32_Server"
    password = "Clave123456"
    wifi.active(True)

```

```

wifi.connect(ssid, password)

try:
    print("Conectando a WiFi...")
    while not wifi.isconnected():
        pass
    print("Conexión WiFi establecida")

    # Conectar al servidor
    client_socket = socket.socket()
    client_socket.connect((SERVER_IP, SERVER_PORT))
    print("Conexión establecida con el servidor")
except Exception as e:
    print("Error al conectar WiFi:", e)
    reset()

# Crear la pantalla
scr = lv.obj()

# Crear los botones de contador
counterBtn = CounterBtn(scr)

# Crear el botón para conectar al WiFi
btn_connect = lv.btn(scr)
btn_connect.set_size(120, 50)
btn_connect.align(lv.ALIGN.TOP_LEFT, 20, 500)
btn_connect.add_event_cb(connect_wifi, lv.EVENT.CLICKED, None)
label_connect = lv.label(btn_connect)
label_connect.set_text("WiFi")
label_connect.center()

# Cargar la pantalla
lv.scr_load(scr)

# Definir etiquetas para mostrar los valores de los sensores

```

```
titulo_label = lv.label(scr)
titulo_label.set_text("UNIVERSIDAD POLITECNICA SALESIANA")
titulo_label.align(lv.ALIGN.TOP_LEFT, 50, 30)
```

```
tema_label = lv.label(scr)
tema_label.set_text("TESIS: IMPLEMENTACION DE UNA INTERFAZ GRAFICA DE
CONTROL DOMOTICO USANDO HMI Y ESP32")
tema_label.align(lv.ALIGN.TOP_LEFT, 50, 50)
```

```
tesistas_label = lv.label(scr)
tesistas_label.set_text("AUTORES: RODRIGUEZ VINCES JERRY ALBERTO - SUAREZ
ARREAGA JORDAN DAVID")
tesistas_label.align(lv.ALIGN.TOP_LEFT, 50, 70)
```

```
sensor1_label = lv.label(scr)
sensor1_label.set_text("Sensores")
sensor1_label.align(lv.ALIGN.TOP_LEFT, 360, 90)
```

```
relays_label = lv.label(scr)
relays_label.set_text("RELES")
relays_label.align(lv.ALIGN.TOP_LEFT, 360, 200)
```

```
sensor1_label = lv.label(scr)
sensor1_label.set_text("Temperatura")
sensor1_label.align(lv.ALIGN.TOP_LEFT, 110, 120)
```

```
# Crear botones de sensores
```

```
btn1 = lv.btn(scr)
btn1.align(lv.ALIGN.TOP_LEFT, 110, 150)
label1 = lv.label(btn1)
label1.set_text("0")
```

```
sensor2_label = lv.label(scr)
sensor2_label.set_text("Humedad")
sensor2_label.align(lv.ALIGN.TOP_LEFT, 240, 120)
```

```
btn2 = lv.btn(scr)
btn2.align(lv.ALIGN.TOP_LEFT, 240, 150)
label2 = lv.label(btn2)
label2.set_text("0")

sensor3_label = lv.label(scr)
sensor3_label.set_text("Humedad Suelo")
sensor3_label.align(lv.ALIGN.TOP_LEFT, 370, 120)

btn3 = lv.btn(scr)
btn3.align(lv.ALIGN.TOP_LEFT, 370, 150)
label3 = lv.label(btn3)
label3.set_text("0")

sensor4_label = lv.label(scr)
sensor4_label.set_text("Movimiento")
sensor4_label.align(lv.ALIGN.TOP_LEFT, 500, 120)

btn4 = lv.btn(scr)
btn4.align(lv.ALIGN.TOP_LEFT, 500, 150)
label4 = lv.label(btn4)
label4.set_text("0")

sensor5_label = lv.label(scr)
sensor5_label.set_text("GAS")
sensor5_label.align(lv.ALIGN.TOP_LEFT, 630, 120)

btn5 = lv.btn(scr)
btn5.align(lv.ALIGN.TOP_LEFT, 630, 150)
label5 = lv.label(btn5)
label5.set_text("0")

SERVER_IP = "192.168.1.1" # Dirección IP del servidor
SERVER_PORT = 80
```

```

wifi = network.WLAN(network.STA_IF)

try:
    print("Hint: Press Ctrl+C to end the program")
    while True:
        if wifi.isconnected():
            url = "http://{}/sensor".format(SERVER_IP)
            try:
                response = requests.get(url)
                if response.status_code == 200:
                    json_data = response.json()
                    label1.set_text(str(json_data['sensor1']))
                    label2.set_text(str(json_data['sensor2']))
                    label3.set_text(str(json_data['sensor3']))
                    label4.set_text(str(json_data['sensor4']))
                    label5.set_text(str(json_data['sensor5']))
                    print("Datos recibidos:", json_data)

            else:
                print("Error: Código de estado", response.status_code)
                response.close()
        except Exception as e:
            print("Error al enviar la solicitud:", e)

        time.sleep(1) # Esperar 10 segundos antes de enviar el siguiente comando
        print("Esperando 1 segundo...")

except KeyboardInterrupt:
    print("The program stopped running, ESP32 has restarted...")
    reset()
finally:
    wifi.disconnect()

```

8.3 Esquemático del ESP32-S3-WROOM-1

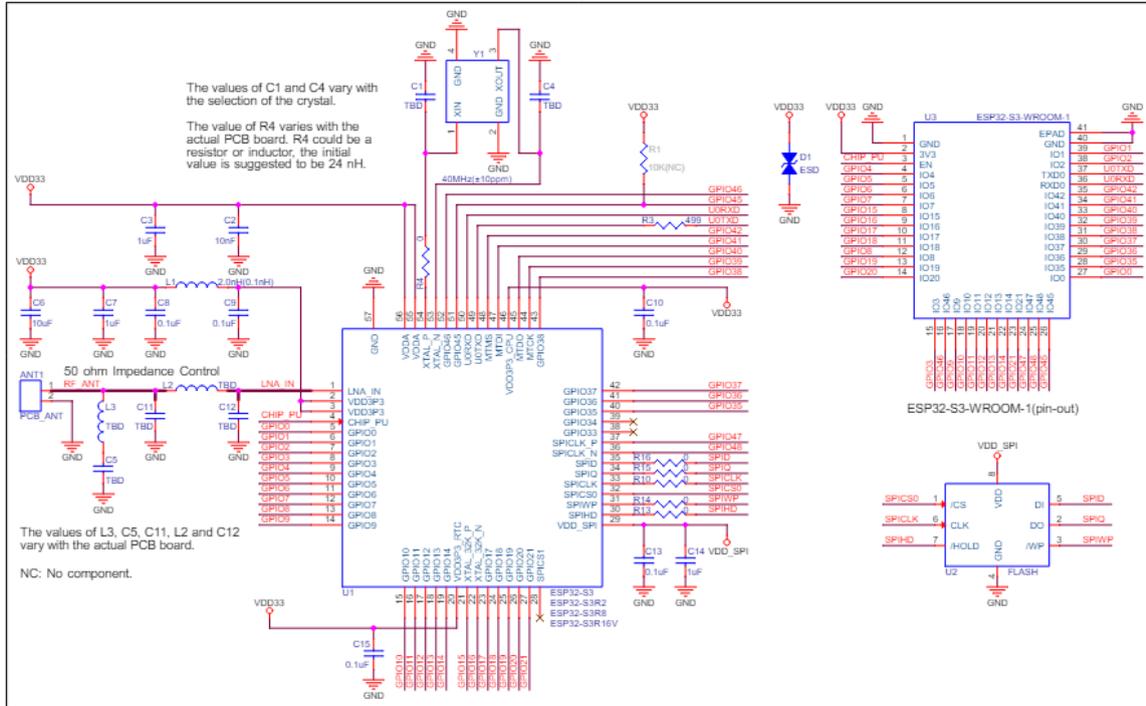


Figura 46 Esquemático del ESP32-S3-WROOM-1

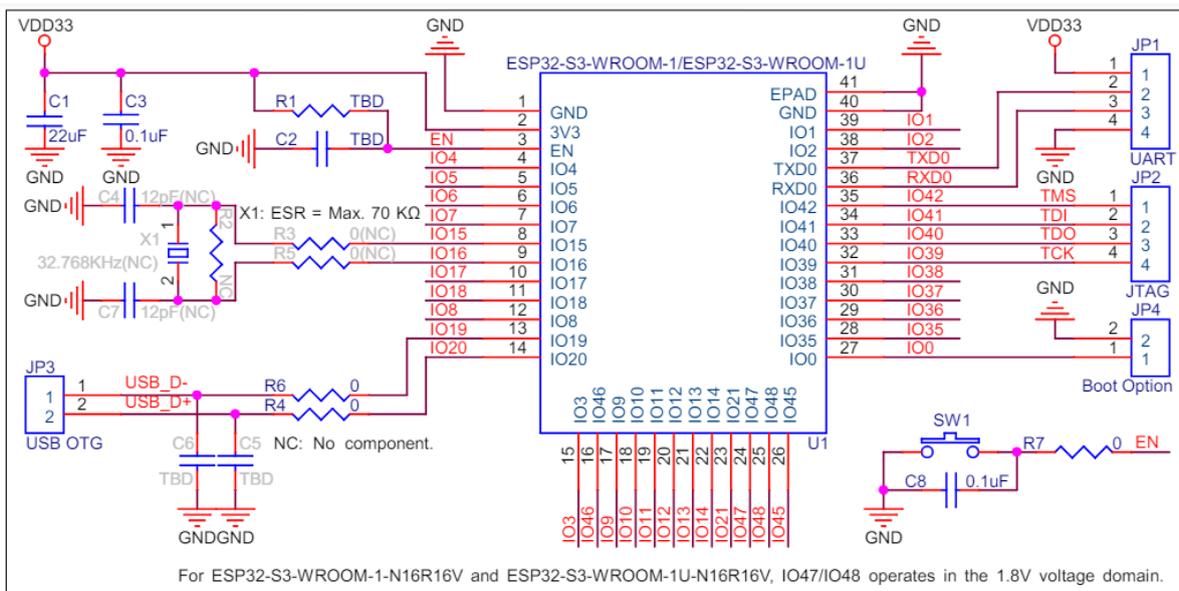


Figura 47 Esquemático de periféricos del ESP32-S3-WROOM-1

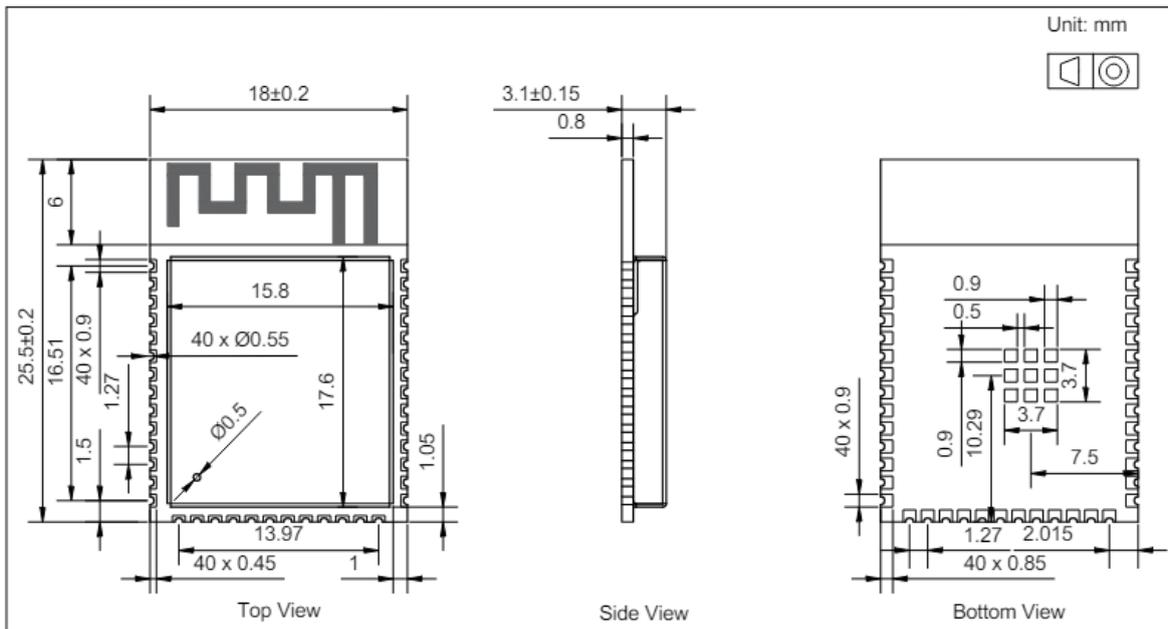


Figura 48 Dimensiones físicas del ESP32-S3-WROOM1

8.4 Boom de materiales

En la tabla 4 se observa el boom de materiales del proyecto.

Tabla 4 Boom de materiales del prototipo

BOOM DE MATERIALES
Elecrow ESP32 HMI
Sensor de Gas MQ2
Sensor PIR Movimiento
Buzzer
Sensor DHT22
Servo SG90
Sensor humedad suelo
Contacto puerta
TARJETA PCB
Cables de conexiones
Case de prototipo