



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE MECATRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN
MULTIPLATAFORMA DE ASISTENTES DE VOZ CON EL PROTOCOLO
LOXONE DENTRO DE UN SMART HOME**

Trabajo de titulación previo a la obtención del
título de Ingeniera en Mecatrónica /
Ingeniero en Mecatrónica

**AUTORES: DANIELA ESTEFANÍA PEZANTEZ CHIMBO
RONAN RODRIGO SOTO VALVERDE**
TUTOR: ING. DIEGO PAÚL CHACÓN TROYA, MDHD.

Cuenca - Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Daniela Estefanía Pezantez Chimbo, con documento de identificación N° 0105901078 y Ronan Rodrigo Soto Valverde con documento de identificación N° 1105778672; 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.

Cuenca, 20 de julio del 2023

Atentamente,



Daniela Estefanía Pezantez Chimbo
0105901078



Ronan Rodrigo Soto Valverde
1105778672

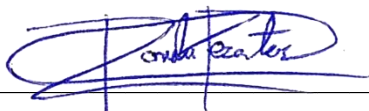
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Daniela Estefanía Pezantez Chimbo con documento de identificación N° 0105901078 y Ronan Rodrigo Soto Valverde con documento de identificación N° 1105778672, 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 técnico: "Diseño e implementación de un sistema de comunicación multiplataforma de asistentes de voz con el protocolo Loxone dentro de un Smart Home", el cual ha sido desarrollado para optar por el título de: Ingeniera en Mecatrónica / Ingeniero en Mecatrónica, 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.

Cuenca, 20 de julio del 2023

Atentamente,



Daniela Estefanía Pezantez Chimbo
0105901078



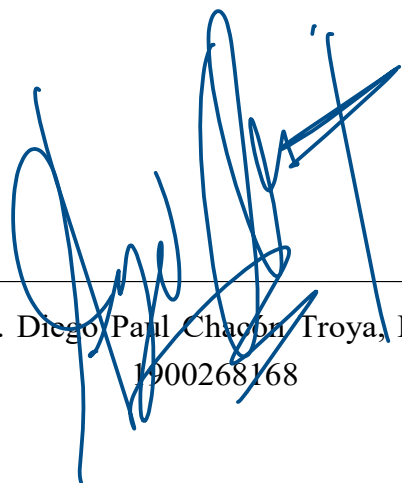
Ronan Rodrigo Soto Valverde
1105778672

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Ing. Diego Paúl Chacón Troya con documento de identificación N° 1900268168, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN MULTIPLATAFORMA DE ASISTENTES DE VOZ CON EL PROTOCOLO LOXONE DENTRO DE UN SMART HOME, realizado por Daniela Estefanía Pezantez Chimbo con documento de identificación N° 0105901078 y Ronan Rodrigo Soto Valverde con documento de identificación N° 1105778672, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 20 de julio del 2023

Atentamente,



Ing. Diego Paul Chacón Troya, Mdhd
1900268168

Dedicatoria

Daniela Estefanía

El presente trabajo va dedicado a mi mamá, mi fuente de inspiración, quien siempre ha sabido salir adelante sin importar las adversidades. Gracias a su valentía, su apoyo incondicional y su amor, este logro no sería posible. Su confianza en mí ha sido mi mayor impulso para enfrentar cada desafío en este camino. Te amo mamá.

A mi hermano, mi ejemplo a seguir, por su apoyo, consejos, guía y cariño. Gracias a él, he llegado hasta donde estoy. Siempre ha estado a mi lado, inspirándome a seguir adelante y a cumplir mis metas.

A Celia, quien ha sido un pilar fundamental en mi vida, me ha enseñado a tener confianza en mí misma. Gracias a su apoyo, descubrí lo fuerte y capaz que puedo ser.

A mi papá, a pesar de ya no estar presente, siento en lo más profundo de mi corazón que estaría orgulloso de mí.

Y a mi pareja, por estar a mi lado siempre animándome a alcanzar mis sueños.

Ronan Rodrigo

Quiero dedicar este trabajo a todas las personas que me apoyaron al transcurso de esta nueva meta en mi vida. En primer lugar, a mis padres Rodrigo y Nancy, quienes a pesar de la distancia supieron apoyarme con cariño de manera emocional e incondicional. Los quiero mucho.

A mis hermanos y mi sobrino, quienes me supieron escuchar y alentar en momentos complicados.

A mi pareja y su familia, les agradezco de corazón por su gran ayuda, por recibirme con alegría en los buenos y malos momentos.

A las amistades que forje al transcurso de esta etapa, por sus valiosos consejos y sus palabras de aliento.

Por último, a todas las demás personas que supieron bríndame su apoyo en cualquier momento de esta meta.

Agradecimientos

Daniela Estefanía

Quiero agradecer a las personas que han sido pilares fundamentales en mi vida y en el logro de esta meta. A mi mamá, gracias por estar siempre para mí, por su paciencia, confianza y amor incondicional que me han impulsado a cumplir mis sueños. A mi hermano, su apoyo y guía han sido fundamentales en este camino. A Celia, Nico y Elif, su presencia y apoyo en cada paso han sido muy importantes para mí. Y a mi pareja, por apoyarme a cumplir mis metas.

Me gustaría además agradecer al Ing. Diego Chacón, por su esfuerzo y dedicación en el desarrollo de este trabajo de titulación. Su orientación, conocimientos y su paciencia han sido elementos esenciales para culminar con éxito este proyecto.

También quiero agradecer a la Ing. Mónica Romero y al Ing. Eugenio Cárdenas, por su empatía, gentileza y bondad.

Ronan Rodrigo

Agradezco principalmente a mis padres Rodrigo y Nancy, por brindarme todo ese apoyo incondicional, paciencia, cariño para lograr esta meta. A mis hermanos y mi sobrino, quienes me supieron apoyar ante cualquier inconveniente. A mi pareja y su familia por su paciencia y apoyo incondicional en el desarrollo de este proyecto. A mis amigos, Johan, Kathleen, Dan, Santiago, Carlos, Jimbo, los cuales me ayudaron en diversos aspectos de mi vida.

Además, me gustaría agradecer al Ing. Diego Chacón, quien nos brindó su apoyo y conocimientos en el desarrollo de este trabajo de titulación.

También a la Ing. Mónica Romero y al Ing. Eugenio Cárdenas, por su comprensión, amabilidad y su gran calidad humana.

A todos ustedes, les estoy sinceramente agradecido por ser parte de este camino y por haber contribuido en el logro de esta meta.

Índice

| | |
|--|-------------|
| Certificado de responsabilidad y autoría del trabajo de titulación | I |
| Certificado de cesión de derechos de autor del trabajo de titulación a la | |
| Universidad Politécnica Salesiana | II |
| Certificado de dirección del trabajo de titulación | III |
| Dedicatoria | IV |
| Agradecimientos | V |
| Resumen | XIII |
| Abstract | XIV |
| 1. Introducción | 1 |
| 2. Problema | 1 |
| 2.1. Antecedentes | 1 |
| 2.2. Descripción del problema | 2 |
| 2.3. Importancia y alcances | 2 |
| 2.4. Delimitación | 2 |
| 2.4.1. Espacial o geográfica | 2 |
| 2.4.2. Temporal | 3 |
| 2.5. Problema General | 3 |
| 2.6. Problemas Específicos | 3 |
| 3. Justificación | 3 |
| 4. Objetivos | 4 |
| 4.1. Objetivo General | 4 |
| 4.2. Objetivos Específicos | 4 |
| 5. Marco Teórico | 4 |
| 5.1. Domótica | 4 |
| 5.1.1. Ventajas de la domótica | 4 |

| | |
|--|-----------|
| 5.2. Protocolo loxone | 5 |
| 5.2.1. Loxone Tree | 6 |
| 5.2.2. Loxone Air | 6 |
| 5.2.3. Loxone Link | 6 |
| 5.2.4. LAN | 6 |
| 5.3. Asistentes de voz | 6 |
| 5.3.1. Alexa | 7 |
| 5.3.2. Siri | 7 |
| 5.3.3. Google Assistant | 7 |
| 5.4. Gateway | 7 |
| 5.5. Microordenadores | 8 |
| 5.6. Node Red | 8 |
| 5.7. Software de diseño mecánico | 8 |
| 5.7.1. Autodesk Inventor | 9 |
| 5.7.2. Fusion 360 | 9 |
| 5.7.3. SolidWorks | 9 |
| 6. Marco metodológico | 10 |
| 6.1. Parámetros iniciales del sistema de comunicación. | 10 |
| 6.1.1. Parámetros de conexión con el protocolo loxone | 10 |
| 6.1.2. Parámetros electrónicos | 11 |
| 6.1.3. Parámetros mecánicos | 11 |
| 6.1.4. Selección de microordenador | 11 |
| 6.1.5. Ambiente de desarrollo | 12 |
| 6.2. Diseño de la estructura externa del módulo. | 14 |
| 6.2.1. Esquema de conexión | 14 |
| 6.2.2. Configuración del ambiente de desarrollo | 15 |
| 6.2.3. Diseño de la estructura de protección externa | 20 |
| 6.2.4. Análisis de temperatura | 23 |
| 6.2.5. Selección del material | 28 |
| 6.3. Programación del sistema de comunicación multiplataforma. | 29 |
| 6.3.1. Reconocimiento de la interfaz y nodos a utilizar | 29 |
| 6.3.2. Nodos de comunicación Google Home y Alexa Smart Home | 36 |
| 6.3.3. Nodo de comunicación HomeKit (Siri) | 38 |
| 6.3.4. Nodo de comunicación Loxone. | 39 |

| | | |
|-----------|---|-----------|
| 6.3.5. | Programación para la conexión de los asistentes de voz. | 41 |
| 6.3.6. | Configuración de comandos de voz Alexa y Google Home. | 51 |
| 6.3.7. | Configuración de comandos de voz Homekit. | 53 |
| 6.4. | Evaluar el funcionamiento del sistema de comunicación multiplataforma. | 54 |
| 6.4.1. | Pruebas de comunicación con los asistentes de Voz | 54 |
| 6.4.2. | Programación dentro de una smart home..... | 56 |
| 7. | Resultados | 60 |
| 7.1. | Parámetros iniciales del sistema de comunicación. | 60 |
| 7.2. | Diseño de la estructura de protección externa del módulo..... | 61 |
| 7.3. | Programación del sistema de comunicación multiplataforma. | 64 |
| 7.4. | Evaluar el funcionamiento del sistema de comunicación multiplataforma. | 65 |
| 8. | Conclusiones | 67 |
| 9. | Recomendaciones | 68 |
| | Referencias | 72 |
| | ANEXOS | 73 |

Lista de Tablas

- 1. Microordenadores para selección 12
- 2. Características generales kit disipadores Raspberry pi 4..... 25
- 3. Características generales del ventilador para la Raspberry pi 4..... 27
- 4. Zonas a implementar el control dentro de un Smart Home 57
- 5. Matriz de decisión..... 61
- 6. Conversión de variables de entrada al miniservidor..... 65
- 7. Conversión de variables para cambio de estado Asistentes de Voz..... 65
- 8. Datos obtenidos al implementar el dispositivo..... 66
- 9. Matriz de consistencia..... 74

Lista de Figuras

| | | |
|-----|--|----|
| 1. | Esquema de conexión | 14 |
| 2. | Instalación sistema operativo | 16 |
| 3. | Configuración del sistema..... | 17 |
| 4. | Suspensión del servicio de bluetooth | 17 |
| 5. | Instalación de Build Essential..... | 18 |
| 6. | Instalación Node.js y Node-Red | 18 |
| 7. | Activación del servicio de Node-Red..... | 19 |
| 8. | Paquete de Homekit (Siri) Node-RED | 19 |
| 9. | Paquete de Alexa y Google Home Node-RED | 19 |
| 10. | Paquete de Loxone Node-RED | 20 |
| 11. | Diseño 1 | 21 |
| 12. | Diseño 2 | 22 |
| 13. | Diseño 3 | 23 |
| 14. | Representación térmica de la placa. | 24 |
| 15. | Kit disipadores Raspberry Pi 4B | 25 |
| 16. | Análisis térmico refrigeración pasiva. | 26 |
| 17. | Ventilador. | 27 |
| 18. | Simulación con ventilador. | 28 |
| 19. | Interfaz de programación Node-RED..... | 30 |
| 20. | Pestañas de flujo Node-RED | 31 |
| 21. | Subflujos Node-RED | 31 |
| 22. | Barra de Herramientas subflujos Node-RED | 32 |
| 23. | Entradas y salidas en subflujos Node-RED | 32 |
| 24. | Nodo estado para subflujos Node-RED | 32 |
| 25. | Nodo Inject Node-RED..... | 33 |
| 26. | Nodo Debug Node-RED | 33 |
| 27. | Nodo Function Node-RED..... | 34 |
| 28. | Nodo Change Node-RED | 34 |
| 29. | Nodo Switch Node-RED | 34 |
| 30. | Nodo Delay Node-RED..... | 35 |
| 31. | Nodo Join Node-RED | 35 |
| 32. | Comunicación nodo Alexa Smart Home | 36 |
| 33. | Nodo Alexa Smart Home | 37 |

| | | |
|-----|---|----|
| 34. | Nodo Alexa Smart Home State | 37 |
| 35. | Nodo Homekit Bridge..... | 38 |
| 36. | Nodo Homekit Service 2..... | 38 |
| 37. | Nodo Service..... | 39 |
| 38. | Nodo Status | 39 |
| 39. | Nodo Loxone Miniserver | 40 |
| 40. | Nodo Control in..... | 40 |
| 41. | Nodo Control out..... | 40 |
| 42. | Nodo Stream in..... | 41 |
| 43. | Estructura general de la programación | 42 |
| 44. | Grupo de Asistentes de Voz..... | 42 |
| 45. | Configuración nodo service | 43 |
| 46. | Configuración nodo Alexa Smart Home..... | 44 |
| 47. | Subflujo Estado/Datos HomeKit..... | 45 |
| 48. | Subflujo Procesamiento de Datos | 46 |
| 49. | Estructura subflujo Procesamiento de Datos | 46 |
| 50. | Programación nodo function Procesamiento de Datos | 47 |
| 51. | Configuración Control Out Loxone | 48 |
| 52. | Configuración Control In Loxone | 49 |
| 53. | Grupo de nodos para el cambio de estado | 49 |
| 54. | Programación nodo function cambio de estado | 50 |
| 55. | Nodos conectados al subflujo cambio de estados de asistentes | 51 |
| 56. | Interfaz de configuración Alexa y Google Home | 52 |
| 57. | Interfaz de configuración de variables para el control por voz para Alexa y Google Home..... | 53 |
| 58. | Configuración de comando de voz Siri | 54 |
| 59. | Pruebas de comunicación de los asistentes de voz..... | 55 |
| 60. | Variables de salida de los nodos | 56 |
| 61. | Variables de Alexa y Google dentro de una smart home..... | 57 |
| 62. | Programación zona cocina | 58 |
| 63. | Programación zona pasillo | 59 |
| 64. | Programación zona escalera..... | 59 |
| 65. | Estructura de protección implementada..... | 62 |
| 66. | Registro de temperatura actual Raspberry Pi | 63 |
| 67. | Registro de temperatura durante 24 horas Raspberry Pi | 64 |

| | | |
|-----|--|----|
| 68. | Tiempos registrados dentro de la programación. | 66 |
| 69. | Temperaturas de operación Raspberry Pi 4B, datasheet. | 75 |
| 70. | Dimensiones de la placa Rapsberry Pi 4B. | 76 |
| 71. | Planos Cubierta Superior. | 77 |
| 72. | Planos Cubierta Inferior. | 78 |

Resumen

El diseño e implementación de un sistema de comunicación multiplataforma de asistentes de voz con el protocolo Loxone dentro de un Smart Home tuvo como finalidad la incorporación de Siri, Alexa y Google permitiendo controlar los dispositivos vinculados con el miniservidor Loxone mediante comandos de voz.

La programación del sistema fue realizada en una Raspberry Pi 4B mediante el uso de Node-Red. Además, se diseñó una estructura de protección con ABS, con el fin de garantizar un producto seguro y resistente. Asimismo, se efectuaron pruebas para asegurar su funcionalidad óptima.

La implementación del módulo se realizó en un smart home, la cual ya disponía de tecnología Loxone en su infraestructura.

Con la implementación en un ambiente real, se comprobó la eficiencia del sistema de comunicación en cuanto al control por voz de los dispositivos y su rápido tiempo de respuesta. Además, se garantizó una correcta adaptación al carril TS35. La incorporación de un ventilador contribuyó a mantener una temperatura estándar, asegurando un funcionamiento óptimo del sistema.

Palabras clave: Asistentes de voz, Alexa, Siri, Google, Protocolo Loxone, Smart Home.

Abstract

The design and implementation of a multiplatform communication system of voice assistants with the Loxone protocol inside a Smart Home was aimed at incorporating Siri, Alexa and Google allowing you to control the devices linked with the Loxone miniserver by voice commands.

The programming of the system was done on a Raspberry Pi 4B using Node-Red. In addition, a protection structure was designed with ABS, to provide a safe and resistant product. Tests were also carried out to ensure optimal functionality.

The module was implemented in a smart home, which already had Loxone technology in its infrastructure.

The implementation in a real environment proved the efficiency of the communication system in terms of voice control of the devices and its fast response time. In addition, a correct adaptation to the TS35 rail was ensured. The addition of a fan helped to maintain a standard temperature, ensuring optimal operation of the system.

Keywords: Voice assistants, Alexa, Siri, Google, Loxone Protocol, Smart Home.

1. Introducción

Con el paso del tiempo, la manera en cómo las personas se relacionan con sus hogares se ha modernizado gracias a la tecnología. La aplicación de asistentes de voz como Alexa, Google Assistant y Siri dentro de la automatización del hogar o domótica, ha adquirido cierta popularidad al momento de gestionar la vida diaria, siendo un factor clave en el manejo del hogar inteligente permitiendo por medio de comandos de voz el control de luces, electrodomésticos y otros dispositivos que dispongan de una instalación domótica.

No obstante, varios sistemas de comunicación de hogares inteligentes que se encuentran disponibles dentro del mercado se limitan a un solo asistente de voz, lo que restringe su capacidad de desempeño y eficiencia. Además, el costo de los sistemas de comunicación comerciales que permiten la conexión entre múltiples asistentes de voz suele ser muy elevado provocando que no sean accesibles para todos los hogares.

El presente proyecto de titulación tiene como objetivo el diseño e implementación de un sistema de comunicación entre múltiples asistentes de voz a través del protocolo Loxone a un bajo costo, ofreciendo una mayor comodidad y sobre todo evitar la necesidad de depender de un solo sistema de comunicación. Por lo cual, se busca satisfacer una demanda real dentro del comercio de hogares inteligentes y lograr así una solución accesible y productiva. Por otro lado, se busca diseñar una estructura de protección externa con carril Din, con el fin de que pueda ser instalada dentro de una smart home aumentando la seguridad y estabilidad de sistema.

2. Problema

2.1. Antecedentes

Los asistentes de voz, dentro de los sistemas domóticos, han permitido mejorar el confort de las personas, mediante el accionamiento de voz para el control de dispositivos conectados dentro de la red domótica, como el accionamiento de un foco, música, información del día a día, sin embargo, los protocolos que se utilizan para desarrollar este tipo de sistemas no suelen tener la conexión directa con todos los asistentes de voz normalmente usados. Dentro del protocolo loxone se han desarrollado módulos externos que permiten la conexión con un

solo asistente de voz, como se menciona en (Velez Palomeque y Azuero Gonzales, 2022), en el cual se utiliza Alexa para el control de dispositivos, por medio de parlantes inteligentes colocados en un lugar específico.

2.2. Descripción del problema

Actualmente, dentro de la comunicación en las casas inteligentes, los asistentes virtuales como Google Home, Alexa, Cortana y Siri, son una tecnología que con el tiempo han ido creciendo con respecto a su uso para el accionamiento de voz de los dispositivos domóticos, sin embargo, el protocolo Loxone al ser el más utilizado en este tipo de sistemas, posee una conexión directa a HomeKit dentro del miniservidor, permitiendo que sea de fácil acceso al uso del asistente de voz Siri perteneciente a Apple; a pesar de ello, esta funcionalidad no se encuentra disponible para otros asistentes dentro del protocolo, siendo necesario el uso de módulos externos como los ofrecidos por 1Home, por lo cual se incorpora esta comunicación mediante un módulo conectado vía LAN al miniservidor, aunque el costo de adquisición es elevado, causando que sea poco accesible al medio actual.

2.3. Importancia y alcances

El proyecto de titulación “Diseño e implementación de un sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone” busca crear la interacción de varios asistentes de voz con el protocolo loxone, sin ser necesario un dispositivo específico o un solo asistente para acceder a la activación de voz, permitiendo la accesibilidad y portabilidad del accionamiento de los dispositivos dentro del hogar, generando confort, seguridad y ahorro energético. El alcance tiene como objetivo a las viviendas que dispongan de un sistema domótico con protocolo loxone y busquen optimizar el control del hogar por medio de comandos de voz, transmitidos por los diferentes asistentes que se encuentran en distintos dispositivos, como teléfonos celulares, parlantes inteligentes, entre otros.

2.4. Delimitación

El problema de estudio se delimitará en las siguientes dimensiones:

2.4.1. Espacial o geográfica

El proyecto se desarrollará para cualquier domicilio dentro de la ciudad de Cuenca que disponga de un sistema domótico con protocolo Loxone.

2.4.2. Temporal

El tiempo estimado para realizar por completo el proyecto de titulación tiene un total de 400 horas, a partir del inicio del periodo 62 hasta la finalización del mismo en la Universidad Politécnica Salesiana.

2.5. Problema General

- ¿Es posible diseñar e implementar sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone?

2.6. Problemas Específicos

- ¿Se podrá reconocer los parámetros iniciales para diseñar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone?
- ¿Es posible diseñar un sistema comunicación multiplataforma de los asistentes de voz con el protocolo loxone?
- ¿Se podrá implementar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone?
- ¿Es posible comprobar el funcionamiento del sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone?

3. Justificación

El uso de la domótica con los asistentes de voz ha ido evolucionando en los últimos años tomando protagonismo en la gestión del hogar. Dentro del protocolo Loxone se han desarrollado proyectos referentes a la conexión de un solo asistente de voz, como Alexa, realizada por Velez Palomeque y Azuero Gonzales (2022), teniendo limitantes físicas y de alcance, del mismo modo, existen dispositivos comerciales que permiten la conexión a múltiples asistentes de voz más conocidos con el protocolo Loxone; sin embargo, el costo de estos módulos es alto, generando gastos excesivos; por lo tanto, el presente proyecto busca desarrollar un sistema de comunicación que permita la conexión externa de varios asistentes de voz al protocolo Loxone a un bajo costo, obteniendo beneficios económicos, ya que el acceso a los servicios de comunicación no requiere de una gran inversión, permitiendo el acceso a distintos equipos

con diferentes asistentes de voz incorporados, brindando confort y eliminando la necesidad de depender de un solo dispositivo para generar una acción dentro de una smart home.

4. Objetivos

4.1. Objetivo General

- Diseñar e implementar un sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone para un smart home.

4.2. Objetivos Específicos

- Reconocer los parámetros iniciales para diseñar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone.
- Diseñar la estructura externa de protección con carril DIN para el sistema de comunicación multiplataforma de los asistentes de voz.
- Implementar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone.
- Comprobar el funcionamiento del sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone.

5. Marco Teórico

5.1. Domótica

“La domótica busca la integración de todos los dispositivos en el hogar, con el fin de que todo funcione en perfecta armonía, con la máxima utilidad y con la mínima intervención por parte del usuario” (Huidobro y Tejedor, 2010). Esto quiere decir que la domótica hace parte de la integración de múltiples tecnologías o conocimientos electrónicos, informáticos y sobre todo de telecomunicaciones aplicadas a una vivienda.

5.1.1. Ventajas de la domótica

Según Sarasúa (2011), existen algunas ventajas dentro de la domótica:

- **Climatización y consumo energético:** Permite programar el encendido y apagado de los artefactos tecnológicos con base a condiciones atmosféricas como las persianas, ventanas, etc. De igual manera, proporciona el autoapagado de la iluminación y los aparatos tecnológicos al momento de que la vivienda se encuentre abandonada, consiguiendo que la energía no sea desperdiciada por descuidos.
- **Seguridad:** Proporciona la seguridad contra robo evitando el ingreso de intrusos, ya que la domótica facilita a las instalaciones de alarmas de seguridad. De la misma forma existe la coordinación con varios sistemas como climatización, control de accesos, incendios, entre otros.
- **Confort:** Se da la gestión automatizada de la climatización, por medio de la incorporación del control de las ventanas, a su vez la vivienda podrá percibir la temperatura, humedad y la dirección del viento mediante una estación meteorológica, permitiendo que dentro de la vivienda se pueda tener un clima y ambiente óptimo. Mediante el sistema de domótica web se logra controlar la smart home desde cualquier parte del mundo. Las redes de multimedia logran que se pueda disfrutar de los equipos de audio y vídeo desde cualquier parte de la casa.
- **Mantenimiento:** Permite la automatización del riego por medio de la programación y lectura de la humedad, así como la automatización de encendido y apagado de los distintos aparatos tecnológicos, luces, aire acondicionado, etc.

5.2. Protocolo loxone

Loxone es una empresa australiana dedicada a la automatización de viviendas, mediante dispositivos de fácil instalación, configuración y bajos costos, lo cual les permitió escalar dentro del mercado domótico. El sistema domótico que utiliza loxone, se basa principalmente en un miniservidor por medio de una conexión cableada o inalámbrica en su versión GO, estos son los encargados de comunicar e integrar los distintos protocolos (Loxone Electronics, 2020).

Dentro del protocolo loxone, existen algunos tipos de extensiones que permiten la conexión con distintos dispositivos según Loxone Electronics (2020):

- Loxone Tree
- Loxone Air

- Loxone Link
- RS232
- LAN
- Modbus
- KNX

5.2.1. Loxone Tree

La implementación de la tecnología Loxone Tree tiene como objetivo reducir la cantidad de cableado necesario en la instalación Loxone. La tecnología Tree funciona como un canal de comunicación tipo “bus” que facilita la integración del sistema de manera sencilla y rápida (Loxone, 2020).

5.2.2. Loxone Air

Loxone Air utiliza la tecnología de malla, la cual asegura que todos los dispositivos Air se encuentren interconectados. De esta manera, cada dispositivo actúa como repetidor de señal, permitiendo una conexión estable y fiable, independientemente de su ubicación en la instalación (Loxone, 2023a).

5.2.3. Loxone Link

A través de la interfaz Loxone Link, es posible conectar extensiones desde cualquier Miniserver disponible (Loxone, 2023b).

5.2.4. LAN

“Una red de área local (LAN) conecta equipos ubicados cerca unos de otros” (Gaviria Gutierrez y Lara Urrego, 2020).

5.3. Asistentes de voz

Los asistentes de voz son herramientas digitales capaces de comprender el habla humana y responder por medio de voces generadas artificialmente. Los asistentes de voz más reconocidos

incluyen a Alexa de Amazon, Siri de Apple y Assistant de Google, los cuales se encuentran integrados teléfonos inteligentes o parlantes inteligentes diseñados para su uso en el hogar. Estos asistentes permiten a los usuarios vivir una experiencia interactiva mediante el uso de su voz, haciendo preguntas, controlando dispositivos en el hogar y accediendo a contenido multimedia, además de gestionar otras tareas básicas como crear listas de tareas, administrar fechas dentro del calendario y enviar correos electrónicos (Hoy, 2018).

5.3.1. Alexa

Alexa se encuentra disponible desde el 2014, es el asistente de voz creado por Amazon, se encuentra disponible para dispositivos Android 4.4, iOS 8.0, Fire OS 5.0 o versiones posteriores, en distintos idiomas (DE y ALEXA, 2020).

“En la actualidad Amazon proporciona una serie de dispositivos que ya incorporan el asistente Alexa o se integran de manera sencilla con Alexa como es el caso de: Echo, Echo Dot, Echo Plus, Echo Spot, Echo show, Echo sub” (DE y ALEXA, 2020).

5.3.2. Siri

“El asistente virtual Siri está integrado en los dispositivos móviles de comunicación producidos por la empresa Apple, la principal característica de este software es que comprende y responde al lenguaje natural humano” (Soto, s.f.).

5.3.3. Google Assistant

“El asistente virtual personal de Google es una tecnología desarrollada con inteligencia artificial, su principal característica es la conversación bidireccional entre los usuarios y Google, donde las preguntas y respuestas se pueden complementar sin restringirse a un solo tema” (García, 2020).

5.4. Gateway

Teniendo en consideración lo mencionado por Harke (2010), un Gateway, pasarela o puerta de enlace, es un dispositivo externo que permite la conversión de protocolos incompatibles,

entre dos o más redes conectadas, permitiendo el envío de datos a una red de destino.

5.5. Microordenadores

Los microordenadores son ordenadores los cuales pueden ser programados libremente, pueden asumir varios trabajos si se dispone los programas adecuados, todos los microordenadores tienen el mismo principio (Bastian, 2001). Algunos ejemplos de microordenadores son:

- **Raspberry Pi:** La Raspberry Pi es una computadora económica, del tamaño de una tarjeta de crédito, su sistema operativo se basa en Linux, por lo cual puede ser conectada a un monitor de computador o a una TV, para luego ser usado por medio de un mouse y teclado, siendo factible para aprender lenguajes de programación como Python. Puede realizar las tareas básicas de un computador de escritorio, desde navegar en internet hasta reproducir juegos (Raspberry, 2022).
- **BeagleBone:** Se basa en el sistema operativo Linux de código abierto, por lo tanto, existe una gran cantidad de potencia informática en donde los desarrolladores tienen acceso a análisis rápidos, con más opciones de almacenamiento de datos, de varias interfases de alta velocidad, se puede crear aplicaciones como robótica autónoma, análisis de visión y vídeo, drones, entre otros (Beagleboard.org, 2022).

5.6. Node Red

Node-RED es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de formas nuevas e interesantes. Proporciona un editor basado en navegador que facilita la conexión de flujos mediante la amplia gama de nodos de la paleta que se pueden implementar en su tiempo de ejecución con un solo clic. (*Node-RED*, s.f.)

5.7. Software de diseño mecánico

Los ingenieros y los diseñadores utilizan el software de diseño e ingeniería mecánica para modelar, validar y comunicar ideas antes de la producción. Hay herramientas adicionales que, en ocasiones, se integran en el software de CAD para productos

de fabricación de una máquina CNC o una impresora 3D. El software de ingeniería mecánica se utiliza en diversos sectores, incluidos el diseño de maquinaria industrial, la automoción y los productos de consumo. (AutoDesk, 2021)

Algunos de los software de diseño mecánico utilizados son:

- Autodesk Inventor
- Fusion 360
- Solidworks

5.7.1. Autodesk Inventor

“Autodesk Inventor es un software para la creación de piezas y conjuntos mecánicos, entre otros, así como la realización de planos de despiece y de conjunto y la simulación de movimiento y montaje y desmontaje de los mismos” (Blanes, 2011). Además, el uso de este software se basa en suscripciones de pago de manera mensual, anual o trimestral.

5.7.2. Fusion 360

Fusion 360 es un software de Autodesk, que posee diferentes tipos de licencia de acuerdo con el uso, es decir, posee licencias de pago a nivel comercial y licencias gratuitas para estudiantes, o proyectos nivel personal con ciertas limitaciones. Según lo descrito por Verma (2018), este software permite realizar modelado en 3D, que implementa paquetes CAM, CAE y CAD en un solo software, lo cual facilita el diseño de elementos, análisis por simulación, entre otros. Además, permite la conexión con múltiples colaboradores dentro de un solo proyecto, con el fin de realizar seguimiento y aportar en el desarrollo del producto final.

5.7.3. SolidWorks

Es un programa que permite realizar el proceso completo de diseño mecánico, desde la idea hasta la realización de planos técnicos necesarios para su fabricación. Mediante herramientas de diseño de pieza, ensamblajes y dibujo. Junto con las herramientas de diseño, SolidWorks incluye herramientas de productividad, gestión de proyectos, presentación y de análisis y simulación. (Vidal, 2015)

Al igual que Autodesk Inventor. SolidWorks requiere la adquisición del software por medio de una compra, a su vez dispone de versiones de estudiantes, siendo limitadas al momento de realizar diseños que requieren muchas más herramientas.

6. Marco metodológico

6.1. Parámetros iniciales del sistema de comunicación.

6.1.1. Parámetros de conexión con el protocolo loxone

Debido a que el protocolo loxone está destinado a las smart home y se busca realizar una comunicación con servicios externos, se debe considerar los tipos de conexiones a internet disponibles dentro de estos ambientes, de manera común existen dos tipos:

- **Conexión por cable:** Según Kurose y Ross (2010) dentro de la conexión realizada por cable usualmente se conoce al protocolo LAN o red de área local por medio de su estándar Ethernet, el cual mediante un cableado denominado RJ45 o par trenzado se comunica con el router, teniendo una velocidad óptima sin tanta mucha de paquetes de información al momento de realizar la comunicación.
- **Conexión inalámbrica:** En el tipo de conexión inalámbrica generalmente se conoce al denominado Wifi; sin embargo, el wifi es un tipo WLAN (Wireless Local Area Network) siendo la más comúnmente usada y se basa en el protocolo LAN. Este tipo conexión, como se menciona por parte de Ramírez Sánchez y Díaz Martínez (2012) elimina la necesidad de usar cables, lo que permite que la red llegue a puntos de difícil acceso a comparación de la red cableado, además cabe mencionar que su velocidad de transferencia de datos se puede ver afectada por interferencias.

Tomando en consideración los puertos que dispone el miniservidor y que el protocolo Loxone se comunica a través de TCP/IP, la configuración “el Miniserver se debe conectar a la red informática local de la instalación mediante su conector LAN, usando una toma RJ45, hacia un switch o un router de la misma instalación” (Loxone, 2023c). Se determinó que para comunicar el dispositivo que contiene el sistema de comunicación se necesita una conexión inalámbrica tipo WLAN (Wifi), permitiendo una instalación mucho más limpia y de fácil acceso a la dirección IP asignada por parte del router al miniordenador; sin embargo, se puede

variar entre una conexión por cable.

6.1.2. Parámetros electrónicos

Para el módulo domótico, es fundamental la elección del microordenador adecuado, por lo que resulta importante considerar diversos parámetros electrónicos con el fin de cumplir los requisitos del proyecto y brinde un rendimiento óptimo. Estos factores incluyen la velocidad y tipo de procesador, la cantidad de memoria RAM, la cantidad y tipos de puertos de entrada/salidas, la disponibilidad de interfaces de red, el rango de voltaje de alimentación y la corriente que requiere, el tamaño, peso y costo.

6.1.3. Parámetros mecánicos

La protección de los módulos en una smart home resulta esencial para prolongar la vida útil de los componentes electrónicos. Se debe tener en cuenta varios factores que pueden afectar la durabilidad de estas protecciones, como el desgaste normal, la humedad y los cambios de temperatura.

6.1.4. Selección de microordenador

Una vez evaluado los parámetros electrónicos y de comunicación con el protocolo loxone, se proponen los siguientes microordenadores:

Tabla 1*Opciones de microcontroladores.*

| Marca | Modelo | Características | Precio |
|--------------|------------------|--|--------|
| Radxa | ROCK 4 SE | Ram= 2 GB LPDDR4 Procesador = Rockchip RK3399-T hexagonal Wifi= WiFi5 | 130\$ |
| Raspberry Pi | Raspberry Pi 4 B | Ram= 2/4 GB Procesador = Cortex-A72 @ 1.5 GHz Wifi=2.4 GHz y 5.0 GHz IEEE 802 | 170\$ |
| Khadas | VIM3 Basic | Ram= 2 GB DDR4 Procesador = Cortex-A73 @ 2.2Ghz Wifi=AP6398S Module 802.11 | 129\$ |
| Orange Pi | Orange Pi 3 LTS | Ram= 2 GB DDR4 Procesador = Cortex-A53 @ 1.8GHz Wifi=AW859A Dual Band WiFi | 60\$ |
| Asus | Tinker Board 2S | Ram= Dual-CH LPDDR4 2GB Procesador = Dual-core Cortex-A72 2GHz Quad-core Arm Cortex-A53 1,5GHz Memoria: 16GB eMMC Wifi= 802.11 | 150\$ |

Nota: Opciones de microordenador recomendadas calidad-precio

Para mejorar la capacidad de procesamiento, la opción más viable es la Raspberry Pi 4 B de 4 GB. Esta placa es la más accesible en cuanto a la comunidad y soporte ante cualquier tipo de falla, además cumple con todos los requerimientos necesarios como intermediario entre los asistentes de voz y el protocolo domótico loxone.

6.1.5. Ambiente de desarrollo

Considerando que el microordenador seleccionado es la Raspberry Pi 4B, se realizó una investigación de los sistemas operativos con los cuales el miniordenador es compatible y se determinaron algunos sistemas operativos que se pueden instalar en Raspberry:

- **Raspberry Pi OS:** “Raspberry Pi OS is a free operating system based on Debian, optimised for the Raspberry Pi hardware, and is the recommended operating system for normal use on a Raspberry Pi” [Raspberry Pi OS es un sistema operativo gratis basado en Debian, optimizado para el hardware de Raspberry Pi y es un sistema operativo recomendado para un uso normal en una Raspberry Pi] (Pi, 2015).
- **DietPi:** “DietPi is extremely lightweight at its core, with features of low process/memory footprint and DietPi-RAMlog installed by default, to get the maximum performance from your device” [DietPi es un sistema extremadamente liviano, con características de bajo consumo de memoria y procesamiento, obteniendo el máximo rendimiento del dispositivo] (DietPi, 2018).

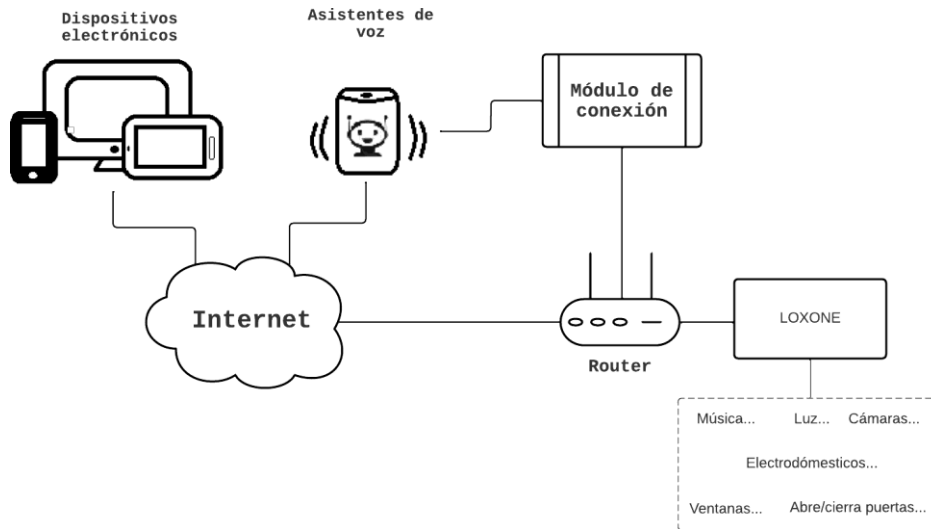
De manera general, Raspberry OS al ser un sistema operativo oficial se dispone de una gran cantidad de información con respecto a sus comandos, además por medio del foro permite solucionar ciertos errores al momento de configurar el dispositivo.

6.2. Diseño de la estructura externa del módulo.

6.2.1. Esquema de conexión

Figura 1

Esquema de conexión



Nota: Se representa la conexión entre el módulo y el protocolo.

La Figura 1 muestra el proceso completo que se lleva a cabo para obtener la respuesta de los asistentes virtuales. A través de la utilización de diversos protocolos de comunicación, estos asistentes virtuales se conectan con servidores en internet, y la información se transmite a los usuarios a través de dispositivos intermediarios, como los routers. Estos dispositivos asignan direcciones IP únicas a cada dispositivo conectado a la red, lo que permite la correcta distribución de la información. El dispositivo en desarrollo tiene la capacidad de interpretar los protocolos recibidos de los asistentes virtuales y comunicarse directamente con el protocolo domótico Loxone, lo que permite una conexión más fluida y eficiente entre los diferentes dispositivos y sistemas.

6.2.2. Configuración del ambiente de desarrollo

Teniendo en consideración los instructivos proporcionados por el fabricante, se debe instalar, el sistema operativo por medio de un SD Card. Esto ocurre debido a que la placa Raspberry Pi 4 B no posee de manera integrada una memoria eMMC.

Para realizar la instalación del sistema Raspberry OS se requiere cargar el sistema operativo a la tarjeta SD, para ello, es necesario un programa que permita ejecutar esta opción. A continuación, se presenta tres opciones de uso libre para llevar a cabo este proceso:

- **Rufus:** “Rufus es una utilidad que le ayuda a formatear y crear soportes USB de arranque, como pendrives, tarjetas de memoria, etcétera” (Batard, s.f.).
- **BalenaEtcher:** “BalenaEtcher es un software de código abierto para flashear imágenes de sistema en unidades externas como USB o tarjetas SD” (Etcher, 2021).
- **Raspberry Pi Imager:** “Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi” [Raspberry Pi Imager es una manera rápida de instalar Raspberry Pi Os y otros sistemas operativos a una tarjeta SD para usar en una Raspberry Pi] (Ltd, 2015).

El software recomendado es Raspberry Pi Imager, debido a la facilidad de uso para la instalación del sistema operativo.

Figura 2

Instalación sistema operativo



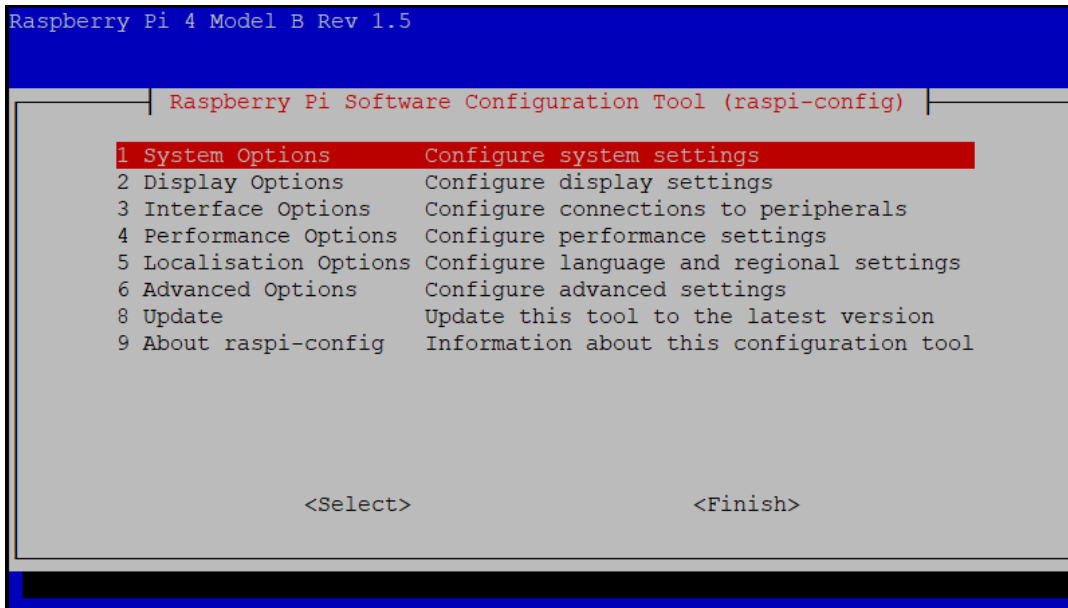
Nota: Uso del software para flashear e instalar el sistema operativo.

En la figura 2, se puede observar que la interfaz del software para realizar la instalación del sistema operativo. En esta placa se utiliza la versión de 64 bits.

Una vez instalado el sistema operativo, se debe realizar algunas configuraciones dentro del sistema, como, las credenciales de acceso que permiten el control del dispositivo y su identificación dentro de la red. Por medio del terminal que proporciona el sistema, se accede a las configuraciones del dispositivo, en donde se presenta una serie de opciones para realizar modificaciones.

Figura 3

Configuración del sistema



Nota: Interfaz gráfica de configuración del sistema del microcontrolador.

Considerando la interfaz mostrada en la figura 3, se configura algunos apartados que vienen por defecto al inicializar por primera vez el microordenador, alguna de estas configuraciones son el hostname para la identificación del dispositivo dentro de la red por lo que, se modifica a "tdomotic". Se realiza el mismo procedimiento para el Password, con el fin de evitar posibles accesos al dispositivo por parte de terceras personas. Asimismo, se ajusta la región y la zona horaria en donde se desarrolla el proyecto.

Figura 4

Suspensión del servicio de bluetooth

```
root@tdomotic:/# sudo systemctl disable bluetooth.service
```

Nota: Comando utilizado para desactivar el servicio de bluetooth.

El servicio de bluetooth se encuentra activado automáticamente cada vez que se inicia el dispositivo según la configuración predeterminada por parte del fabricante. No obstante, para

el desarrollo del proyecto, el uso de bluetooth no es necesario. Por lo tanto, se usa el comando mostrado en la figura 4, para deshabilitar el servicio y liberar la capacidad de procesamiento que esta asignada para el bluetooth.

Figura 5

Instalación de Build Essential

```
root@tdomotic:/# sudo apt install build-essential git curl
```

Nota: Comando usado para descargar Build-Essential. (Node-RED, 2013).

Para el proceso de instalación de Node-red dentro de la placa Raspberry pi 4B, se siguen una serie de pasos proporcionados en la página web de Node-RED. El primer paso mencionado en la plataforma es la instalación de Build Essential, siendo una librería necesaria para la compilación de programas, para ello se utiliza el código que se muestra en la figura 5.

Figura 6

Instalación Node.js y Node-Red

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Nota: Comando usado para instalar Node.js y Node-Red. (Node-RED, 2013).

Una instalación adicional es la de Node.js, debido a que es fundamental para el uso de Node-Red. Para ello, se utiliza el siguiente comando, que se muestra en la figura 6. Este comando descarga un script que detecta las versiones instaladas de Node.js y npm, o realiza la instalación por defecto si no están presentes. Además, el script descarga el paquete de Node-Red y permite agregarlo como servicio de inicio del sistema.

Figura 7

Activación del servicio de Node-Red

```
root@domotic:/# sudo systemctl enable nodered.service
```

Nota: Comando usado para activar el servicio de Node-Red. (Node-RED, 2013).

Para activar el servicio de Node-Red, es necesario utilizar el código que se muestra en la figura 7. Este código permite habilitar el servicio, el cual por defecto se encuentra desactivado al finalizar la instalación.

Una vez que el entorno de desarrollo para la programación ha sido instalado, se deben instalar los paquetes necesarios para comenzar con la programación del sistema de comunicación. Para ello, se consideran oportunos los siguientes paquetes en Node-Red:

Figura 8

Paquete de Homekit en Node-RED



node-red-contrib-homekit-bridged
1.5.0

Nota: Paquete de instalación de Homekit in Node-RED. (Node-RED, s.f.).

- **NRCHKB:** Teniendo en cuenta lo que menciona NRCHKB (2021), este nodo fue desarrollado para la simulación de accesorios de Homekit en el ámbito de Node-RED.

Figura 9

Paquete de Alexa y Google Home en Node-RED



node-red-contrib-alexa-smart-home
0.4.68

Nota: Paquete de instalación de Alexa y Google Home in Node-RED. (Node-RED, s.f.).

- **Node-RED Smart Home Control:** "Node-RED Smart Home Control permite llevar rápidamente el control por voz a los flujos de Node-RED, utilizando Amazon Alexa y/o Google Assistant". (Control, s.f.)

Figura 10

Paquete de Loxone en Node-RED



Nota: Paquete de Loxone Node-RED. (Node-RED, s.f.).

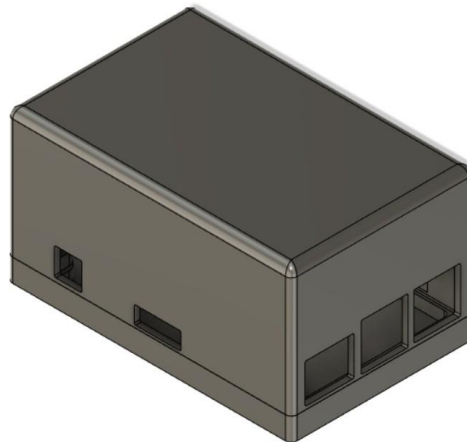
- **Nodo Loxone:** Considerando lo mencionado por Utecht (2017), este nodo permite la conexión entre el Miniservidor Loxone y Node-RED por medio del uso del Websocket.

6.2.3. Diseño de la estructura de protección externa

Con respecto al diseño de la estructura de protección externa, se optó por utilizar la herramienta Fusion 360, el cual permite utilizar el modelado 3D y diseño paramétrico para crear una protección adaptada a los requisitos y dimensiones de la placa. Se consideraron 3 diseños con el propósito de seleccionar el diseño más adecuado utilizando una matriz de decisión como referencia.

Figura 11

Diseño 1

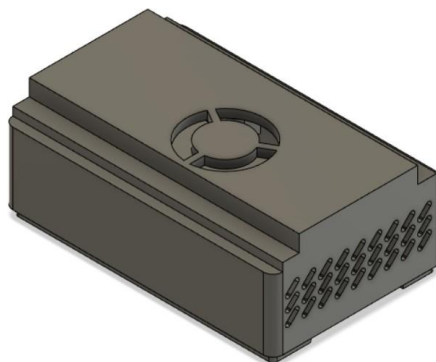


Nota: Protección del sistema en forma rectangular.

El diseño inicial presenta una forma rectangular con bordes redondeados, dividido en dos partes, la base y la tapa. Ambas partes se ensamblan junto a la placa Raspberry mediante 4 tornillos M2 de acuerdo a los insertos existentes en el mercado, debido a que el proceso de manufactura es impresión 3D.

Figura 12

Diseño 2

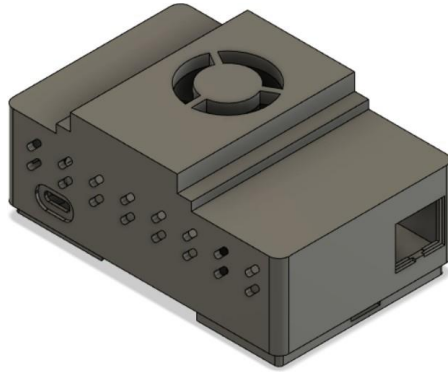


Nota: Protección del sistema horizontal.

En el segundo diseño al igual que el diseño anterior se ha optado por una forma rectangular, compuesta de dos partes: la base y la tapa. Estas dos partes se ensamblan mediante 4 tornillos M2. Sin embargo, se ha tenido en cuenta la necesidad de proporcionar ventilación a la placa, como resultado se tiene en cuenta la entrada y salida de aire en el sistema de protección. Además, se ha diseñado con un sobresaliente a lo largo de la tapa con el objetivo de instalar un ventilador. Asimismo, se ha diseñado con el fin de proporcionar la compatibilidad del sistema con el carril DIN, permitiendo una ubicación horizontal del sistema de protección.

Figura 13

Diseño 3



Nota: Protección del sistema vertical.

En caso del diseño 3 presenta similitudes con el diseño 2 con diferencia en el sobresaliente destinado a la instalación del ventilador que, en este caso, se encuentra a lo ancho de la tapa. Además, se diseñó el acople para el carril DIN de manera que el sistema de protección se ubique de manera vertical.

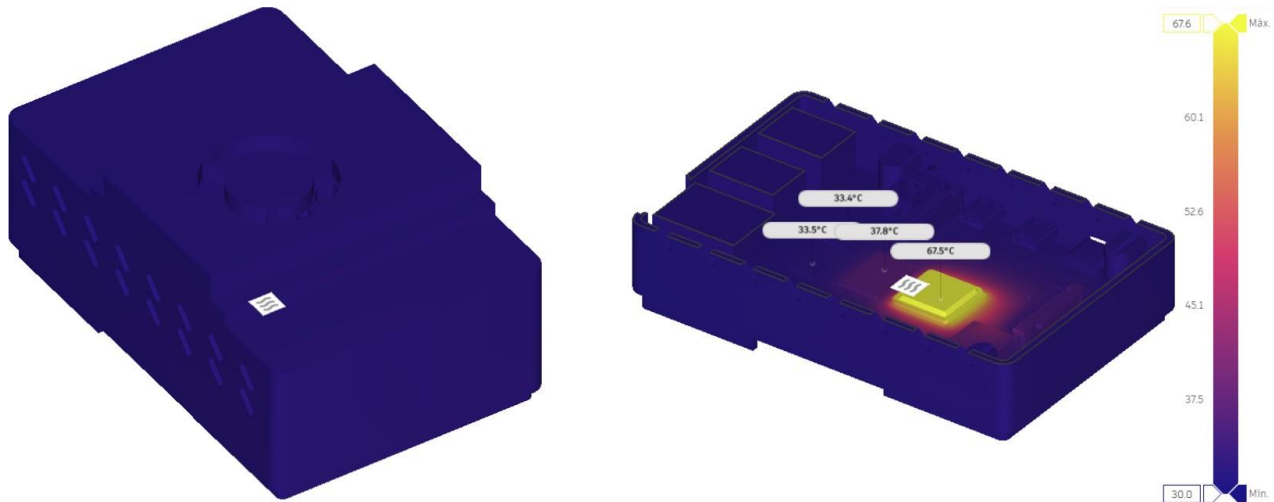
El diseño seleccionado es el diseño 2, el cual fue elegido después de realizar un análisis detallado utilizando una matriz de decisión adjunta en la sección de resultados.

6.2.4. Análisis de temperatura

Considerando las especificaciones de operación de la placa Raspberry Pi 4B, se indica que está diseñada para funcionar dentro de un rango de temperatura de 0°C a 50°C. No obstante, en la práctica, se ha observado que el sensor incorporado en el CPU registra una temperatura cercana a los 60°C.

Figura 14

Representación térmica de la placa.



a) Temperatura Estructura de protección

b) Temperatura Raspberry Pi 4B

Nota: Representación térmica de la placa Raspberry Pi 4B.

- Al examinar la figura 14a, se observa que la estructura de protección no experimenta una variación significativa en relación con la temperatura ambiente. Este resultado se debe a que el diseño fue especialmente creado para optimizar el flujo de aire de entrada y salida, asegurando así una temperatura estable en su entorno.
- En la figura 14b, se representa la temperatura aproximada de los componentes de la carcasa de la placa Raspberry Pi 4B. Se observa un aumento de temperatura en la CPU Broadcom BCM2711 con una aproximación de 68°C dentro de la simulación. Para este estudio se consideran valores aproximados de calor interno entre 2 a 5W para la CPU, así como, una temperatura ambiente de 30°C.

Figura 15

Kit disipadores Raspberry Pi 4B



Nota: Disipadores para la placa Raspberry Pi 4B.

Considerando que la temperatura de operación de la placa es de hasta 50°C, se propone el uso de refrigeración pasiva, es decir, el uso de disipadores de calor para la disminución de la temperatura del dispositivo. Para la selección de los disipadores se consideran los recomendados dentro del kit suministrado junto a la Raspberry Pi 4B, a su vez, de la disponibilidad de manera local.

Tabla 2

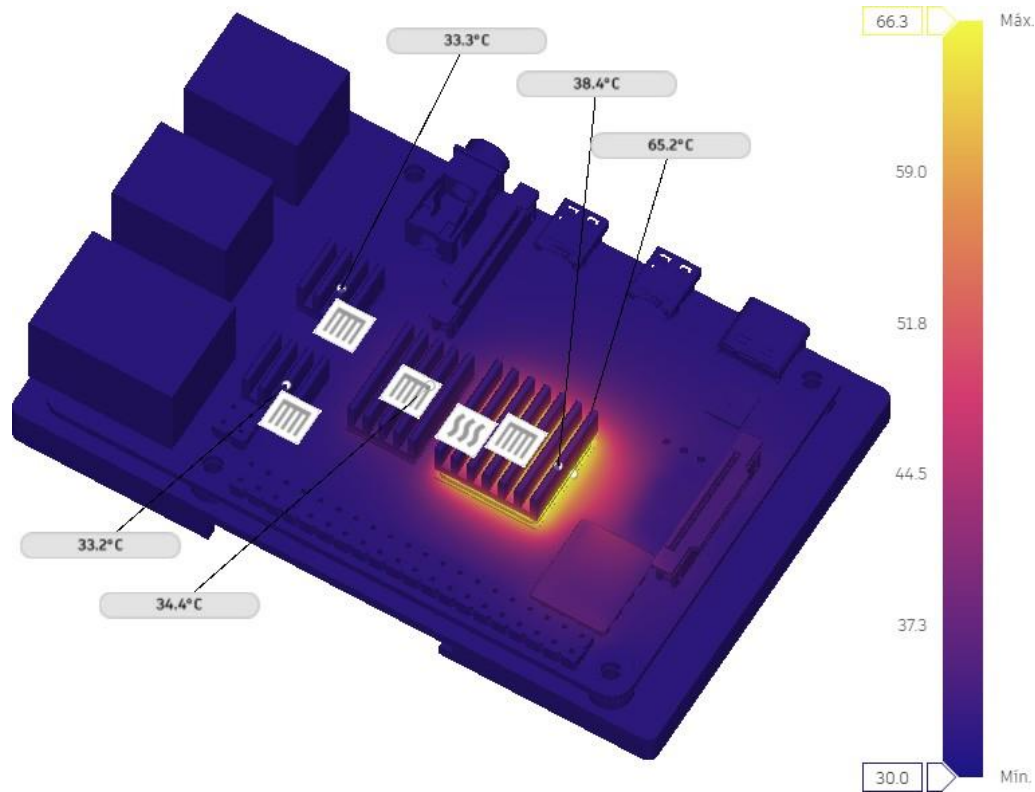
Características generales kit disipadores Raspberry pi 4.

| Características de los disipadores | |
|------------------------------------|------------------------------------|
| Dimensiones CPU | 14mm X 14mm X 6mm |
| Dimensiones RAM | 14mm X 10mm X 6mm |
| Dimensiones LAN | 9mm X 9mm X 6mm |
| Dimensiones GPU | 7mm X 7mm X 6mm |
| Material | Aluminio |
| Contacto | Cinta adhesiva conductora de calor |

Nota: Características de los disipadores

Figura 16

Análisis térmico refrigeración pasiva.



Nota: Análisis térmico para la placa Raspberry Pi 4B.

Al tener las características del disipador se lleva a cabo un nuevo análisis referente a la refrigeración pasiva en cada uno de los cuatro chips. Como se observa en la figura 16, la temperatura de los componentes no presenta un descenso significativo, mostrando una disminución de 3°C a 4°C.

Estos resultados recomiendan implementar un distinto tipo de refrigeración al dispositivo, ya que, al tener solo disipación pasiva, no permite que la placa se encuentre dentro del rango de operación óptimo.

Para la implementación de la refrigeración activa para la Raspberry Pi, se recomienda el uso de un ventilador de acuerdo con la página oficial y foros de Raspberry. Teniendo en cuenta que la placa opera con 5V, se debe seleccionar un ventilador proporcionado por Raspberry

que cumpla con ese rango de voltaje y con un flujo de aire de 1.4 CFM (pies cúbicos por minuto).

Figura 17

Ventilador.



Nota: Ventilador disponible de la placa Raspberry Pi 4B.

Teniendo en cuenta la recomendación de Raspberry, se presenta el ventilador seleccionado en la figura 17 disponible de manera local. A continuación, en la tabla 3 se presenta las características del ventilador.

Tabla 3

Características generales del ventilador para la Raspberry pi 4.

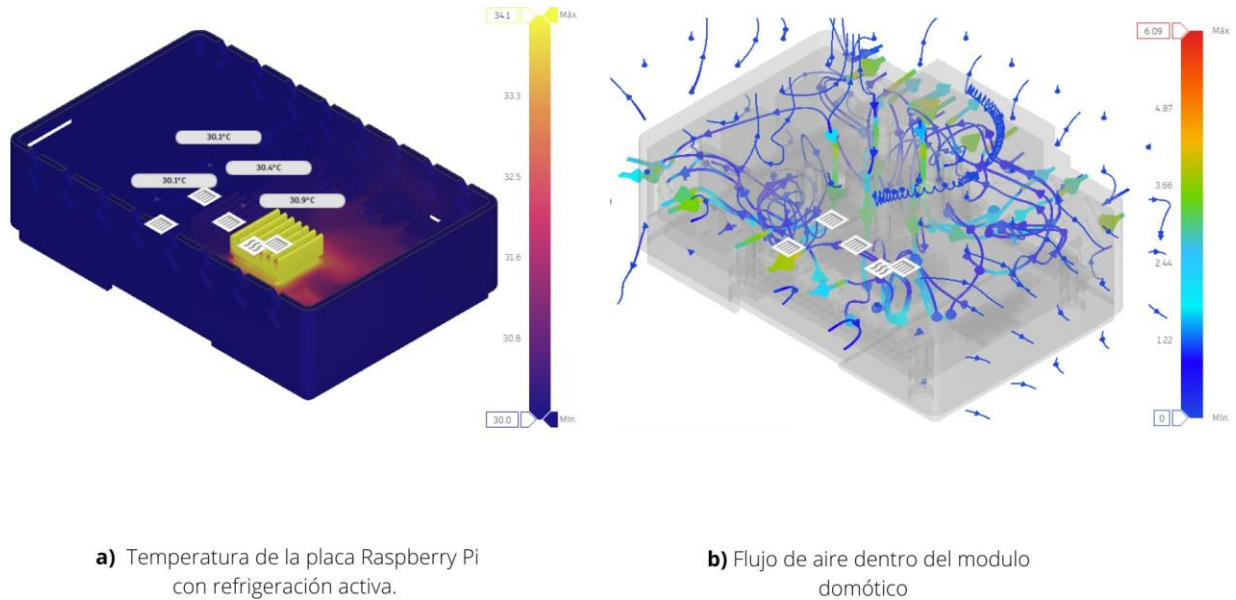
| Características del ventilador | |
|--------------------------------|-------------------|
| Volumen de aire (CFM) | 3.1 CFM |
| Material | ABS |
| Voltaje | 5 V |
| Dimensiones | 30mm x 30mm x10mm |

Nota: Características del ventilador.

Utilizando la información proporcionada sobre los dispositivos y características mencionadas, se procede a realizar un nuevo análisis.

Figura 18

Simulación con ventilador.



Nota: Simulación adicionado con el ventilador.

Tras completar la simulación, se derivan los siguientes puntos clave para el análisis.

- En la figura 18a, se muestra la temperatura de los componentes al aplicar una convección forzada o refrigeración activa, evidenciando una temperatura máxima de 34°C, siendo una temperatura apta para el funcionamiento óptimo de la placa.
- En la figura 18b, se observa el flujo de aire generado por el ventilador hacia los disipadores del CPU y RAM, presentando una velocidad media de $2 \frac{ft}{s}$. En las rejillas de ventilación la velocidad máxima de flujo de aire puede alcanzar hasta $4.87 \frac{ft}{s}$.

6.2.5. Selección del material

Para llevar la selección del material adecuado, se toma como base el diseño previamente seleccionado y los parámetros mecánicos mencionados en el primer objetivo. Después de evaluar estas consideraciones, se ha optado por utilizar ABS. A continuación, se presentan las características del material ABS, según lo indicado por *Colorplus3D* (2023):

- **Características físicas:**

- Es un material rígido.
- Material tenaz y resistentes a los impactos

■ **Propiedades mecánicas:**

- La tensión a la rotura es de 40 MPa.
- Alargamiento hasta la rotura del 30%.
- Su módulo elástico es de 2443 MPa.
- Su resistencia a la flexión es de 68 MPa.
- La resistencia de impacto es de 45 kJ/m².
- La densidad es de 1.06 g/cm³.

■ **Propiedades térmicas:**

- La temperatura de extrusor es de 230 a 260 °C.
- La temperatura máxima es de 100°C.
- La velocidad de impresión es de 20-50 mm/s.

Tras evaluar las características del material ABS, se puede afirmar que cumple con los parámetros mecánicos necesarios y presenta una excelente resistencia a la temperatura de trabajo de funcionamiento óptima del módulo, sin presentar inconvenientes.

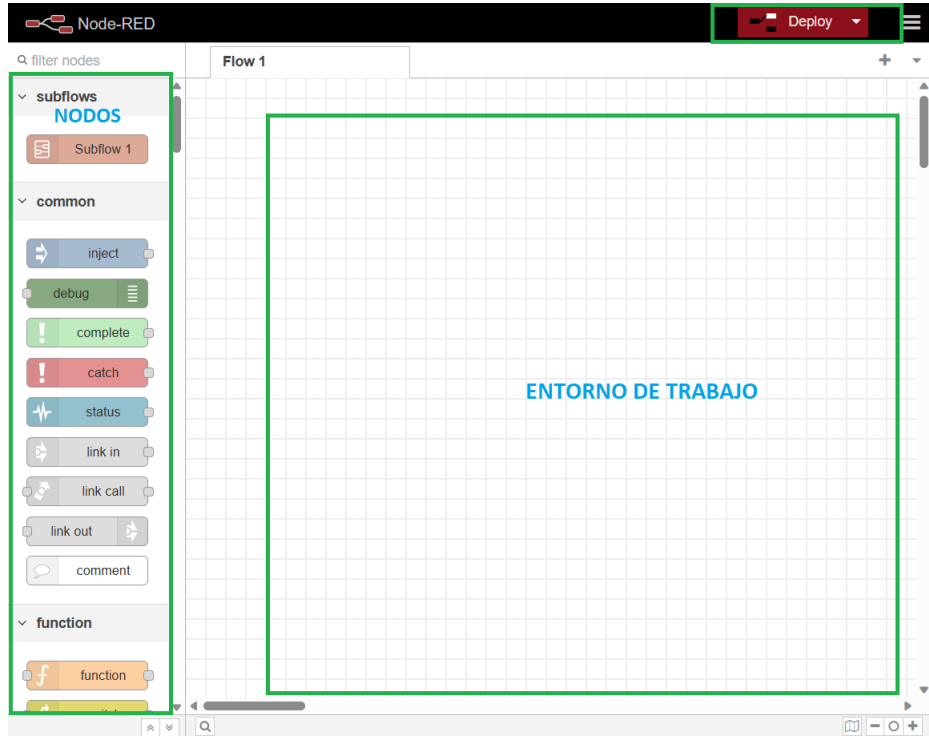
6.3. Programación del sistema de comunicación multiplataforma.

6.3.1. Reconocimiento de la interfaz y nodos a utilizar

Una vez preparado el ambiente de desarrollo, que permite la programación a través de un host local en el navegador, es importante considerar la interfaz de trabajo y las opciones utilizadas para ordenar correctamente los grupos de nodos con el fin de no generar posibles confusiones.

Figura 19

Interfaz de programación Node-RED



Nota: Interfaz de programación en Node-RED.

La figura 11 describe las zonas más importantes a usar en el transcurso de la programación, consistiendo en dos:

- **Nodos:** Son bloques funcionales que permiten ejecutar acciones específicas según el nodo seleccionado.
- **Entorno de trabajo:** Es el lugar en donde se desarrolla la programación por medio de los flujos constituidos por nodos.

Figura 20

Pestañas de flujo Node-RED

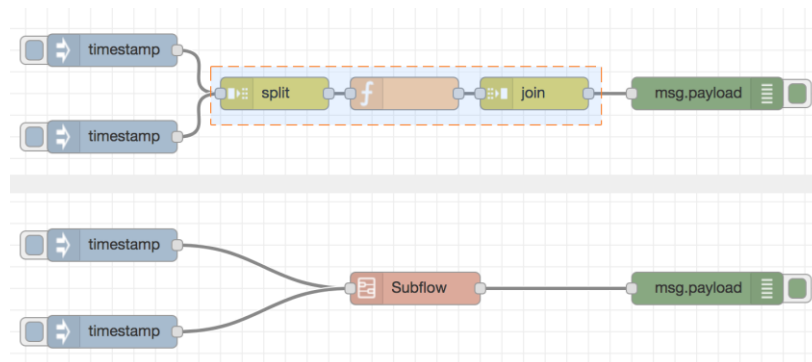


Nota: Pestañas de flujo Node-RED. (Node-RED, s.f.-b)

En la figura 12, se observa una pestaña de flujos, según Node-RED (s.f.-b) lo describe de la siguiente manera, “un flujo se representa como una pestaña dentro del espacio de trabajo del editor y es la forma principal para organizar los nodos.”

Figura 21

Subflujos Node-RED



Nota: Subflujos Node-RED. (Node-RED, s.f.)

Una de las opciones importantes a la hora de desarrollar la programación por medio de flujo es la organización de los flujos, de manera que en la figura 13 se presentan los subflows o subflujos, los cuales según Node-RED (s.f.) dice, “un subflujo es una colección de nodos que se contraen en un solo nodo del área de trabajo. Se pueden utilizar para reducir la complejidad visual de un flujo”, además de que permite reutilizar el subflujo en cualquier pestaña, debido a que se encontrara disponible como un nodo más dentro de la paleta de nodos.

Figura 22

Barra de Herramientas subflujos Node-RED



Nota: Herramientas dentro de los Subflujos Node-RED. (Node-RED, s.f.)

Asimismo, es importante conocer la barra de herramientas como lo muestra la figura 14, la cual que proporciona nodos dentro de los subflujos, siendo los siguientes:

Figura 23

Entradas y salidas en subflujos Node-RED



Nota: Nodos de entrada y salida de datos para los Subflujos Node-RED. (Node-RED, s.f.)

- **Nodo de entrada y salida:** Considerando lo mencionado por Node-RED (s.f.), el nodo gris denominado “input” representa la entrada y el nodo con etiqueta “output” la salida del subflujo. Se considera que el subflujo puede tener únicamente una entrada de datos; sin embargo, en el caso de la salida, pueden existir tantos nodos de salida como se requiera.

Figura 24

Nodo estado para subflujos Node-RED



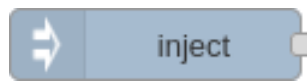
Nota: Nodo estado para Subflujos Node-RED. (Node-RED, s.f.)

- **Nodo estado:** Teniendo en cuenta a Node-RED (s.f.), el nodo estado, permite conocer cuál es la condición de la salida de uno de los nodos que conforman el subflujo.

Además del área de trabajo y las opciones disponibles en Node-RED para programar, es importante considerar el uso de los nodos nativos de Node-RED en el desarrollo de la programación. Se presenta la funcionalidad de cada uno de los nodos que se pueden utilizar:

Figura 25

Nodo inject Node-RED

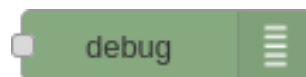


Nota: Nodo inject Node-RED. (Node-RED, s.f.-a).

- **Nodo Inject:** “The Inject node can be used to manual trigger a flow by clicking the node’s button within the editor. It can also be used to automatically trigger flows at regular intervals” [El nodo Inyect se puede utilizar para activar manualmente un flujo haciendo clic en el botón del nodo dentro del editor. También se puede utilizar para activar automáticamente los flujos a intervalos regulares] (Node-RED, s.f.-a).

Figura 26

Nodo Debug Node-RED



Nota: Nodo debug en Node-RED. (Node-RED, s.f.-a).

- **Nodo Inject:** “The Debug node can be used to display messages in the Debug sidebar within the editor” [El nodo Debug se puede utilizar para mostrar mensajes en la barra lateral Debug dentro del editor] (Node-RED, s.f.-a).

Figura 27

Nodo Function Node-RED

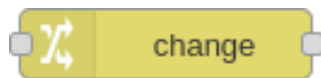


Nota: Nodo function en Node-RED. (Node-RED, s.f.-a).

- **Nodo Function:** “The Function node allows JavaScript code to be run against the messages that are passed through it” [El nodo Función permite ejecutar código JavaScript en los mensajes que pasan a través.] (Node-RED, s.f.-a).

Figura 28

Nodo Change Node-RED

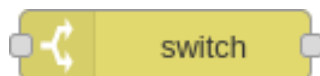


Nota: Nodo change en Node-RED. (Node-RED, s.f.-a).

- **Nodo Change:** “The Change node can be used to modify a message’s properties and set context properties without having to resort to a Function node.” [El nodo Change se puede utilizar para modificar las propiedades de un mensaje y establecer propiedades de contexto sin tener que recurrir a un nodo Function] (Node-RED, s.f.-a).

Figura 29

Nodo Switch Node-RED



Nota: Nodo switch en Node-RED. (Node-RED, s.f.-a).

- **Nodo Switch:** “The Switch node allows messages to be routed to different branches of a flow by evaluating a set of rules against each message” [El nodo Switch permite que los mensajes se direccionen a diferentes ramas de un flujo mediante la evaluación de un conjunto de reglas para cada mensaje] (Node-RED, s.f.-a).

Figura 30

Nodo Delay Node-RED

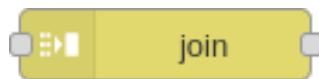


Nota: Nodo delay en Node-RED. (Node-RED, s.f.-a).

- **Nodo Delay:** “Delays each message passing through the node or limits the rate at which they can pass” [Retrasa cada mensaje que pasa a través del nodo o limita la velocidad a la que pueden pasar] (Node-RED, s.f.-a).

Figura 31

Nodo Join Node-RED



Nota: Nodo join en Node-RED. (Node-RED, s.f.-a).

- **Nodo Join:** “Joins sequences of messages into a single message” [Une secuencias de mensajes en un solo mensaje] (Node-RED, s.f.-a).

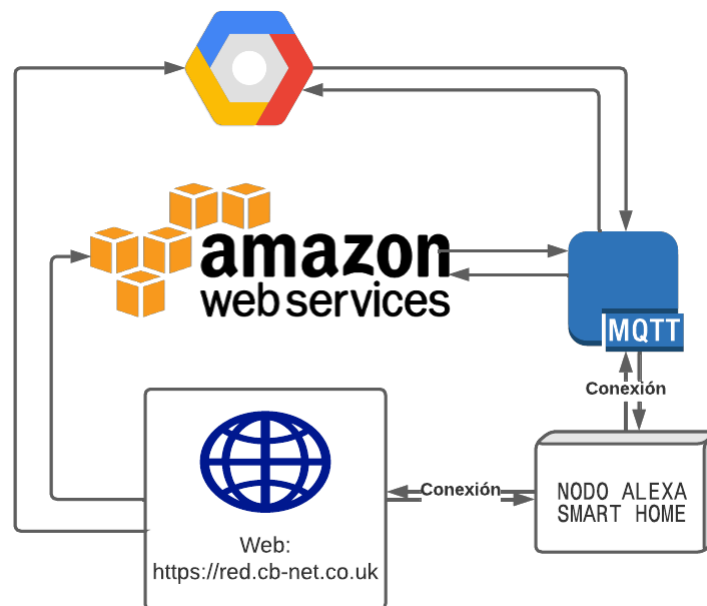
Los nodos descritos son fundamentales para el desarrollo de la programación, debido a que permiten la adquisición de datos, el procesamiento y la visualización de los mismos mientras se ejecuta el flujo de trabajo.

6.3.2. Nodos de comunicación Google Home y Alexa Smart Home

Considerando los nodos adicionales instalados como complementos, se describe el funcionamiento del paquete Alexa Smart Home, el cual funciona como puente de comunicación para los asistentes de Alexa y Google Home.

Figura 32

Comunicación nodo Alexa Smart Home



Nota: Proceso de comunicación del nodo alexa smart home con los servicios de Amazon y Google.

En la figura 24, se observa cómo funciona el nodo con respecto a la integración de los dos asistentes virtuales Alexa y Google Assistant. El nodo se direcciona al uso de una página web vinculada que permite acceder a la configuración de los dispositivos, ya sea para agregar o eliminar, comunicando cada acción con el servicio de Amazon como de Google, a su vez, se emplea el protocolo MQTT para enviar los cambios de estado y las acciones de cada dispositivo configurado a los respectivos servicios correspondientes. A continuación, se describe cada servicio y el protocolo MQTT:

- **AWS:** También conocida como Amazon Web Service, la cual, es un servicio en la nube

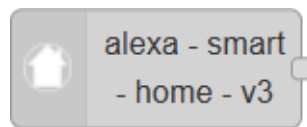
que ofrece, “tecnologías de infraestructura como cómputo, almacenamiento y bases de datos, hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas” (Amazon, 2010).

- **Home Graph:** “Google Home Graph, una base de datos que almacena y proporciona datos contextuales sobre la casa y sus dispositivos” (Google, 2022).
- **MQTT:** “MQTT es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para la comunicación de un equipo a otro.” (Amazon, 2022).

El uso de estos protocolos y plataformas conforman el paquete de Alexa Smart Home, en el cual existen diversos nodos, los cuales se describen según su función:

Figura 33

Nodo Alexa Smart Home

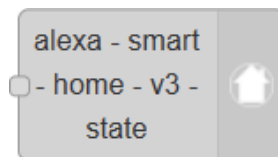


Nota: Nodo Alexa Smart Home en Node-RED.

- **Alexa Smart Home:** “This node provides a Device endpoint for the Alexa/ Google Assistant device bridge” [Este nodo proporciona un punto de enlace de dispositivo para el puente de dispositivo Alexa/Asistente de Google] (Coldfire, 2018).

Figura 34

Nodo Alexa Smart Home State



Nota: Nodo Alexa Smart Home State en Node-RED.

- Alexa Smart Home State: “This node provides state reporting to the Web App” [Este nodo proporciona informes de estado a la aplicación web] (Coldfire, 2018).

6.3.3. Nodo de comunicación HomeKit (Siri)

Una vez descrito como se realiza la comunicación de los asistentes Alexa y Google, se considera el nodo homekit bridged descargado para simular dispositivos en node red. Dentro de este paquete existen los siguientes nodos:

Figura 35

Nodo Homekit Bridge

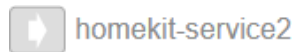


Nota: Nodo Homekit Bridge en Node-RED.

- **HomeKit Bridge:**“The Bridge node is a configuration node, specifying the bridge that iOS sees, i.e. the device that is manually being added by the user. All accessories behind a bridge noded are then automatically added by iOS” [El nodo Bridge es un nodo de configuración, que especifica el puente que ve iOS, es decir, el dispositivo que el usuario agrega manualmente. Todos los accesorios detrás de un puente de enlace son agregados automáticamente por iOS] (NRCHKB, 2021).

Figura 36

Nodo Homekit Service 2



Nota: Nodo Homekit Service 2 en Node-RED.

- **HomeKit Service 2:**“The Service node represents the single device you want to control or query. Every service node creates its own HAP accessory to keep things simple” [El nodo Servicio representa el único dispositivo que desea controlar o consultar. Cada nodo de servicio crea su propio accesorio HAP para simplificar las cosas.] (NRCHKB, 2021).

Figura 37

Nodo Service



Nota: Nodo Service en Node-RED.

- **Service:** “The Service node represents the single device you want to control or query. Every service node creates its own HAP accessory to keep things simple” [El nodo Servicio representa el único dispositivo que desea controlar o consultar. Cada nodo de servicio crea su propio accesorio HAP para simplificar las cosas] (NRCHKB, 2021).

Figura 38

Nodo Status



Nota: Nodo Status en Node-RED.

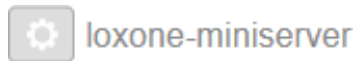
- **Status:** “This node outputs status of the HomeKit Service” [Este nodo muestra el estado del servicio HomeKit.] (NRCHKB, 2022).

6.3.4. Nodo de comunicación Loxone.

Una vez que se han descrito los nodos que permiten establecer conexión de los asistentes de voz, se requiere establecer un vínculo entre el miniservidor Loxone y Node-RED, con el fin de manipular los datos de entrada y de salida de cada una de las categorías, así como, controlar los dispositivos configurados previamente dentro del miniservidor. Se describe a continuación los nodos que conforman el paquete loxone dentro de Node-RED:

Figura 39

Nodo Loxone Miniserver

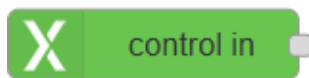


Nota: Nodo Loxone Miniserver en Node-RED.

- **Loxone Miniserver:** “Configure a miniserver connection used by every other node” [configuración de una conexión del miniservidor utilizada por todos los demás nodos.] (Utecht, 2017).

Figura 40

Nodo Control in

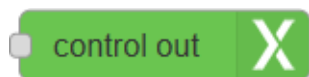


Nota: Nodo Control in en Node-RED.

- **Control in:** “Select a control and a state to hook an event which then gets passed to node-red on occurrence” [Selección de un control y estado para enlazar un evento que luego pase por node-red al ocurrir] (Utecht, 2017).

Figura 41

Nodo Control out



Nota: Nodo Control out en Node-RED.

- **Control out:** “Select a control and feed it commands” [Selección de un control y envíe comandos al miniservidor] (Utecht, 2017).

Figura 42

Nodo Stream in



Nota: Nodo Stream in en Node-RED.

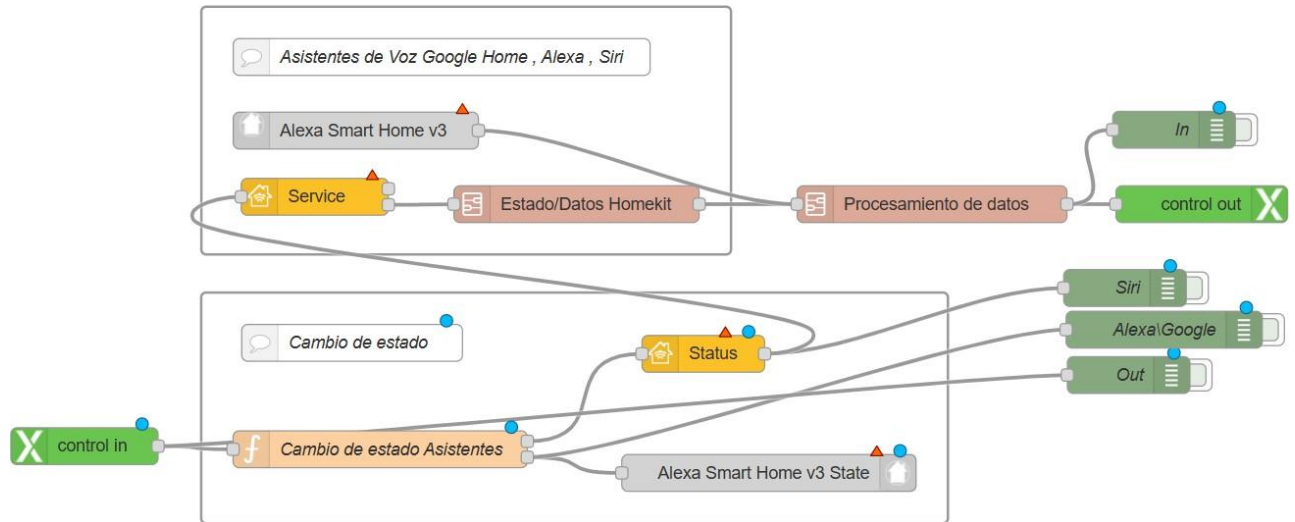
- **Stream in:** “Receive all warning occurring events from a selected room and/or category” [Recibir todos los eventos que se produzcan una advertencia en una sala y/o categoría seleccionada] (Utecht, 2017).

6.3.5. Programación para la conexión de los asistentes de voz.

Teniendo en consideración los nodos de comunicación de los asistentes Google Assistant, Alexa y Siri, se debe establecer la comunicación con el protocolo loxone.

Figura 43

Estructura general de la programación

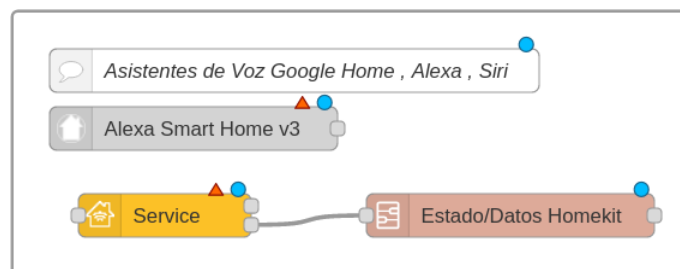


Nota: Estructura general del flujo de programación en Node-RED.

En la figura 42, se observa de manera general el diseño planteado para la comunicación de los asistentes por medio de nodos, en donde se evidencia el uso de los paquetes de Alexa, Smart Home, Homekit, Loxone y el uso de los subflujos. Además, se divide dentro de grupos los nodos según su función.

Figura 44

Grupo de Asistentes de Voz



Nota: Grupo de nodos destinado a los asistentes de voz.

- En la figura 45 se evidencia el grupo de nodos destinado a intercambiar datos entre los servidores de AWS y Google Cloud para el caso del nodo Alexa Smart Home y mediante la emulación de dispositivos a través del nodo Homekit.

Figura 45

Configuración nodo service

The image shows two side-by-side configuration panels for a HomeKit node. Panel (a) is titled 'Configuración nodo service' and contains the following fields: Service Hierarchy (Parent), Service (Choose...), Host Type (Bridge), Bridge (Bridge), Manufacturer (NRCHKB), Serial Number (Default Serial Number), Model (1.5.0), Firmware Revision (1.5.0), Hardware Revision (1.5.0), Software Revision (1.5.0), Topic (Topic), Name (Name), and Characteristic Properties ({}). Panel (b) is titled 'Configuración bridge' and contains: Pin Code (988-08-636), Port (Leave blank to auto assign), Advertiser (BONJOUR), Allow Insecure Request (checkbox), Manufacturer (NRCHKB), Serial Number (Default Serial Number), Model (1.5.0), Firmware Revision (1.5.0), Hardware Revision (1.5.0), Software Revision (1.5.0), Name (Name), Allow Message Passthrough (checkbox), and Custom MDNS Configuration (checkbox).

Nota: Configuración nodo service y bridge.

- Una vez ubicado los nodos correspondientes a cada asistente, se realiza la respectiva configuración referente a los asistentes de voz. En la figura 45, se muestra la interfaz del nodo service perteneciente al paquete de Homekit, en donde la figura 45a. presenta la interfaz del nodo, siendo los principales apartados a configurar el dispositivo a emular y conectar por medio de mDNS. Dentro de las listas desplegables se puede seleccionar el servicio de dispositivo que se pretende utilizar, ya sea un código de acceso como una Tv. Otro punto importante a considerar es el expuesto en la figura 45b. en donde se selecciona el tipo de host que se usará para comunicarse entre el dispositivo emulado y la app de Home de Apple, normalmente se realiza la conexión al puente por medio del Pin-Code, siendo necesario configurarlo una sola vez para todos los dispositivos.

Figura 46

Configuración nodo Alexa Smart Home

a) Configuración nodo Alexa Smart Home

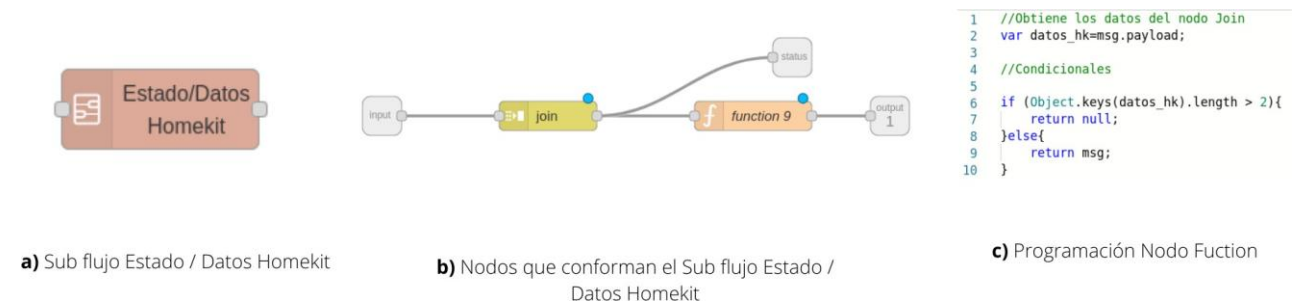
b) Configuración cuenta Node-RED Smart Home Control

Nota: Ajustes dentro del nodo Alexa Smart Home.

- Dentro del grupo que conforman los asistentes de voz, el nodo Alexa Smart Home, permite la comunicación de Alexa y Google Assistant. Este nodo permite la selección de varios dispositivos, los cuales se listan en una pestaña desplegable denominada “Device” como se muestra en la figura 46a, a su vez, permite la opción de Auto Acknowledge, la cual permite enviar una señal a la página de Node-Red Smart Home Control confirmando la ejecución de cualquier comando de voz previamente configurado. En la figura 46b, se puede observar los parámetros requeridos para la pestaña de “Account”, el “Username” y “Password” se deben crear de manera previa dentro de la página web en donde se encuentra toda la documentación del nodo “<https://red.cb-net.co.uk/>”. De igual manera, existe el apartado de MQTT, tanto el “Hostname”, como el “API”, debido a que se usará el servicio gratuito que ofrece el nodo, no se requiere configurar estos apartados; sin embargo, en caso de ejecutar una versión propia del nodo con un nuevo hosting y un propio servidor MQTT, se recomienda cambiar estos apartados.

Figura 47

Subflujo Estado/Datos HomeKit



Nota: Estructura del Subflujo Estado/Datos HomeKit.

- Por último, dentro de este grupo de nodos se encuentra el subflujo encargado de recibir únicamente los datos provenientes del nodo Homekit, como se muestra en la figura 47a, debido a que la cantidad de datos enviados por parte del nodo “Service” dependerá de la configuración adicional que se le proporcione al mismo, es decir, en caso de una bombilla que permita la regulación del brillo, la respuesta del nodo al activar dicho dispositivo enviará un dato en forma de Objeto en distintos milisegundos con diferentes propiedades, como lo puede ser “On” y “Brightness”, de tal manera el nodo “Join” mostrado en la figura 47b, permite unir estos mensajes de manera efectiva en menos de 0.01 milisegundos. Este dato tipo objeto es monitoreado dentro del ambiente de programación por medio del nodo “status” y el dato tipo objeto es receptado por el nodo función. Dentro del nodo función que se presenta en la figura 47c, se encuentra el código utilizado para enviar únicamente los datos provenientes del nodo “Service” dentro de los estados configurados se consideraron el encendido y apagado, debido a esto el condicional busca que el tamaño del objeto sea mayor a 2. Además, la función de este nodo es filtrar datos provenientes de otro grupo de nodos encargados de realizarun cambio de estado en los asistentes, gracias a esto el subflujo recibe dos conexiones distintas.

Figura 48

Subflujo Procesamiento de Datos

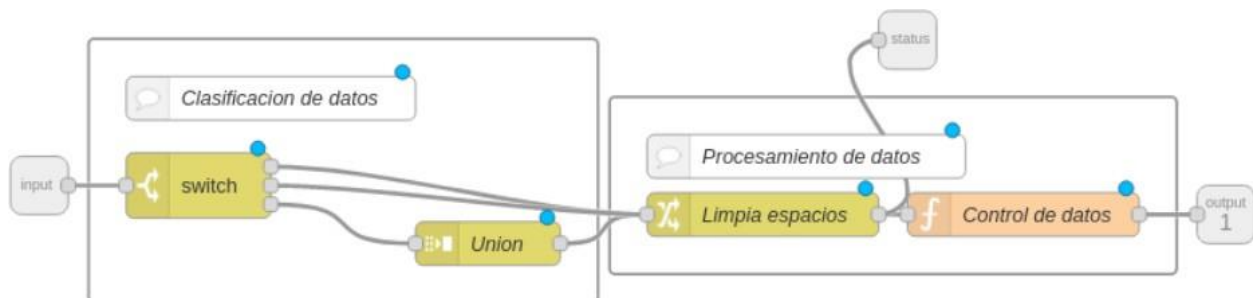


Nota: Subflujo Procesamiento de Datos en Node-RED.

El nodo consecutivo al grupo de nodos “Estado/Datos HomeKit”, es el “Procesamiento de datos” como se ve en la figura 48. Este nodo es el encargado de obtener los datos provenientes del subflujo mencionado anteriormente, traducirlos en una cadena de texto “String” o generar datos “Numéricos”, dependiendo del control que requiera el dispositivo.

Figura 49

Estructura subflujo Procesamiento de Datos



Nota: Se presenta la estructura interna del subflujo Procesamiento de Datos.

- El subflujo está conformado por varios nodos como se presenta en la figura 49, dentro de la cual, existe una agrupación para cada tipo de función interna. En el grupo de clasificación de datos, por medio del nodo “Switch”, se configuran diferentes salidas dependiendo del tipo de dato de entrada, es decir, en caso de existir un dato tipo “String” o “Numérico” hace referencia al nodo Alexa Smart Home, el cual está conformado por los asistentes de Alexa y Google. En el caso de Siri, el dato obtenido siempre será de tipo “Objeto”.

- En el caso de la agrupación de nodos procesamiento de datos, consta únicamente de dos nodos, el nodo “Change” denominado “Limpieza de datos” busca los espacios existentes del msg.payload proveniente de la agrupación “Clasificación de datos” y elimina estos espacios con el fin de entregar los datos limpios al nodo Función (Control de datos Salida).

Figura 50

Programación nodo function Procesamiento de Datos

```

1 //Variables
2 var datos = msg.payload;//Variable que recibe los datos
3
4 //Condicion para identificar si el dato es del tipo objeto
5 //Si es objeto pertenece a HomeKit
6 //Si no es objeto pertenece a Google Home/Alexa
7
8 //Google Home/Alexa
9 if (typeof datos != 'object') {
10
11     //Identifica si se envio un cambio de estado ON/OFF
12     if (typeof datos == "string" && datos == "ON") {
13         msg.payload = "On";
14     } else { //
15         msg.payload = "Off";
16     }
17 } else {
18     //HomeKit
19     //Condicion de encendido
20     if (datos.On == true) {
21         msg.payload = "On";
22     } else {
23         msg.payload = "Off";
24     }
25 }
26 return msg;

```

Nota: Se presenta la programación dentro del nodo fuction.

- Considerando la figura 50, se muestra la programación del nodo “Control de datos Salida”, en la cual consta de variables que reciben los datos y sirven como salida del nodo enviando diferentes estados. Además de existir una serie de condicionales para cada asistente, esto se da a causa de los distintos tipos de datos que ingresan al nodo, debido a que se requiere controlar el encendido y apagado, la salida proporcionada se genera por medio del msg.payload con el estado de acuerdo al que receptorá el nodo Control out.

Figura 51

Configuración Control Out Loxone

a) Configuración nodo Control out

b) Configuración miniservidor loxone

Nota: Interfaz de configuración Control Out Loxone.

- Una vez los datos pasen por todos los subflujos y grupos de nodos, se debe configurar el nodo que recibe los datos hacia el miniservidor Loxone, por medio de TCP/IP. Para ello, en la figura 51a, se debe seleccionar los campos requeridos como el nombre, categoría, y el nombre del dispositivo que se quiere controlar. No obstante, para obtener las configuraciones de los dispositivos que existen dentro del miniservidor, se debe establecer conexión mediante IP como se muestra en la figura 51b, para obtener toda información se debe preguntar directamente a la persona que configuró el miniservidor o realizar una búsqueda dentro de la red local del hogar. De manera predeterminada, el puerto al que se conecta el miniservidor es el 80 y la IP la asigna directamente el router de internet o puede ser modificada dentro de Loxone Config, se recomienda que sea estática, debido a que, si se configura una dinámica, en caso de existir algún tipo de corte de energía la IP cambiará. Así mismo, el “Username” y “Password” por defecto es admin, pero se puede variar dependiendo de los requisitos del usuario.
- Otro punto a tener en cuenta en la figura 51b, es la encriptación que tiene el miniservidor el cual, al no contar con un cifrado SSL (Seguridad en la capa de transporte) debido a su hardware, se propone el uso de tokens como medida de encriptación de datos predeterminada. Para el keepalive se establece un tiempo de 30 segundos debido a que pueden existir modificaciones en el estado de los dispositivos, lo cual requiere una

respuesta temprana por parte del miniservidor.

Figura 52

Configuración Control In Loxone

The screenshot shows the 'Properties' configuration window for 'Control In'. It includes the following fields:

- Miniserver: A dropdown menu with a red border and a pencil icon.
- Room: A dropdown menu.
- Category: A dropdown menu.
- Control: A dropdown menu with a gear icon.
- State: A dropdown menu with a checkbox icon.
- Name: A text input field containing the word 'Name'.

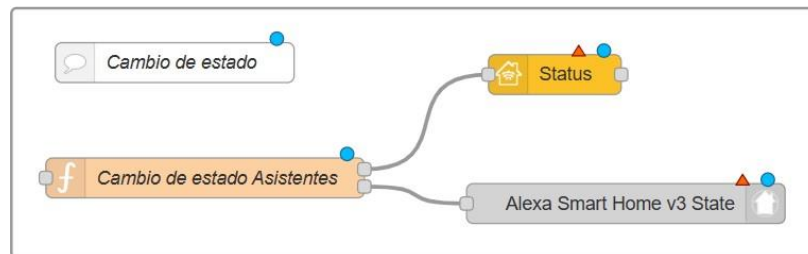
At the bottom of the window, a red error message reads: **miniserver not connected**.

Nota: Interfaz de configuración Control In Loxone.

Para empezar con el cambio de estado dentro de cada asistente, se debe recibir la información por parte del miniservidor referente al dispositivo al cual se envió un cambio de estado. Para ello, el nodo control in envía la información actual del dispositivo seleccionado, al tratarse del encendido y apagado envía datos numéricos, en donde, se puede evidenciar si existe un cambio en las variables del dispositivo.

Figura 53

Grupo de nodos para el cambio de estado



Nota: Se presentan los nodos utilizados para el cambio de estado dentro de los asistentes de voz.

En la figura 53 se observa el grupo de nodos encargado de realizar un cambio de estado visible en las aplicaciones de Smart Home de cada asistente por medio de nodos provenientes de los paquetes de los asistentes y procesando los datos que entrega el nodo control in.

Figura 54

Programación nodo function cambio de estado

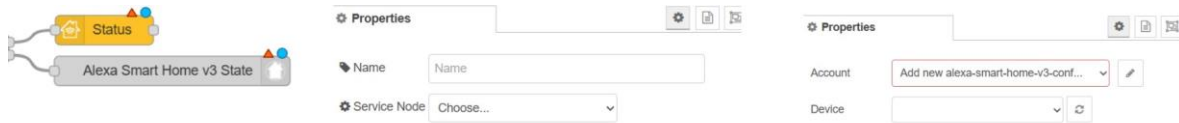
```
1 //Obtiene como dato el objeto enviado del nodo status
2 var d_status = msg.payload;
3
4 //Variables
5 var hkt;//Variable para homekit
6 var a_g;//Variable para alexa y google home
7
8 ///Loxone
9 var l_control = d_status;
10
11 // Condicionales
12 //Encendido
13 if (l_control === 1) {
14     //Homekit
15     hkt = {
16         payload: { "On": true }
17     };
18     //Google y Alexa
19     a_g = { payload: { "state": { "power": "ON" } } };
20     return [hkt, a_g];
21 }
22
23 // Apagado
24 else if (l_control === 0) {
25     hkt = {
26         payload: { "On": false }
27     };
28     //Google y Alexa
29     a_g = { payload: { "state": { "power": "OFF" } } };
30     return [hkt, a_g];
31 }
```

Nota: Se presentan la programación del nodo function denominado cambio de estado asistentes.

- En la figura 54, se muestra la programación del nodo function, el cual obtiene los datos provenientes del nodo control in de loxone y los clasifica por medio de condicionales, en este caso únicamente busca si es encendido o apagado de los dispositivos. Envía datos con el formato requerido para Alexa y Google al siguiente nodo de function, a su vez, envía una señal de salida con el formato “Objeto” característico del nodo usado para HomeKit.

Figura 55

Nodos conectados al subflujo cambio de estados de asistentes



a) Nodos Salida

b) Configuración nodo status

c) Configuración nodo alexa smart home state

Nota: Se observan los nodos conectados a la salida de todo el proceso.

- Considerando la figura 55a, se observan los nodos status y alexa smart home state, que van conectados a la salida del subflujo. A la vez, la salida que se encuentra en dirección al nodo status debe ir a la entrada del nodo service mencionado anteriormente, ya que puede existir un cambio en el estado del dispositivo configurado en caso de que el comando inicial proporcionado por el dispositivo fue enviado por los asistentes de Alexa o Google, complementando la funcionalidad del subflujo subflow “Estado/Datos Homekit”.
- La función principal del nodo status es conseguir la información del estado del dispositivo configurado, como se ve en la figura 55b, donde se selecciona únicamente el dispositivo del que se requiere la información del estado, para esto debe coincidir con el seleccionado en el nodo service, debido a que se complementa la información adquirida por el nodo status con la entrada del subflujo “Estado/Datos Homekit”.
- En el caso del nodo alexa smart home state, se debe seleccionar el dispositivo de destino al cual se le ejecutara un cambio de estado, como se ve en la figura 55c. Las entradas de este nodo dependerán de las condiciones del nodo anterior, es decir en caso de existir solo el comando de encendido (“ON”) o apagado (“OFF”), se envía el dato únicamente en la salida conectada. Por medio de este nodo se cambia el estado dentro de Google Home y Alexa en caso de que se realice un cambio de estado con Siri.

6.3.6. Configuración de comandos de voz Alexa y Google Home.

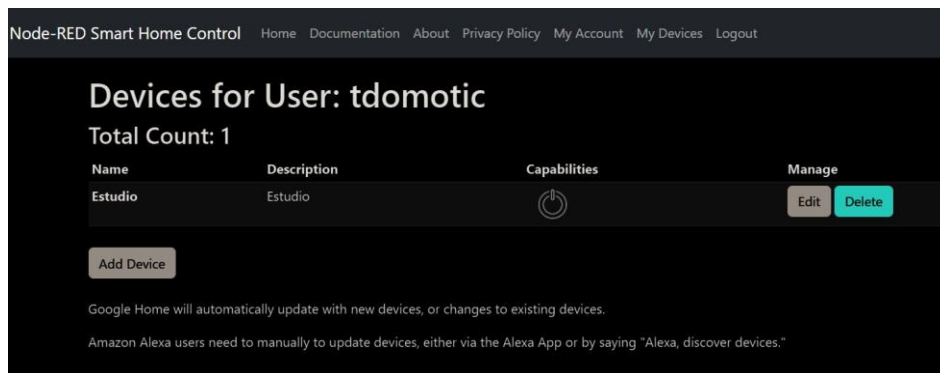
Para la activación por voz de los diversos dispositivos configurados dentro del miniservidor Loxone, se debe realizar un establecimiento previo de las palabras clave para activar o

desactivar los dispositivos mediante el reconocimiento de dicho comando por parte de los asistentes virtuales.

En el caso de los asistentes Alexa y Google Home, al coexistir dentro de la librería usada para el proyecto, se requiere de una única configuración de cada comando de voz para estos dos asistentes. Para realizar esta configuración, el paquete de Node-RED llamado node-red-alexa-smart-home ofrece una plataforma web que permite establecer parámetros de cada dispositivo al que se pretenda controlar y cada uno de los dispositivos configurados se mostrarán dentro de la configuración del nodo alexa-smart-home-v3.

Figura 56

Interfaz de configuración Alexa y Google Home

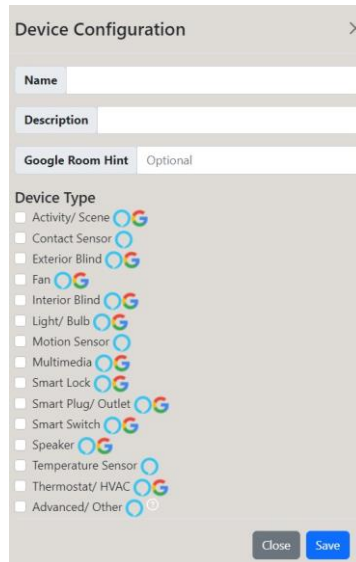


Nota: Se observa la interfaz de los dispositivos registrados previamente.

En la figura 56, se puede observar la interfaz en donde se realiza la configuración de los comandos de voz y una lista de los dispositivos registrados previamente en la cuenta creada dentro de la plataforma. Además, se puede identificar la categoría o capacidad de cada uno de los equipos, es decir, se muestra el tipo de activación de cada dispositivo.

Figura 57

Interfaz de configuración de variables para el control por voz para Alexa y Google Home



Nota: Se observa la configuración de los dispositivos con su respectivo comando de voz.

Considerando la figura 57, se muestra como es la interfaz para agregar un nuevo dispositivo, en donde el apartado llamado “Name” hace referencia al nombre de la variable por el cual, se debe activar dicho equipo configurado. Además, permite seleccionar el tipo de dispositivo, existen algunas restricciones para Google Home en caso del sensor de contacto y temperatura. De manera que con esta interfaz sencilla se puede configurar una gran cantidad de dispositivos a controlar por medio de Alexa y Google Home, cada equipo agregado se ve reflejado dentro del nodo alexa-smart-home-v3.

6.3.7. Configuración de comandos de voz Homekit.

Para los comandos de voz para Siri, se necesita realizar una configuración dentro de la programación, es decir, se genera un dispositivo por programación en donde se establece el nombre del dispositivo dentro del nodo y este se debe conectar a un denominado “Bridge” que permite la comunicación con los dispositivos Apple como lo puede ser iPhone u HomePod.

Figura 58

Configuración de comando de voz Siri

The image shows a configuration window titled "Properties" for a device. The "Service" dropdown is highlighted in green and set to "Lightbulb". Below it, "Host Type" is set to "Bridge". The "Name" field is also highlighted in green and contains the text "Name". At the bottom, the "Characteristic Properties" field is highlighted in green and is empty. Other fields include "Manufacturer" (NRCHKB), "Serial Number" (Default Serial Number), "Model" (1.5.0), "Firmware Revision" (1.5.0), "Hardware Revision" (1.5.0), and "Software Revision" (1.5.0). There is also a "Topic" field set to "Topic" and a "Wait for Setup message" checkbox.

Nota: Se observa los apartados para la configuración del dispositivo.

En la figura 58, se observa los apartados a modificar para el registro del comando de voz y las características del dispositivo, en donde el más importante es el apartado “Name” el cual hace referencia al comando de voz del dispositivo configurado. Al igual que con Alexa y Google Home se puede seleccionar el tipo de equipo a configurar y en el caso de que se pretenda controlar más de una variable del equipo se puede agregar mediante código en el apartado “Characteristic Properties”. Considerando esto, se debe establecer estos parámetros para cada uno de los equipos que se pretendan controlar por medio de comandos de voz para el asistente Siri.

6.4. Evaluar el funcionamiento del sistema de comunicación multiplataforma.

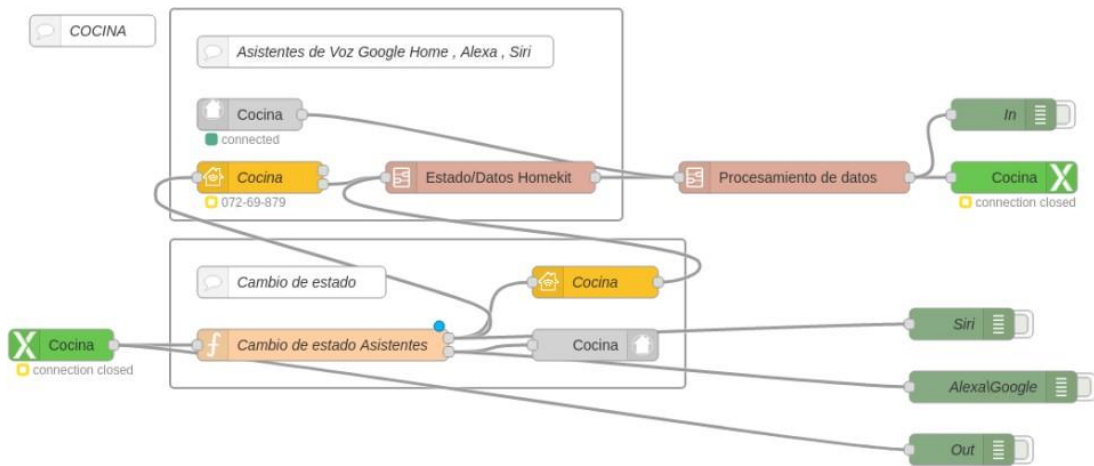
6.4.1. Pruebas de comunicación con los asistentes de Voz

Una vez planteada la estructura general de la programación para realizar la comunicación con cada uno de los asistentes de voz, se requiere probar dentro de un entorno real, es decir,

conectando el miniservidor Loxone y activando o desactivando dispositivos.

Figura 59

Pruebas de comunicación de los asistentes de voz



Nota: Se observa un ejemplo de comunicación de la comunicación Loxone - Asistentes de Voz

En la figura 59, se observa la conexión al miniservidor Loxone y el uso de un dispositivo, en este caso se realizó la prueba para la cocina, para ello se establece tanto el comando de voz "Cocina" dentro de la programación y la plataforma para Alexa y Google Home. Además, se puede observar el estado de conexión tanto de los asistentes como del miniservidor Loxone.

Figura 60

Variables de salida de los nodos

```
9/7/2023, 13:14:11 node: In
msg.payload : string[2]
"0n"
-----
9/7/2023, 13:14:11 node: Out
Cocina : msg.payload : number
1
-----
9/7/2023, 13:14:11 node: Siri
msg.payload : Object
▶ { On: true }
-----
9/7/2023, 13:14:11 node: Alexa/Google
msg.payload : Object
▶ { state: object }
```

Nota: Mensajes de variables de salida del nodo debug.

Analizando la figura 60 se tiene los siguientes resultados:

- Considerando la programación general planteada, el subflujo llamado “Proceso de datos” muestra la salida “On”, la cual es aceptada por el miniservidor loxone. Esto demuestra cómo se realizó la conversión de los datos provenientes de cualquiera de los tres asistentes de voz a un formato compatible con loxone.
- Otro punto por analizar es la respuesta del nodo control out, el cual reacciona de manera inmediata ante el cambio de estado ingresado al miniservidor referente a la variable “Cocina”, a su vez, se comprueba la correcta conversión de datos loxone al formato de cada uno de los asistentes de voz. Durante las pruebas realizadas, se ha constatado que el tiempo de respuesta y el tiempo necesario para realizar cambios de estado de cada asistente es de aproximadamente 1 segundo.

6.4.2. Programación dentro de una smart home

Una vez comprobado el funcionamiento del sistema, se implementa dentro de una vivienda que disponga del miniservidor Loxone, para ello, se identifican las zonas para los cuales se requiere realizar un control por voz dentro de la vivienda.

Tabla 4

Zonas a implementar el control dentro de una smart home.














| Zonas del hogar | Parámetro a controlar |
|-----------------|-----------------------|
| Cocina | Encendido/Apagado Luz |
| Pasillo | Encendido/Apagado Luz |
| Escaleras | Encendido/Apagado Luz |

Nota: Áreas a controlar por medio de comandos de voz dentro de una smart home

Considerando las zonas que se muestran en la tabla 4, se toma en cuenta la cantidad de dispositivos de cada una de las áreas y el tipo de activación que se requiere según lo programado dentro del miniservidor loxone, con el fin de establecer los flujos de trabajo en donde se configuran los dispositivos para la activación de voz.

Figura 61

Variables de Alexa y Google dentro de una smart home

| Name | Description | Capabilities | Manage |
|-------------|-----------------|---|-------------|
| Cocina | Luz Cocina |  | Edit Delete |
| Comedor | Luz Comedor |  | Edit Delete |
| Desayunador | Luz desayunador |  | Edit Delete |
| Entrada | Luz Entrada |  | Edit Delete |
| Escalera 1 | Luz escalera 1 |  | Edit Delete |
| Estudio | Luz estudio |  | Edit Delete |
| Estudio 1 | Luz estudio 1 |  | Edit Delete |
| Pasillo 1 | Luz pasillo 1 |  | Edit Delete |
| Pasillo 2 | Luz pasillo 2 |  | Edit Delete |
| Patio | Luz Patio |  | Edit Delete |
| Sala | Luz Sal |  | Edit Delete |
| Sala 1 | Luz sala 1 |  | Edit Delete |
| Salida | Luz Salida |  | Edit Delete |

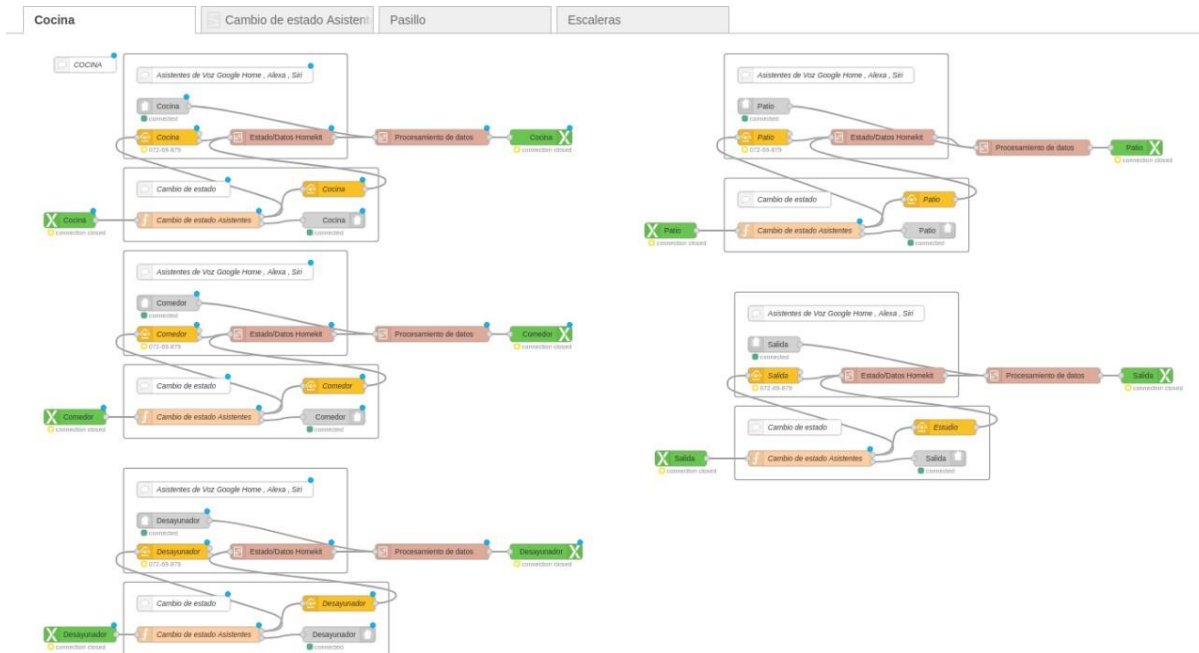
Nota: Variables para el control de voz de Alexa y Google.

Al conocer la cantidad de dispositivos dentro de cada zona, se debe configurar las variables para el control por voz del nodo alexa smart home, para ello se establecen 13 dispositivos detectados con su respectiva descripción y el parámetro a controlar.

Por consiguiente, se debe establecer la configuración para el asistente de voz Siri dentro de los flujos de trabajos con sus respectivos nombres y el parámetro a controlar.

Figura 62

Programación zona cocina



Nota: Estructura del flujo de programación para la zona de la cocina.

Figura 63

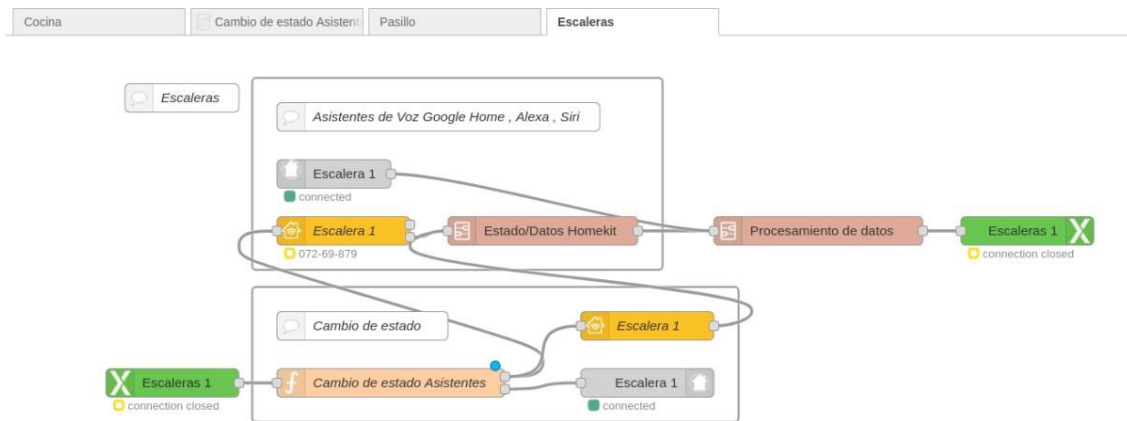
Programación zona pasillo



Nota: Estructura del flujo de programación para la zona del pasillo.

Figura 64

Programación zona escalera



Nota: Estructura del flujo de programación para la zona de escalera.

7. Resultados

7.1. Parámetros iniciales del sistema de comunicación.

Los parámetros mecánicos, electrónicos y de conexión considerados, son de gran importancia para la selección del microordenador, entre algunos de los parámetros más relevantes son:

- Conexión Wifi.
- Conexión Ethernet.
- Memoria RAM.
- Procesador.
- Rango de temperatura de operación.
- Ambiente de desarrollo.
- Durabilidad.

En cuanto a los parámetros de conexión, como lo son el Wifi y Ethernet, son las opciones fundamentales para establecer una comunicación con el miniservidor Loxone a través de la red local a la cual se encuentra conectada.

Además de los parámetros electrónicos para la selección del microordenador, como lo son la RAM y el procesador, se consideran la disponibilidad de módulos Wifi y Ethernet para establecer una comunicación con el miniservidor y con la red de internet con el fin de enviar y recibir datos de los asistentes de voz. Debido a estos parámetros el Raspberry PI 4B es un microordenador que posee todos estos parámetros y permite la creación de nuevos sistemas. Gracias a sus módulos de conectividad y las características que presentan sus componentes internos.

Por otro parte, los parámetros mecánicos como la temperatura se encuentran estrechamente relacionados con la temperatura de operación del Raspberry PI 4B. Según lo menciona, su datasheet se encuentra en un rango de entre 0°C a 50°C, la cual se debe considerar en la selección del material y con ello asegurar la durabilidad del dispositivo.

7.2. Diseño de la estructura de protección externa del módulo.

Para el diseño de la estructura de protección, se considera que el material seleccionado es ABS, el cual presenta buenas propiedades mecánicas aptas para la fabricación de elementos de protección, sabiendo esto, se ha elaborado una matriz de decisión, presentada en la tabla 5, en donde, se tomaron en cuenta cinco aspectos con el fin de calificar cada uno en función de su importancia. La puntuación más alta establece que el diseño 3 es el más idóneo para su construcción.

Tabla 5

Matriz de decisión.

| | Atractivo | | Sujeción de tapas | | Comodidad | | Facilidad de fabricación | | Costo | | Rango |
|-----------------------|-----------|-----|-------------------|-----|-----------|------|--------------------------|-----|-------|-----|-------|
| Factor de ponderación | 0.10 | | 0.20 | | 0.15 | | 0.25 | | 0.30 | | 1 |
| Diseño 1 | 5 | 0.5 | 4 | 0.8 | 1 | 0.15 | 6 | 1.5 | 4 | 1.2 | 4.5 |
| Diseño 2 | 9 | 0.9 | 8 | 1.6 | 7 | 1.05 | 8 | 2 | 6 | 1.8 | 7.35 |
| Diseño 3 | 8 | 0.8 | 8 | 1.6 | 7 | 1.05 | 8 | 2 | 8 | 2.4 | 7.85 |

Nota: Se realiza la evaluación de cada diseño dentro de una matriz de decisión.

Con respecto a la fabricación de la estructura de protección para el sistema de comunicación, se considera el proceso de manufactura aditiva con el fin de minimizar gastos y realizar diversos prototipos, garantizando la integridad del dispositivo domóticos.

Figura 65

Estructura de protección implementada



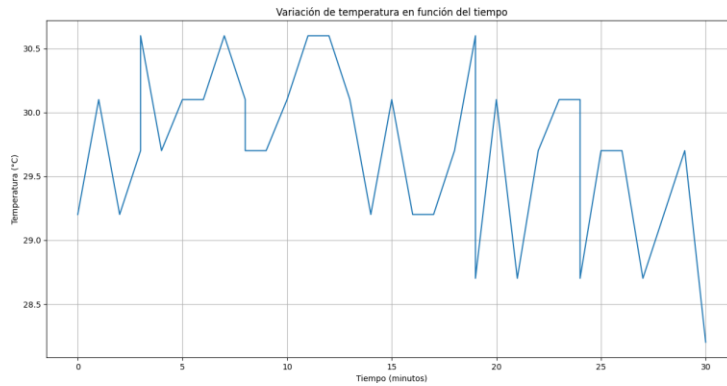
Nota: Estructura de protección para la Raspberry Pi 4B.

En la figura 65, se observa el resultado de la estructura de protección del dispositivo fabricada por impresión 3D, con su respectiva base adaptada para el carril DIN TS35, sus rejillas de ventilación, el acceso a la conexión Ethernet y el puerto alimentación del dispositivo.

Referente al análisis de refrigeración para componentes electrónicos, para comprobar la eficiencia referente a la reducción de temperatura de la Raspberry pi 4b, se realiza un monitoreo durante un periodo de tiempo de 30 minutos, como resultado se obtiene el siguiente gráfico.

Figura 66

Registro de temperatura actual Raspberry Pi

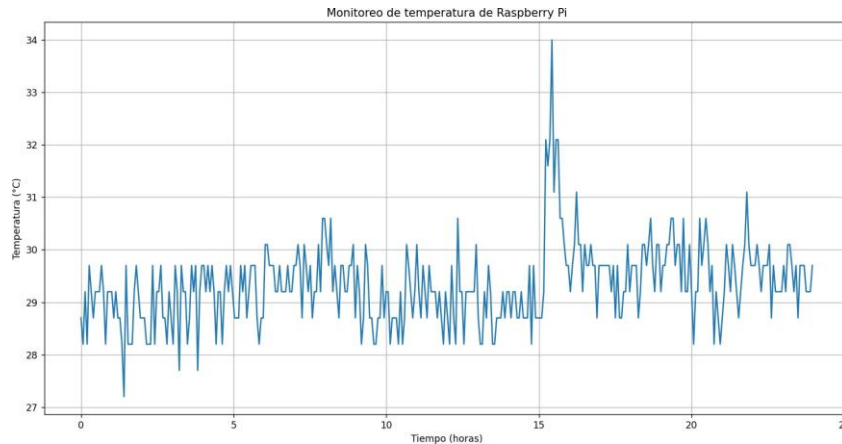


Nota: Registro en tiempo real de la temperatura del Raspberry Pi 4B.

Se observa que existe un pico máximo de temperatura de 30.6°C y una temperatura promedio de 29 °C, es decir, estas pequeñas diferencias pudieron ser ocasionadas por alguna variación de temperatura dentro de la habitación en donde se realizó el monitoreo.

Figura 67

Registro de temperatura durante 24 horas Raspberry Pi



Nota: Registro en tiempo real de la temperatura al transcurso de 24 del Raspberry Pi 4B.

Se monitoreó durante 24 horas y se obtuvo una temperatura media aproximada es de 30°C. Esta temperatura se encuentra dentro del rango de operación de 0-50°C, según lo menciona el datasheet para la placa Raspberry Pi 4B. Cabe mencionar que presenta la temperatura aproximada que se analizó dentro de la simulación de refrigeración del dispositivo.

Todos estos datos permiten comprobar que el diseño implementado es válido para un óptimo funcionamiento del microordenador Raspberry Pi 4B.

7.3. Programación del sistema de comunicación multiplataforma.

El sistema de comunicación multiplataforma desarrollado permite el control tanto del encendido o apagado, mediante voz de diversos dispositivos configurados previamente dentro del miniservidor Loxone mediante la reutilización de los diversos grupos de nodos, los cuales convierten los tipos de mensajes provenientes de cada uno de los asistentes a un formato aceptado por Loxone.

Tabla 6

Conversión de variables entrada al miniservidor.

| Asistentes de Voz | Tipo de Variables | Variables de destino |
|-------------------|-------------------|----------------------|
| Alexa | String | String |
| Google Home | String | String |
| Siri | Object | String |

Nota: Conversión de variables de cada asistente de voz hacia el miniservidor loxone.

En la tabla 6, se observa los tipos de datos dentro de la etapa inicial proporcionada por los asistentes de voz, los cuales se traducen a un formato aceptado por el miniservidor Loxone al finalizar todo el proceso.

Tabla 7

Conversión de variables para cambio de estado Asistentes de Voz.

| Tipo de Variable Loxone | Tipo de Variable Alexa | Tipo de Variable Google Home | Tipo de Variable Siri |
|-------------------------|------------------------|------------------------------|-----------------------|
| Number | Object | Object | Object |

Nota: Conversión de variables de loxone a cada uno de los asistentes de voz.

Dentro de la etapa final, para el cambio de estado existe nuevamente una conversión de datos desde el formato Loxone a uno aceptado por cada uno de los asistentes de voz configurados como se ve en la tabla 7, con el fin de que, si sucede algún evento, ya sea por medio de la activación por voz o manual, sea registrado por cada asistente.

Con los cambios que presentan cada variable en los diferentes procesos, demuestra que el sistema de comunicación multiplataforma es compatible y es eficiente con la recepción y el envío de datos tanto al miniservidor loxone como a cada uno de los asistentes de voz.

7.4. Evaluar el funcionamiento del sistema de comunicación multiplataforma.

Al implementar el sistema dentro de un ambiente real, como lo es una smart home, existen diversos datos que pueden evidenciar el correcto funcionamiento del sistema de comunicación

multiplataforma al activar o desactivar los dispositivos múltiples veces con el uso de los asistentes de voz.

Figura 68

Tiempos registrados dentro de la programación.

```

9/7/2023, 13:57:34 node: In
msg.payload : string[2]
"0n"

9/7/2023, 13:57:34 node: Out
Estudio : msg.payload : number
1

9/7/2023, 13:57:34 node: Siri
msg.payload : Object
> { On: true }

9/7/2023, 13:57:34 node: AlexaGoogle
msg.payload : Object
> { state: object }

```

Nota: Tiempos de respuesta del sistema de comunicación.

Tabla 8

Datos obtenidos al implementar el dispositivo.

| Indicador | Unidad de medida | Resultado |
|---|------------------|-------------------|
| Tiempo de respuesta de activación | Segundos | Menor a 1 segundo |
| Conversión datos Alexa y Google | String \String | Exitoso |
| Conversión de datos Siri | Object\String | Exitoso |
| Tiempo de cambio de estado Alexa y Google | Segundos | Menor a 1 segundo |
| Tiempo de cambio de estado Siri | Segundos | Menor a 1 segundo |
| Temperatura máxima registrada | Celsius | 36 °C |
| Tiempo de reinicio del dispositivo | Segundos | 30 s |
| Consumo energético | Vatio | 3 W |

Nota: Datos recopilados luego de implementar el dispositivo en una smart home.

Considerando la tabla, se pueden analizar los siguientes puntos como resultado de la implementación:

- El sistema de comunicación presenta una respuesta temprana menor a 1 segundo para la activación de los dispositivos dentro de una smart home, es decir, el intercambio de información, así como la conversión de datos por parte de Alexa, Google y Siri, es eficiente para la comunicación con el miniservidor Loxone.
- Con respecto al cambio de estado se refleja de manera simultánea en los tres asistentes de voz. Esto implica que se registre de manera temprana e imperceptible el nuevo estado de los dispositivos dentro de la aplicación celular de cada asistente.
- En cuanto a las características físicas del dispositivo, se presenta una temperatura estándar dentro del ambiente normal de operación, sin llegar a presentar problemas de deformación por temperatura del material. A su vez se acopla perfectamente al carril DIN TS35 con cuál el diseño es válido para las instalaciones domóticas.
- En caso de existir algún problema con la red eléctrica a la cual esté conectada el dispositivo, la configuración que se establece en el microordenador permite que el dispositivo en caso de un reinicio forzado retome su funcionalidad completamente, evitando la pérdida de las configuraciones de los dispositivos dentro de la smart home.
- Con respecto al consumo energético del microordenador se estima que consume aproximadamente 4W, ya que al tratarse de un sistema que funciona en segundo plano, no existe la apertura de otras aplicaciones y con ello existe la reducción del uso del procesador junto con la disminución de la temperatura. Con esto se estima un gasto anual de 3.5 USD/kWh, lo que resultado como un consumo bajo para el sistema de comunicación.

8. Conclusiones

- Los parámetros iniciales electrónicos y de conexión son esenciales en la selección del microordenador que permita ejecutar correctamente el enlace con los asistentes más usados Alexa, Google y Siri, con el sistema Loxone. Además, las variables mecánicas planteadas son importantes tanto en el diseño como en la selección de materiales compatibles con los

rangos de operación del dispositivo, de manera que el sistema de comunicación pueda operar correctamente.

- Como resultado del diseño de la estructura de protección, el uso de un análisis para la refrigeración del microordenador es crucial para la selección de componentes adecuados, ya sean disipadores o ventiladores. Además, mediante la impresión 3D como proceso de manufactura se pueden elaborar diversos prototipos que permitan optimizar el diseño sin generar grandes costos de producción.
- El sistema de comunicación multiplataforma permite la integración de los asistentes de voz más utilizados, como lo es Alexa, Google y Siri, para el control por voz de dispositivos previamente configurados dentro del miniservidor Loxone, para ello el uso de diversos canales de conexión como Wifi, Ethernet y los protocolos de comunicación TCP/IP son de suma importancia para el intercambio de información entre los servicios configurados de manera local y los que se encuentran a cargo de la gestión de los asistentes de voz.
- Se evaluó el funcionamiento del sistema de comunicación multiplataforma con el miniservidor Loxone dentro de una smart home, agregando una nueva funcionalidad para la gestión de los dispositivos por medio de comandos de voz, lo cual genera un mayor confort. Todos los datos recopilados y las pruebas de funcionalidad presentados se efectuaron con los asistentes de voz en diversos dispositivos como lo son Amazon Echo, teléfonos celulares Android y Apple. En conclusión, se ha comprobado que el módulo permite el de distintos dispositivos, ya sean teléfonos inteligentes, altavoces, audífonos, entre muchos otros.

9. Recomendaciones

- En caso de agregar nuevas funcionalidades, se recomienda modificar el código o los bloques de cada subflujo de trabajo, ya que esto optimiza el tiempo y permite que toda la programación se pueda reutilizar. Se debe considerar el tipo de dispositivo a implementar tanto en el paquete alexa-smart-home y homekit, ya que el tipo de datos que envían dependerán del tipo de dispositivo.

Dentro de la programación, en el caso de Alexa y Google, al configurar dispositivos que realicen más de una acción, considerar que el cambio de estado se ve afectado, de manera que se debe realizar acciones consecutivas en un orden específico para realizar una retroalimentación adecuada a los asistentes de voz.

En cuanto a homekit se recomienda identificar con un nombre en específico el puente de comunicación, con el fin de evitar posibles confusiones en el caso de usar más de un puente.

- Para mantener funcionando correctamente el dispositivo, se recomienda llevar a cabo un mantenimiento preventivo, el cual consta principalmente de realizar un cambio de ventilador de manera anual debido a que se puede generar una acumulación de polvo o desajuste que impida el correcto funcionamiento del dispositivo encargado de la refrigeración forzada. Además de efectuar una limpieza cada 6 meses dentro y fuera de la estructura de protección para prevenir bloqueos en las rejillas de ventilación y daños en el microordenador.

Referencias

- Amazon. (2010, 12). *¿Qué es AWS?* Descargado de <https://aws.amazon.com/es/what-is-aws/>
- Amazon. (2022, 10). *¿Qué es el MQTT? - Explicación del protocolo MQTT - AWS.* Descargado de <https://aws.amazon.com/es/what-is/mqtt/>
- AutoDesk. (2021, 9). *Software de diseño e ingeniería mecánica 3D | Autodesk.* Descargado de <https://www.autodesk.es/solutions/3d-mechanical-engineering>
- Bastian, P. (2001). *Electrotecnia* (Vol. 1). Ediciones Akal.
- Batard, P. (s.f.). *Rufus - Cree unidades USB arrancables fácilmente.* Descargado de <https://rufus.ie/es/>
- Beagleboard.org. (2022, 11). *BeagleBone® AI-64.* Descargado de <https://www.beagleboard.org/boards/beaglebone-ai-64>
- Blanes, C. S. (2011). *Diseño mecánico con autodesk inventor paso a paso.* Editorial Club Universitario.
- Coldfire. (2018, 12). *node-red-contrib-alexa-smart-home.* Descargado de <https://flows.nodered.org/node/node-red-contrib-alexa-smart-home>
- Colorplus3D. (2023, 6). Descargado de <https://www.colorplus3d.com/>
- Control, N.-R. S. H. (s.f.). *Home | Node-RED Smart Home Control.* Descargado de <https://red.cb-net.co.uk/>
- DE, U. G. P. L. I., y ALEXA, A. C. (2020). Guía de despliegue y uso de asistentes de voz en el hogar del mayor.
- DietPi. (2018, 4). *Lightweight justice for your SBC!* Descargado de <https://dietpi.com/>
- Etcher, B. (2021, 6). *Preguntas frecuentes - Balena Etcher.* Descargado de <https://balenaetcher.online/preguntas-frecuentes/#:~:text=balenaEtcher%20es%20un%20software%20de,%2C%20macOS%2C%20e%20incluso%20Linux.>
- García, N. H. (2020). Propuesta de arquitectura para el desarrollo e implementación de un módulo de nuevos comandos e instrucciones para google assistant. *Revista Aristas*, 8 (15), 128–132.
- Gaviria Gutierrez, M., y Lara Urrego, J. C. (2020). Diseño de una red lan para la empresa heavens fruit sas.
- Google. (2022, 10). *Home Graph.* Descargado de <https://developers.home.google.com/cloud-to-cloud/primer/home-graph?hl=es-419#:~:text=Home%20Graph%20is%20essentially%20a,that%20you%20are%20currently%20in.>
- Harke, W. (2010). *Domotica para viviendas y edificios.* Marcombo.

- Hoy, M. B. (2018). *Alexa, siri, cortana, and more: an introduction to voice assistants* (Vol. 37) (n.º 1). Taylor Francis.
- Huidobro, J. M., y Tejedor, R. J. M. (2010). *Manual de domotica*. creaciones copyright sl.
- Kurose, J., y Ross, K. W. (2010). *Redes de computadoras* (Vol. 5). Pearson educación.
- Loxone. (2020, 3). *Loxone Tree - ESES Loxone*. Descargado de <https://www.loxone.com/eses/kb/loxone-tree/>
- Loxone. (2023a, 2). *Air Base Extension*. Descargado de <https://www.loxone.com/eses/kb/air-base-extension/#:~:text=Loxone%20Air%20incluye%20una%20tecnolog%C3%ADa,fija%2C%20que%20funcionan%20como%20repetidores.>
- Loxone. (2023b, 2). *Link*. Descargado de <https://www.loxone.com/eses/kb/link/>
- Loxone. (2023c, 1). *Miniserver*. Descargado de <https://www.loxone.com/eses/kb/miniserver/>
- Loxone Electronics. (2020, 3). *Información sobre el cableado*. Descargado de <https://www.loxone.com/eses/kb/informacion-sobre-cableado/>
- Ltd, R. P. (2015, 4). *Raspberry Pi OS – Raspberry Pi*. Descargado de <https://www.raspberrypi.com/software/>
- Node-RED. (s.f.-a). *The Core Nodes : Node-RED*. Descargado de <https://nodered.org/docs/user-guide/nodes>
- Node-RED. (s.f.-b). *Flows : Node-RED*. Descargado de <https://nodered.org/docs/user-guide/editor/workspace/flows>
- Node-RED*. (s.f.). Descargado de <https://nodered.org>
- Node-RED. (s.f.). *Subflows : Node-RED*. Descargado de <https://nodered.org/docs/user-guide/editor/workspace/subflows>
- Node-RED. (2013, 9). *Running on Raspberry Pi : Node-RED*. Descargado de <https://nodered.org/docs/getting-started/raspberrypi>
- NRCHKB. (2021, 4). *HomeKit all the things!* Descargado de <https://nrchkb.github.io/>
- NRCHKB. (2022, 7). *Status Node*. Descargado de <https://nrchkb.github.io/wiki/nodes/status-node/>
- Pi, R. (2015, 4). *Raspberry Pi Documentation - Raspberry Pi OS*. Descargado de <https://www.raspberrypi.com/documentation/computers/os.html>
- Pi, R. (2019, 6). *Raspberry-pi-4-datasheet*. Descargado de <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>
- Ramírez Sánchez, J., y Díaz Martínez, J. (2012). Las redes inalámbricas, más ventajas que desventajas. *Revista Ciencia Administrativa [Internet]*, 2, 85–89.
- Raspberry. (2022, 4). *¿que es ?* Descargado de <https://raspberrypi.cl/que-es>

-raspberry/

Sarasúa, J. C. (2011, 7). *Domótica. Un factor importante para la arquitectura sostenible*. Descargado de https://revistascientificas.cuc.edu.co/moduloarquitecturacuc/article/view/141/pdf_72

Soto, G. A. E. (s.f.). *La interacción de las personas con los asistentes virtuales: Estudio sobre los usos, percepción e impactos del asistente siri de apple*. sf.

Utecht, D. (2017, 3). *node-red-contrib-loxone*. Descargado de <https://flows.nodered.org/node/node-red-contrib-loxone>

Velez Palomeque, C. S., y Azuero Gonzales, E. A. (2022). *Diseño y construction de un modulo para la comunicacin entre el dispositivo alexa y el protocolo loxone* (B.S. thesis).

Verma, G. (2018). *Autodesk fusion 360 black book*. BPB Publications.

Vidal, C. R. (2015). *Diseño mecánico con solidworks 2015*. Grupo Editorial RA-MA.

ANEXOS

Anexo A: Matriz de Consistencia Lógica

Tabla 9

Matriz de consistencia.

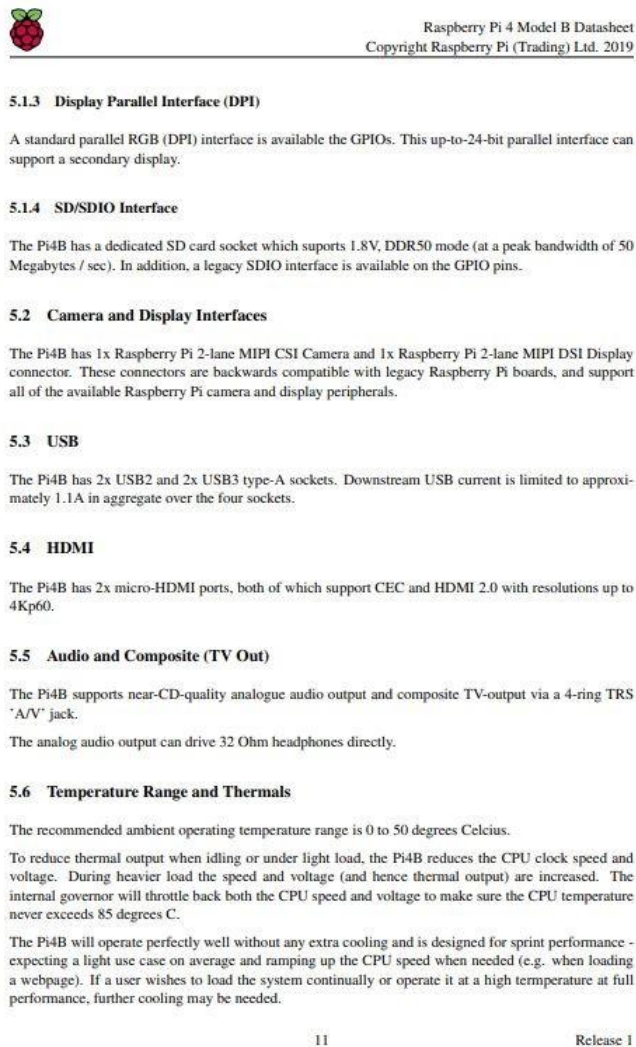
| MATRIZ DE CONSISTENCIA | | |
|--|--|-------------------|
| PROBLEMA GENERAL | OBJETIVO GENERAL | MARCO TEÓRICO |
| ¿Es posible diseñar e implementar un sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone dentro de un smart home? | Diseñar e implementar un sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone dentro de un smart home. | Protocolo Loxone |
| PROBLEMAS ESPECÍFICOS | OBJETIVOS ESPECÍFICOS | MARCO TEÓRICO |
| ¿Se podrá reconocer los parámetros iniciales para diseñar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone dentro de un smart home? | Reconocer los parámetros iniciales para diseñar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone dentro de un smart home. | Asistentes de voz |
| ¿Es posible diseñar la estructura externa de protección con carril DIN para el sistema de comunicación multiplataforma de los asistentes de voz? | Diseñar la estructura externa de protección con carril DIN para el sistema de comunicación multiplataforma de los asistentes de voz. | Gateway |
| ¿Se podrá implementar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone dentro de un smart home? | Implementar un sistema de comunicación multiplataforma de los asistentes de voz con el protocolo loxone dentro de un smart home. | Node-Red |
| ¿Es posible comprobar el funcionamiento de un sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone dentro de un smart home? | Comprobar el funcionamiento del sistema de comunicación multiplataforma de asistentes de voz con el protocolo loxone dentro de un smart home. | |

Nota: Matriz de consistencia para la identificación de objetivos generales y específicos.

Anexo B: Datasheet de Raspberry Pi 4B

Figura 69

Temperaturas de operación Raspberry Pi 4B, datasheet.



Nota: Datos del rango de operación de la placa. Pi (2019).

Figura 70

Temperaturas de operación Raspberry Pi 4B, datasheet.



2.3 Software

- ARMv8 Instruction Set
- Mature Linux software stack
- Actively developed and maintained
 - Recent Linux kernel support
 - Many drivers upstreamed
 - Stable and well supported userland
 - Availability of GPU functions using standard APIs

3 Mechanical Specification

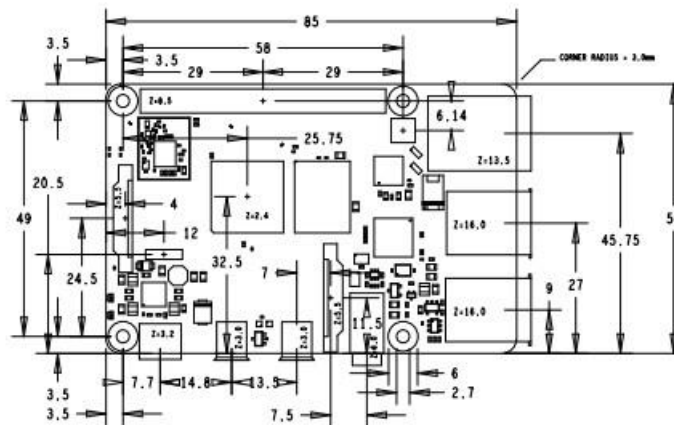


Figure 1: Mechanical Dimensions

4 Electrical Specification

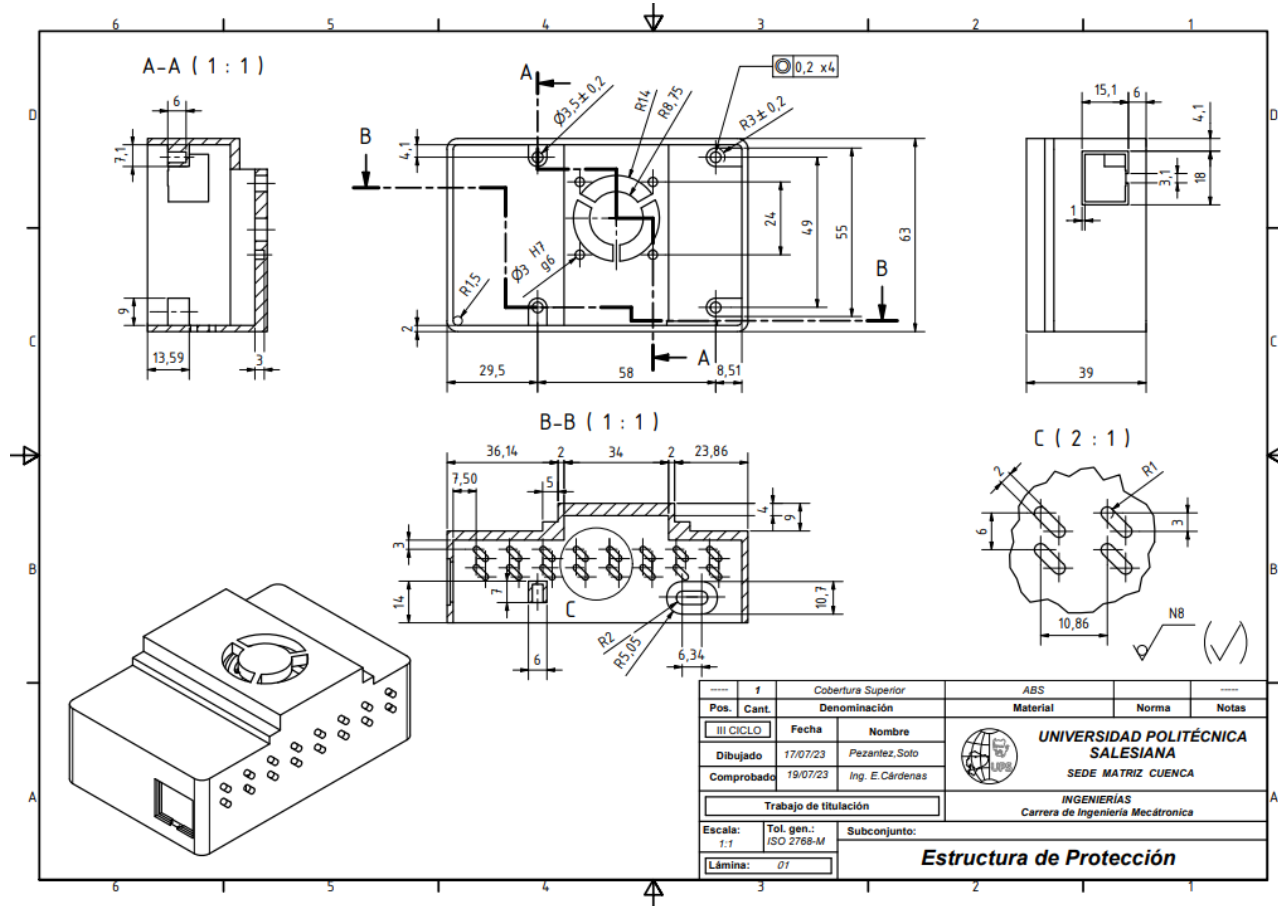
Caution! Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Nota: Dimensiones de la placa Rapsberry Pi 4B. Pi (2019).

Anexo C: Planos Estructura de Protección

Figura 71

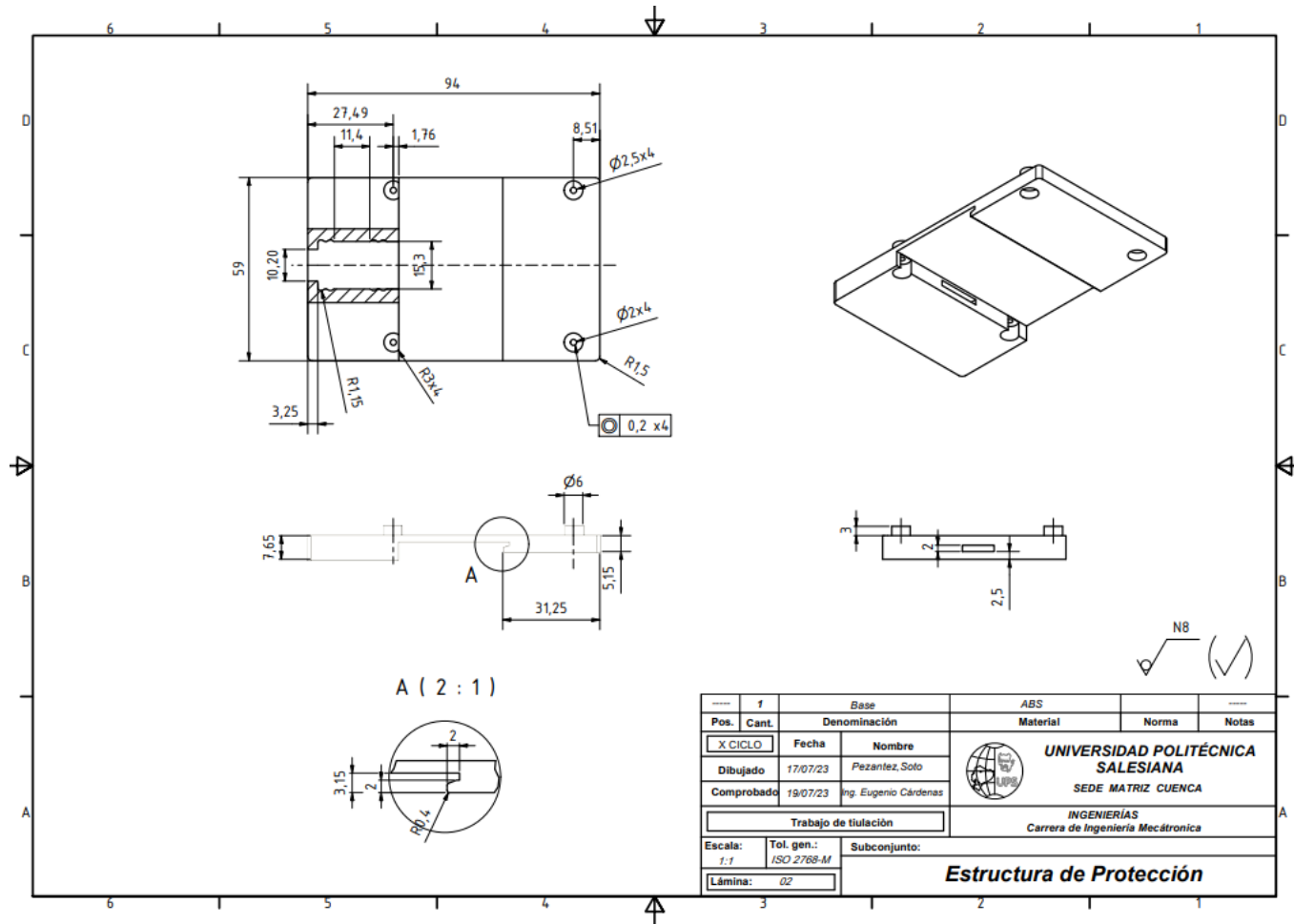
Planos Cubierta Superior.



Nota: Planos para la elaboración de la cubierta superior.

Figura 72

Planos Cubierta Inferior.



Nota: Planos para la elaboración de la cubierta inferior.