



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE COMANDOS  
DE VOZ

CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS

MACHALA  
2021



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE  
COMANDOS DE VOZ

CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS

MACHALA  
2021



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN  
PROPUESTAS TECNOLÓGICAS

GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE COMANDOS DE VOZ

CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS

HERNANDEZ ROJAS DIXYS LEONARDO

MACHALA, 26 DE ABRIL DE 2021

MACHALA  
2021

# Tesis K

---

## INFORME DE ORIGINALIDAD

---

0%

INDICE DE SIMILITUD

0%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL  
ESTUDIANTE

---

## FUENTES PRIMARIAS

---

1

[libros.cecar.edu.co](http://libros.cecar.edu.co)

Fuente de Internet

<1%

2

[ccu.maz.uasnet.mx](http://ccu.maz.uasnet.mx)

Fuente de Internet

<1%

---

Excluir citas

Activo

Excluir coincidencias < 30 words

Excluir bibliografía

Activo

## CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, CUEVA GALLEGOS KARLA ANDREA, en calidad de autora del siguiente trabajo escrito titulado GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE COMANDOS DE VOZ, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 26 de abril de 2021

  
CUEVA GALLEGOS KARLA ANDREA  
0704653625

UNIVERSITAS  
MAGISTRO-  
RUM  
ET SCHOLARUM



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL  
CARRERA DE INGENIERÍA DE SISTEMAS**

**GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE  
COMANDOS DE VOZ**

**CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS**

**MACHALA**

**2021**



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL  
CARRERA DE INGENIERÍA DE SISTEMAS**

**GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE  
COMANDOS DE VOZ**

**CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS**

**MACHALA**

**2021**



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL  
CARRERA DE INGENIERÍA DE SISTEMAS**

**TRABAJO DE TITULACIÓN  
PROPUESTAS TECNOLÓGICAS**

**GESTIÓN DE SISTEMAS DOMÓTICOS LOCAL MEDIANTE  
COMANDOS DE VOZ**

**CUEVA GALLEGOS KARLA ANDREA  
INGENIERA DE SISTEMAS**

**HERNANDEZ ROJAS DIXYS LEONARDO**

**MACHALA, 2021**

**MACHALA**

**2021**





## **DEDICATORIA**

Hace cinco años inicié el sueño y compromiso de obtener mi título universitario en la Carrera de Ingeniería de Sistemas; el presente trabajo está dedicado primeramente al forjador de mi camino, el que siempre me acompaña en cada paso que doy, mi padre celestial, Dios.

A mi madre, por ser un ejemplo de constancia y mi mayor apoyo en cada momento de dificultad que me ha tocado pasar en todo este proceso.

A mi familia, que con su amor y cariño han sido piezas fundamentales en cada momento; a mis amigos, por estar presentes en las buenas y en las malas; a todas aquellas personas que fueron parte en este proceso.

## **AGRADECIMIENTO**

Ser mamá sin duda es la profesión más difícil que pueda existir, no hay horarios ni vacaciones, los constantes sacrificios que solo una madre puede hacer para que sus hijos cumplan sus sueños.

Es por ello que quiero empezar agradeciendo a Dios por permitirme tener una gran madre; trabajadora y dedicada, que con su amor ha sido un pilar fundamental en cada decisión y proyecto, con su apoyo y consejos he podido superar cada uno de los obstáculos que se me han presentado,

Agradezco a mi familia y amigos; por el apoyo incondicional que me han brindado día a día; a mi tutor de tesis el Ing. Dixys Leonardo Hernández Rojas que ha impartido sus conocimientos en el desarrollo del presente trabajo de titulación para obtener mi título universitario como Ingeniera de Sistemas.

## **RESUMEN**

El Internet de las Cosas hace referencia a la conexión de cualquier dispositivo electrónico con internet, desde un foco hasta diversos electrodomésticos; recibiendo y enviando datos de manera inalámbrica sin la necesidad de que el ser humano interfiera entre el dispositivo y el internet; datos que se convertirán en información para optimizar procesos de la vida cotidiana del usuario.

Desde hace más de tres décadas el ser humano busca la manera de optimizar el tiempo y es ahí donde entra el concepto de domótica, que se caracteriza por hacer que cualquier dispositivo electrónico del hogar se conecte a la nube, esta gestión de dispositivos inteligentes desde cualquier herramienta es lo que se denomina como casa inteligente o Home Automation, siendo una de las aplicaciones más importantes y populares dentro del Internet de las cosas.

Actualmente se puede controlar una vivienda a través del reconocimiento de voz, el cual permite una comunicación entre ser humano y computador. Es por ello que surge la necesidad de integrarlos a un sistema domótico que cuente con protocolos de comunicación que permitan su integración, haciendo fácil el trabajo de monitoreo y control; gracias al acceso a internet el usuario controla el sistema dentro y fuera de su red local.

Gracias a que la domótica permite el control de dispositivos inteligentes, el presente trabajo ha considerado manejarlos a través de comandos de voz mediante micrófonos y asistentes de voz incorporados dentro de la plataforma domótica, con la finalidad de intercambiar directamente información enviada por el usuario mediante los comandos de voz al sistema domótico que gestiona las peticiones a los dispositivos.

El presente trabajo se basa en la gestión de sistemas domóticos local, para lo cual se implementó la plataforma domótica Home Assistant donde se integran dispositivos inteligentes que podrán ser gestionados y controlados con las herramientas que ofrece esta plataforma, también se implementa un modelo 3D del hogar en donde se realiza el encendido y apagado de los dispositivos; para el reconocimiento de comandos de voz se dispone de la plataforma domótica Rhasspy donde se configura los micrófonos que sirven de receptor del comando de voz para él envió de eventos hacia Home Assistant que se encarga de ejecutar la petición, se

debe configurar los sistemas para que se comuniquen entre sí y realizar las peticiones.

Como todo sistema está sujeto a pruebas, en este proyecto se establecieron parámetros de distancia dentro de un rango de 1 a 4 metros y tonos de voz pertenecientes a hombres, mujeres, niños y ancianos; empleados en los micrófonos USB y ReSpeaker 4-Mic Array que se encuentran en la plataforma domótica Rhasspy, también se hizo uso de los asistentes de voz Alexa Amazon y Google Assistant, los resultados de la evaluación al prototipo propuesto, indican que el primer escenario de confirmación en ambiente sin ruido es favorable por presentar un porcentaje mayor al 50% de satisfacción a nivel general del conjunto de pruebas realizadas, en cambio, el escenario con ambiente de ruido presenta falencias ya que los recursos utilizados son de baja arquitectura, mientras que las pruebas de latencia reflejan que los tiempos obtenidos en milisegundos son de menor porcentaje ocasionando que la comunicación entre los dispositivos, micrófonos, asistentes de voz y las plataformas domóticas sea más veloz, siendo un sistema capaz de reconocer comandos de voz y ejecutar la petición de manera óptima.

**Palabras claves:**

IoT, Domótica, Gestión, Plataformas Domóticas, Home Assistant, Rhasspy, Micrófonos, Asistentes de voz

## **ABSTRACT**

The Internet of Things refers to the connection of any electronic device to the internet, from a light bulb to various household appliances; receiving and sending data wirelessly without the need for humans to interfere between the device and the internet; data that will become information to optimize processes of the user's daily life.

For more than three decades the human being has been looking for ways to optimize time and that is where the concept of home automation comes in, which is characterized by making any electronic device in the home connect to the cloud, this management of intelligent devices from any tool is what is called smart home or Home Automation, being one of the most important and popular applications within the Internet of things.

Currently a home can be controlled through voice recognition, which allows communication between human and computer. That is why there is a need to integrate them into a home automation system that has communication protocols that allow their integration, making easy the work of monitoring and control; thanks to Internet access the user controls the system inside and outside the local network.

Thanks to the fact that home automation allows the control of intelligent devices, the present work has considered managing them through voice commands by means of microphones and voice assistants incorporated within the home automation platform, with the purpose of correctly exchanging information sent by the user through voice commands to the home automation system that manages the requests to the devices.

The present work is based on the management of local home automation systems, for which the Home Assistant home automation platform was implemented where intelligent devices are integrated that can be managed and controlled with the tools offered by this platform, also a 3D model of the home is implemented where the switching on and off of the devices is performed; For the recognition of voice commands, the Rhasspy home automation platform is used to configure the microphones that serve as voice command receivers to send events to Home Assistant, which is responsible for executing the request.

As any system is subject to testing, in this project, distance parameters were established within a range of 1 to 4 meters and voice tones belonging to men, women, children and the elderly; employed in USB microphones and ReSpeaker 4-Mic Array found in the Rhasspy home automation platform, also made use of Alexa Amazon and Google Assistant voice assistants, the results of the evaluation to the proposed prototype, indicate that the first scenario of confirmation in environment without noise is favorable by presenting a percentage greater than 50% satisfaction at the overall level of the set of tests performed, on the other hand, the scenario with noise environment presents shortcomings since the resources used are of low architecture, while the latency tests reflect that the times obtained in milliseconds are of lower percentage causing the communication between devices, microphones, voice assistants and home automation platforms to be faster, being a system capable of recognizing voice commands and execute the request in an optimal way.

**Keywords:**

IoT, Domótica, Gestión, Plataformas Domóticas, Home Assistant, Rhasspy, Micrófonos, Asistentes de voz

## **CONTENIDO**

<b>DEDICATORIA</b>	<b>1</b>
<b>AGRADECIMIENTO</b>	<b>2</b>
<b>RESUMEN</b>	<b>3</b>
<b>ABSTRACT</b>	<b>5</b>
<b>INTRODUCCIÓN</b>	<b>13</b>
<b>1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS</b>	<b>14</b>
1.1.    Ámbito de aplicación: descripción del contexto y hechos de interés	14
1.2.    Establecimiento de Requerimientos	15
1.3.    Justificación del requerimiento a satisfacer	15
<b>2. CAPÍTULO II. DESARROLLO DEL PROYECTO</b>	<b>16</b>
2.1.    Definición del prototipo tecnológico	16
2.2.    FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO	17
2.2.1.    Internet de las cosas	17
2.2.1.1.    Dispositivos IOT	18
2.2.2.    Domótica	18
2.2.2.1.    Protocolos de comunicación	19
2.2.2.1.1.    SSH	19
2.2.2.2.    Plataformas Domóticas	19
2.2.2.2.1.    Home Assistant	19
2.2.2.2.2.    Smart Life	20
2.2.2.2.3.    Rhasspy	20
2.2.2.3.    Dispositivos domóticos o inteligentes	20
2.2.2.3.1.    Nexxt	20
2.2.2.3.2.    Raspberry	21
2.2.2.3.3.    ReSpeaker	21
2.2.2.3.4.    Echo Dot	21
2.3.    OBJETIVOS DEL PROTOTIPO	22
2.3.1.    Objetivo General	22
2.3.2.    Objetivos Específicos	22
2.4.    DISEÑO DEL PROTOTIPO	22
2.4.1.    Diseño del plano del hogar en Sweet Home 3D y GIMP	25
2.5.    EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO	28
2.5.1.    Instalación de Home Assistant en VirtualBox	28
2.5.2.    Integración de dispositivos IoT a Home Assistant	31
2.5.2.1.    Integración de dispositivos inteligente Nexxt	31



2.5.2.2.	Integración de dispositivos IoT a Home Assistant	32
2.5.2.3.	Integración de Home Assistant Cloud	34
2.5.2.4.	Integración de Amazon Alexa	39
2.5.3.	Pantalla de inicio de Home Assistant con dispositivos integrados	41
2.5.4.	Instalación de Rhasspy	41
2.5.4.1.	Preparación de tarjeta Micro SD 32gb	41
2.5.4.2.	Instalación de Raspbian	43
2.5.4.3.	Instalación de ReSpeaker 4-Mic Array	44
2.5.4.4.	Instalación de Rhasspy con Docker	47
2.5.4.5.	Configuración del sistema Rhasspy	47
2.5.5.	Conexión rhasspy con hassio	51
2.5.6.	Integración del diseño 3D en Home Assistant	53
2.5.6.1.	Instalación de Add-on Samba en Home Assistant	53
2.5.6.2.	Implementación del modelo 3D en Home Assistant	55
<b>3.</b>	<b>CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO</b>	<b>57</b>
3.1.	PLAN DE EVALUACIÓN	57
3.1.1.	Aplicación en Node-Red	58
3.1.1.1.	Instalación del Add-on Node-Red en Home Assistant	58
3.1.1.2.	Aplicación para toma de tiempos	60
3.1.2.	Toma de tiempos reconocimiento de comando voz	61
3.1.2.1.	Micrófono USB	62
3.1.2.2.	Micrófono ReSpeaker 4-Mic Array	63
3.1.3.	Toma de tiempos de peticiones	64
3.1.3.1.	Micrófono USB	64
3.1.3.2.	Micrófono ReSpeaker 4-Mic Array	66
3.1.3.3.	Asistente de voz Alexa Amazon	68
3.1.3.4.	Asistente de voz Google Assistant	70
3.1.4.	Toma de tiempos en la confirmación de peticiones	72
3.1.4.1.	Micrófono USB	73
3.1.4.2.	Micrófono ReSpeaker 4-Mic Array	74
3.1.4.3.	Asistente de voz Alexa Amazon	75
3.1.4.4.	Asistente de voz Google Assistant	76
3.2.	RESULTADOS DE LA EVALUACIÓN	77
3.2.1.	Comparación de dispositivos usados	77
3.2.1.1.	Micrófono USB	78
3.2.1.2.	Micrófono ReSpeaker 4-Mic Array	79

3.2.1.3.	Asistente de voz Alexa Amazon y Google Assistant	80
3.2.1.4.	Comparación de los micrófonos en ambiente de ruido	81
3.2.2.	Promedio de latencia total	82
3.2.2.1.	USB	83
3.2.2.1.1.	Latencia de en Rhasspy	83
3.2.2.1.2.	Latencia de Home Assistant	83
3.2.2.1.3.	Latencia del dispositivo	83
3.2.2.2.	ReSpeaker 4-Mic Array	84
3.2.2.2.1.	Latencia de Rhasspy	84
3.2.2.2.2.	Latencia de Home Assistant	84
3.2.2.2.3.	Latencia del dispositivo	84
3.2.2.3.	Google Assistant	85
3.2.2.3.1.	Latencia de Google Asistant a Home Assistant	85
3.2.2.3.2.	Latencia de Home Assistant al dispositivo	85
3.2.2.3.3.	Latencia del dispositivo a Home Assistant	85
3.2.2.4.	Alexa Amazon	86
3.2.2.4.1.	Latencia de Alexa Amazon a Home Assistant	86
3.2.2.4.2.	Latencia de Home Assistant al dispositivo	86
3.2.2.4.3.	Latencia del dispositivo a Home Assistant	86
3.2.3.	Comparación de latencia entre los dispositivos evaluados	87
3.3.	CONCLUSIONES	88
3.4.	RECOMENDACIONES	89
<b>BIBLIOGRAFÍA</b>		<b>90</b>

## ÍNDICE DE TABLAS

Tabla 1	Comparación de intentos - USB	78
Tabla 2	Comparación de Intentos – ReSpeaker 4-Mic	79
Tabla 3	Comparación de Intentos - Alexa Amazon y Google Assistant	80
Tabla 4	Comparación de los micrófonos en ambiente de ruido	81
Tabla 5	Promedio latencia total - USB	83
Tabla 6	Promedio latencia USB – Reconocimiento de comando en Rhasspy	83
Tabla 7	Promedio latencia USB – Petición de Rhasspy a Home Assistant	83
Tabla 8	Promedio latencia USB - Petición de Home Assistant a Rhasspy	83
Tabla 9	Promedio latencia ReSpeaker 4-Mic Array	84
Tabla 10	Promedio latencia 4-Mic - Reconocimiento de comando en Rhasspy	84
Tabla 11	Promedio latencia 4-Mic - Petición de Rhasspy a Home Assistant	84

Tabla 12 Promedio latencia 4-Mic - Petición de Home Assistant al dispositivo	84
Tabla 13 Promedio latencia Google Assistant	85
Tabla 14 Promedio latencia Google Assistant - Petición de Home Assistant	85
Tabla 15 Promedio latencia Google Assistant - Petición de Home Assistant al dispositivo	85
Tabla 16 Promedio latencia Google Assistant - Confirmación del dispositivo a Home Assistant	85
Tabla 17 Promedio latencia Alexa Amazon	86
Tabla 18 Promedio latencia Alexa Amazon - Petición de Alexa a Home Assistant	86
Tabla 19 Promedio latencia Alexa Amazon - Petición de Home Assistant al dispositivo	86
Tabla 20 Promedio latencia Alexa Amazon - Confirmación del dispositivo a Home Assistant	86
Tabla 21 Comparación de latencia - Promedio total de los dispositivos	87

## ÍNDICE DE FIGURAS

Figura 1 Arquitectura del proyecto	17
Figura 2 Prototipo de Interfaz de control de Home Assistant	23
Figura 3 Prototipo de Interfaz de control de Rhasspy	23
Figura 4 Prototipo de Interfaz de control de Node-RED	24
Figura 5 Prototipo de Interfaz del plano del hogar 3D	24
Figura 6 Diseño plano 3D - Diseño del plano del hogar en Sweet Home 3D	25
Figura 7 Diseño plano del hogar 3D - Prototipo Final del hogar	25
Figura 8 Diseño plano del hogar 3D - Renderización del plano con luz On día/noche	26
Figura 9 Diseño plano del hogar 3D – Incorporación de imágenes en GIMP	26
Figura 10 Diseño plano del hogar 3D - Luz encendida	27
Figura 11 Diseño plano del hogar 3D - Exportación de imágenes procesadas	27
Figura 12 Instalación Home Assistant - Creación de máquina virtual	28
Figura 13 Instalación Home Assistant – Selección del disco duro	28
Figura 14 Instalación Home Assistant – Habilitar EFI en Configuración	29
Figura 15 Instalación Home Assistant – Selección del adaptador puente	29
Figura 16 Instalación Home Assistant – Credenciales de Home Assistant	30
Figura 17 Instalación Home Assistant – Configuración de Home Assistant	30
Figura 18 Instalación Home Assistant - Página de inicio de Home Assistant	31
Figura 19 Integración de dispositivos Nexxt - Registro del dispositivo en Smart Life	31
Figura 20 Integración dispositivos Nexxt – Pestaña dispositivos agregados en Smart Life	32
Figura 21 Integración dispositivos a Home Assistant - Pestaña de integración	32
Figura 22 Integración dispositivos a Home Assistant - Búsqueda de integración Tuya	33
Figura 23 Integración dispositivos a Home Assistant - Credenciales de Smart Life	33
<i>Figura 24 Integración dispositivos a Home Assistant - Dispositivos por entidades</i>	34
<i>Figura 25 Integración dispositivos a Home Assistant - Presentación de dispositivos</i>	34
<i>Figura 26 Integración dispositivos a Home Assistant - Integraciones</i>	35
<i>Figura 27 Integración dispositivos a Home Assistant - Creación de cuenta</i>	35
<i>Figura 28 Integración dispositivos a Home Assistant - Mensaje de confirmación</i>	35
<i>Figura 29 Integración dispositivos a Home Assistant - Iniciar sesión</i>	36
<i>Figura 30 Integración dispositivos a Home Assistant - Selección de Asistente de Google</i>	36
<i>Figura 31 Integración dispositivos a Home Assistant - Configurar dispositivos</i>	37
<i>Figura 32 Integración dispositivos a Home Assistant - Selección de Home Assistant Cloud</i>	37
<i>Figura 33 Integración dispositivos a Home Assistant – Ingreso de credenciales</i>	38

<i>Figura 34 Integración dispositivos a Home Assistant - Presentación de dispositivos</i>	38
<i>Figura 35 Integración Alexa – Integraciones</i>	39
<i>Figura 36 Integración Alexa – Ingreso de credenciales</i>	39
<i>Figura 37 Integración Alexa – Integraciones Home Assistant dentro de Amazon Alexa</i>	40
<i>Figura 38 Integración Alexa – Ingreso de Credenciales de Home Assistant Cloud</i>	40
<i>Figura 39 Integración dispositivos – Pantalla inicio</i>	41
<i>Figura 40 Preparación de tarjeta Micro SD 32gb - Plataforma BalenaEtcher</i>	41
<i>Figura 41 Preparación de tarjeta Micro SD 32gb – Selección del disco</i>	42
<i>Figura 42 Preparación de tarjeta Micro SD 32gb - Selección del objetivo</i>	42
<i>Figura 43 Preparación de tarjeta Micro SD 32gb - Finalización</i>	43
<i>Figura 44 Instalación de Raspbian - Selección de la ubicación</i>	43
<i>Figura 45 Instalación de Raspbian – Ingreso de contraseña</i>	44
<i>Figura 46 Instalación de ReSpeaker 4-Mic Array - Actualización de versión</i>	45
<i>Figura 47 Instalación de ReSpeaker 4-Mic Array - Instalación del servidor de git</i>	45
<i>Figura 48 Instalación de ReSpeaker 4-Mic Array - Creación de usuario de git</i>	45
<i>Figura 49 Instalación de ReSpeaker 4-Mic Array - Instalación de las tarjetas de voz seeed</i>	45
<i>Figura 50 Instalación de ReSpeaker 4-Mic Array - Ingreso a la carpeta seeed-voicecard</i>	46
<i>Figura 51 Instalación de ReSpeaker 4-Mic Array – Reinicio de raspberry pi</i>	46
<i>Figura 52 Instalación de ReSpeaker 4-Mic Array - Instalación correcta del dispositivo</i>	46
<i>Figura 53 Instalación de ReSpeaker 4-Mic Array - Funcionamiento correcto del dispositivo</i>	47
<i>Figura 54 Instalación de Rhasspy con Docker - Comando de instalación</i>	47
<i>Figura 55 Configuración del sistema Rhasspy - Pantalla de inicio de Rhasspy</i>	48
<i>Figura 56 Configuración del sistema Rhasspy - Configuración para acceso a dispositivos</i>	48
<i>Figura 57 Configuración del sistema Rhasspy - Descarga de perfil</i>	49
<i>Figura 58 Configuración del sistema Rhasspy - Reconocimiento de comandos</i>	49
<i>Figura 59 Configuración del sistema Rhasspy - Comandos de voz a reconocer</i>	50
<i>Figura 60 Configuración del sistema Rhasspy – Palabras a entrenar</i>	50
<i>Figura 61 Configuración del sistema Rhasspy - Selección del micrófono</i>	51
<i>Figura 62 Conexión rhasspy con hassio - Creación de token en Home Assistant</i>	51
<i>Figura 63 Conexión rhasspy con hassio - Conexión a Home Assistant desde Rhasspy</i>	52
<i>Figura 64 Conexión rhasspy con hassio - Código HTTP</i>	52
<i>Figura 65 Conexión rhasspy con hassio – Envío de peticiones desde Rhasspy a Home Assistant</i>	52
<i>Figura 66 Conexión rhasspy con hassio - Home Assistant receptando peticiones</i>	53
<i>Figura 67 Instalación de Samba - Búsqueda del Add-on</i>	53
<i>Figura 68 Instalación de Samba - Instalación del Add-on</i>	54
<i>Figura 69 Instalación de Samba - Configuración del Add-on</i>	54
<i>Figura 70 Instalación de Samba – Add-on ejecutado</i>	55
<i>Figura 71 Instalación de Samba - Ingreso a las carpetas de Home Assistant</i>	55
<i>Figura 72 Implementación modelo 3D - Selección de tarjeta “Elementos de Imagen”</i>	56
<i>Figura 73 Implementación modelo 3D - Creación de pestaña Floorplan</i>	56
<i>Figura 74 Implementación modelo 3D – Creación del código parte 1</i>	56
<i>Figura 75 Implementación modelo 3D – Creación del código parte 2</i>	57
<i>Figura 76 Implementación modelo 3D - Funcionamiento</i>	57
<i>Figura 77 Instalación Node-Red - Instalación del Add-on</i>	58
<i>Figura 78 Instalación Node-Red - Instalación del Add-on</i>	59
<i>Figura 79 Instalación Node-Red - Instalación del Add-on</i>	59
<i>Figura 80 Desarrollo aplicación - Selección de nodos</i>	60
<i>Figura 81 Desarrollo aplicación - Configuración de nodos</i>	60

Figura 82 Desarrollo aplicación - Configuración de nodos	61
Figura 83 Encendido de foco - Toma de tiempo de comando de voz de Rhasspy	62
Figura 84 Apagado de foco – Toma de tiempo de comando de voz de Rhasspy	62
Figura 85 Encendido de foco - Toma de tiempo de comando de voz de Rhasspy	63
Figura 86 Encendido de foco – Toma de tiempo de comando de voz de Rhasspy	63
Figura 87 Prueba de tiempo - Encendido de foco desde USB - Rhasspy a Hassio	64
Figura 88 Prueba de tiempo - Encendido de foco desde USB – Hassio a Dispositivo	65
Figura 89 Prueba de tiempo - Apagado de foco desde USB - Rhasspy a Hassio	65
Figura 90 Prueba de tiempo - Apagado de foco desde USB - Hassio a Dispositivo	66
Figura 91 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic - Rhasspy a Hassio	66
Figura 92 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic– Hassio a dispositivo	67
Figura 93 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic – Rhasspy a Hassio	67
Figura 94 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic – Hassio a dispositivo	68
Figura 95 Prueba de tiempo - Encendido de foco desde Alexa Amazon - Rhasspy a Hassio	68
Figura 96 Prueba de tiempo - Encendido de foco desde Alexa Amazon - Hassio a dispositivo	69
Figura 97 Prueba de tiempo - Apagado de foco desde Alexa Amazon – Rhasspy a Hassio	69
Figura 98 Prueba de tiempo - Apagado de foco desde Alexa Amazon – Hassio a dispositivo	70
Figura 99 Prueba de tiempo - Encendido de foco desde Google Assistant - Rhasspy a Hassio	70
Figura 100 Prueba de tiempo - Encendido de foco desde Google Assistant - Hassio a dispositivo	71
Figura 101 Prueba de tiempo - Apagado de foco desde Google Assistant – Rhasspy a Hassio	71
Figura 102 Prueba de tiempo - Apagado de foco desde Google Assistant – Hassio a dispositivo	72
Figura 103 Prueba de tiempo - Encendido de foco desde USB	73
Figura 104 Prueba de tiempo - Apagado de foco desde USB	73
Figura 105 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic Array	74
Figura 106 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic Array	74
Figura 107 Prueba de tiempo – Encendido de foco desde Alexa Amazon	75
Figura 108 Prueba de tiempo – Apagado de foco desde Alexa Amazon	75
Figura 109 Prueba de tiempo – Encendido de foco desde Google Assistant	76
Figura 110 Prueba de tiempo – Encendido de foco desde Google Assistant	76
Figura 111 Comparación de intentos - USB	78
Figura 112 Comparación de Intentos – ReSpeacker 4-Mic	79
Figura 113 Comparación de Intentos - Alexa Amazon y Google Assistant	80
Figura 114 Comparación de los micrófonos en ambiente de ruido	81

## INTRODUCCIÓN

Con el pasar de los años el ser humano se ha visto en la necesidad de optimizar tiempo en sus tareas cotidianas, es así como dentro del internet de las cosas (IoT) nace una de las aplicaciones más importantes, la domótica, que es la conversión de dispositivos como focos, swicht, entre otros, a dispositivos inteligentes, los cuales se gestionan dentro de una plataforma domótica donde envían y receptan datos que convierten en información para solucionar las actividades diarias; a esta aplicación se la conoce como casa inteligente o smart home [1].

La domótica es un aporte importante para la gestión del hogar por lo que reduce responsabilidades y esfuerzos en el usuario, ya que están siendo gestionados por el sistema domótico mediante el internet; aportando accesibilidad hacia los dispositivos sin necesidad del contacto físico con el mismo, haciendo la vida más fácil para los miembros del hogar además ayuda en la gestión del hogar a personas con impedimentos físicos [2].

Los dispositivos domóticos o inteligentes, son fabricados por distintas marcas y cada una cuenta con su propio sistema de gestión domótico lo que es una limitante para incorporar un dispositivo de una marca diferente; es ahí donde surge la necesidad de identificar una plataforma domótica open source, que facilite el proceso de la integración de dispositivos indistintamente del fabricante para facilitar la gestión u optimizar los dispositivos domóticos del hogar.

El presente trabajo de titulación describe el análisis de los tiempos de respuesta de las peticiones que existe entre la comunicación de los dispositivos domóticos para su monitoreo y control mediante el uso de la plataforma domótica Home Assistant; como receptor de los comandos de voz local se hace uso de la plataforma Rhasspy la misma que envía eventos a Home Assistant.

Este informe presenta la siguiente estructura:

En el **Capítulo 1**, describe el ámbito de aplicación del sistema domótico, los requerimientos para su implementación, los antecedentes y el enfoque del proyecto, igual de importante la justificación para la solución a la necesidad planteada.

El **Capítulo 2**, describe la definición del prototipo tecnológico utilizado para la gestión del sistema domóticos, la fundamentación teórica la cual dio las pautas necesarias para llegar a la solución, los objetivos, diseño y ejecución del prototipo.

El **Capítulo 3**, presenta el plan y los resultados de la evaluación, además describe las conclusiones y recomendaciones que surgieron al realizar la solución planteada.

## **1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS**

### **1.1. Ámbito de aplicación: descripción del contexto y hechos de interés**

El internet de las cosas (IoT) nace de la necesidad de optimizar tiempo en la gestión de distintos dispositivos, es decir pueden conectar dispositivos como focos, swicht, entre otros, mediante el uso del internet, los mismos que envían y receptan datos que convierten en información sin que el ser humano esté involucrado [3] [4].

Dentro del internet de las cosas existen varias aplicaciones una de las más importantes es la domótica que da la facilidad de gestionar y optimizar los dispositivos eléctricos dentro de la vivienda, brindando comodidad a sus habitantes aparte de ser amigable con el medio ambiente [5] [6].

Existen múltiples fabricantes de dispositivos domóticos los cuales cuentan con su propia plataforma para la gestión de sus dispositivos, haciendo que el usuario instale distintas aplicaciones para manejar los dispositivos de distintas marcas, es ahí donde aparecen las plataformas domóticas que integran los dispositivos indistintamente del fabricante y haciendo la vida del usuario más fácil [7].

Según [8] en la actualidad la domótica es una aplicación del IoT pensada en el confort de los miembros del hogar mediante la gestión de los dispositivos físicos que garantiza al usuario gestionarlos sin necesidad de grandes conocimientos tecnológicos, ya que se encuentran gestionados por el sistema domótico mediante el internet evitando responsabilidades y esfuerzos físicos al usuario.

El propósito de la siguiente propuesta tecnológica es la gestión de un sistema domótico que permita evaluar los tiempos de respuesta de las peticiones que

existe entre la comunicación de los dispositivos domóticos, mediante el uso de la plataforma domótica Home Assistant para el monitoreo y control del sistema domótico; también se hace uso de la plataforma Rhasspy como receptor de los comandos de voz que envía eventos a Home Assistant.

## **1.2. Establecimiento de Requerimientos**

Un sistema debe estar validado mediante la implementación de un prototipo, que muestre el ciclo de comunicación de los dispositivos dentro del sistema domótico, es decir la intervención de las plataformas domóticas y los dispositivos con el usuario, de esta manera se conoce el funcionamiento del sistema.

La implementación de un sistema domótico es a conveniencia del usuario ya que existen múltiples plataformas para diferentes sistemas operativos y distintas arquitecturas; en los sistemas domóticos se implementa dispositivos inteligentes de diferente marca que permite la comunicación entre ellos, todo esto sucede gracias a los protocolos de comunicación que tiene la plataforma.

La domótica, siendo una de las aplicaciones más importantes está enfocada en facilitar la vida del ser humano; optimizando múltiples tareas diarias del hogar y es ahí donde implementa un asistente de voz que realiza las peticiones que viajan desde una plataforma domótica al sistema domótico que tiene integrados los dispositivos para lo cual se deben realizar configuraciones que conectan los sistemas que intervendrán.

Un sistema domótico no se caracteriza sólo por ser eficiente en el desarrollo de las tareas que el usuario le asigne, si no, más bien debe ser eficaz, con esto se quiere decir que mientras más rápida y fluida sea la comunicación entre los dispositivos que conformen el sistema, mejor experiencia de uso va a tener el usuario, para esto se utilizan herramientas para la captura de envíos de peticiones y confirmaciones, existentes dentro del sistema domótico con el fin de saber la latencia existente entre la comunicación de los dispositivos.



### **1.3. Justificación del requerimiento a satisfacer**

La necesidad de implementar un sistema domótico con asistente de voz hace posible el diseño del prototipo que especifica las tecnologías a implementar de manera remota y por medio del internet, así mismo muestra el flujo de comunicación entre usuario – plataforma domótica – dispositivos inteligentes para una percepción visual del diseño del sistema.

Los dispositivos inteligentes al ser de distintos fabricantes crean la necesidad de integrarlos a un sistema domótico que cuente con protocolos de comunicación que permitan su integración, haciendo fácil el trabajo de monitoreo y control, a su vez dentro de la plataforma se puede visualizar el tiempo de respuesta a los comandos de voz enviados; gracias al acceso a internet el usuario controla el sistema dentro y fuera de su red local [9].

El propósito de la siguiente propuesta tecnológica es la gestión de un sistema domótico local mediante comandos de voz, que permita al usuario la integración de los dispositivos inteligentes indistintamente del fabricante, mediante la plataforma domótica Home Assistant instalado en virtualbox se podrá controlar y monitorear los dispositivos dentro y fuera de la red local, así mismo se podrá analizar el tiempo de respuesta en las peticiones realizadas al servidor; también se hace uso de la plataforma Rhasspy instalada en una raspberry como alternativa local del sistema de voz Alexa o Google Home, donde se establece comandos de voz para el manejo de los dispositivos, los mismos que son enviados a Home Assistant.

## **2. CAPÍTULO II. DESARROLLO DEL PROYECTO**

### **2.1. Definición del prototipo tecnológico**

Todo sistema de gestión, recibe datos que se convertirán en información para la toma de decisiones; así mismo son los sistemas domóticos, recolectan datos mediante los dispositivos inteligentes que se convierten en información y la almacenan para realizar las acciones que previamente se configuran.

En la figura 1 se muestra la solución a la propuesta tecnológica de la gestión de sistemas domóticos locales mediante comandos de voz, haciendo uso de

Home Assistant y Rhasspy que permiten al usuario controlar e integrar dispositivos inteligentes.

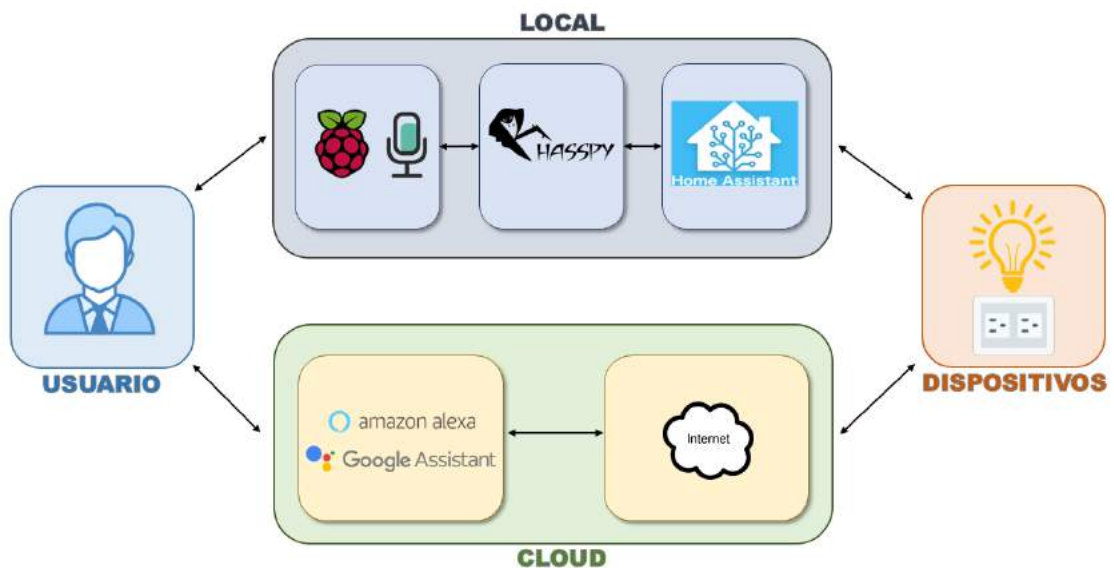


Figura 1 Arquitectura del proyecto

Fuente: Elaboración Propia

La solución a la propuesta tecnológica es de manera local y en la cloud:

- El funcionamiento local se realiza mediante Rhasspy instalado en un contenedor dentro de Raspberry, también se instalará el micrófono para escuchar los comandos a configurar, desde Rhasspy se enviaron los comandos al sistemas domóticos Home Assistant instalado en VirtualBox donde se administra los dispositivos inteligentes desde una computadora.
- El funcionamiento haciendo uso de la cloud, es un proceso de integración de los dispositivos inteligentes en Alexa y Google Assistant; se ha considerado dicha integración para comparaciones del funcionamiento.

## 2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO

### 2.2.1. Internet de las cosas

En los últimos años la tecnología avanzado de manera muy rápida, desafiando al ser humano al desarrollo de múltiples aplicaciones y dispositivos que faciliten las actividades diarias; es así cómo aparece el término de Internet de las cosas que consiste en el manejo de dispositivos conectados a una red, ya sea local o en la nube, dichos dispositivos

interactúan entre sí mediante protocolos de comunicación, los mismos que se deben configurar de manera correcta y a conveniencia del usuario, garantizando que cada una de las tareas que previamente asignó el usuario ya no tendrá necesidad de hacerlo de manera física ya que el sistema lo hará en el tiempo requerido [9].

Se define al internet de las cosas como una estructura diseñada por el usuario para recibir órdenes o a su vez establecer horarios remotos que cumplan las tareas, esta red permite la interacción sistemas embebidos y los dispositivos inteligentes los mismos que cuentan con una dirección ip para su reconocimiento [10] [11].

Según [12] smart home es una ventana al mundo virtual; convirtiendo en dispositivos inteligentes a los dispositivos no inteligentes, los mismos que envían datos para ser analizados por la red o en la nube, permitiendo al usuario gestionar las tareas a través del internet.

#### **2.2.1.1. Dispositivos IOT**

Estos dispositivos permiten que la vida diaria del ser humano sea más sencilla; haciendo que los dispositivos sean inteligentes enviando datos a sistemas que convierten esos datos en información que el usuario pueda manejar. Al mencionar el término dispositivo IoT hace referencia a un foco, un interruptor, entre otros, los mismos que sirven para gestionar las tareas de una casa o también dispositivos que permiten el manejo de un automóvil sin intervención física, es decir existen muchos dispositivos inteligentes; en el mercado se encuentran de diversas marcas y valor económico, ya depende del usuarios cuál de ellos se ajusta a sus necesidades [13] [14] [15].

Según [16] [17] los dispositivos inteligentes que engloban el término de Internet de las Cosas cuentan con sistemas de gestión haciendo que el usuario programe sus tareas de manera dinámica, ordenada y fácil; todos estos dispositivos tienen que estar conectados a la misma red mediante una dirección ip asignada por el router y de esta manera conectarse al internet, es así cómo se envían datos a procesar y se convierte en información.

### **2.2.2. Domótica**

El apagado, encendido y regulación de dispositivos que hemos realizado por años de manera manual es un mecanismo que está quedando obsoleto ya que no es amigable con el medio ambiente, es así como en el mundo de la tecnología aparece el concepto de domótica facilitando las tareas diarias del ser humano fin de automatizar una vivienda, un edificio o un negocio; trabajando en conjunto con Internet de las Cosas (IoT), garantizando que el usuario pueda gestionar el uso de los dispositivos, la energía y la comunicación entre el mundo virtual y el mundo real [18].

#### **2.2.2.1. Protocolos de comunicación**

Los dispositivos inteligentes deben estar conectados a la red para ser gestionados por un sistema a través del usuario; se necesita de un mecanismo de interacción entre ellos; es decir, se hace uso de un protocolo de comunicación que le permita receptor datos y envíe información de las tareas a realizar [19].

##### **2.2.2.1.1. SSH**

El protocolo de comunicación ssh es un mecanismo remoto de protección de los datos; los usuarios que no cuenten con los permisos necesarios no podrán hacer uso de la información que se comparte; por lo cual se establece dentro de la red de internet una comunicación directa y segura [20].

Según [21] es un protocolo cliente-servidor que está obligado a realizar una conexión cifrada que evita el robo de información; es decir, el cliente envía una carpeta de datos con una clave y únicamente el servidor destinado tendrá la clave para abrir la carpeta y hacer uso de los datos.

#### **2.2.2.2. Plataformas Domóticas**

Los dispositivos inteligentes son fabricados por múltiples empresas las mismas que cuentan con su propia plataforma domótica para ser manejados dentro ellas, es decir si se pretende implementar dispositivos de otras marcas causa dificultades de comunicación entre ellos; por lo que es recomendable que el usuario utilice plataformas domóticas open source, ya que estas facilitan el intercambio de información entre sí [22].

#### **2.2.2.2.1. Home Assistant**

Según [23] Home Assistant es un sistema domótico open source que automatiza y controla las tareas del hogar, está diseñado para hacer más fácil la vida del ser humano ya que dentro de la plataforma se puede programar el encendido y apagado de los dispositivos inteligentes que se incorpore dentro de ella.

Home Assistant es una plataforma que se instala en una máquina virtual o en una Raspberry Pi, siendo un supervisor central que ayuda al control y monitoreo de los dispositivos desde una interfaz móvil compatible con la plataforma; el usuario puede programar el encendido de las luces cuando una persona llega al hogar, en un determinado horario, regular la intensidad de las luces, entre otras tareas [24].

#### **2.2.2.2.2. Smart Life**

Según [25] fue desarrollada por la compañía Tuya y es open source, su interfaz gráfica es muy similar a Tuya; para la integración de dispositivos inteligentes debe crear una cuenta e ingresarlos de manera individual; una vez almacenados los dispositivos se observa si ID y su clave para el posterior ingreso a una plataforma domótica.

#### **2.2.2.2.3. Rhasspy**

Según [26] es un asistente de voz local que funciona sin estar conectado a internet; garantizado al usuario la privacidad de su información, convirtiéndolo en el software más completo ya que su arquitectura está basado en la arquitectura de Jasper.

#### **2.2.2.3. Dispositivos domóticos o inteligentes**

Según [27] dentro de IoT existe el concepto de manejar dispositivos electrónicos y que conecten entre sí, es así como nace la idea de crear dispositivos inteligentes que intercambien datos entre ellos mediante señal wifi o por cable para cumplir una función.

Los dispositivos domóticos o inteligentes al enviar la información mediante wifi o cable, se pueden controlar dentro y fuera de una red local ya que gracias al internet se almacenan en varios servicios de la web [28].

#### **2.2.2.3.1. Nexxt**

Según [29] Nexxt es una marca que se caracteriza en la fabricación de múltiples dispositivos inteligentes de malla Wi-Fi para la automatización del hogar, también cuenta con una plataforma domótica para integrar dichos dispositivos domóticos, los mismos que son almacenados en la nube.

#### **2.2.2.3.2. Raspberry**

Funciona perfectamente como una computadora ya que puede conectar un mouse, teclado y a una pantalla de tv o un monitor; también cuenta con sistema operativo para ser usada y todo esto a un costo accesible para el usuario, normalmente suele ser tomado como ambiente de pruebas o para gestionar sistemas domóticos [30].

En la actualidad la microcomputadora Raspberry pi está siendo utilizada para múltiples proyectos en un ambiente de pruebas por su bajo costo y adaptabilidad, contando con un software aceptable e intuitivo para el usuario; uno de los usos frecuentes es como servidor de video y/o audio en tiempo real [31] [32].

#### **2.2.2.3.3. ReSpeaker**

Según [33] ReSpeaker proporciona dispositivos que permiten al usuario construir sistemas de voz de acuerdo a las configuraciones que se desarrollen, permitiendo interactuar con los distintos dispositivos inteligentes que integremos a proyectos de inteligencia artificial o aplicaciones de voz ya que es una placa de expansión que cuenta con cuatro micrófonos incorporados.

Este tipo de dispositivos independientemente de la marca pueden ser integrados en una placa Raspberry Pi o en Arduino, según los requerimientos que soliciten cada placa; gracias a los beneficios que brinda el internet de las cosas marcando tendencia en aplicaciones del hogar [34].

#### **2.2.2.3.4. Echo Dot**

Echo Dot es un parlante que al conectarse con Alexa se convierte en un sistema de voz inteligente que mediante sus funciones denominadas skills

realiza múltiples tareas como reproducir música, controlar dispositivos IoT, responder preguntas entre otras. Puede ser integrado a plataformas domóticas para la gestión de dispositivos del hogar por comandos de voz a una distancia considerablemente de 12 metros [35].

## **2.3. OBJETIVOS DEL PROTOTIPO**

### **2.3.1. Objetivo General**

Implementar sistemas de gestión domótico local mediante el control de los dispositivos IoT por comandos de voz, para el análisis de latencia en el intercambio de información de los dispositivos en la plataforma domótica.

### **2.3.2. Objetivos Específicos**

- Diseñar el prototipo del sistema de gestión domótico para la selección de las plataformas domóticas.
- Instalar y ejecutar la plataforma domótica Home Assistant, para la gestión e integración dispositivos inteligentes.
- Instalar y ejecutar la plataforma domótica Rhasspy, para reconocimiento de comandos de voz.
- Implementación de herramientas para la toma de latencia en el envío y la confirmación de peticiones.

## **2.4. DISEÑO DEL PROTOTIPO**

Para el diseño del prototipo del sistema de gestión se hace uso de la plataforma domótica Home Assistant instalada en una máquina virtual, usando el programa VirtualBox de Oracle; en la cual se ha incorporado los dispositivos inteligentes como focos y switches desarrollados por la empresa Nexxt; donde también muestra el estado actual de los dispositivos, si están encendidos o apagados, como se muestra en la figura 2.

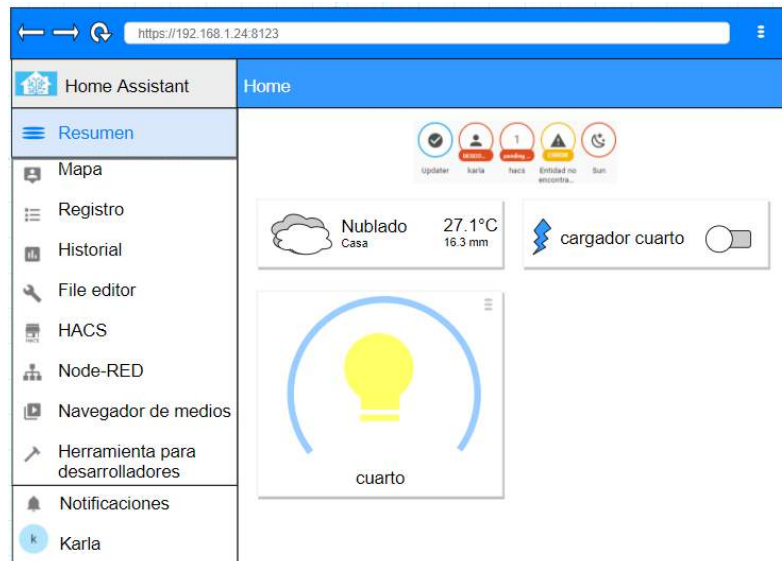


Figura 2 Prototipo de Interfaz de control de Home Assistant

Fuente: Elaboración Propia

En el prototipo que se muestra en la figura 3, utiliza la plataforma domótica Rhasspy para el reconocimiento de los comandos de voz mediante micrófono ReSpeaker 4-Mic Array; donde se enviaran las peticiones directamente a Home Assistant mediante comandos HTTP.

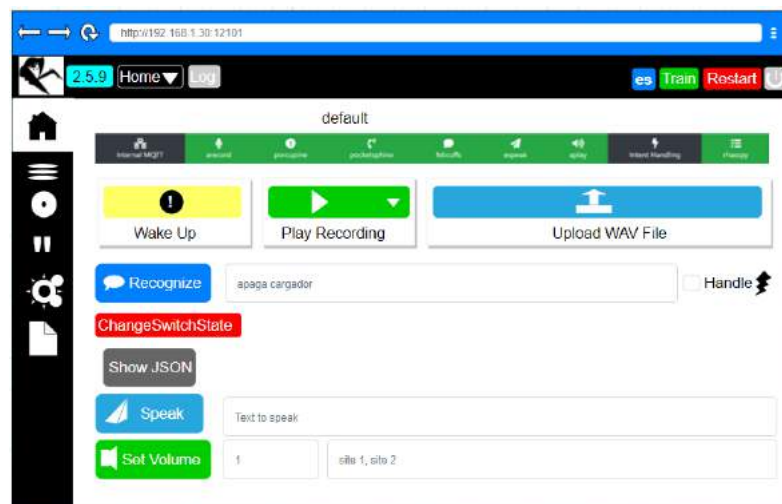


Figura 3 Prototipo de Interfaz de control de Rhasspy

Fuente: Elaboración Propia

Se implementó Node Red dentro de Home Assistant para conocer el tiempo de respuesta que se tiene en los dispositivos domóticos, de esta manera se



identifica los tiempos de peticiones que se emplean, así como se visualiza en la figura 4:

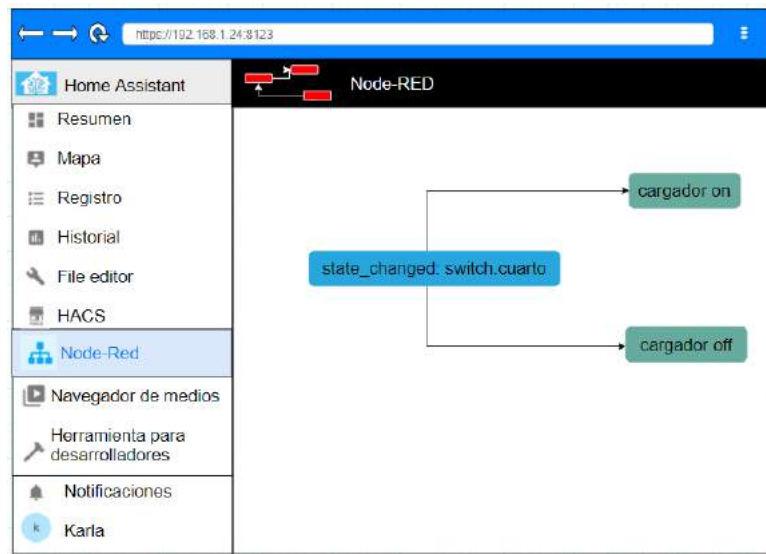


Figura 4 Prototipo de Interfaz de control de Node-RED

Fuente: Elaboración Propia

En la figura 5 se muestra el prototipo de gestión domótica se ha considerado conveniente implementar un diseño en 3D del hogar, el mismo que está diseñado en Sweet Home 3D y GIMP, donde permite visualizar los dispositivos encendidos y apagados.

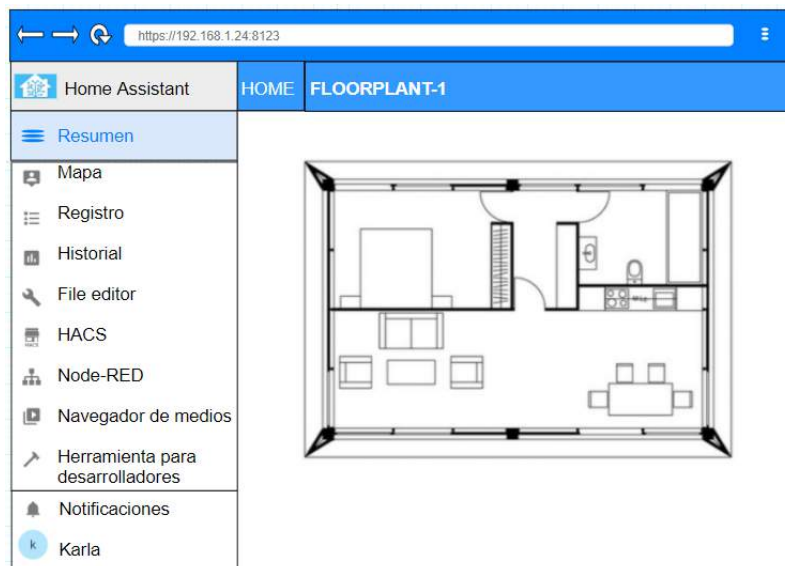


Figura 5 Prototipo de Interfaz del plano del hogar 3D

Fuente: Elaboración Propia

### 2.4.1. Diseño del plano del hogar en Sweet Home 3D y GIMP

En la figura 6 se visualiza el sistema domótico implementado en un modelo 3D del hogar, incorporando objetos del hogar todo esto es posible gracias a la herramienta de modelado Sweet Home 3D.

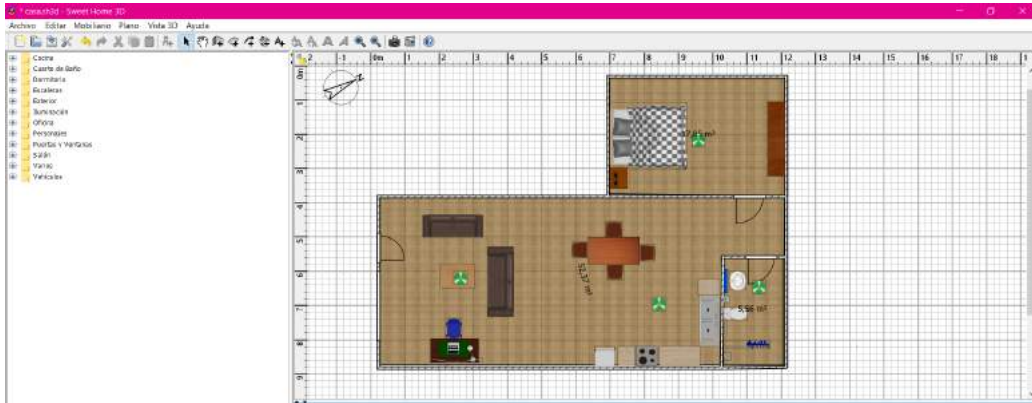


Figura 6 Diseño plano 3D - Diseño del plano del hogar en Sweet Home 3D

*Fuente: Elaboración Propia*

En la figura 7 se observa el modelo 3D del hogar, en modo de ejecución donde se ha incorporado objetos del hogar que hacen más real el comportamiento gracias a la herramienta de modelado Sweet Home 3D.

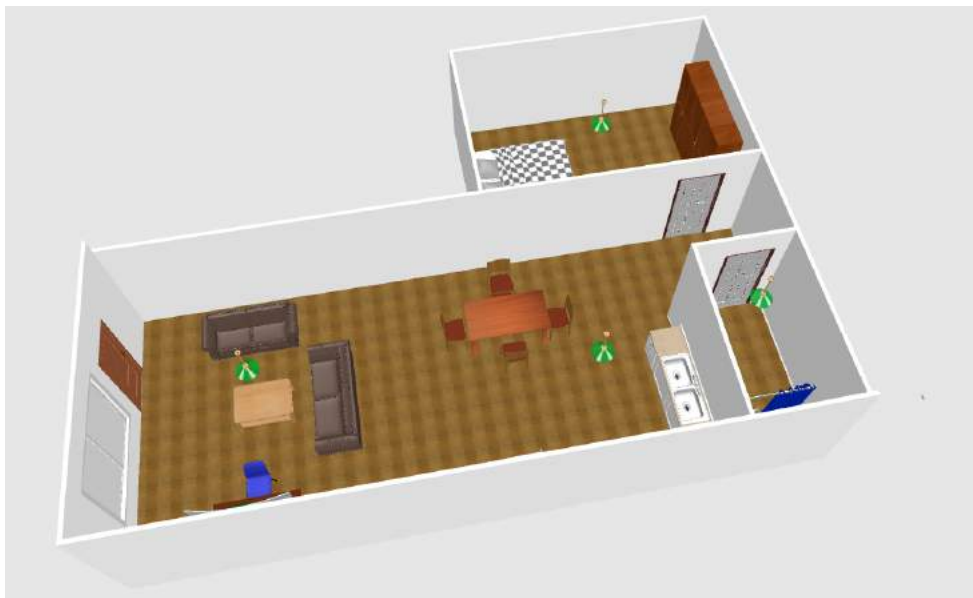


Figura 7 Diseño plano del hogar 3D - Prototipo Final del hogar

*Fuente: Elaboración Propia*

Se debe realizar la renderización en horarios tanto para el día y la noche en Sweet Home 3D, para el proceso de encendido de luces en el sistema domótico Home Assistant, como se visualiza en la figura 8.

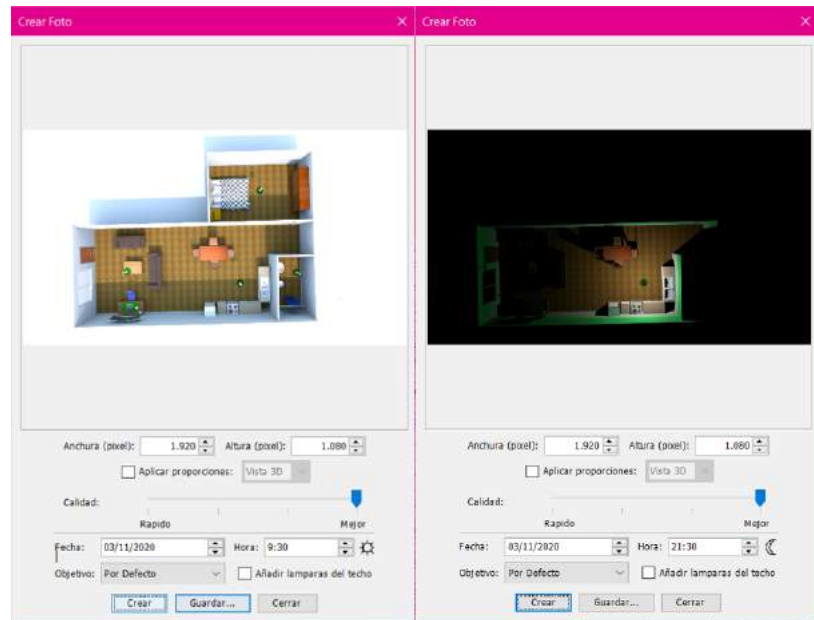


Figura 8 Diseño plano del hogar 3D - Renderización del plano con luz On día/noche

Fuente: Elaboración Propia

Como se muestra en la figura 9, la imagen renderizada en plano 3D en un estado de apagado, posterior ir realizando los estados de encendido de los dispositivos.

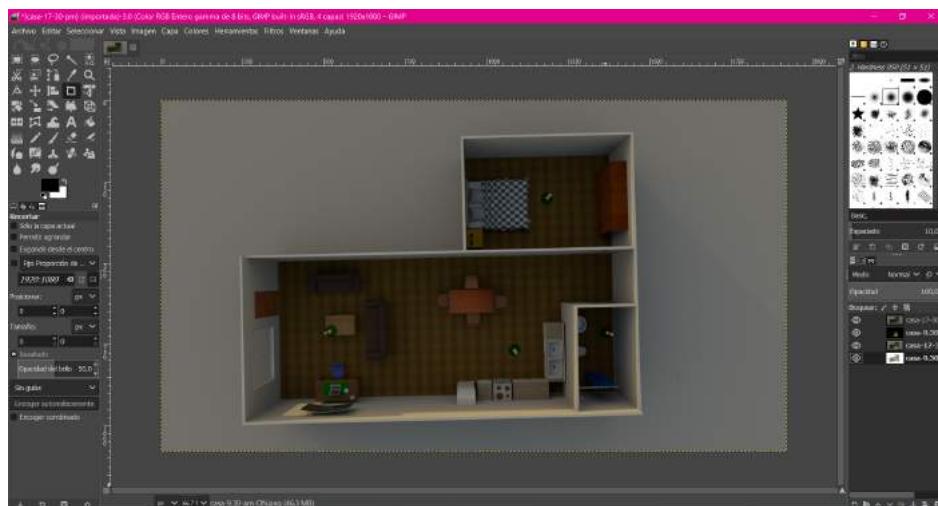


Figura 9 Diseño plano del hogar 3D – Incorporación de imágenes en GIMP

Fuente: Elaboración Propia

En la figura 10 se observa el proceso de selección del área donde estará el dispositivo que realiza la acción de encendido y apagado de las luces, para transportarlas a Home Assistant mediante la herramienta de edición GIMP.

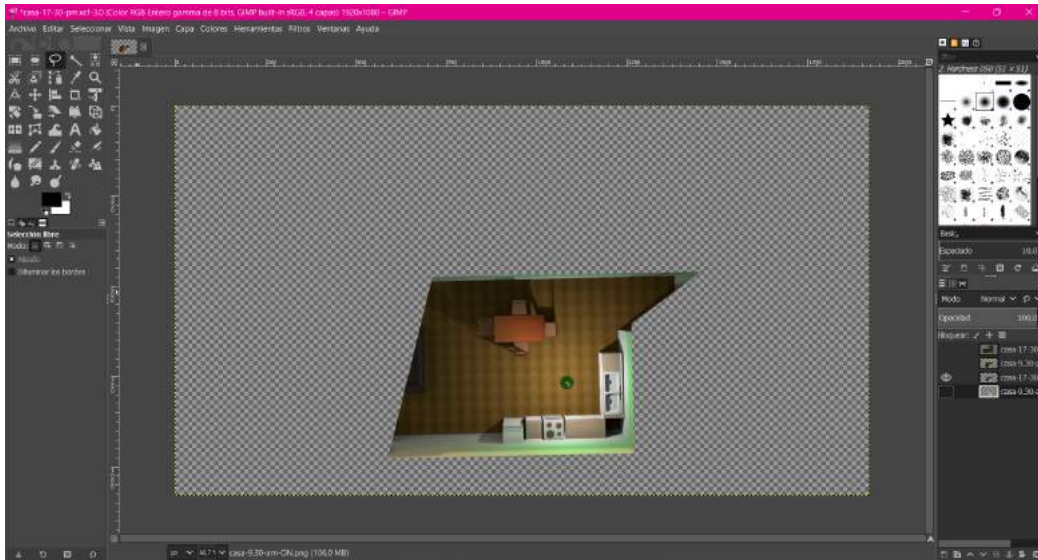


Figura 10 Diseño plano del hogar 3D - Luz encendida

Fuente: Elaboración Propia

Las imágenes se exportan una a una para obtener los distintos estados de encendido y apagado como se muestra en la Figura 11.

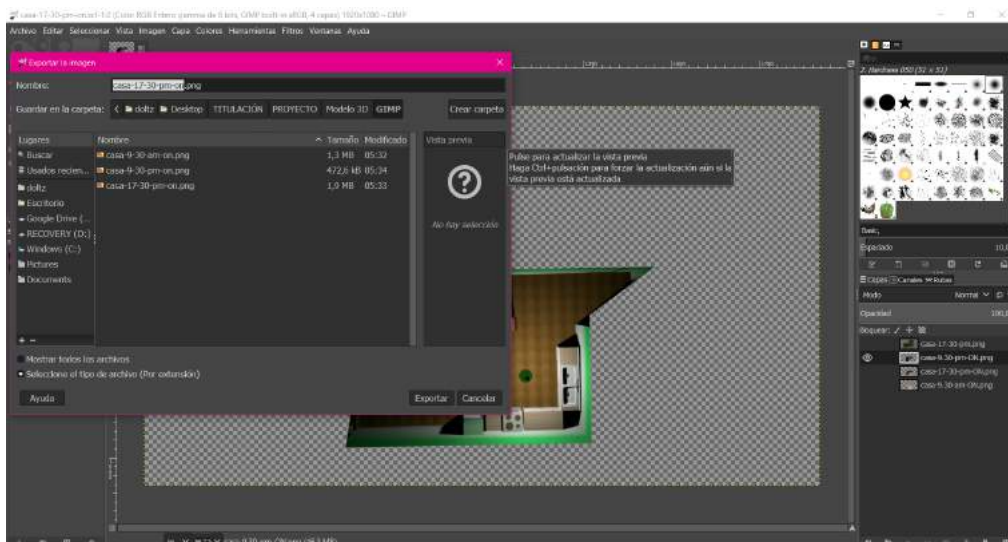


Figura 11 Diseño plano del hogar 3D - Exportación de imágenes procesadas

Fuente: Elaboración Propia

## 2.5. EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO

### 2.5.1. Instalación de Home Assistant en VirtualBox

La instalación de Home Assistant se puede realizar en una Raspberry o en una máquina virtual de preferencia por el usuario. Para la presente instalación se hizo uso de VirtualBox.

Debe ingresar a la página oficial de Home Assistant, donde podrá verificar los requisitos para la instalación del archivo VDI, el mismo que se descarga de la página.

Como primer paso en el VirtualBox se crea nuestra máquina virtual estableciendo nombre, la carpeta destino, el sistema operativo y la versión como se puede ver en la Figura 12.

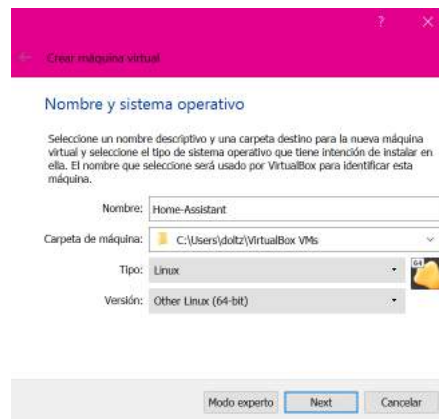


Figura 12 Instalación Home Assistant - Creación de máquina virtual

Fuente: Elaboración Propia

Una vez creada la máquina virtual, selecciona el archivo VDI, para empezar la instalación de la plataforma domótica a utilizar en el presente trabajo de titulación como se muestra en la Figura 13.

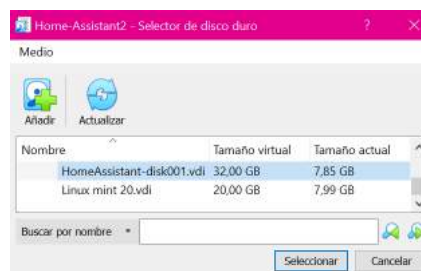


Figura 13 Instalación Home Assistant – Selección del disco duro

Fuente: Elaboración Propia

En la Figura 14 se muestra la instalación del archivo VDI y ejecución de Home Assistant, se configura las siguientes opciones:

- Memoria base: 2048MB
- Pestaña Sistema:
  - Características extendidas: Habilitar EFI (solo SO especiales)
- Pestaña Red: Adaptador Puesto (Figura 15).

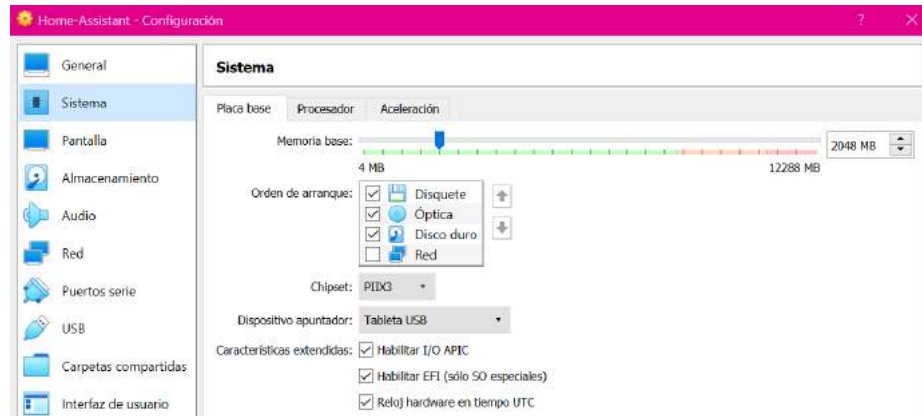


Figura 14 Instalación Home Assistant – Habilitar EFI en Configuración

*Fuente: Elaboración Propia*




Figura 15 Instalación Home Assistant – Selección del adaptador puente

*Fuente: Elaboración Propia*

Las configuraciones realizadas sirven para que el sistema doméstico instalado funcione correctamente en ejecución. Ingresamos al navegador y como ruta usamos la dirección ip que el sistema asigna de manera automática a Home Assistant y el puerto correspondiente para el sistema, ejemplo 192.198.1.24:8123



Como se muestra en la Figura 16, dentro del sistema le aparece la siguiente ventana donde se crea una cuenta para el acceso a la plataforma domótica.

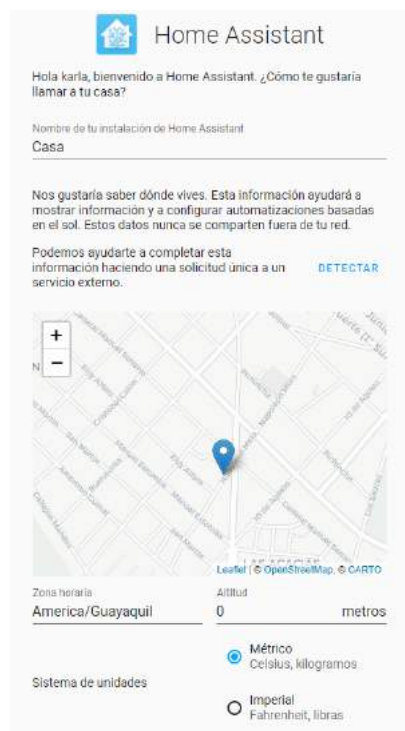


The screenshot shows the 'Home Assistant' account creation interface. At the top, there is a blue house icon with a plus sign and the text 'Home Assistant'. Below this, a message asks if the user is ready to set up their home and join a community of thinkers. The main section is titled 'Comencemos creando una cuenta de usuario.' and contains four input fields: 'Nombre', 'Nombre de usuario', 'Contraseña', and 'Confirmar contraseña'. A blue button labeled 'CREAR UNA CUENTA' is positioned below the fields. At the bottom, there is a link that reads 'Alternativamente, puedes restaurar desde una copia de seguridad anterior.'

Figura 16 Instalación Home Assistant – Credenciales de Home Assistant

Fuente: Elaboración Propia

Ya creada la cuenta debe seleccionar la ubicación, zona horaria y sistema de unidades como se ve en la Figura 17.



The screenshot shows the 'Home Assistant' configuration screen. At the top, there is a blue house icon with a plus sign and the text 'Home Assistant'. Below this, a message says 'Hola karla, bienvenido a Home Assistant. ¿Cómo te gustaría llamar a tu casa?'. The 'Nombre de tu instalación de Home Assistant' field is filled with 'Casa'. A message explains that location information is used for automation and is not shared. A 'DETECTAR' button is available. Below this is a map showing a location in Guayaquil, Ecuador. The 'Zona horaria' is set to 'America/Guayaquil' and 'Altitud' is '0 metros'. The 'Sistema de unidades' is set to 'Métrico' (Celsius, kilograms) with radio buttons for 'Imperial' (Fahrenheit, pounds) also visible.

Figura 17 Instalación Home Assistant – Configuración de Home Assistant

Fuente: Elaboración Propia

La Figura 18 muestra la pantalla principal de Home Assistant donde se visualiza las entidades que se incorporan para la automatización.

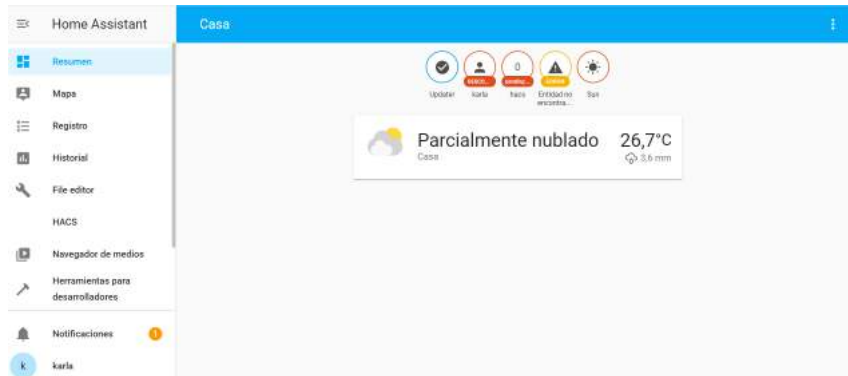


Figura 18 Instalación Home Assistant - Página de inicio de Home Assistant

*Fuente: Elaboración Propia*

## 2.5.2. Integración de dispositivos IoT a Home Assistant

Para la integración de los dispositivos IoT primero se los debe registrar en la aplicación Smart Life en Android; una vez realizado este paso se ingresa a la plataforma domótica Home Assistant, en el apartado de configuración está la opción de integraciones donde se incorpora cada dispositivo de manera individual.

### 2.5.2.1. Integración de dispositivos inteligente Nexxt

En la Figura 19 y 20 se observa el registro de los dispositivos domóticos de la marca Nexxt es sencillo y prácticamente intuitivo por ejemplo para la integración de un foco selecciona iluminación.





Figura 19 Integración de dispositivos Nextt - Registro del dispositivo en Smart Life

Fuente: Elaboración Propia



Figura 20 Integración dispositivos Nextt – Pestaña dispositivos agregados en Smart Life

Fuente: Elaboración Propia

### 2.5.2.2. Integración de dispositivos IoT a Home Assistant

Dentro de Home Assistant como muestra la Figura 21, en el menú de configuración está la opción de integraciones donde seleccionamos el botón de añadir integración.

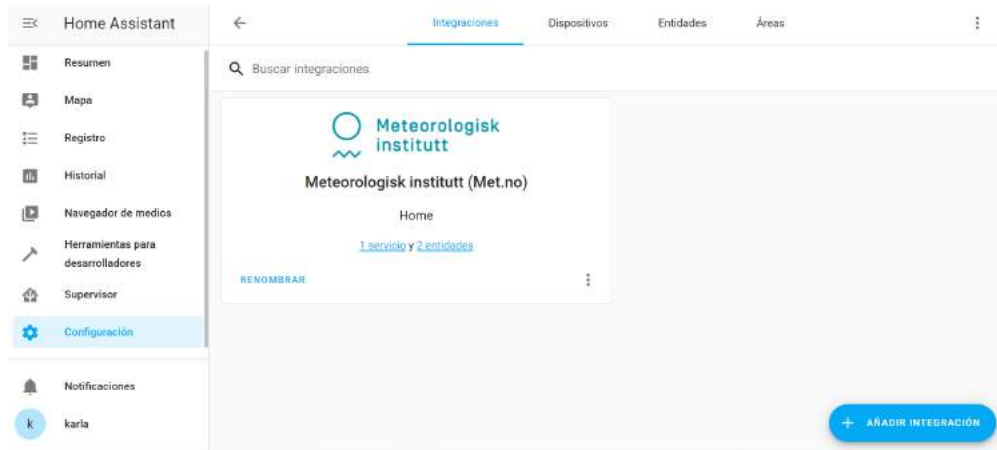


Figura 21 Integración dispositivos a Home Assistant - Pestaña de integración

Fuente: Elaboración Propia

Buscamos la integración tuya así como se muestra en la Figura 22.

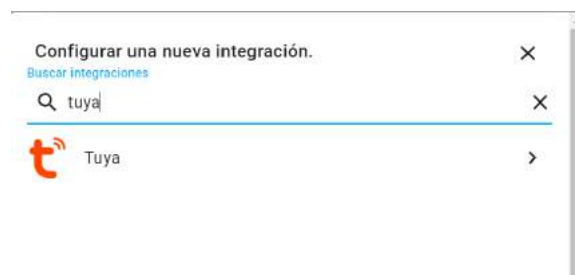


Figura 22 Integración dispositivos a Home Assistant - Búsqueda de integración Tuya

Fuente: Elaboración Propia

En la Figura 23 muestra el ingreso de las mismas credenciales de la aplicación Android Smart Life para establecer la conexión.

**Tuya** X

Introduce tu credencial Tuya.

Usuario  
 karlacueva94@gmail.com

Contraseña  
 |..... 👁

Código de país de tu cuenta (por ejemplo, 1 para USA o 86 p...  
 593

La aplicación en la cual registraste tu cuenta  
 Smart Life ▼

**ENVIAR**

Figura 23 Integración dispositivos a Home Assistant - Credenciales de Smart Life

Fuente: Elaboración Propia

Dentro de Home Assistant como se ve en la Figura 24 ingresamos al apartado resumen donde añadiremos los dispositivos que tenemos en la aplicación; de esta manera ya tenemos los dispositivos integrados como se visualiza en la Figura 25.

¿Qué tarjeta te gustaría añadir a tu vista "Home" ?

POR TARJETA
**POR ENTIDAD**

🔍 Buscar entidades:

<input type="checkbox"/>	↑ Entidad
<input checked="" type="checkbox"/>	⚡ cargador cuarto switch.eb8b4ff6670ac8bebzmy
<input type="checkbox"/>	🏠 Casa zone.home
<input type="checkbox"/>	☁ Casa weather.casa
<input type="checkbox"/>	🔔 config entry discovery persistent_notification.config_entry_discovery
<input checked="" type="checkbox"/>	💡 cuarto light.520487802462ab32bf88
<input type="checkbox"/>	📡 hacs sensor.hacs

**CANCELAR CONTINUAR**

Figura 24 Integración dispositivos a Home Assistant - Dispositivos por entidades

Fuente: Elaboración Propia

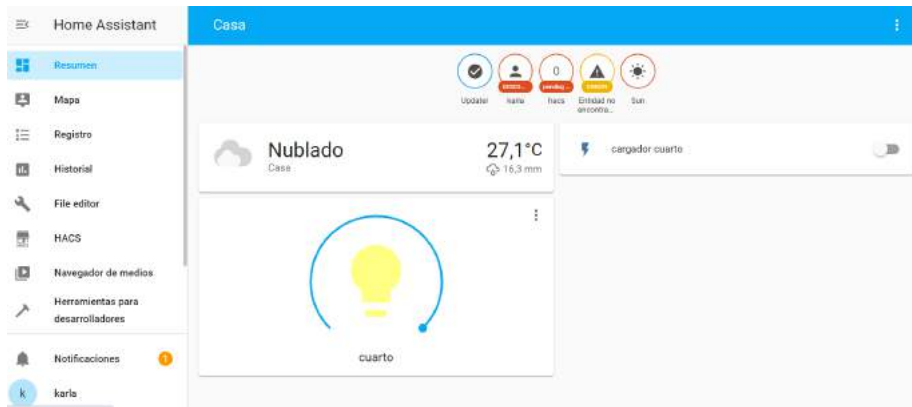


Figura 25 Integración dispositivos a Home Assistant - Presentación de dispositivos

Fuente: Elaboración Propia

### 2.5.2.3. Integración de Home Assistant Cloud

En el apartado de configuración como se ve en la Figura 26 seleccionamos la opción Home Assistant Cloud donde creamos una cuenta con el correo personal y una contraseña como se puede observar en la Figura 27.



Figura 26 Integración dispositivos a Home Assistant - Integraciones

Fuente: Elaboración Propia

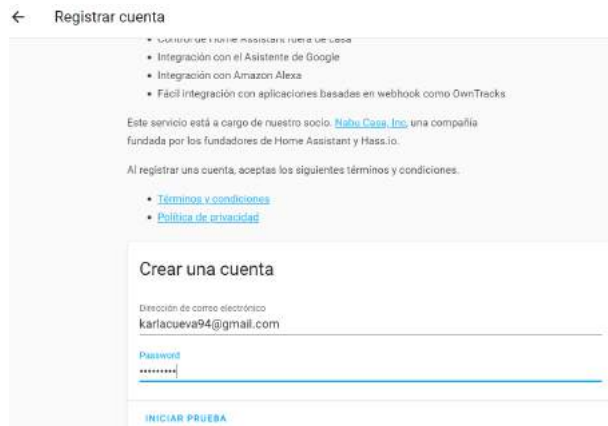


Figura 27 Integración dispositivos a Home Assistant - Creación de cuenta

Fuente: Elaboración Propia

Como se visualiza en la Figura 28, una vez creada la cuenta debemos recibir un correo de confirmación para poder hacer uso del servicio.

## Confirm your email address on Nabu Casa

Hello! We want to verify that [karlacueva94@gmail.com](mailto:karlacueva94@gmail.com) is your email address, and then you can get started with Home Assistant Cloud, powered by Nabu Casa.

[Confirm Email](#)

### Didn't request this email?

Don't worry. Someone may have entered your address by mistake. You can ignore or delete this email, nothing further will happen.

Figura 28 Integración dispositivos a Home Assistant - Mensaje de confirmación

Fuente: Elaboración Propia

Ingresamos a la cuenta creada donde ingresamos las credenciales como se muestra en la Figura 29, habilitamos la opción de Asistente de Google, luego se debe hacer la configuración en el Smartphone como se observa en la Figura 30.

## Home Assistant Cloud

fundada por los fundadores de Home Assistant y Hass.io.

Home Assistant Cloud es un servicio de suscripción con una prueba gratuita de un mes. No se necesita información de pago.

[Aprende más sobre Home Assistant Cloud](#)

### Inicia sesión

Correo electrónico  
karlacueva94@gmail.com

Contraseña  
.....

[INICIA SESIÓN](#) [¿Se te olvidó tu contraseña?](#)

Figura 29 Integración dispositivos a Home Assistant - Iniciar sesión

Fuente: Elaboración Propia

## Asistente de Google

Con la integración del Asistente de Google para Home Assistant Cloud, podrás controlar todos tus dispositivos Home Assistant a través de cualquier dispositivo habilitado para Asistente de Google.

- [Activa la skill de Home Assistant para el Asistente de Google](#)
- [Documentación de configuración](#)

### Activar informes de estado

Si activas los informes de estado, Home Assistant enviará todos los cambios de estado de las entidades expuestas a Google. Esto te permite ver siempre los últimos estados en la aplicación de Google.

### Dispositivos de seguridad

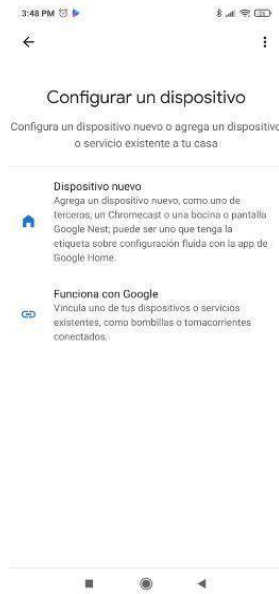
Por favor, introduce un pin para interactuar con los dispositivos de seguridad. Los dispositivos de seguridad son puertas, puertas de garaje y cerraduras. Se te pedirá que digas/introduzcas este pin cuando interactúes con dichos dispositivos a través del Asistente de Google.

Pin de dispositivos de seguridad

Figura 30 Integración dispositivos a Home Assistant - Selección de Asistente de Google

Fuente: Elaboración Propia

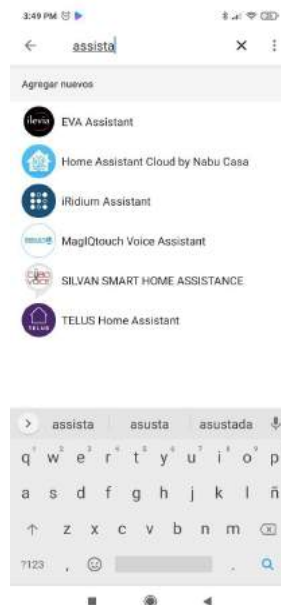
En el Smartphone como se visualiza en la Figura 31 se ingresa a la aplicación del asistente de google para realizar las configuraciones pertinentes, aparecerá un menú “Agregar y administrar” donde damos clic en la opción “Configurar dispositivo” y seleccionamos “Funciona con Google”.



*Figura 31 Integración dispositivos a Home Assistant - Configurar dispositivos*

*Fuente: Elaboración Propia*

Como se observa en la figura 32 en la pantalla se busca “Home Assistant Cloud by Nabu Casa”, luego ingresamos con las credenciales y aparecerán los dispositivos domóticos que se han ingresado en la plataforma Home Assistant.



*Figura 32 Integración dispositivos a Home Assistant - Selección de Home Assistant Cloud*

*Fuente: Elaboración Propia*

Ingresar las credenciales para autenticar como se muestra en la Figura 33, realizamos el inicio de sesión en Home Assistant Cloud, de esta manera se incorporan los dispositivos.

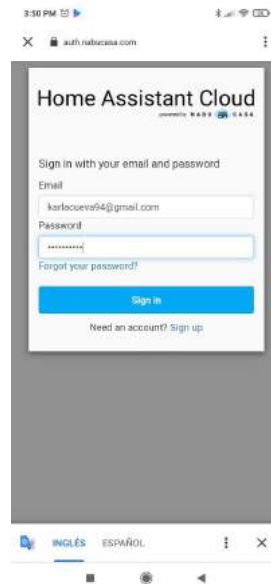


Figura 33 Integración dispositivos a Home Assistant – Ingreso de credenciales

*Fuente: Elaboración Propia*

Como se muestra en la Figura 34 una vez ingresadas las credenciales los dispositivos se integran de manera automática para ser manejados por medio de comandos de voz del Google Assistant.



Figura 34 Integración dispositivos a Home Assistant - Presentación de dispositivos

*Fuente: Elaboración Propia*



#### 2.5.2.4. Integración de Amazon Alexa

Home Assistant cuenta con un apartado configuración donde se debe activar la opción de “Alexa” de la misma manera que se realizó la integración de Google Assistant como se puede observar en la Figura 35.

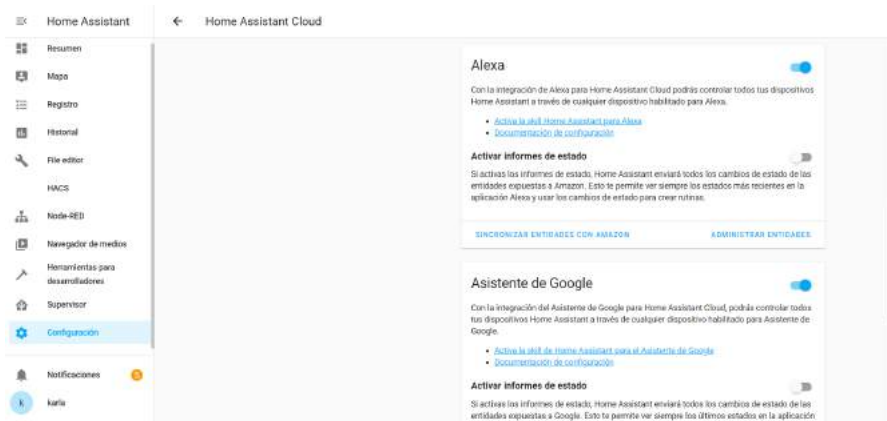


Figura 35 Integración Alexa – Integraciones

Fuente: Elaboración Propia

En la Figura 36 se observa el ingreso de las credenciales del perfil creado en la aplicación Amazon Alexa del sistema operativo Android, para el control desde el Home Assistant.

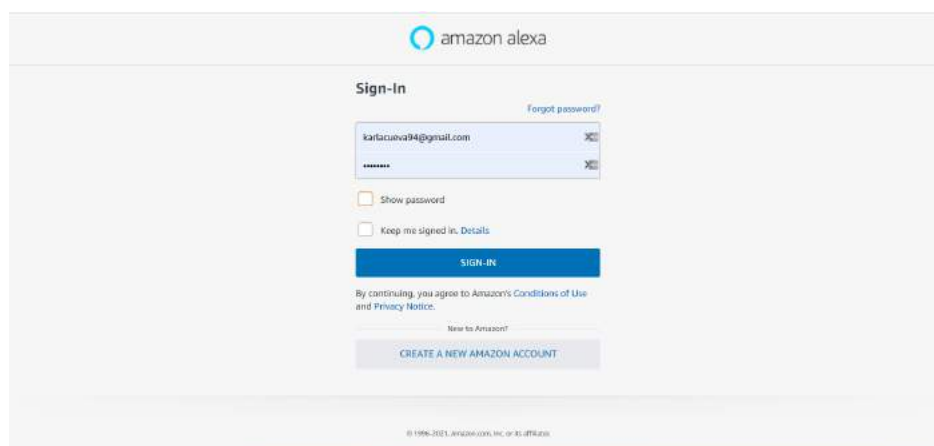


Figura 36 Integración Alexa – Ingreso de credenciales

Fuente: Elaboración Propia

En la aplicación de Amazon Alexa que se encuentra en el Smartphone hay un apartado llamado “Skills y juegos”; al ingresar se busca Home Assistant para integrarlo como se visualiza en la Figura 37.

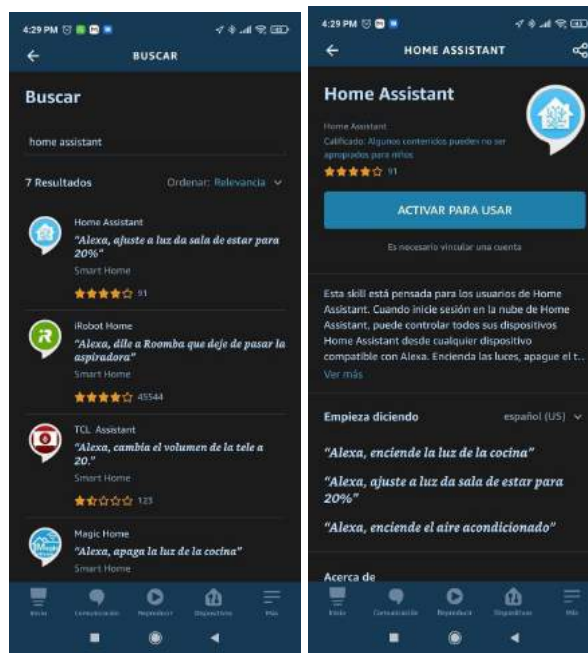


Figura 37 Integración Alexa – Integraciones Home Assistant dentro de Amazon Alexa

Fuente: Elaboración Propia

En la Figura 39 solicita el ingreso de las credenciales del Home Assistant Cloud; de esta manera se tiene el acceso de los dispositivos por medio de Amazon Alexa.



Figura 38 Integración Alexa – Ingreso de Credenciales de Home Assistant Cloud

Fuente: Elaboración Propia

### 2.5.3. Pantalla de inicio de Home Assistant con dispositivos integrados

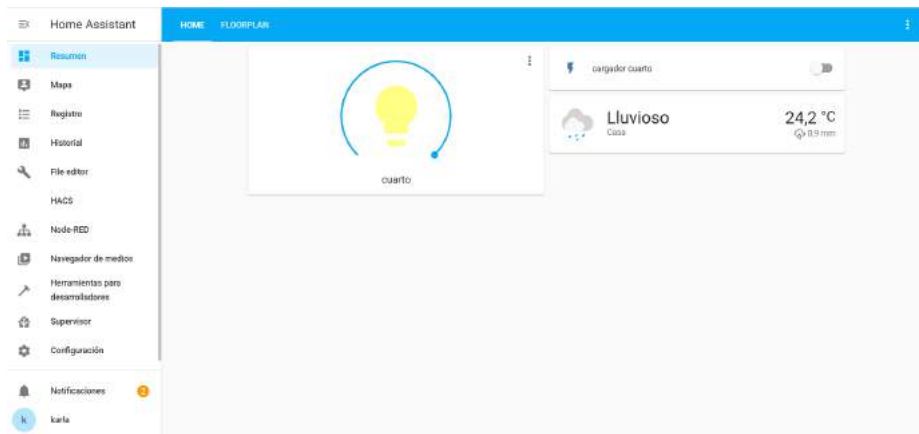


Figura 39 Integración dispositivos – Pantalla inicio

Fuente: Elaboración Propia

### 2.5.4. Instalación de Rhasspy

El proceso de la instalación se ha considerado realizarlo en una Raspberry Pi por su arquitectura; es necesario ya que dentro de esta plataforma se realiza el reconocimiento de los comandos de voz local que son enviados a Home Assistant para realizar las peticiones a los dispositivos.

#### 2.5.4.1. Preparación de tarjeta Micro SD 32gb

En la Figura 40 empieza el primer paso se debe preparar la tarjeta Micro SD 32gb, haciendo uso de la plataforma BalenaEtcher para grabar la imagen del sistema operativo como se muestra en la Figura 41, el cual se descarga de la página oficial de raspberry pi.

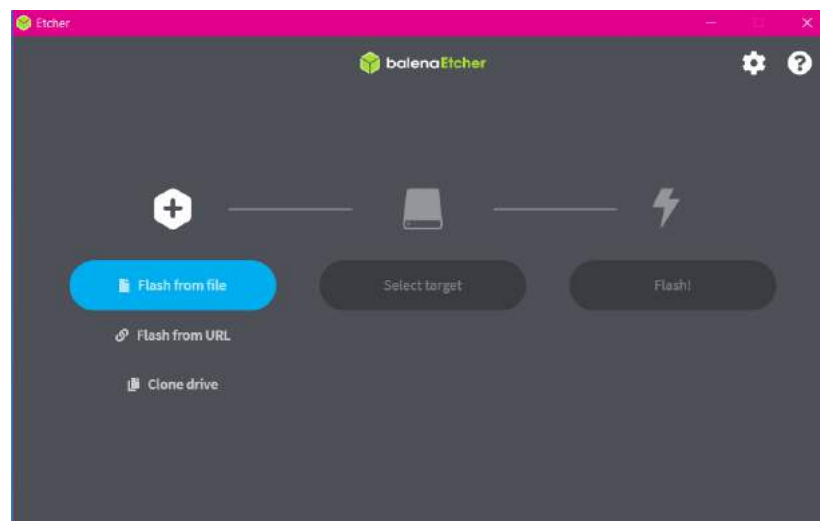


Figura 40 Preparación de tarjeta Micro SD 32gb - Plataforma BalenaEtcher

*Fuente: Elaboración Propia*

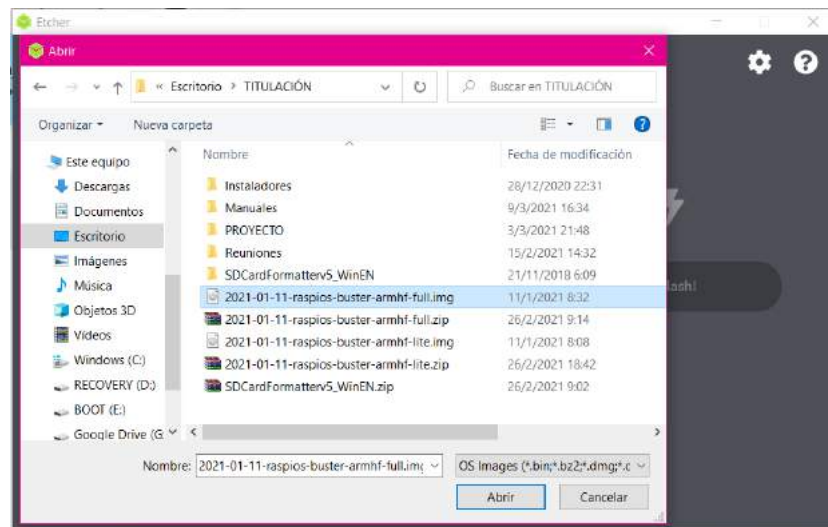


Figura 41 Preparación de tarjeta Micro SD 32gb – Selección del disco

*Fuente: Elaboración Propia*

Una vez seleccionada la imagen ISO del sistema operativo Raspbian como en la Figura 42, se guarda los archivos de configuración para la instalación de Raspbian y realiza el último paso de finalización de la instalación así como se observa en la Figura 43.

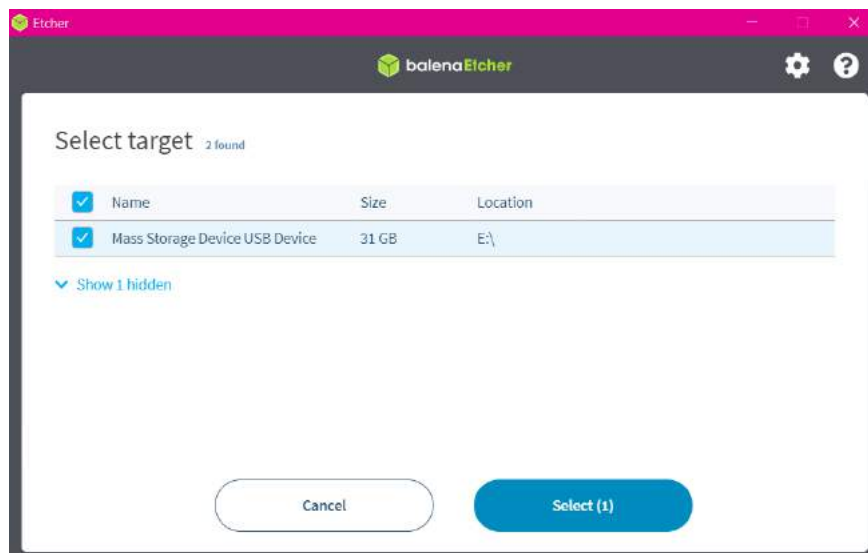


Figura 42 Preparación de tarjeta Micro SD 32gb - Selección del objetivo

*Fuente: Elaboración Propia*

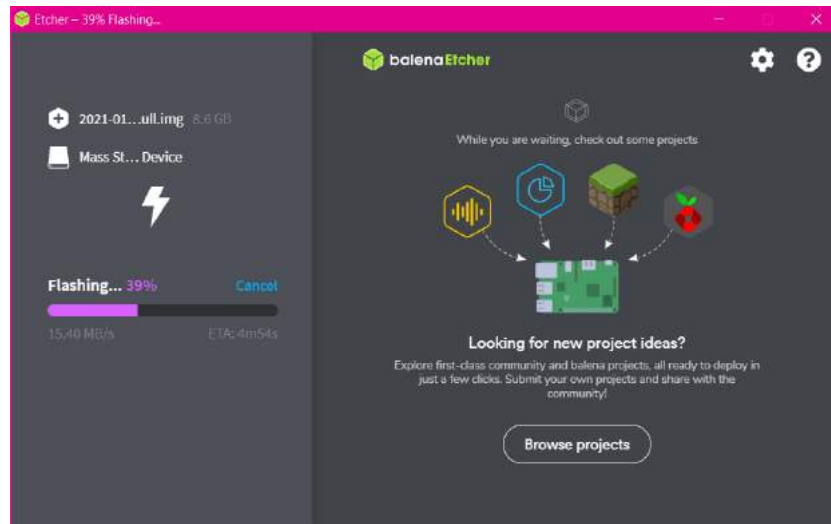


Figura 43 Preparación de tarjeta Micro SD 32gb - Finalización

Fuente: Elaboración Propia

La tarjeta raspberry pi necesita el siguiente archivo para su funcionamiento:

- **ssh:** mediante este protocolo de autenticación la raspberry accede al internet de manera remota.

#### 2.5.4.2. Instalación de Raspbian

Raspberry pi es considerada una computadora de bajo presupuesto por lo que necesita dispositivos de entrada (mouse y teclado) y dispositivo de salida (monitor) para la instalación de Raspbian en modo gráfico.

Como se observa en la Figura 44 la instalación es como cualquier sistema operativo, seleccionar país, idioma, zona horaria.

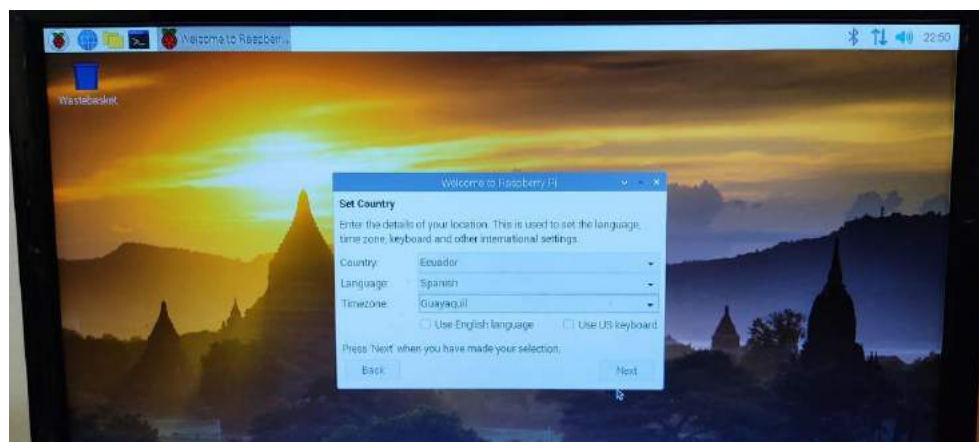
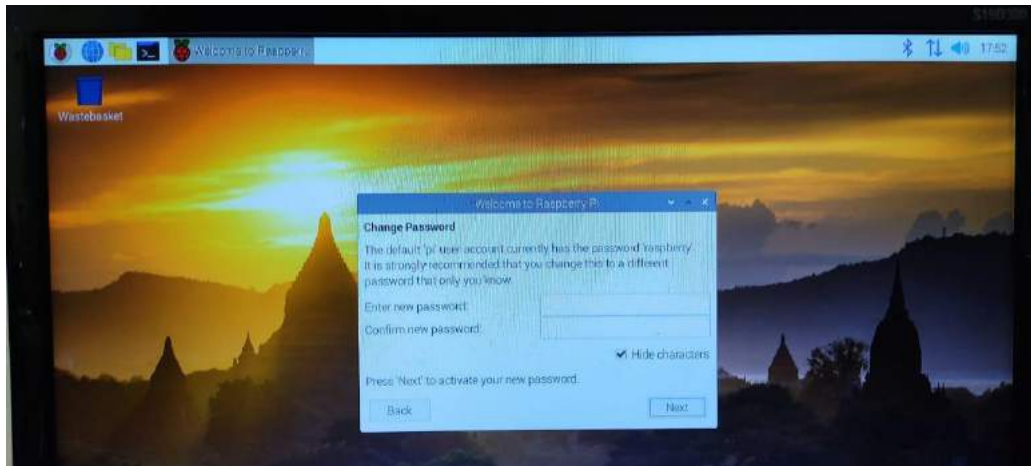


Figura 44 Instalación de Raspbian - Selección de la ubicación

Fuente: Elaboración Propia

A continuación se ingresa una contraseña la cual debe confirmar para ser validada así como se observa en la Figura 45. Una vez instalado se procede a colocar los micrófonos a la Raspberry pi para instalar los archivos que permiten el uso de los mismos



*Figura 45 Instalación de Raspbian – Ingreso de contraseña*

*Fuente: Elaboración Propia*

#### **2.5.4.3. Instalación de ReSpeaker 4-Mic Array**

En el desarrollo de presente proyecto se usó dos micrófonos para la realización del proyecto los cuales se describen a continuación:

- **ReSpeaker 4-Mic Array:** es una placa desarrollada para proyectos de inteligencia artificial y aplicaciones de voz porque es muy conveniente aplicarlo en sistemas de domótica controlados por comandos de voz; como su nombre lo dice consta de cuatro micrófonos para su manejo en raspberry pi es necesario instalar complementos.
- **Mini microphone:** dispositivo para aplicaciones de voz que se puede incorporar en una raspberry pi sin necesidad de instalar complementos.

Es conveniente aclarar que Raspberry pi hace uso de los comandos de Linux para las configuraciones mediante consola.

Antes de realizar una instalación en estos sistemas operativos se verifica que tenga la última versión del sistema operativo Raspbian para lo cual se ejecuta el siguiente código como se observa en la Figura 46:

```

pi@raspberrypi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

```

Figura 46 Instalación de ReSpeaker 4-Mic Array - Actualización de versión

Fuente: *Elaboración Propia*

En la Figura 47 se puede observar la instalación del servidor de git que ayuda al almacenamiento de la configuración del proyecto.

```

pi@raspberrypi:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man libcurl3-gnutls liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn

```

Figura 47 Instalación de ReSpeaker 4-Mic Array - Instalación del servidor de git

Fuente: *Elaboración Propia*

Desde el terminal de comandos como se visualiza en la Figura 48 se configura el proyecto para lo cual necesita de un usuario que se crea con el siguiente comando:

```

pi@raspberrypi:~$ sudo adduser git
Adding user 'git' ...
Adding new group 'git' (1001) ...
Adding new user 'git' (1001) with group 'git' ...
Creating home directory '/home/git' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for git
Enter the new value, or press ENTER for the default
  Full Name []: git
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y

```

Figura 48 Instalación de ReSpeaker 4-Mic Array - Creación de usuario de git

Fuente: *Elaboración Propia*

Las tarjetas de voz seeed cuentan con un código fuente que lo ayuda a su funcionamiento por lo que se usa el siguiente comando para obtenerlo en la raspberry pi como en la Figura 49.

```

pi@raspberrypi:~$ git clone https://github.com/respeaker/seeed-voicecard.git
Cloning into 'seeed-voicecard'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 982 (delta 15), reused 23 (delta 10), pack-reused 950
Receiving objects: 100% (982/982), 1.39 MiB | 62.00 KiB/s, done.
Resolving deltas: 100% (619/619), done.

```

Figura 49 Instalación de ReSpeaker 4-Mic Array - Instalación de las tarjetas de voz seeed

Fuente: *Elaboración Propia*



Luego de descargar el código fuente ingresa a la carpeta seeed-voicecard como se observa en la Figura 50 para realizar la instalación de los complementos para la tarjeta de voz; como se muestra en la Figura 50:

```
pi@raspberrypi:~ $ cd seeed-voicecard
pi@raspberrypi:~/seeed-voicecard $ sudo ./install.sh

### will compile with the latest kernel...

### Install required tool packages
Get:1 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Hit:3 http://archive.raspberrypi.org/debian buster InRelease
Hit:4 https://download.docker.com/linux/raspbian buster InRelease
Get:5 http://packages.microsoft.com/repos/code stable/main armhf Packages [10.8 kB]
Get:6 http://packages.microsoft.com/repos/code stable/main amd64 Packages [10.4 kB]
Get:7 http://packages.microsoft.com/repos/code stable/main arm64 Packages [10.9 kB]
Fetched 42.5 kB in 6s (7,357 B/s)
Reading package lists... 9%
```

Figura 50 Instalación de ReSpeaker 4-Mic Array - Ingreso a la carpeta seeed-voicecard

Fuente: Elaboración Propia

Cuando se realiza una instalación es recomendable reiniciar raspberry pi para su correcta compilación y que ejecute sin inconvenientes; lo que llamaremos una instalación limpia como en la Figura 51.

```
pi@raspberrypi:~/seeed-voicecard $ sudo reboot
pi@raspberrypi:~/seeed-voicecard $ Connection to 192.168.1.28 closed by remote host.
Connection to 192.168.1.28 closed.
```

Figura 51 Instalación de ReSpeaker 4-Mic Array – Reinicio de raspberry pi

Fuente: Elaboración Propia

En la Figura 52 se visualiza por medio de los siguientes comandos la comprobación de que los complementos del dispositivo se han instalado correctamente.

```
pi@raspberrypi:~ $ cd seeed-voicecard
pi@raspberrypi:~/seeed-voicecard $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835-i2s-wm8960-hifi wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~/seeed-voicecard $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835-i2s-wm8960-hifi wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
Subdevices: 0/1
  Subdevice #0: subdevice #0
```

Figura 52 Instalación de ReSpeaker 4-Mic Array - Instalación correcta del dispositivo

Fuente: Elaboración Propia



Dentro de la misma consola se verifica que el dispositivo de audio funciona correctamente mediante el sistema de audio Audacity como se muestra en la Figura 53.

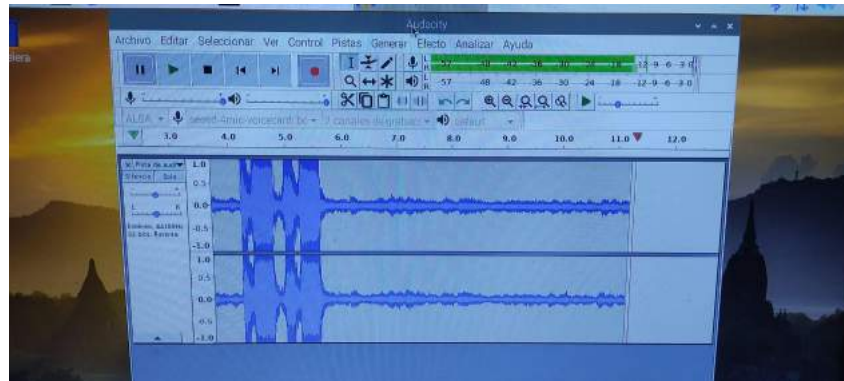


Figura 53 Instalación de ReSpeaker 4-Mic Array - Funcionamiento correcto del dispositivo

Fuente: Elaboración Propia

#### 2.5.4.4. Instalación de Rhasspy con Docker

Una de las maneras más simple para la instalación de Rhasspy es haciendo uso de Docker, se realizan los siguientes pasos:

Verificar que se encuentre docker instalado y que se encuentre el usuario, luego en segundo plano se debe se inicia la imagen del sistema Rhasspy como en la Figura 54.

```
pi@raspberrypi:~$ docker run -d -p 12101:12101 \
> --name rhasspy \
> --restart unless-stopped \
> -v "$HOME/.config/rhasspy/profiles:/profiles" \
9c8b7a02ad90: Downloading [>] 3.221MB/171.5MB
> --device /dev/snd:/dev/snd \
> rhasspy/rhasspy \
> --user-profiles /profiles \
> --profile en
Unable to find image 'rhasspy/rhasspy:latest' locally
9c8b7a02ad90: Downloading [>] 1.603MB/171.5MB
```

Figura 54 Instalación de Rhasspy con Docker - Comando de instalación

Fuente: Elaboración Propia

#### 2.5.4.5. Configuración del sistema Rhasspy

Para verificar la correcta instalación accede a la interfaz web de Rhasspy mediante la dirección ip seguido del puerto 12101; se observa en la Figura 55 la pantalla principal donde se realizan las configuraciones para nuestro sistema doméstico.

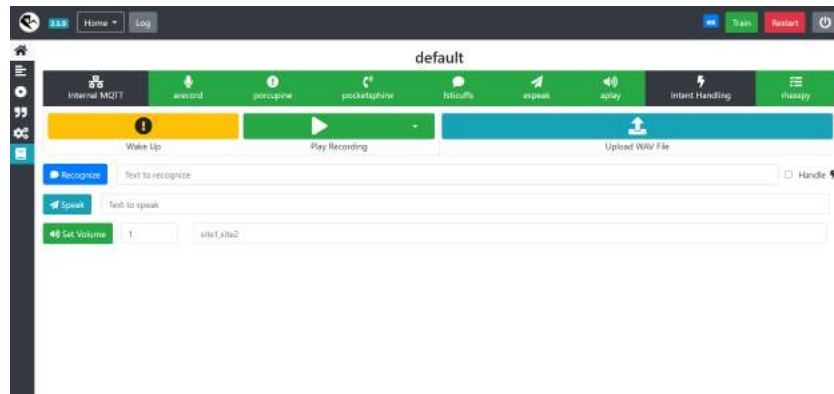


Figura 55 Configuración del sistema Rhasspy - Pantalla de inicio de Rhasspy

Fuente: Elaboración Propia

Se configura los parámetros de la pestaña “Settings” de acuerdo a las necesidades del proyecto. En este caso se realiza el reconocimiento de comandos mediante voz mediante los micrófonos antes mencionados por lo que debe de quedar como se observa en la Figura 56, cabe recalcar que estas configuraciones están acorde a los micrófonos usados y se basó en la documentación oficial de Rhasspy.

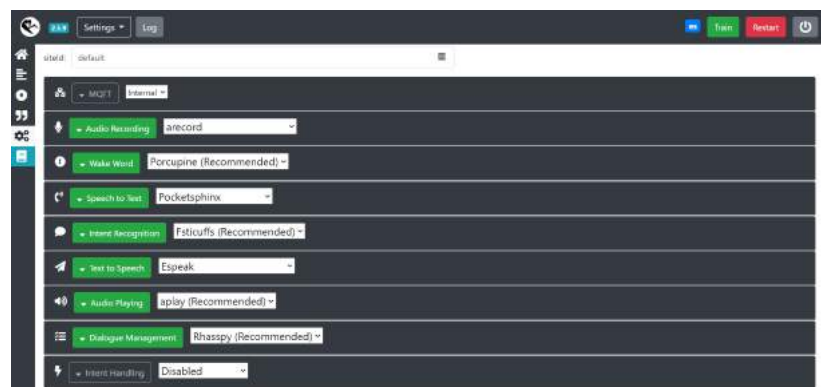


Figura 56 Configuración del sistema Rhasspy - Configuración para acceso a dispositivos

Fuente: Elaboración Propia

Una vez realizada la configuración pide la descarga del perfil, es importante realizarlo ya que descarga los últimos archivos de Rhasspy para continuar con el desarrollo de la práctica como en la Figura 57.

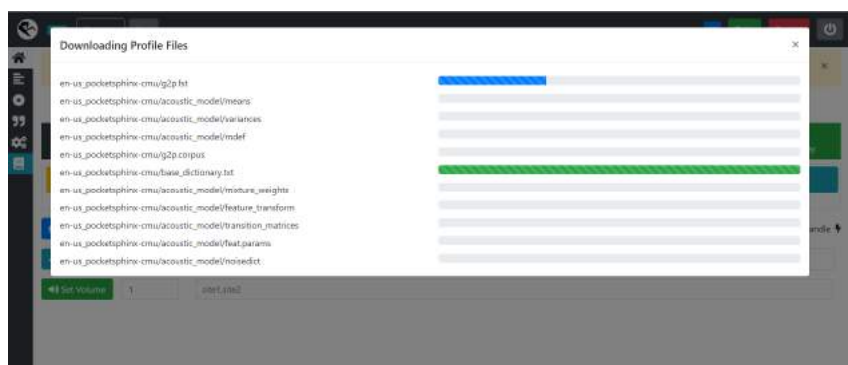


Figura 57 Configuración del sistema Rhasspy - Descarga de perfil

Fuente: Elaboración Propia

La Figura 58 muestra el correcto reconocimiento del comando con el que se ha entrenado el perfil; al reconocer el comando Rhasspy muestra la transcripción y un archivo Json que da aviso del reconocimiento del comando.

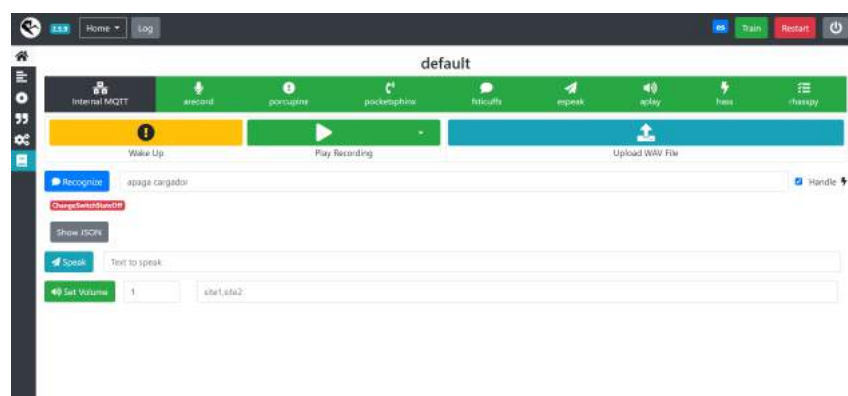


Figura 58 Configuración del sistema Rhasspy - Reconocimiento de comandos

Fuente: Elaboración Propia

Como todo sistema de reconocimiento se debe entrenar con los comandos a usar, Rhasspy tiene un apartado “Sentences” donde se necesita almacenar de manera escrita los comandos de voz los mismos que se agrupan por intensidad como en la Figura 59.

Los comandos de Rhasspy se describen a continuación:

- **ChangeLightState:** es usado para iluminación de los focos y después de State se agrega “On” u “Off” independientemente de la acción a realizar.

- **ChangeSwitchState:** sirve para dispositivos como interruptores, de igual manera después de State se agrega “On” u “Off” independientemente de la acción a realizar.

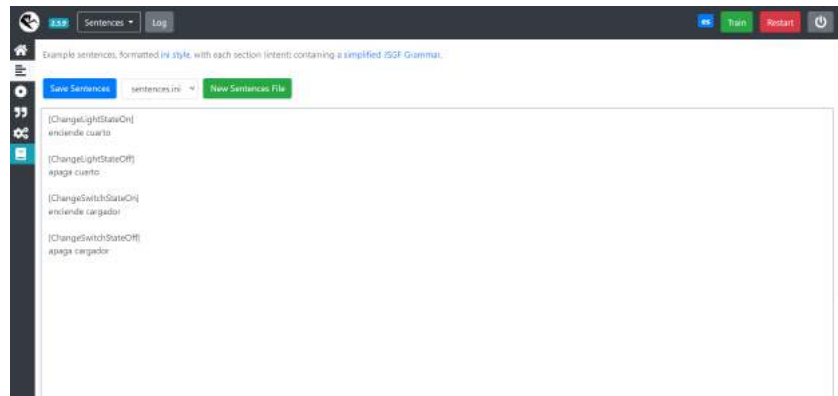


Figura 59 Configuración del sistema Rhasspy - Comandos de voz a reconocer

Fuente: Elaboración Propia

En la Figura 60 visualiza el apartado “Words” también se debe entrenar, pero en este caso será solo palabra por palabra sin necesidad de comandos específicos de Rhasspy.

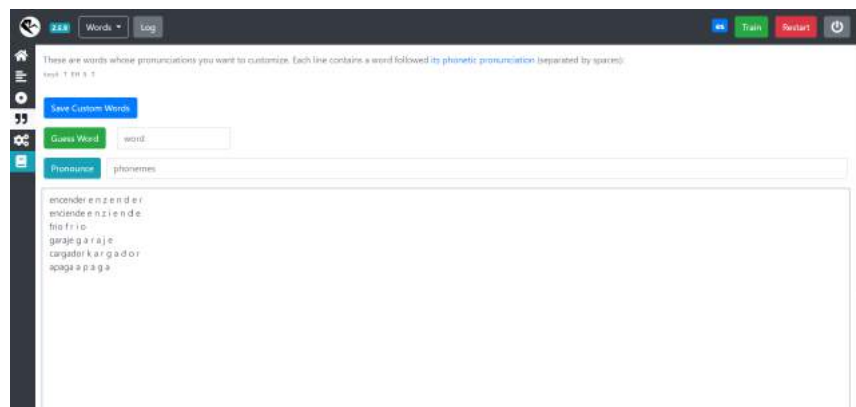


Figura 60 Configuración del sistema Rhasspy – Palabras a entrenar

Fuente: Elaboración Propia

Dentro de “Settings” se configura cada pestaña de acuerdo a las necesidades del usuario. Como se visualiza la Figura 61 en la pestaña “Audio Recording” se debe seleccionar el micrófono por el que escucha el comando.

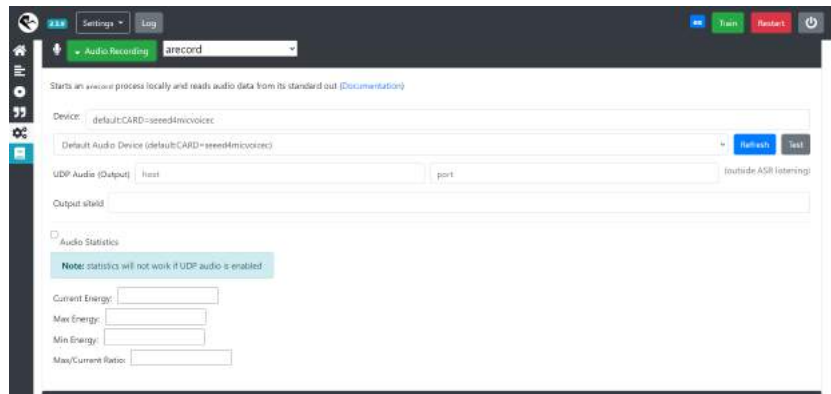


Figura 61 Configuración del sistema Rhasspy - Selección del micrófono

Fuente: Elaboración Propia

### 2.5.5. Conexión rhasspy con hassio

Como primer paso dentro de Home Assistant, crea un token el cual es un código cifrado que ayuda a la comunicación entre Home Assistant y Rhasspy como en la Figura 62.

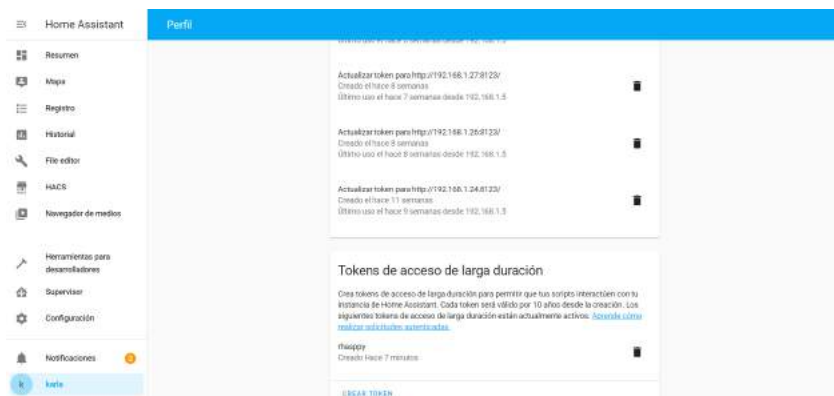


Figura 62 Conexión rhasspy con hassio - Creación de token en Home Assistant

Fuente: Elaboración Propia

Dentro de Rhasspy en la pestaña “Setting”, dirigirse al parámetro “Intent Handling” y seleccionar la opción “Home Assistant”; pedirá ingresar la url de Home Assistant, el token previamente creado y marcar la opción “Send events to Home Assistant (/api/events)”. De este modo se crea la comunicación entre los sistemas mencionados como se observa en la Figura 63.



En la Figura 66 se visualiza la orden enviada mediante el comando de voz empleado.

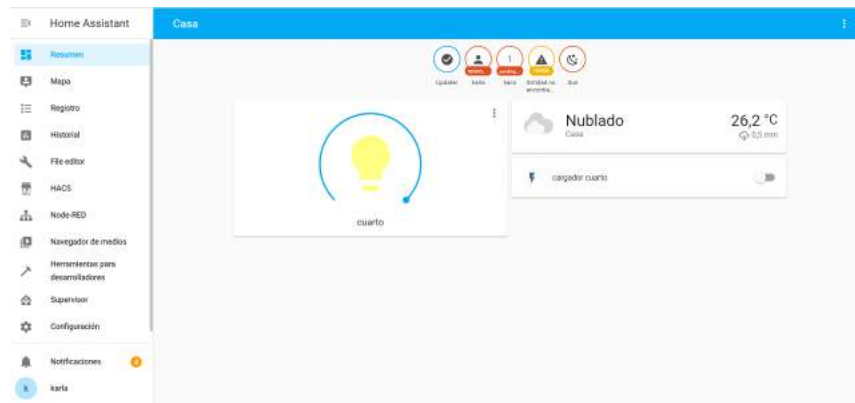


Figura 66 Conexión rhaspy con hassio - Home Assistant receptando peticiones

Fuente: Elaboración Propia

## 2.5.6. Integración del diseño 3D en Home Assistant

Para la implementación del diseño en 3D, desde la tienda de Home Assistant se instala un Add-on llamado Samba, que a futuro permite compartir carpetas existentes en la plataforma domótica, donde se almacena las imágenes para el plano en 3D.

### 2.5.6.1. Instalación de Add-on Samba en Home Assistant

Samba es un Add-on que facilita el compartir carpetas como se observa en la Figura 67 para añadir los documentos con los que se va a trabajar, ya que permite acceder a las carpetas de la plataforma domótica Home Assistant.

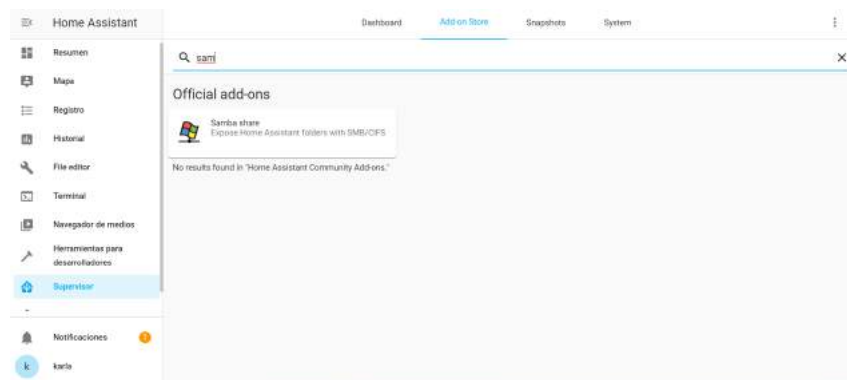


Figura 67 Instalación de Samba - Búsqueda del Add-on

Fuente: Elaboración Propia

Una vez encontrado Samba se procede a realizar la instalación, siendo un Add-on fácil de instalar como se muestra en la Figura 68.

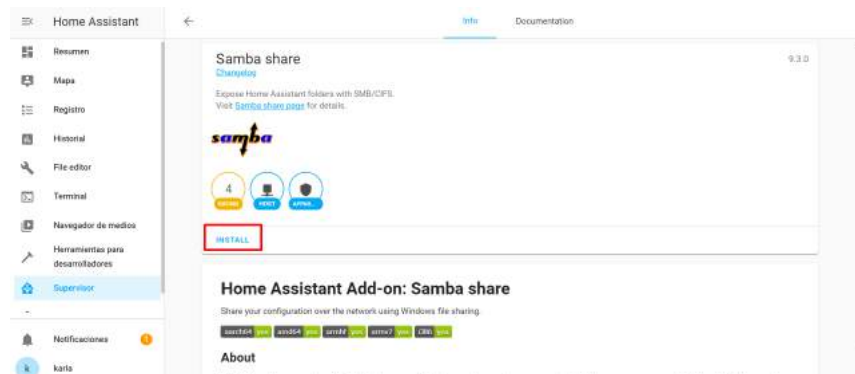


Figura 68 Instalación de Samba - Instalación del Add-on

Fuente: Elaboración Propia

Como se muestra en la Figura 69 dentro de la configuración de Samba se establece las credenciales para el acceso a las carpetas compartidas de Home Assistant.

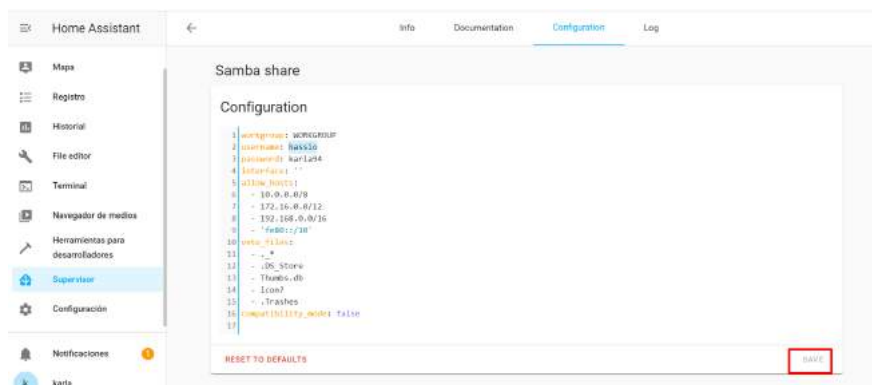


Figura 69 Instalación de Samba - Configuración del Add-on

Fuente: Elaboración Propia

Para el correcto funcionamiento de Samba debe ejecutar el servidor como se visualiza en la Figura 70.



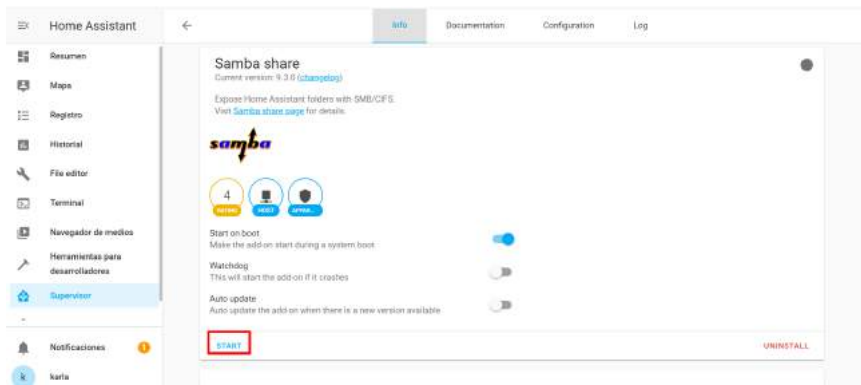


Figura 70 Instalación de Samba – Add-on ejecutado

Fuente: Elaboración Propia

En la Figura 71 se muestra el acceso a las carpetas compartidas de Home Assistant desde la máquina anfitrión.

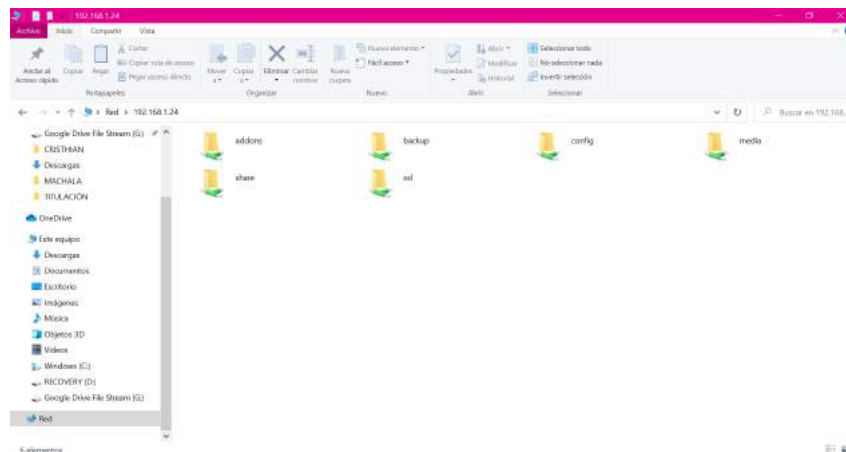


Figura 71 Instalación de Samba - Ingreso a las carpetas de Home Assistant

Fuente: Elaboración Propia

### 2.5.6.2. Implementación del modelo 3D en Home Assistant

Dentro de Home Assistant como se observa en la Figura 72 se crea una nueva pestaña que contendrá el modelo 3D, implementando una tarjeta en blanco para escribir el código que hace funcionar el modelo 3D con el encendido y apagado de los dispositivos, en este caso las luces.

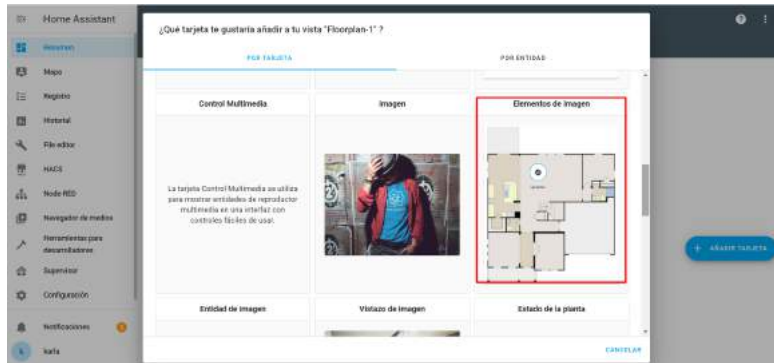


Figura 72 Implementación modelo 3D - Selección de tarjeta "Elementos de Imagen"

Fuente: Elaboración Propia

En la creación de la pestaña se establece un nombre, las demás opciones son opcionales, se guarda el cambio realizado como se puede ver en la Figura 73.

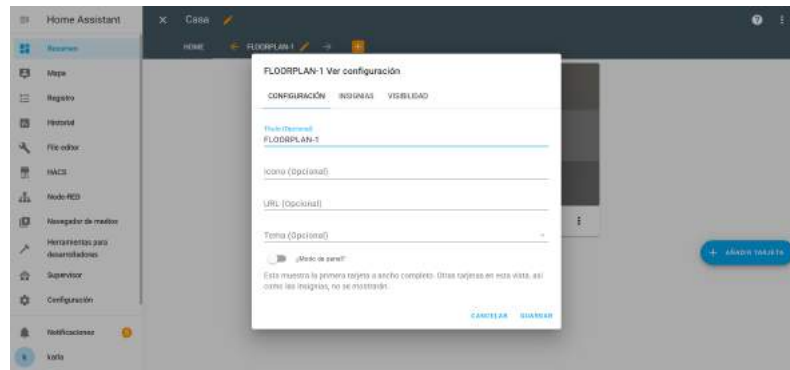


Figura 73 Implementación modelo 3D - Creación de pestaña Floorplan

Fuente: Elaboración Propia

En la Figura 74 y 75 se visualiza la pantalla donde escribe el código que especifica la acción encendido y apagado mediante las imágenes para la gestión de los dispositivos.

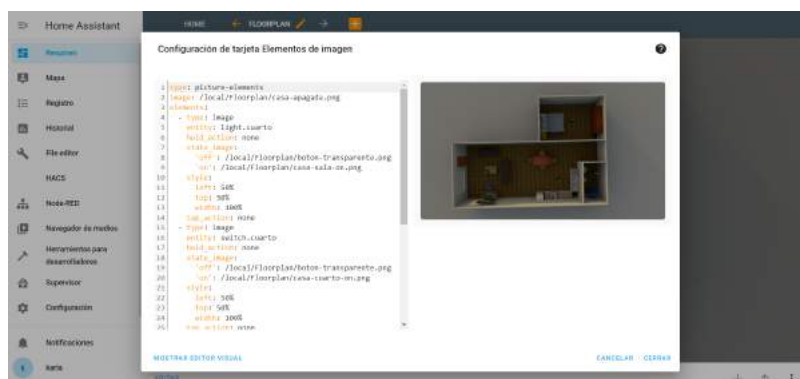


Figura 74 Implementación modelo 3D – Creación del código parte 1

Fuente: Elaboración Propia

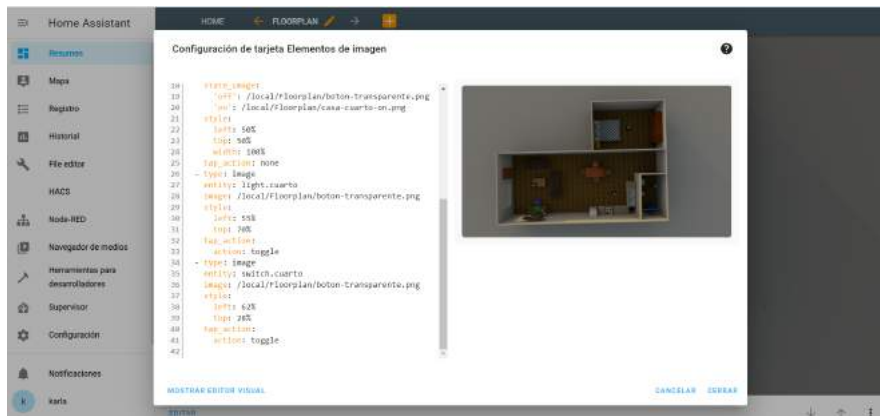


Figura 75 Implementación modelo 3D – Creación del código parte 2

Fuente: Elaboración Propia

En la figura 76 se puede visualizar el correcto funcionamiento del código antes empleado, donde se realiza la acción de encendido de los distintos dispositivos integrados en el sistema domótico.



Figura 76 Implementación modelo 3D - Funcionamiento

Fuente: Elaboración Propia

### 3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

#### 3.1. PLAN DE EVALUACIÓN

Para el desarrollo de las pruebas de la propuesta tecnológica planteada se ha considerado conveniente implementar las siguientes herramientas como fuente de apoyo para el análisis:

- Archivo JSON de Rhasspy
- Wireshark
- Node-Red

Se ha considerado realizar pruebas que analicen los tiempos en milisegundos de las peticiones enviadas desde Rhasspy a Home Assistant en relación al encendido y apagado de los dispositivos IoT.

Con el archivo JSON se conocerá el tiempo de envío de la petición de Rhasspy a Home Assistant; Wireshark ayuda a conocer los tiempos de recepción de Home Assistant al dispositivo IoT y con Node-Red se desarrolló una pequeña aplicación que ayuda a la medición de los tiempos de encendido de los dispositivos. Las pruebas realizadas sirvieron de apoyo para el análisis de los datos mediante una tabla comparativa.

### 3.1.1. Aplicación en Node-Red

El tiempo de confirmación de las peticiones se las puede tomar desde Node-Red mediante el desarrollo de una pequeña aplicación, Node-Red es un Add-on que se encuentra dentro de Home Assistant, su descarga e instalación es rápida y fácil.

#### 3.1.1.1. Instalación del Add-on Node-Red en Home Assistant

Como todo Add-on la instalación se hace por medio del Supervisor, es así como se hace uso de Node-Red para el desarrollo de una aplicación que tome la latencia de los dispositivos IoT como se observa en la Figura 77.

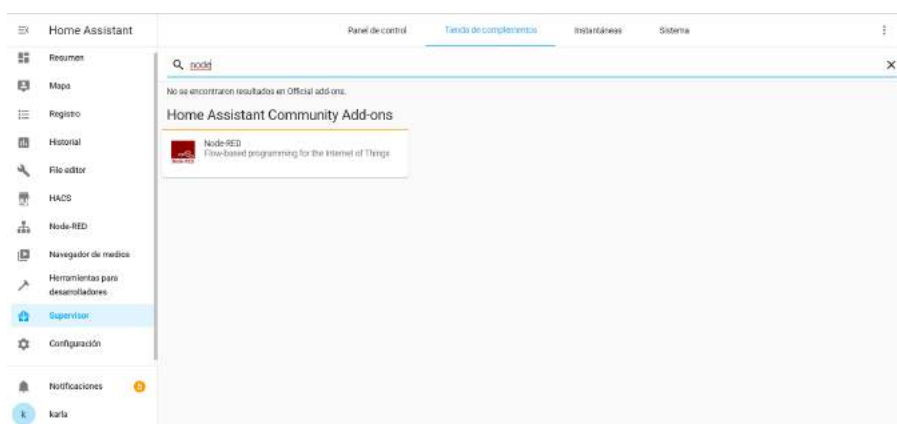


Figura 77 Instalación Node-Red - Instalación del Add-on

Fuente: Elaboración Propia

Una vez encontrado Node-Red se procede a realizar la instalación como se observa en la Figura 78, siendo un Add-on fácil de instalar. Para el correcto funcionamiento de Node-Red debe ejecutar el servidor.

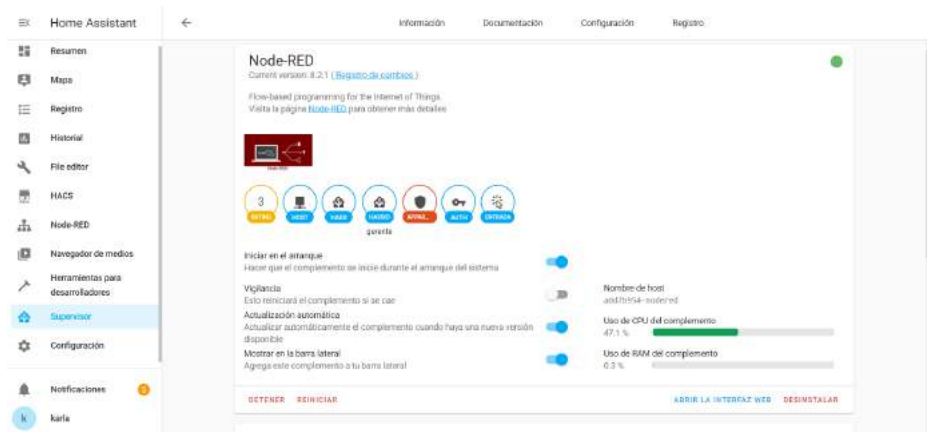


Figura 78 Instalación Node-Red - Instalación del Add-on

Fuente: Elaboración Propia

En la Figura 79 se observa cómo se crea Node-Red como una nueva pestaña, al dar clic muestra un panel de trabajo donde se puede realizar las aplicaciones que se desee; en este caso se realiza una aplicación para conocer los tiempos de petición.

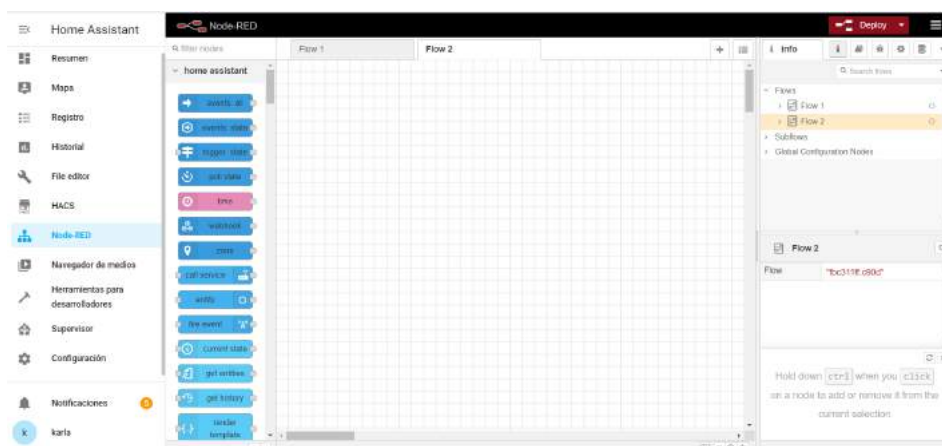


Figura 79 Instalación Node-Red - Instalación del Add-on

Fuente: Elaboración Propia

### 3.1.1.2. Aplicación para toma de tiempos

Node-Red es una aplicación fácil e intuitiva para los usuarios ya que programar dentro de ella es fácil, se basa en una programación de nodos por lo que no presenta mayor complejidad; y se encuentra dentro de los Add-on de la plataforma domótica Home Assistant.

Para conocer los tiempos de las peticiones se establecieron tres nodos, un nodo para verificar el cambio de estado de la bombilla que se va a analizar, y dos debug los cuales permiten saber el tiempo que demora en cambiar de estado como se observa en la Figura 80.

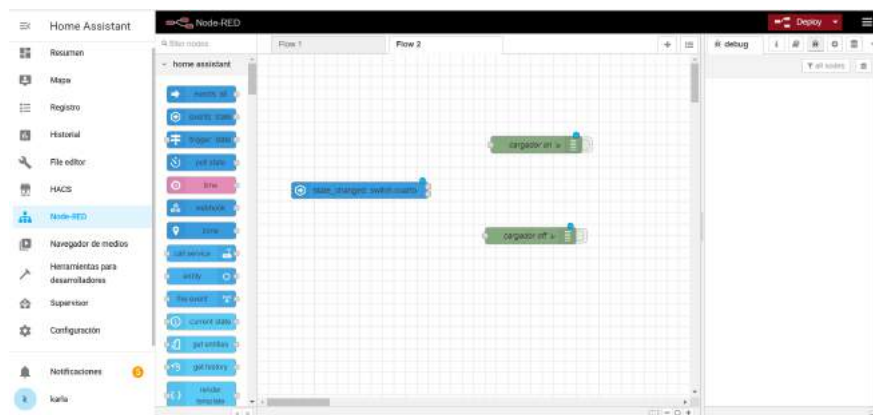


Figura 80 Desarrollo aplicación - Selección de nodos

Fuente: Elaboración Propia

En la Figura 81 se configura el estado de encendido y apagado que toma cada nodo para la lectura de latencia y se establece un el nombre del nodo es decir el nodo “encendido” y el nodo “apagado”; de esta manera se hace la lectura ordenada y rápida por su identificación.

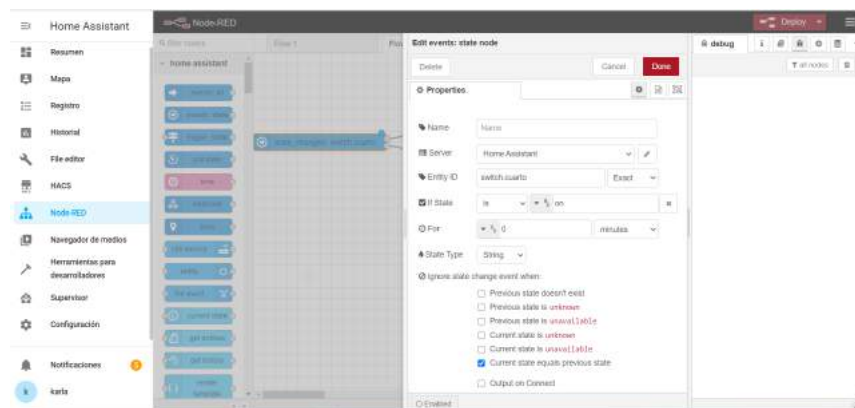


Figura 81 Desarrollo aplicación - Configuración de nodos

Fuente: Elaboración Propia

En la Figura 82 se muestra la aplicación completamente realizada para la toma de tiempos; los tiempos de respuesta se dan una vez realizadas las peticiones.

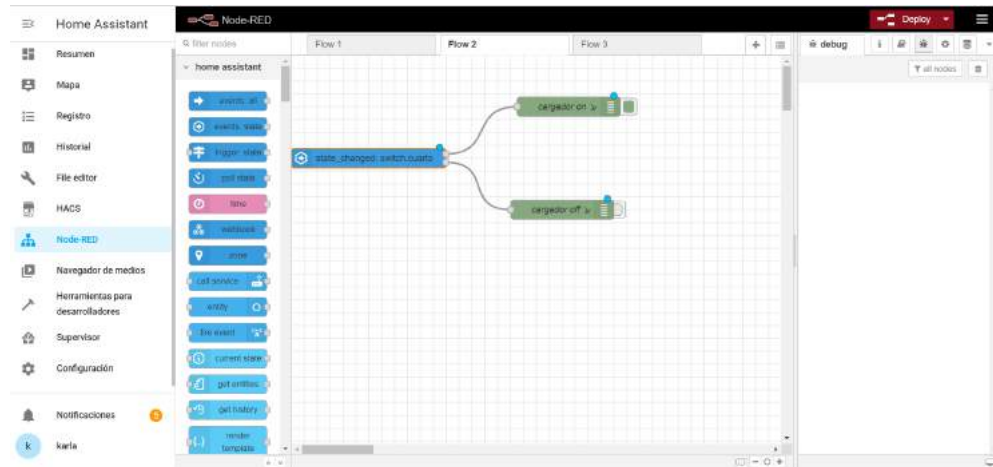


Figura 82 Desarrollo aplicación - Configuración de nodos

Fuente: Elaboración Propia

### 3.1.2. Toma de tiempos reconocimiento de comando voz

En la toma de los tiempos de respuesta de los distintos sistemas se ha considerado el archivo JSON proporcionado por Rhasspy, para conocer el tiempo del proceso de reconocimiento de los comandos de voz de los micrófonos utilizados:

- USB
- ReSpeaker 4-Mic Array

Para el presente análisis se ha considerado el dispositivo foco que dentro del sistema domótico Home Assistant se denomina “Cuarto”; se mide el tiempo de latencia en su estado de encendido y apagado, las mismas que se detallan en las siguientes imágenes.

### 3.1.2.1. Micrófono USB

En la Figura 83 se observa el tiempo de encendido en milisegundos del micrófono USB proporcionado de archivo JSON de Rhasspy.

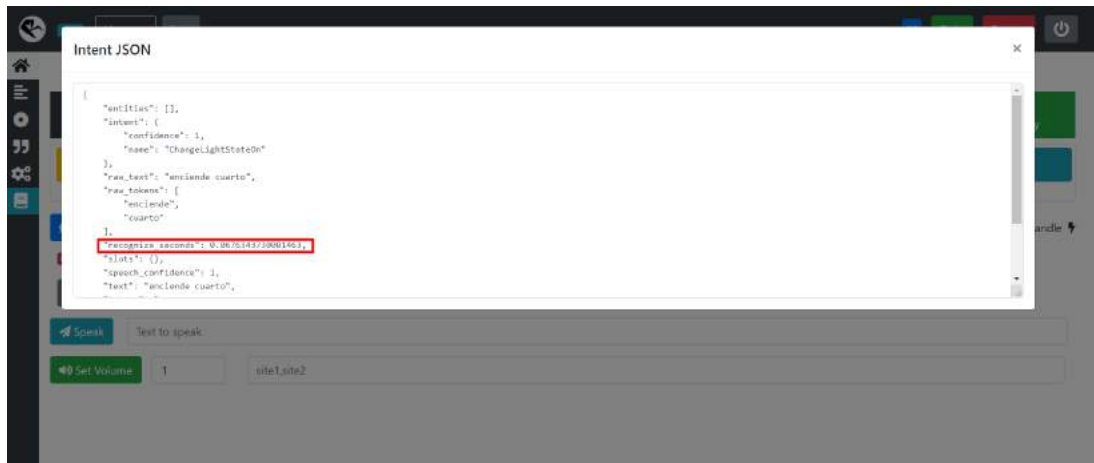


Figura 83 Encendido de foco - Toma de tiempo de comando de voz de Rhasspy

Fuente: Elaboración Propia

En la Figura 84 se observa el tiempo de apagado en milisegundos del micrófono USB proporcionado de archivo JSON de Rhasspy.

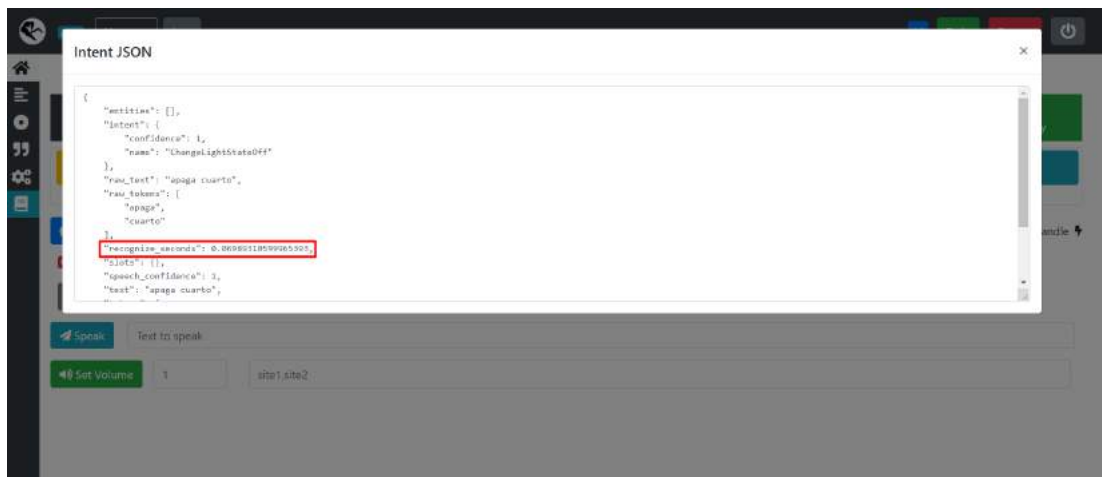


Figura 84 Apagado de foco – Toma de tiempo de comando de voz de Rhasspy

Fuente: Elaboración Propia



### 3.1.2.2. Micrófono ReSpeaker 4-Mic Array

En la Figura 85 se observa el tiempo de encendido en milisegundos del micrófono ReSpeaker 4-Mic Array proporcionado de archivo JSON de Rhasspy.

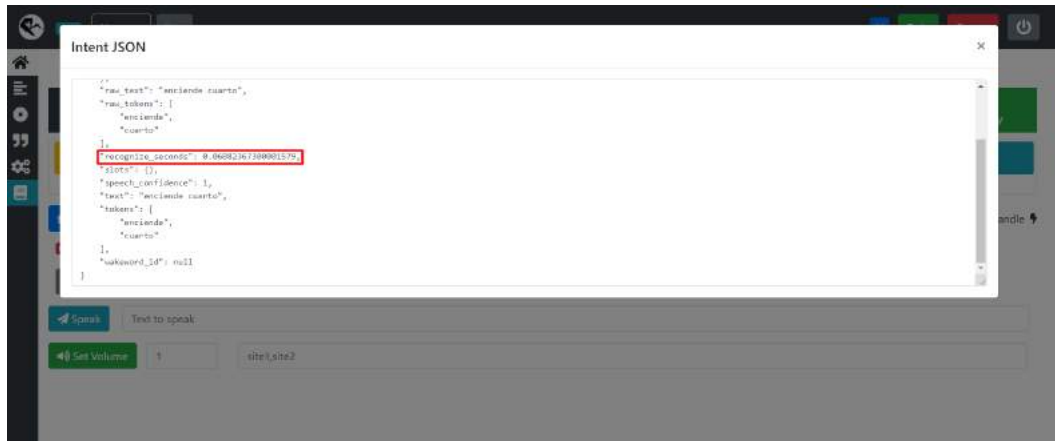


Figura 85 Encendido de foco - Toma de tiempo de comando de voz de Rhasspy

Fuente: Elaboración Propia

En la Figura 86 se observa el tiempo de apagado en milisegundos del micrófono ReSpeaker 4-Mic Array proporcionado de archivo JSON de Rhasspy.

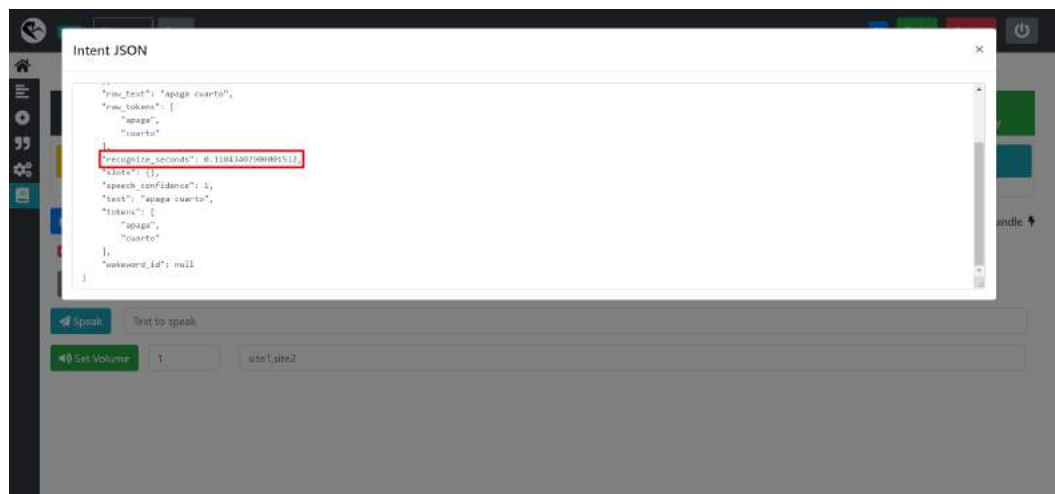


Figura 86 Encendido de foco – Toma de tiempo de comando de voz de Rhasspy

Fuente: Elaboración Propia

### 3.1.3. Toma de tiempos de peticiones

En la toma de los tiempos de respuesta de los distintos sistemas se ha considerado a Wireshark, ya que realiza el análisis de tramas, para conocer el tiempo del proceso de toma de datos en los micrófonos utilizados:

- USB
- ReSpeaker 4-Mic Array

Asistentes de voz:

- Alexa Amazon
- Google Assistant

Para el presente análisis se ha considerado el dispositivo foco en su estado de encendido y apagado, las mismas que se detallan en las siguientes imágenes.

#### 3.1.3.1. Micrófono USB

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de encendido mediante el micrófono USB como visualiza en la Figura 87.

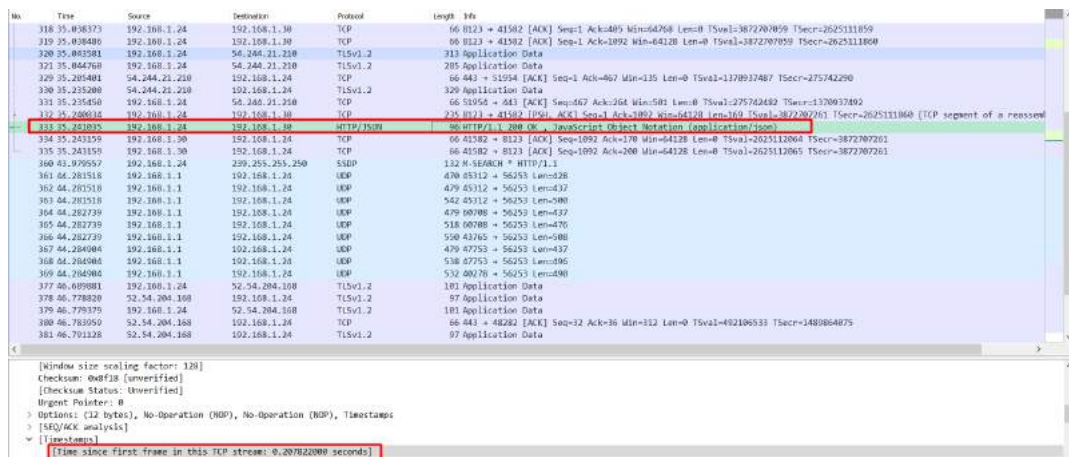


Figura 87 Prueba de tiempo - Encendido de foco desde USB - Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de encendido mediante el micrófono USB como se visualiza en la Figura 88.

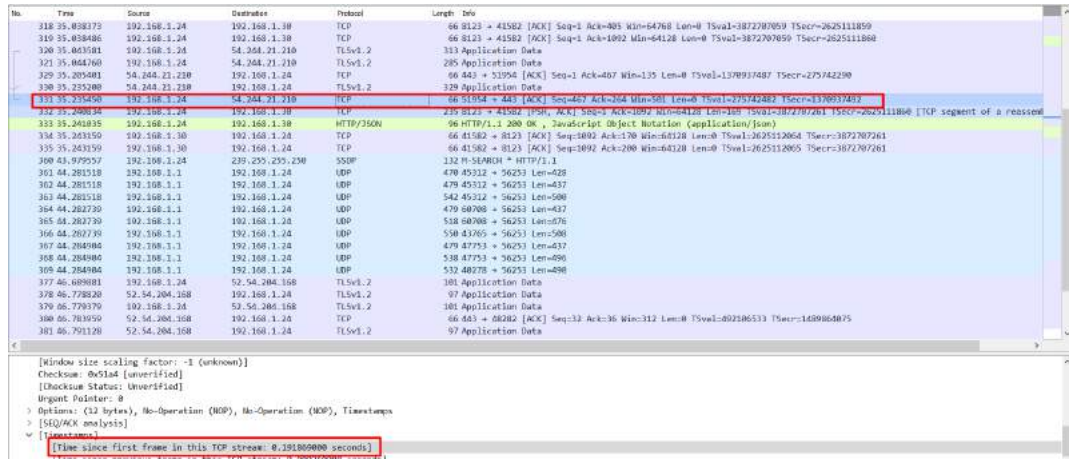


Figura 88 Prueba de tiempo - Encendido de foco desde USB – Hassio a Dispositivo

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de apagado mediante el micrófono USB como se visualiza en la Figura 89.

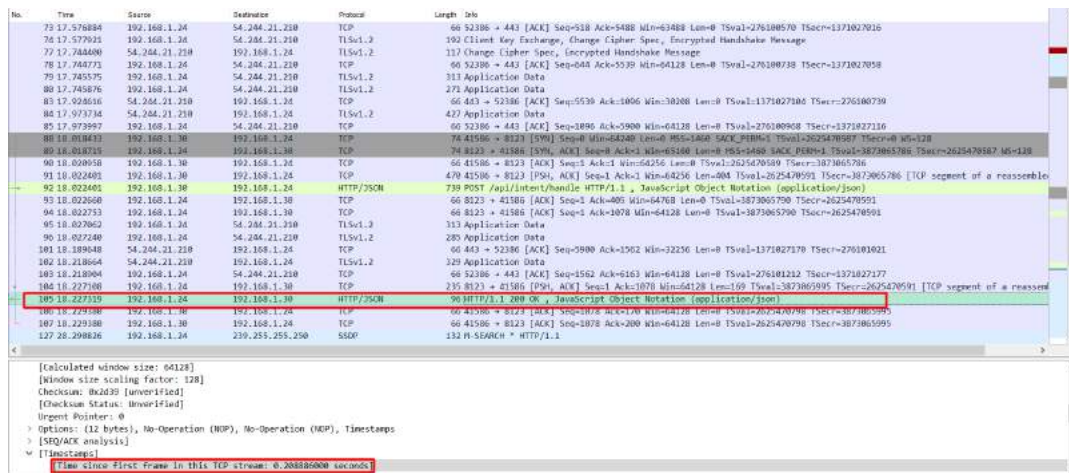


Figura 89 Prueba de tiempo - Apagado de foco desde USB - Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de apagado mediante el micrófono USB como se visualiza en la Figura 90.

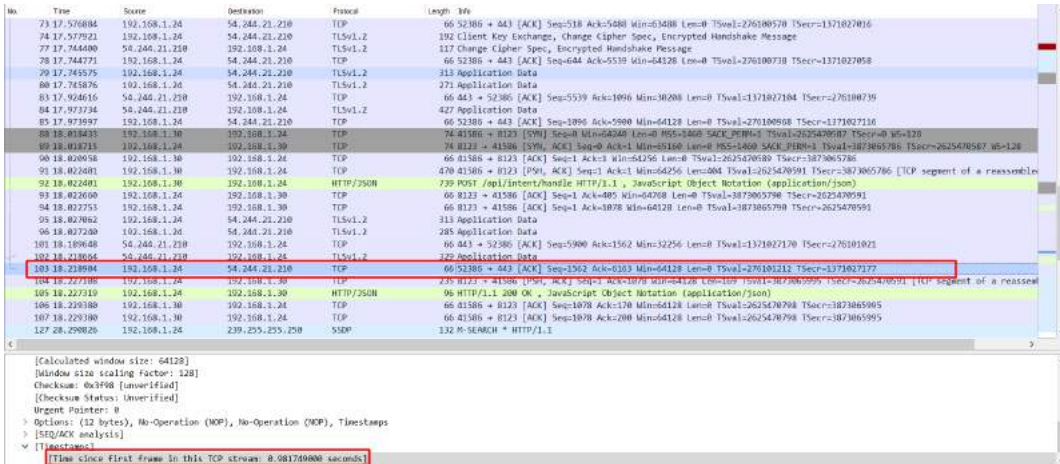


Figura 90 Prueba de tiempo - Apagado de foco desde USB - Hassio a Dispositivo

Fuente: Elaboración Propia

### 3.1.3.2. Micrófono ReSpeaker 4-Mic Array

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de encendido mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 91.

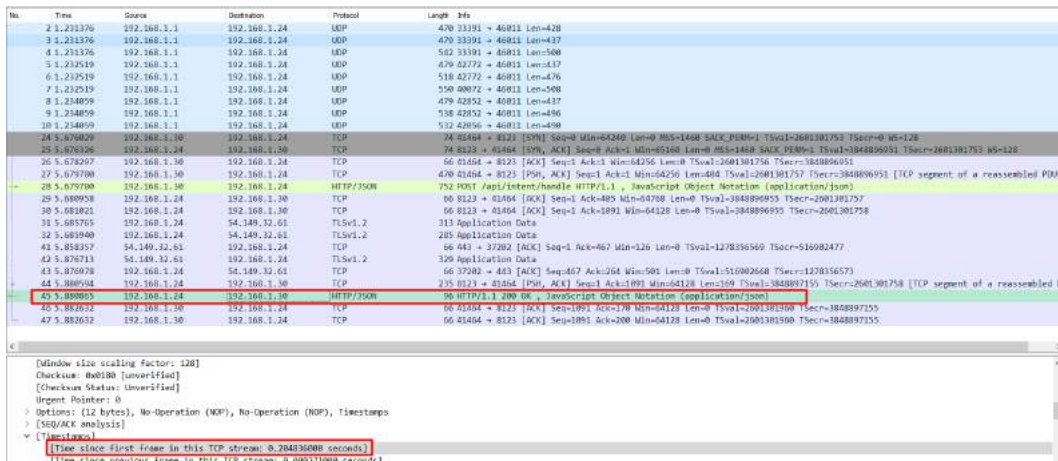


Figura 91 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic - Rhasspy a Hassio

Fuente: Elaboración Propia



En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de encendido mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 92.

No.	Time	Source	Destination	Protocol	Length	Info
2	1.231376	192.168.1.1	192.168.1.24	UDP	479	33391 → 40811 Len=437
3	1.231376	192.168.1.1	192.168.1.24	UDP	542	33391 → 40811 Len=500
4	1.232519	192.168.1.1	192.168.1.24	UDP	479	42772 → 40811 Len=437
5	1.232519	192.168.1.1	192.168.1.24	UDP	538	42772 → 40811 Len=476
6	1.232519	192.168.1.1	192.168.1.24	UDP	538	40872 → 40811 Len=500
7	1.234059	192.168.1.1	192.168.1.24	UDP	479	43852 → 40811 Len=437
8	1.234059	192.168.1.1	192.168.1.24	UDP	538	43852 → 40811 Len=496
9	1.234059	192.168.1.1	192.168.1.24	UDP	532	42856 → 40811 Len=490
10	1.234059	192.168.1.1	192.168.1.24	UDP	78	41464 → 8123 [PSH] Seq=1 Win=4128 Len=8 TSVol=3681301751 TSecr=0 Wd=128
15	5.609858	192.168.1.30	192.168.1.24	TCP	66	8123 → 41464 [ACK] Seq=1 Acks=0 Win=6768 Len=8 TSVol=3681301751 TSecr=268128757
25	5.609336	192.168.1.24	192.168.1.30	TCP	66	8123 → 41464 [ACK] Seq=1 Acks=1 Win=6536 Len=8 TSVol=3681301751 TSecr=2681301751 Wd=128
20	5.679297	192.168.1.30	192.168.1.24	TCP	66	41464 → 8123 [ACK] Seq=1 Acks=1 Win=64256 Len=8 TSVol=2681301750 TSecr=384890951
27	5.679780	192.168.1.30	192.168.1.24	TCP	479	41464 → 8123 [PSH, ACK] Seq=1 Acks=1 Win=64256 Len=804 TSVol=2681301750 TSecr=384890951 [TCP segment of a reassembled PDU]
28	5.679780	192.168.1.30	192.168.1.24	HTTP/JSON	752	POST /api/intent/pandle HTTP/1.1, JavaScript Object Notation [Application/JSON]
29	5.680058	192.168.1.30	192.168.1.24	TCP	66	8123 → 41464 [ACK] Seq=1 Acks=0 Win=6768 Len=8 TSVol=3681301751 TSecr=268128757
30	5.681021	192.168.1.24	192.168.1.30	TCP	66	8123 → 41464 [ACK] Seq=1 Acks=1891 Win=64128 Len=8 TSVol=384890955 TSecr=2681301750
31	5.685765	192.168.1.24	54.149.32.61	TLSv1.2	313	Application Data
32	5.685940	192.168.1.24	54.149.32.61	TLSv1.2	285	Application Data
41	5.838237	54.149.32.61	192.168.1.24	TCP	66	443 → 33302 [ACK] Seq=1 Acks=67 Wd=126 Len=8 TSVol=1278253566 TSecr=51602477
42	5.876713	54.149.32.61	192.168.1.24	TLSv1.2	329	Application Data
43	5.876978	192.168.1.24	54.149.32.61	TCP	66	19201 → 443 [ACK] Seq=487 Acks=264 Min=501 Len=8 TSVol=516026668 TSecr=1278150573
45	5.880584	192.168.1.24	192.168.1.30	TCP	235	8123 → 41464 [PSH, ACK] Seq=1 Acks=1891 Win=64128 Len=168 TSVol=3848917155 TSecr=2681301751 [TCP segment of a reassembled PDU]
46	5.882612	192.168.1.30	192.168.1.24	HTTP/JSON	66	41464 → 8123 [ACK] Seq=1891 Acks=170 Win=64128 Len=8 TSVol=2681301750 TSecr=3848907155
47	5.882612	192.168.1.30	192.168.1.24	TCP	66	41464 → 8123 [ACK] Seq=1891 Acks=200 Win=64128 Len=8 TSVol=2681301750 TSecr=3848907155

Figura 92 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic- Hassio a dispositivo

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de apagado mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 93.

No.	Time	Source	Destination	Protocol	Length	Info
54	6.928956	54.244.21.210	192.168.1.24	TCP	66	443 → 41570 [ACK] Seq=1 Acks=518 Win=28160 Len=8 TSVol=1365132733 TSecr=252523257
55	6.928956	54.244.21.210	192.168.1.24	TLSv1.2	1514	Server Hello
56	6.928956	54.244.21.210	192.168.1.24	TCP	1514	443 → 41570 [ACK] Seq=1449 Acks=518 Win=28160 Len=1448 TSVol=1365132733 TSecr=252523257 [TCP segment of a reassembled PDU]
57	6.928956	54.244.21.210	192.168.1.24	TCP	1514	443 → 41570 [ACK] Seq=2897 Acks=518 Win=28160 Len=1448 TSVol=1365132733 TSecr=252523257 [TCP segment of a reassembled PDU]
58	6.930137	192.168.1.24	54.244.21.210	TCP	66	41570 → 443 [ACK] Seq=518 Acks=345 Win=62952 Len=8 TSVol=2525232426 TSecr=1365132733
59	6.931171	54.244.21.210	192.168.1.24	TLSv1.2	1289	Certificate, Server Key Exchange, Server Hello Done
60	6.931451	192.168.1.24	54.244.21.210	TCP	66	41570 → 443 [ACK] Seq=518 Acks=388 Win=64128 Len=8 TSVol=2525232426 TSecr=1365132733
61	6.932276	192.168.1.24	54.244.21.210	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
62	7.104863	54.244.21.210	192.168.1.24	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
63	7.104861	192.168.1.24	54.244.21.210	TCP	66	41570 → 443 [ACK] Seq=644 Acks=579 Win=64128 Len=8 TSVol=2525232601 TSecr=1365132777
64	7.105023	192.168.1.24	54.244.21.210	TLSv1.2	313	Application Data
65	7.105013	192.168.1.24	54.244.21.210	TLSv1.2	285	Application Data
66	7.274219	54.244.21.210	192.168.1.24	TCP	66	443 → 41570 [ACK] Seq=5339 Acks=1119 Win=30008 Len=8 TSVol=1365132819 TSecr=252523602
67	7.204577	54.244.21.210	192.168.1.24	TCP	329	Application Data
68	7.204867	192.168.1.24	54.244.21.210	TCP	66	41570 → 443 [ACK] Seq=1110 Acks=5802 Win=64128 Len=8 TSVol=2525237091 TSecr=1365132824
69	7.208721	192.168.1.24	192.168.1.30	TCP	235	8123 → 41464 [PSH, ACK] Seq=1 Acks=1878 Win=64128 Len=168 TSVol=3848908509 TSecr=2681301754 [TCP segment of a reassembled PDU]
70	7.208836	192.168.1.24	192.168.1.30	HTTP/JSON	76	HTTP/1.1 200 OK, JavaScript Object Notation [Application/JSON]
71	7.308810	192.168.1.30	192.168.1.24	TCP	66	41464 → 8123 [ACK] Seq=1078 Acks=178 Win=64128 Len=8 TSVol=2681301754 TSecr=3848908509
72	7.308918	192.168.1.30	192.168.1.24	TCP	66	41464 → 8123 [ACK] Seq=1078 Acks=200 Win=64128 Len=8 TSVol=2681301754 TSecr=3848908509
73	18.005146	192.168.1.24	239.255.255.250	SNDP	112	M-SNDP (*) HTTP/1.1
76	18.005404	192.168.1.1	192.168.1.24	UDP	470	37311 → 48289 Len=428
77	18.005404	192.168.1.1	192.168.1.24	UDP	470	37311 → 48289 Len=437
78	18.005404	192.168.1.1	192.168.1.24	UDP	542	37311 → 48289 Len=500
79	18.010773	192.168.1.1	192.168.1.24	UDP	479	44565 → 48289 Len=437
80	18.010773	192.168.1.1	192.168.1.24	UDP	518	44565 → 48289 Len=476
81	18.010773	192.168.1.1	192.168.1.24	UDP	530	55317 → 48289 Len=500

Figura 93 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic - Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de apagado mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 94.

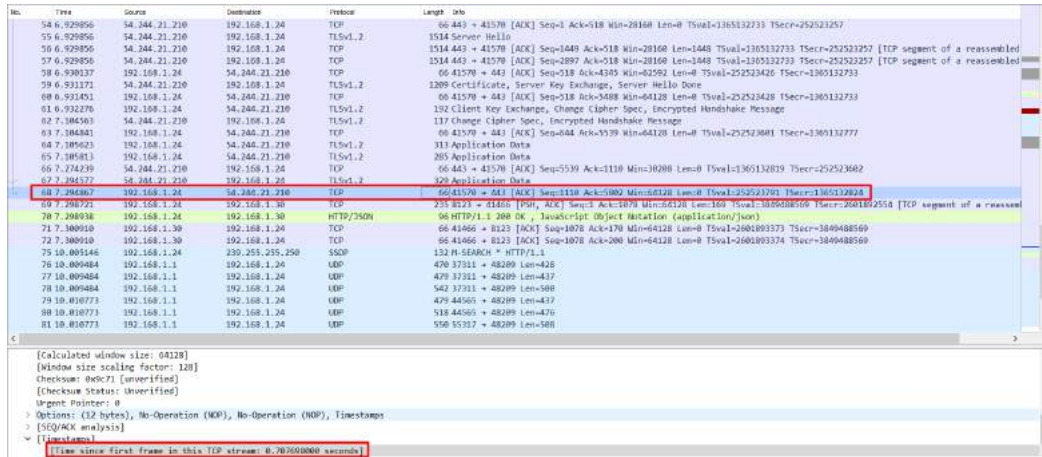


Figura 94 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic – Hassio a dispositivo

Fuente: Elaboración Propia

### 3.1.3.3. Asistente de voz Alexa Amazon

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de encendido mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 95.

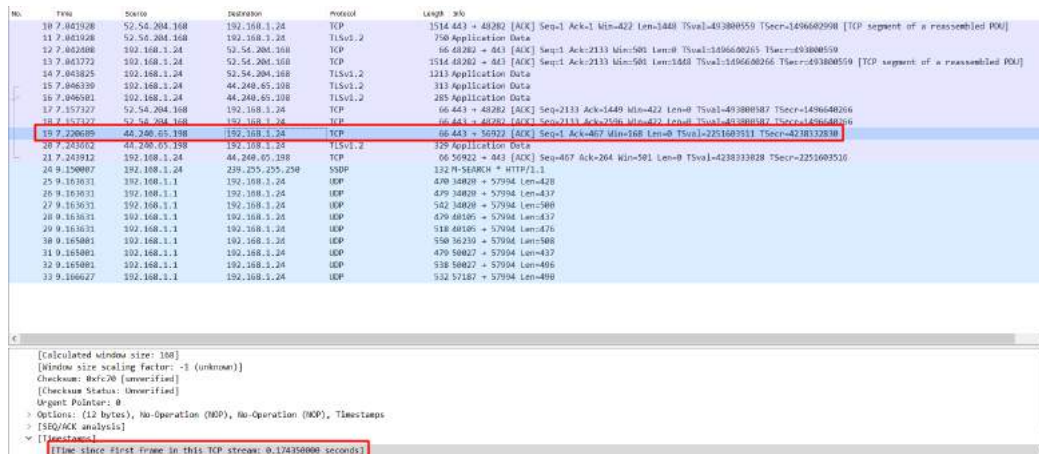


Figura 95 Prueba de tiempo - Encendido de foco desde Alexa Amazon - Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de encendido mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 96.

No.	Time	Source	Destination	Protocol	Length	Info
10	7.041928	52.54.204.168	192.168.1.24	TCP	3514	443 → 8282 [ACK] Seq=1 Ack=1 Win=422 Len=1448 Tsvol=493800559 Tsecr=1496602998 [TCP segment of a reassembled PDU]
11	7.041938	52.54.204.168	192.168.1.24	TLSv1.2	790	Application Data
12	7.042088	192.168.1.24	52.54.204.168	TCP	66	48282 → 443 [ACK] Seq=1 Ack=2133 Win=581 Len=0 Tsvol=1496640265 Tsecr=493800550
13	7.043772	192.168.1.24	52.54.204.168	TCP	3514	48282 → 443 [ACK] Seq=1 Ack=2133 Win=581 Len=1448 Tsvol=1496640266 Tsecr=493800550 [TCP segment of a reassembled PDU]
14	7.043825	192.168.1.24	52.54.204.168	TLSv1.2	1213	Application Data
15	7.046339	192.168.1.24	44.240.65.198	TLSv1.2	313	Application Data
16	7.046501	192.168.1.24	44.240.65.198	TLSv1.2	285	Application Data
17	7.157327	52.54.204.168	192.168.1.24	TCP	66	443 → 8282 [ACK] Seq=2133 Ack=1449 Win=422 Len=0 Tsvol=493800567 Tsecr=1496648260
18	7.157327	52.54.204.168	192.168.1.24	TCP	66	443 → 8282 [ACK] Seq=2133 Ack=2598 Win=422 Len=0 Tsvol=493800567 Tsecr=1496648260
19	7.226589	44.240.65.198	192.168.1.24	TCP	66	443 → 50622 [ACK] Seq=1 Ack=467 Win=168 Len=0 Tsvol=225168511 Tsecr=421813150
20	7.243662	44.240.65.198	192.168.1.24	TLSv1.2	920	Application Data
21	7.243922	192.168.1.24	44.240.65.198	TCP	66	50622 → 443 [ACK] Seq=467 Ack=204 Win=901 Len=0 Tsvol=421813150 Tsecr=251685516
24	9.158067	192.168.1.24	239.255.255.250	SSDP	132	M:SSDP → m1:1:1
25	9.163631	192.168.1.1	192.168.1.24	ICMP	478	34820 → 57994 Len=428
26	9.163631	192.168.1.1	192.168.1.24	ICMP	479	34820 → 57994 Len=437
27	9.163631	192.168.1.1	192.168.1.24	ICMP	542	34820 → 57994 Len=500
28	9.163631	192.168.1.1	192.168.1.24	ICMP	479	40085 → 57994 Len=437
29	9.163631	192.168.1.1	192.168.1.24	ICMP	518	40085 → 57994 Len=476
30	9.165001	192.168.1.1	192.168.1.24	ICMP	558	36239 → 57994 Len=588
31	9.165001	192.168.1.1	192.168.1.24	ICMP	479	50627 → 57994 Len=437
32	9.165001	192.168.1.1	192.168.1.24	ICMP	538	50627 → 57994 Len=496
33	9.166627	192.168.1.1	192.168.1.24	ICMP	532	57187 → 57994 Len=490

[Calculated window size: 901]  
 [Window size scaling factor: -1 (unknown)]  
 [Checksum: 0x951 (unverified)]  
 [Checksum Status: Unverified]  
 [Urgent Pointer: 0]  
 [Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps]  
 [SEQ/ACK analysis]  
 [Timestamps]  
 [Time since first frame in this TCP stream: 0.19751000 seconds]

Figura 96 Prueba de tiempo - Encendido de foco desde Alexa Amazon - Hassio a dispositivo

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de apagado mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 97.

No.	Time	Source	Destination	Protocol	Length	Info
18	6.890918	52.54.204.168	192.168.1.24	TCP	3514	443 → 8282 [ACK] Seq=1 Ack=1 Win=422 Len=1448 Tsvol=494586900 Tsecr=1490740177 [TCP segment of a reassembled PDU]
19	6.890918	52.54.204.168	192.168.1.24	TLSv1.2	751	Application Data
20	6.891283	192.168.1.24	52.54.204.168	TCP	66	48282 → 443 [ACK] Seq=1 Ack=2134 Win=581 Len=0 Tsvol=1490785644 Tsecr=494586900
21	6.892616	192.168.1.24	52.54.204.168	TCP	3514	48282 → 443 [ACK] Seq=1 Ack=2134 Win=581 Len=1448 Tsvol=1490785645 Tsecr=494586900 [TCP segment of a reassembled PDU]
22	6.892671	192.168.1.24	52.54.204.168	TLSv1.2	1393	Application Data
23	6.895236	192.168.1.24	54.244.21.238	TLSv1.2	313	Application Data
24	6.895398	192.168.1.24	54.244.21.238	TLSv1.2	285	Application Data
25	7.084319	52.54.204.168	192.168.1.24	TCP	66	443 → 8282 [ACK] Seq=2134 Ack=1445 Win=422 Len=0 Tsvol=494586929 Tsecr=1499785645
26	7.084319	52.54.204.168	192.168.1.24	TCP	66	443 → 8282 [ACK] Seq=2134 Ack=2778 Win=422 Len=0 Tsvol=494586929 Tsecr=1499785645
27	7.095958	54.244.21.210	192.168.1.24	TCP	66	443 → 50514 [ACK] Seq=1 Ack=467 Win=120 Len=0 Tsvol=1373428793 Tsecr=285075589
28	7.096025	54.244.21.210	192.168.1.24	TCP	225	Application Data
29	7.880487	192.168.1.24	54.244.21.238	TCP	66	50514 → 443 [ACK] Seq=467 Ack=204 Win=901 Len=0 Tsvol=285075694 Tsecr=1373428798

[Calculated window size: 126]  
 [Window size scaling factor: -1 (unknown)]  
 [Checksum: 0x1137 (unverified)]  
 [Checksum Status: Unverified]  
 [Urgent Pointer: 0]  
 [Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps]  
 [SEQ/ACK analysis]  
 [Timestamps]  
 [Time since first frame in this TCP stream: 0.154714000 seconds]

Figura 97 Prueba de tiempo - Apagado de foco desde Alexa Amazon – Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de apagado mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 98.

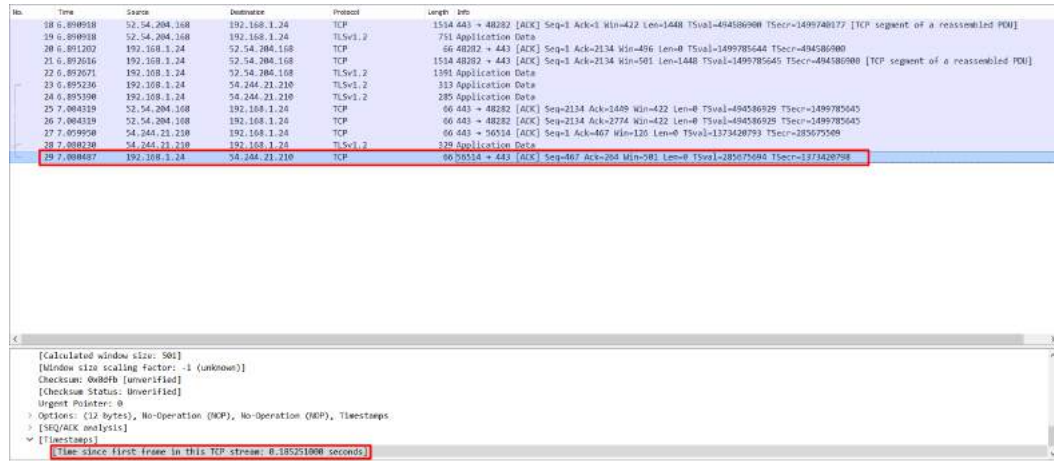


Figura 98 Prueba de tiempo - Apagado de foco desde Alexa Amazon – Hassio a dispositivo

Fuente: Elaboración Propia

### 3.1.3.4. Asistente de voz Google Assistant

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de encendido mediante el asistente de voz Google Assistant como se visualiza en la Figura 99.

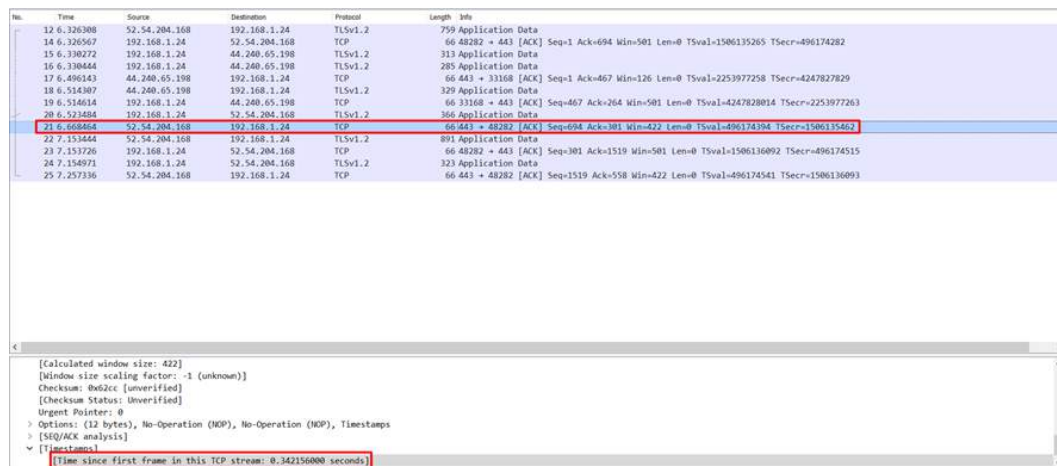


Figura 99 Prueba de tiempo - Encendido de foco desde Google Assistant - Rhasspy a Hassio

Fuente: Elaboración Propia



En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de encendido mediante el asistente de voz Google Assistant como se visualiza en la Figura 100.

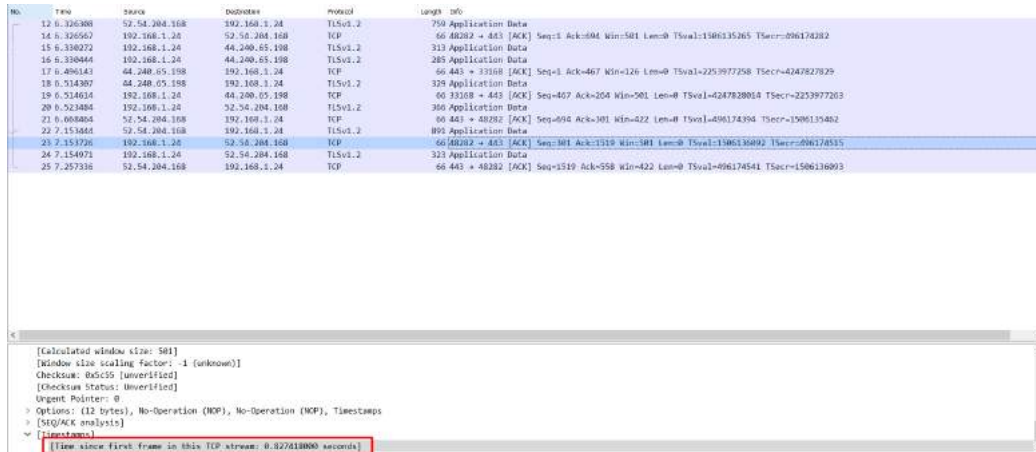


Figura 100 Prueba de tiempo - Encendido de foco desde Google Assistant - Hassio a dispositivo

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta de Rhasspy para la acción de apagado mediante el asistente de voz Google Assistant como se visualiza en la Figura 101.

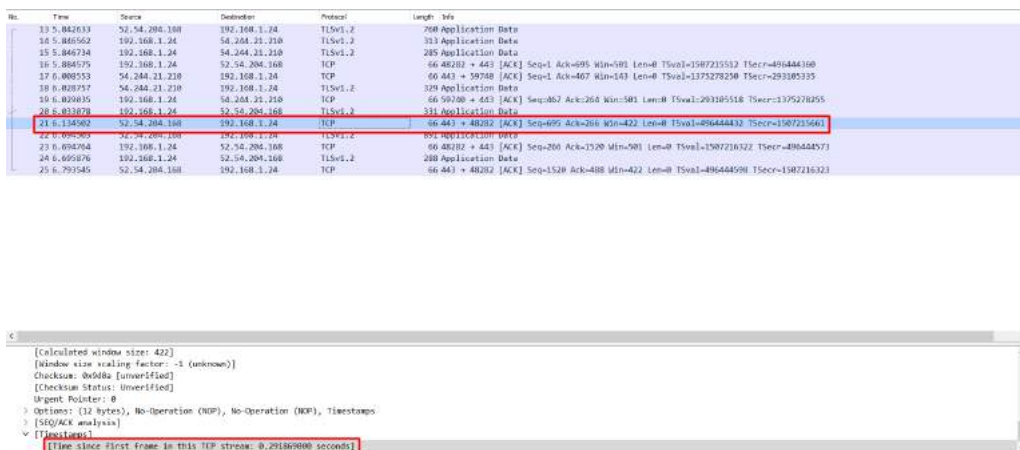


Figura 101 Prueba de tiempo - Apagado de foco desde Google Assistant – Rhasspy a Hassio

Fuente: Elaboración Propia

En la herramienta Wireshark se obtiene el tiempo de respuesta del sistema domótico Home Assistant para la acción de apagado mediante el asistente de voz Google Assistant como se visualiza en la Figura 102.

No.	Time	Source	Destination	Protocol	Length	Info
13	5.862633	52.54.204.108	192.168.1.24	TLSv1.2	788	Application Data
14	5.86552	192.168.1.24	54.244.21.210	TLSv1.2	353	Application Data
15	5.866734	192.168.1.24	54.244.21.210	TLSv1.2	285	Application Data
16	5.888575	192.168.1.24	52.54.204.108	TCP	66	66 88282 → 443 [ACK] Seq=1 Ack=185 Win=503 Len=0 TSval=1507215512 TSecr=496444368
17	5.888553	54.244.21.210	192.168.1.24	TCP	66	66 443 → 59750 [ACK] Seq=1 Ack=457 Win=143 Len=0 TSval=1375278250 TSecr=29395335
18	5.893757	54.244.21.210	192.168.1.24	TLSv1.2	329	Application Data
19	5.898935	192.168.1.24	54.244.21.210	TCP	66	66 59740 → 443 [ACK] Seq=407 Ack=204 Win=503 Len=0 TSval=293189518 TSecr=1375278255
20	5.833078	192.168.1.24	52.54.204.108	TLSv1.2	333	Application Data
21	5.134302	52.54.204.108	192.168.1.24	TCP	66	66 443 → 48282 [ACK] Seq=695 Ack=286 Win=422 Len=0 TSval=496444432 TSecr=1507215601
22	5.694501	52.54.204.108	192.168.1.24	TLSv1.2	402	Application Data
23	5.694764	192.168.1.24	52.54.204.108	TCP	66	66 48282 → 443 [ACK] Seq=206 Ack=1520 Win=503 Len=0 TSval=2507216322 TSecr=496444573
24	5.093876	192.168.1.24	52.54.204.108	TLSv1.2	288	APPLICATION DATA
25	5.793543	52.54.204.108	192.168.1.24	TCP	66	66 443 → 48282 [ACK] Seq=1520 Ack=488 Win=422 Len=0 TSval=496444598 TSecr=1507216323

Figura 102 Prueba de tiempo - Apagado de foco desde Google Assistant – Hassio a dispositivo

Fuente: Elaboración Propia

### 3.1.4. Toma de tiempos en la confirmación de peticiones

En la toma de los tiempos de respuesta de los distintos sistemas se ha considerado aplicación desarrollada en la herramienta Node-Red, para conocer el tiempo del proceso de la petición, de los siguientes micrófonos utilizados:

- USB
- ReSpeaker 4-Mic Array

Asistentes de voz:

- Alexa Amazon
- Google Assistant

Para el presente análisis se ha considerado el dispositivo foco en su estado de encendido y apagado, las mismas que se detallan en las siguientes imágenes

### 3.1.4.1. Micrófono USB

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de encendido mediante el micrófono USB como se visualiza en la Figura 103.

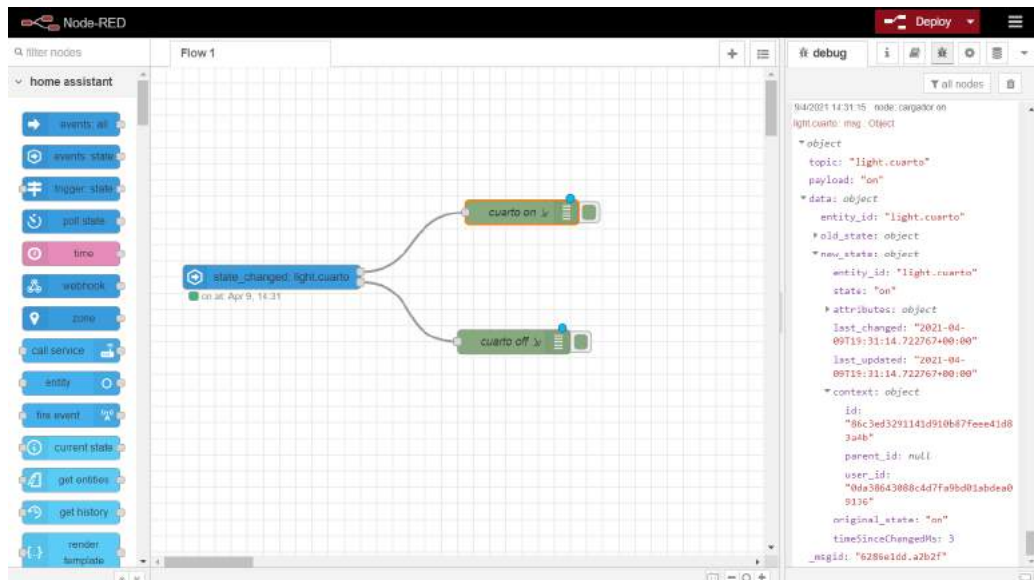


Figura 103 Prueba de tiempo - Encendido de foco desde USB

Fuente: Elaboración Propia

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de apagado mediante el micrófono USB como se visualiza en la Figura 104.

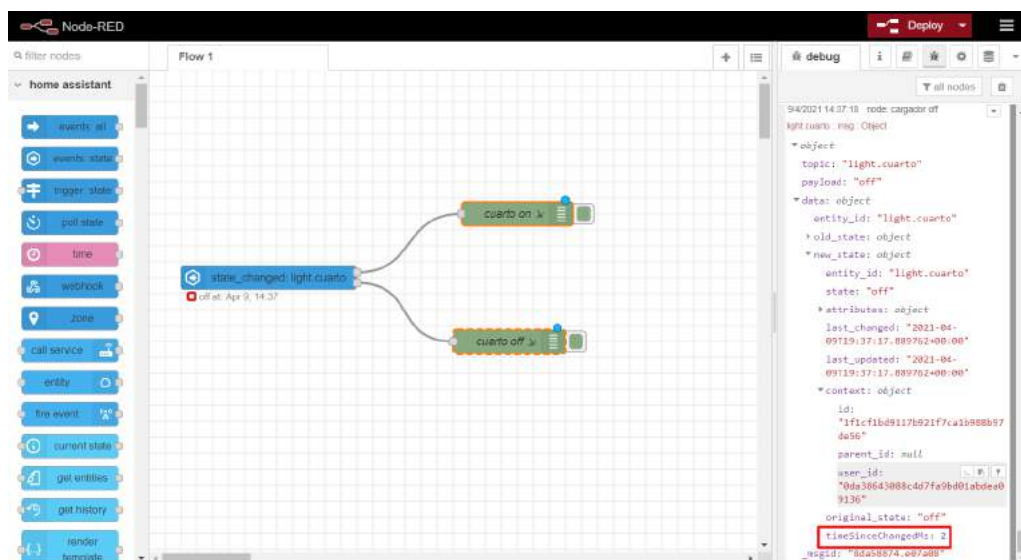


Figura 104 Prueba de tiempo - Apagado de foco desde USB

Fuente: Elaboración Propia

### 3.1.4.2. Micrófono ReSpeaker 4-Mic Array

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de encendido mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 105.

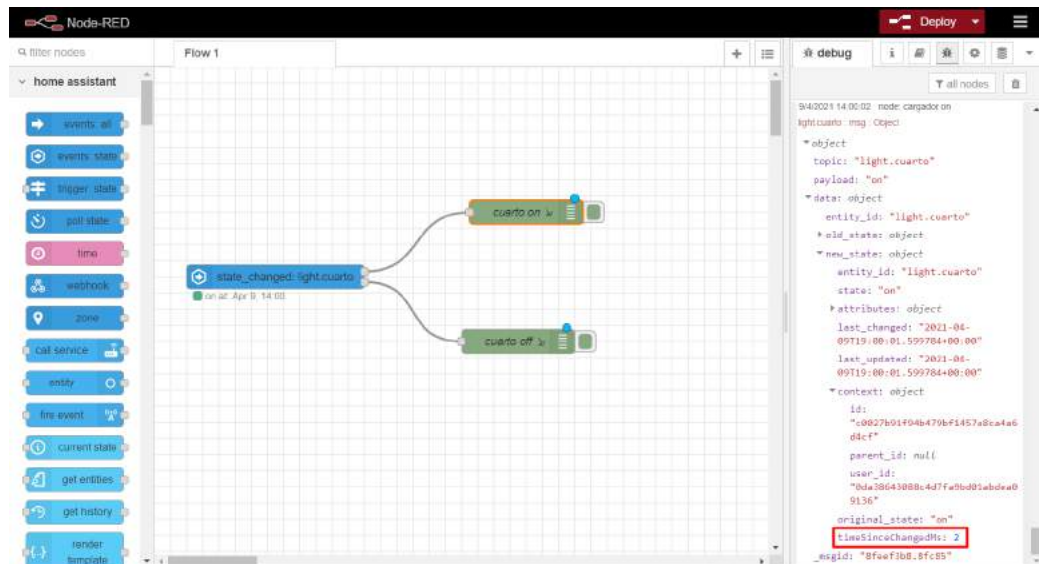


Figura 105 Prueba de tiempo - Encendido de foco desde ReSpeaker 4-Mic Array

Fuente: Elaboración Propia

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de apagado mediante el micrófono ReSpeaker 4-Mic Array como se visualiza en la Figura 106.

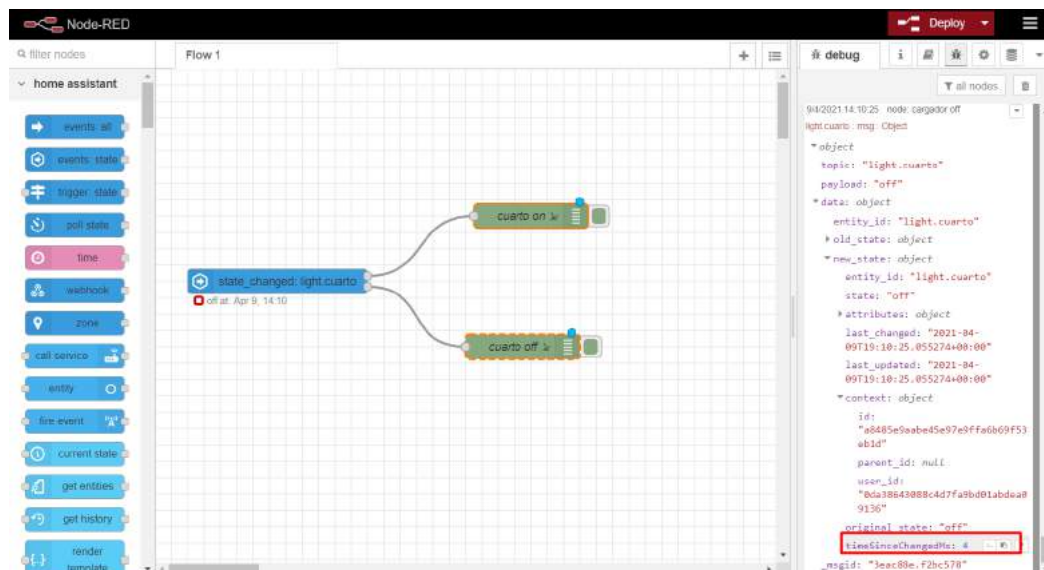


Figura 106 Prueba de tiempo - Apagado de foco desde ReSpeaker 4-Mic Array

Fuente: Elaboración Propia

### 3.1.4.3. Asistente de voz Alexa Amazon

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de encendido mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 107.

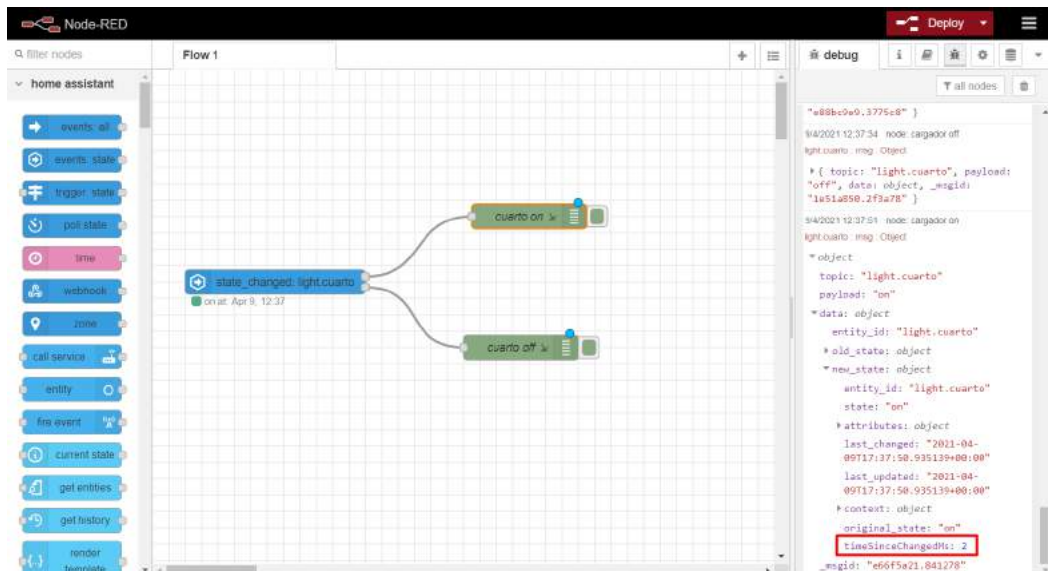


Figura 107 Prueba de tiempo – Encendido de foco desde Alexa Amazon

Fuente: Elaboración Propia

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de apagado mediante el asistente de voz Alexa Amazon como se visualiza en la Figura 108.

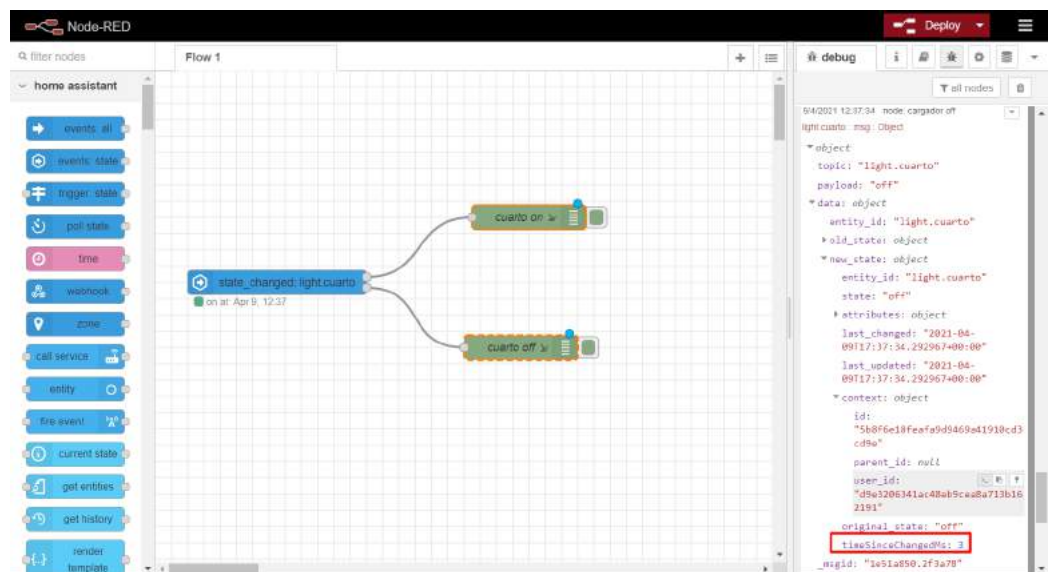


Figura 108 Prueba de tiempo – Apagado de foco desde Alexa Amazon

Fuente: Elaboración Propia



### 3.1.4.4. Asistente de voz Google Assistant

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de encendido mediante el asistente de voz Google Assistant como se visualiza en la Figura 109.

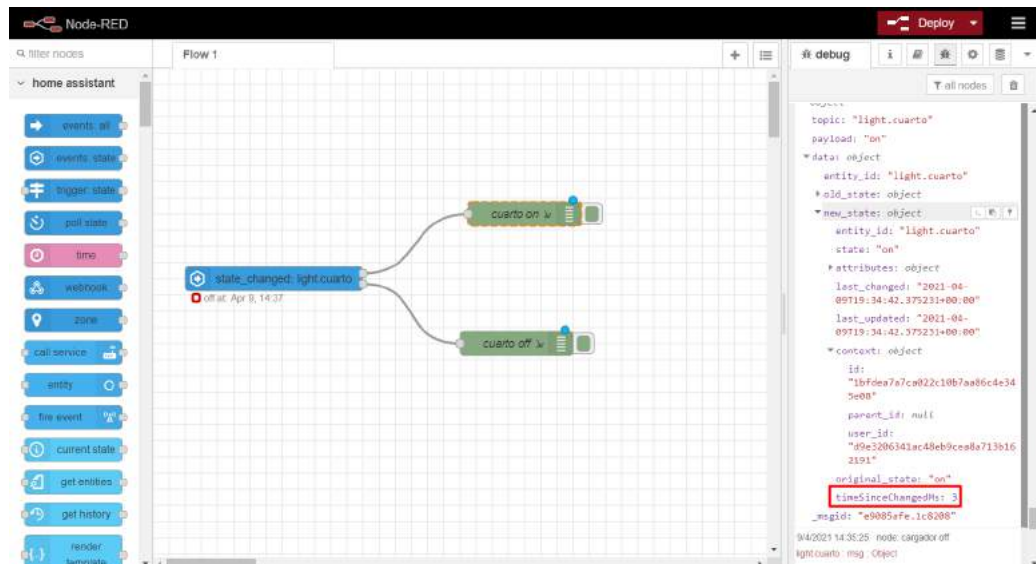


Figura 109 Prueba de tiempo – Encendido de foco desde Google Assistant

Fuente: Elaboración Propia

En la herramienta Node-Red de Home Assistant se obtiene el tiempo de confirmación para la acción de apagado mediante el asistente de voz Google Assistant como se visualiza en la Figura 110.

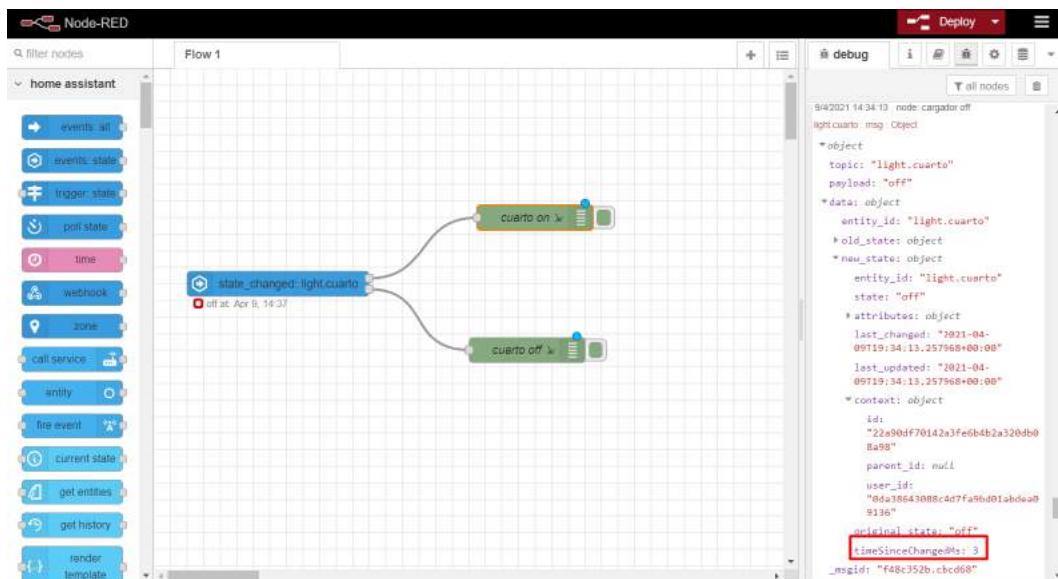


Figura 110 Prueba de tiempo – Encendido de foco desde Google Assistant

Fuente: Elaboración Propia

### **3.2. RESULTADOS DE LA EVALUACIÓN**

El análisis de las tramas realizado con la herramienta Wireshark, ha dejado como resultado el tiempo de envío y confirmación de peticiones en milisegundos, tiempo que es tomado desde el servidor a los dispositivos IoT que se encuentran integrados en el sistema domótico Home Assistant, tomando el estado de encendido y apagado del dispositivo foco.

Para el correcto análisis de los tiempos se ha realizado cien pruebas con cada uno de los micrófonos y los sistemas integrados, para establecer los parámetros de las pruebas se ha considerado:

- Tiempo de reconocimiento de comando de voz en Rhasspy.
- Tiempo de confirmación de petición Rhasspy al sistema domótico.
- Tiempo de confirmación de petición al sistema domótico al dispositivo.
- Acciones de confirmación y fallos.

El total de latencia es la sumatoria de los tres tiempos que da como resultado en una acción empleada en el dispositivo domótico mediante el envío de información dentro del sistema domótico.

Las pruebas son tomadas desde distintos ambientes es decir con ruido y sin ruido y con distintos timbres de voz que se encuentran en hombre, mujer, niño y anciano a una distancia de 1 metro y 4 metros con el propósito de medir el alcance de los micrófonos empleados en la propuesta tecnológica.

El almacenamiento y manejo de los datos es realizado en la herramienta de Microsoft Office, Excel, que permite obtener promedio de los tiempos y a su vez el análisis mediante gráficos estadísticos.

#### **3.2.1. Comparación de dispositivos usados**

Una de las pruebas realizadas en el proyecto fue la respuesta ante distintos timbres de voz, para lo cual se ha tomado las siguientes consideraciones:

- Timbres de voz (hombre, mujer, niño y anciano).
- Distancia (1m y 4m).
- Ambiente con ruido y sin ruido

Estas pruebas ayudaran a medir el cumplimiento de respuesta mediante estado de encendido (ON) y apagado (OFF); se aplican 100 pruebas de si y no para los micrófonos USB y ReSpeaker 4-Mic Array, y para Alexa Amazon y Google Assistant se aplican 50 pruebas con los mismos parámetros de los otros micrófonos.

### 3.2.1.1. Micrófono USB

Tabla 1 Comparación de intentos - USB

Distancia	Estado	Intentos	Hombre	Mujer	Niño	Adulto
1 Metro	On	Válidos	71	50	45	25
		Fallidos	29	50	55	75
	Off	Válidos	74	48	43	27
		Fallidos	26	52	57	73
4 Metros	On	Válidos	51	41	12	12
		Fallidos	49	59	88	88
	Off	Válidos	53	38	14	11
		Fallidos	47	62	86	89
Total			400	400	400	400

Fuente: Elaboración Propia

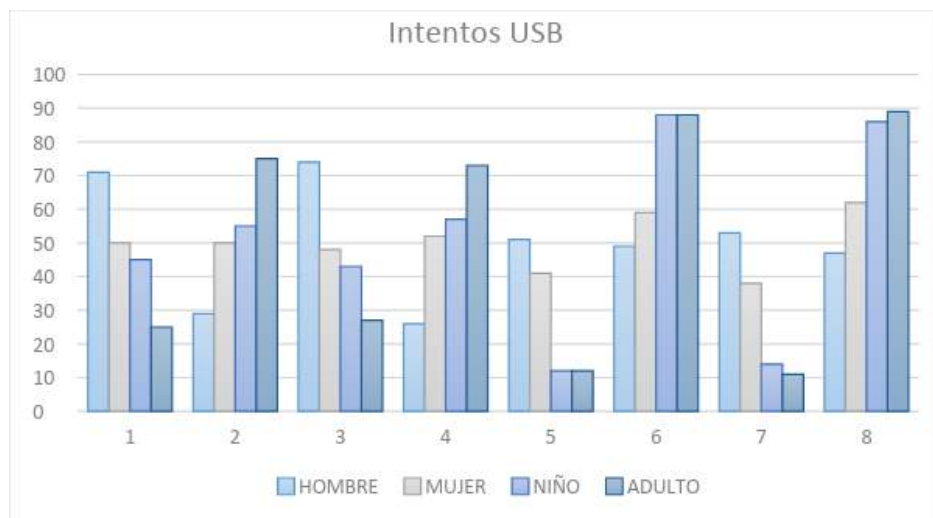


Figura 111 Comparación de intentos - USB

Fuente: Elaboración Propia

**Análisis:** Los resultados de las pruebas en un ambiente sin ruido, dice que un timbre de voz más alto y grave como el de hombre tiene una mayor aceptación ya que tiene un 71 respuestas válidas en el estado on y para off 74, seguido por el de un timbre de una mujer que es de 50 en on y 48 en off, para un niño es de 45 en on y en off de 43 y para un anciano



el 25 y 27, en relación a 1m; pero a 4m solo la voz masculina tiene más aceptación ya que cuenta con un 51 en on y en off con 47, seguido de la voz de una mujer con un 41 en estado de on y 38 en off; por lo que se puede concluir diciendo que el micrófono USB acepta las voces graves y fuertes en relación a las voces finas y bajas.

### 3.2.1.2. Micrófono ReSpeaker 4-Mic Array

Tabla 2 Comparación de Intentos – ReSpeaker 4-Mic

Distancia	Estado	Intentos	Hombre	Mujer	Niño	Adulto
1 Metro	On	Válidos	84	74	66	51
		Fallidos	16	26	34	49
	Off	Válidos	86	79	74	53
		Fallidos	14	21	26	47
4 Metros	On	Válidos	78	62	54	36
		Fallidos	22	38	46	64
	Off	Válidos	70	61	58	38
		Fallidos	30	39	42	62
Total			400	400	400	400

Fuente: Elaboración Propia

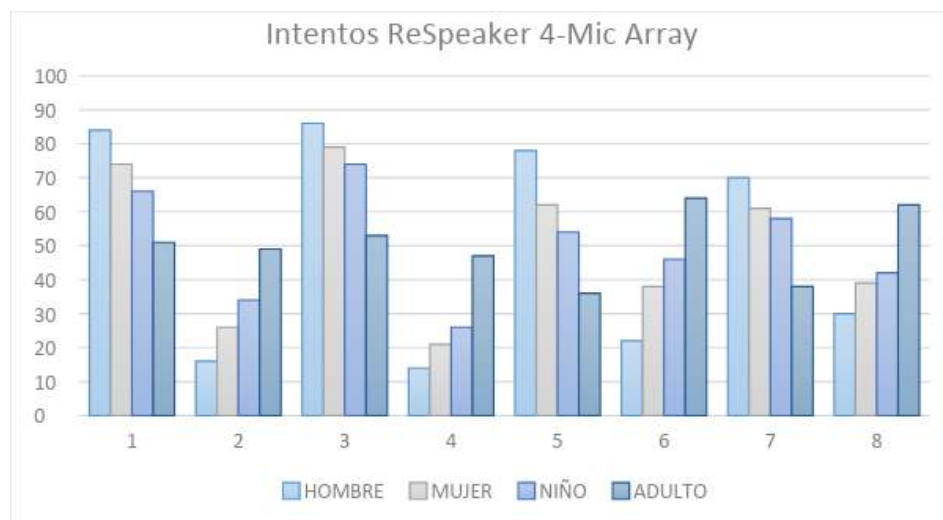


Figura 112 Comparación de Intentos – ReSpeaker 4-Mic

Fuente: Elaboración Propia

**Análisis:** Al revisar los resultados en las pruebas realizadas en un ambiente sin ruido da como resultado que un timbre de voz más alto y grave tiene mayor aceptación en respuestas válidas 84 en el estado on y para off 86, seguido por el de un timbre de una mujer que es de 71 en on

y 79 en off, para un niño es de 66 en on y en off de 74 y para un anciano el 51 y 53 respectivamente, en relación a 1m; pero a 4m solo la voz masculina tiene más aceptación ya que cuenta con un 78 en on y en off con 70; por lo que se puede concluir diciendo que el micrófono 4-Mic acepta las voces graves y fuertes en relación a las voces finas y bajas.

### 3.2.1.3. Asistente de voz Alexa Amazon y Google Assistant

Tabla 3 Comparación de Intentos - Alexa Amazon y Google Assistant

Distancia	Estado	Intentos	Hombre	Mujer	Niño	Adulto
1 Metro	On	Válidos	50	50	50	50
		Fallidos	0	0	0	0
	Off	Válidos	50	50	50	50
		Fallidos	0	0	0	0
4 Metros	On	Válidos	50	50	50	50
		Fallidos	0	0	0	0
	Off	Válidos	50	50	50	50
		Fallidos	0	0	0	0
		Total	200	200	200	200

Fuente: Elaboración Propia

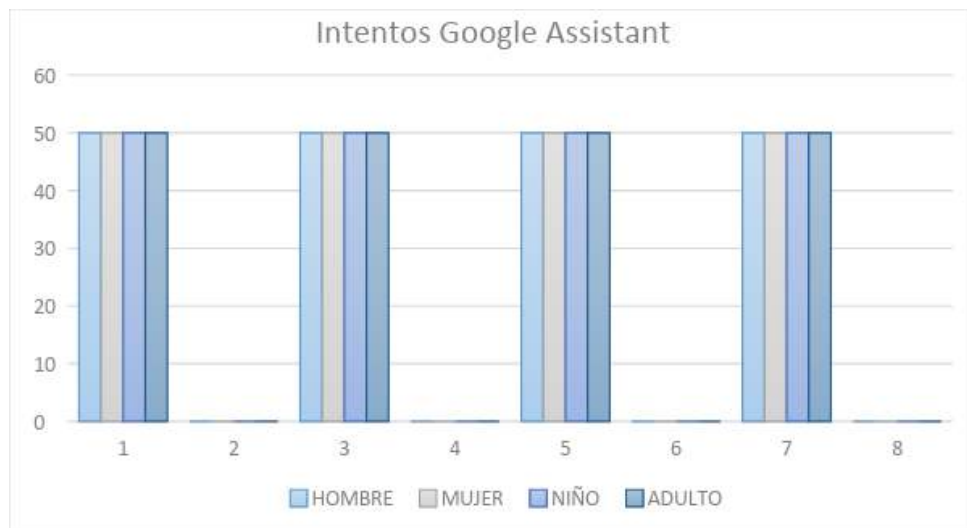


Figura 113 Comparación de Intentos - Alexa Amazon y Google Assistant

Fuente: Elaboración Propia

**Análisis:** Se realizó 50 pruebas tanto en Alexa Amazon y Google Assistant, a una distancia de 1m y 4m para cada dispositivo, lo cual dio como resultado que ante los distintos timbres de voz como hombre, mujer, niño y anciano no presenta problemas ni dificultades para el

reconocimiento de la tarea que se les asigna ante ninguna distancia, ya que estos dispositivos presentan una fuerte sensibilidad ante el reconocimiento del comando más que del timbre de voz.

### 3.2.1.4. Comparación de los micrófonos en ambiente de ruido

Tabla 4 Comparación de los micrófonos en ambiente de ruido

Estado	Distancia	Micrófono	Intentos		Total
			Válidos	Fallidos	
On	1 Metro	Usb	14	86	100
		Mic-4	54	46	100
		G. Assistant	100	0	100
		Alexa	100	0	100
	4 Metro	Usb	12	88	100
		Mic-4	45	55	100
		G. Assistant	100	0	100
		Alexa	100	0	100
Off	1 Metro	Usb	11	89	100
		Mic-4	49	51	100
		G. Assistant	100	0	100
		Alexa	100	0	100
	4 Metro	Usb	15	85	100
		Mic-4	49	51	100
		G. Assistant	100	0	100
		Alexa	100	0	100

Fuente: Elaboración Propia

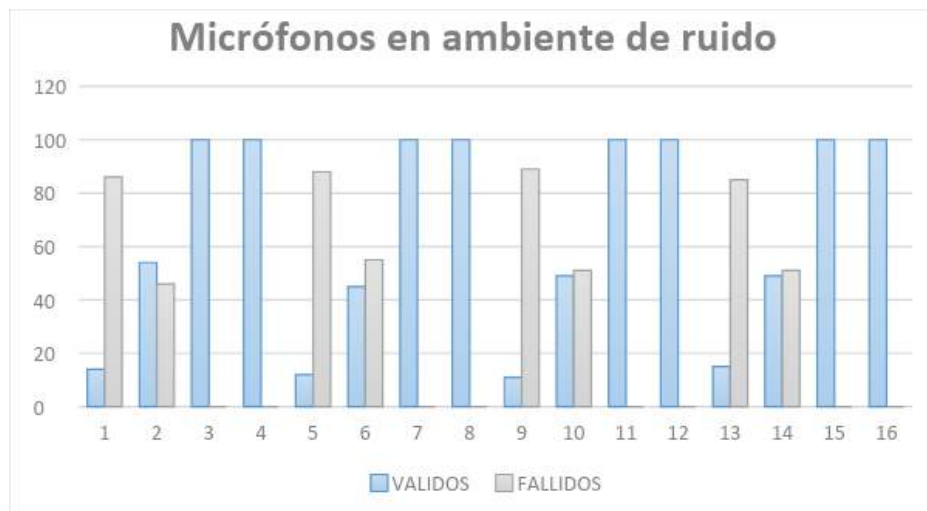


Figura 114 Comparación de los micrófonos en ambiente de ruido

Fuente: Elaboración Propia

**Análisis:** Mediante la prueba anterior se determinó que la mayor parte de aciertos la tiene el timbre de voz grave y fuerte correspondiente a un hombre, es por ello que la presente prueba se toma ese timbre de voz pero en un ambiente de ruido con una distancia de 1m y 4m para lo cual Alexa Amazon y Google Assistant tienen un porcentaje de 100% de aciertos; Mic-4 con 54 aciertos y USB con 14 a una distancia de 1m en on y en off 49 y 11 correspondientes y en relación a la distancia de 4m Alexa Amazon y Google Assistant cuentan con un porcentaje de 100% de aciertos; Mic-4 con 45 aciertos y USB con 12 en el estado de on y para el estado de off 49 para Mic-4 y 15 en USB.

### **3.2.2. Promedio de latencia total**

Para calcular la latencia total primero se obtiene los valores de cada uno de los tiempos mediante el envío de la información que se produce mediante el estado de encendido y apagado del dispositivo.

Se calcula los siguientes tiempos para los micrófonos USB y ReSpeaker 4-Mic Array:

- t1: Tiempo de reconocimiento del comando
- t2: Tiempo de Rhasspy a Home Assistant
- t3: Tiempo de Home Assistant al Dispositivo
- t4: Tiempo del Dispositivo a Home Assistant

Para los asistentes de voz Google Assistant y Alexa Amazon se calcula los siguientes tiempos:

- t2: Tiempo de Google Assistant y/o Alexa Amazon a Home Assistant
- t3: Tiempo de Home Assistant al Dispositivo
- t4: Tiempo del Dispositivo a Home Assistant

Se calcula estos tres tiempos ya que el dispositivo envía la petición directo a la nube sin pasar por el sistema domótico Rhasspy ya que únicamente llega la información a Home Assistant para que el dispositivo domótico realice el estado de encendido o apagado, según la petición.

### 3.2.2.1. USB

Tabla 5 Promedio latencia total - USB

Estado	t1(ms)	t2(ms)	t3(ms)	t4(ms)	Total (ms)
On	453,19	381,197	359,519	1,89	1195,796
Off	351,125	382,124	356,417	1,9	1091,566

Fuente: Elaboración Propia

Las pruebas con el dispositivo foco dan como tiempo de latencia total un promedio de 1195,796 ms en el estado de on y 1091,566 ms en off.

#### 3.2.2.1.1. Latencia de en Rhasspy

Tabla 6 Promedio latencia USB – Reconocimiento de comando en Rhasspy

Estado	t1(ms)	t2(ms)	Total(ms)
On	453,19	381,197	834,387
Off	351,125	382,124	733,249

Fuente: Elaboración Propia

El total de latencia para el tiempo de respuesta en Rhasspy es de 834,387 ms en on y 733,249 en off.

#### 3.2.2.1.2. Latencia de Home Assistant

Tabla 7 Promedio latencia USB – Petición de Rhasspy a Home Assistant

Estado	t3(ms)	t4(ms)	Total(ms)
On	359,519	1,89	361,409
Off	356,417	1,9	358,317

Fuente: Elaboración Propia

El tiempo de respuesta de Home Assistant es de 361,409 ms en on y 358,317 ms en off.

#### 3.2.2.1.3. Latencia del dispositivo

Tabla 8 Promedio latencia USB - Petición de Home Assistant a Rhasspy

Estado	t4(ms)
On	1,89
Off	1,9

Fuente: Elaboración Propia

El dispositivo envía a Home Assistant la respuesta de que se cumplió la petición dando un promedio de 1,89 ms para on y 1,9 ms para off.

### 3.2.2.2. ReSpeaker 4-Mic Array

Tabla 9 Promedio latencia ReSpeaker 4-Mic Array

Estado	t2(ms)	t2(ms)	t2(ms)	t2(ms)	Total(ms)
On	644,23	351,17	332,86	2,41	1330,67
Off	677,70	331,04	312,95	2,45	650,50

Fuente: Elaboración Propia

El tiempo promedio total de latencia para ReSpeaker 4-Mic Array en on es de 1330,67 ms y en off de 650,50 ms.

#### 3.2.2.2.1. Latencia de Rhasspy

Tabla 10 Promedio latencia 4-Mic - Reconocimiento de comando en Rhasspy

Estado	t1(ms)	t2(ms)	Total(ms)
On	644,23	351,17	995,40
Off	677,70	331,04	335,10

Fuente: Elaboración Propia

El tiempo promedio de respuesta de Rhasspy es de 995,40 ms en on y 335,10 ms en off.

#### 3.2.2.2.2. Latencia de Home Assistant

Tabla 11 Promedio latencia 4-Mic - Petición de Rhasspy a Home Assistant

Estado	t3(ms)	t4(ms)	Total(ms)
On	332,86	2,41	335,27
Off	312,95	2,45	315,40

Fuente: Elaboración Propia

El protocolo de comunicación ssh es un y segura.

#### 3.2.2.2.3. Latencia del dispositivo

Tabla 12 Promedio latencia 4-Mic - Petición de Home Assistant al dispositivo

Estado	t4(ms)
On	2,41
Off	2,45

Fuente: Elaboración Propia

Para el promedio de latencia como respuesta del dispositivo es de 2,41 ms en on y 2,45 ms en off.

### 3.2.2.3. Google Assistant

Tabla 13 Promedio latencia Google Assistant

Estado	t2(ms)	t3(ms)	t4(ms)	Total(ms)
On	189,99	209,40	2,47	401,87
Off	194,90	214,74	2,40	412,04

Fuente: Elaboración Propia

El tiempo de promedio total de latencia, para el estado de encendido es de 401,87 ms y para apagado es de 412,04 ms.

#### 3.2.2.3.1. Latencia de Google Assistant a Home Assistant

Tabla 14 Promedio latencia Google Assistant - Petición de Home Assistant

Estado	t2(ms)
On	189,99
Off	194,90

Fuente: Elaboración Propia

El tiempo de latencia es de un promedio de 189,99 ms para on y 194,90 ms para off.

#### 3.2.2.3.2. Latencia de Home Assistant al dispositivo

Tabla 15 Promedio latencia Google Assistant - Petición de Home Assistant al dispositivo

Estado	t3(ms)	t4(ms)	Total(ms)
On	209,41	2,47	211,88
Off	214,74	2,40	217,14

Fuente: Elaboración Propia

La latencia promedio de respuesta de Home Assistant es de 211,88 ms para on y 217,14 para off.

#### 3.2.2.3.3. Latencia del dispositivo a Home Assistant

Tabla 16 Promedio latencia Google Assistant - Confirmación del dispositivo a Home Assistant

Estado	t4(ms)
On	2,47
Off	2,40

Fuente: Elaboración Propia

El promedio de tiempo de latencia del dispositivo a Home Assistant para on es de 2,47 ms y para off es de 2,40 ms.

### 3.2.2.4. Alexa Amazon

Tabla 17 Promedio latencia Alexa Amazon

Estado	t2(ms)	t3(ms)	t4(ms)	Total(ms)
On	172,51	186,23	2,00	360,74
Off	172,91	186,93	1,93	361,77

Fuente: Elaboración Propia

El tiempo total de promedio de latencia para Alexa Amazon es de 360,74 ms en el estado de on y para el estado de off es de 361,77 ms.

#### 3.2.2.4.1. Latencia de Alexa Amazon a Home Assistant

Tabla 18 Promedio latencia Alexa Amazon - Petición de Alexa a Home Assistant

Estado	t2(ms)
On	172,51
Off	172,91

Fuente: Elaboración Propia

El promedio de latencia desde Alexa Amazon a Home Assistant es de 172,51 ms para on y 172,91 ms para off.

#### 3.2.2.4.2. Latencia de Home Assistant al dispositivo

Tabla 19 Promedio latencia Alexa Amazon - Petición de Home Assistant al dispositivo

Estado	t3(ms)	t4(ms)	Total(ms)
On	186,23	2,00	188,23
Off	186,93	1,93	188,86

Fuente: Elaboración Propia

El promedio total de latencia entre Home Assistant al dispositivo es de 188,23 ms para on y 188,86 ms para off.

#### 3.2.2.4.3. Latencia del dispositivo a Home Assistant

Tabla 20 Promedio latencia Alexa Amazon - Confirmación del dispositivo a Home Assistant

Estado	t4(ms)
On	2,00
Off	1,93

Fuente: Elaboración Propia

El promedio de latencia del dispositivo para on es de 2,00 ms y para off de 1,93 ms.



### 3.2.3. Comparación de latencia entre los dispositivos evaluados

Tabla 21 Comparación de latencia - Promedio total de los dispositivos

Estado	Dispositivo	t1(ms)	t2(ms)	t3(ms)	t4(ms)	TOTAL(ms)
On	USB	453,19	381,20	359,52	1,89	1195,80
	4-Mic	644,23	351,17	332,86	2,41	1330,67
	G. Assistant	NA	189,99	209,40	2,47	401,87
	Alexa	NA	172,51	186,23	2,00	360,74
Off	USB	351,13	382,12	356,42	1,90	1091,57
	4-Mic	677,70	331,04	312,95	2,45	1324,14
	G. Assistant	NA	194,90	214,74	2,40	412,04
	Alexa	NA	172,91	186,93	1,93	361,77

Fuente: Elaboración Propia

Las pruebas que se realizaron a los sistemas domóticos dan como resultado el tiempo de latencia que interviene en cada uno de los micrófonos con los dispositivos domóticos empleados en el proyecto, los cuales se pueden administrar de manera remota o de manera local ya que están instalados localmente los dispositivos y las plataformas domóticas utilizadas.

El micrófono que tiende a responder de manera más tardía es el ReSpeaker 4-Mic Array con un promedio de 1330,67 ms con respecto al encendido del dispositivo foco y 1324,14 ms para apagar el mismo, a diferencia del USB que tiene 1195,80 ms para on y 1091,57 para off ya que estos son los micrófonos que hemos empleado para el manejo de los sistemas domóticos locales y están siendo probados.

A diferencia de Google Assistant y Alexa Amazon que tienen una diferencia considerable ya que están siendo enviados los datos directamente a la nube con la variante de que estos son sistemas de voz perfectamente trabajados dando como resultado 401,87 ms para el encendido del dispositivo foco y 412,04 ms para el apagado del foco para Google Assistant; y para Alexa Amazon 360,74 ms en on y en off 361,77 ms.

El sistema domótico en cuanto a funcionamiento tiene un buen resultado ya que prácticamente es creado con hardware a bajo costo y en cuanto a la comunicación de los dispositivos lo realiza en un tiempo relativamente corto,

los sistemas domóticos están instalados para el funcionamiento de manera local.

### **3.3. CONCLUSIONES**

- Se logró ampliar los conocimientos de sistemas domóticos para el desarrollo de la propuesta tecnológica gracias a la investigación bibliográfica desarrollada y el apoyo de tesis realizadas con la aplicación de domótica que forma parte de Internet de las Cosas.
- El prototipo de la presente propuesta tecnológica concluyó con la solución al problema como guía del manejo de los sistemas domóticos utilizados y conocer el flujo de los datos enviados desde el usuario hasta los dispositivos domóticos.
- En el proyecto por motivo de pruebas se usó una máquina virtual para la instalación y ejecución de la plataforma domótica Home Assistant donde se administran los dispositivos IoT, a nivel profesional genera problemas, al apagar la computadora que contiene la máquina virtual los servicios se apagan, la solución a estos inconvenientes es instalar el sistema domótico en una placa Raspberry Pi.
- Se dispone de un sistema de reconocimiento de comandos de voz, Rhasspy, en una placa de Raspberry Pi, el cual es un sistema de herramientas desplegadas donde se establece los comandos de voz a utilizar los mismos que son enviados al sistema domótico Home Assistant mediante servidores externos vía HTTP que ordenan a los dispositivos IoT realizar una petición.
- Para hacer el sistema domótico más atractivo al usuario se implementó un modelo en 3D del plano del hogar con la finalidad de encender y apagar los dispositivos desde el mismo, se trabajó imágenes del modelo del hogar en el estado de encendido y apagado que forman parte del código del modelo en 3D dentro del sistema domótico Home Assistant, en el estado de pruebas se comprobó que dicho modelo no afecta al tiempo de respuesta ante el envío de información dentro del sistema.
- En el ambiente de pruebas para este proyecto se utilizó distintos timbres de voz (Hombre, mujer, niño y anciano) empleados en

micrófonos y asistentes de voz para el análisis de latencia mediante herramienta Wireshark y la aplicación desarrollada en Node-Red garantizando al usuario un manejo satisfactorio a la hora de gestionar los dispositivos IoT por comandos de voz.

### **3.4. RECOMENDACIONES**

- Home Assistant es un sistema domótico que integra dispositivos IoT indistintamente del fabricante, por medio de su estandarización se comunican los dispositivos que intercambian datos; se recomienda instalar dicho sistema domótico en una placa Raspberry Pi para el acceso a los dispositivos vía internet sin la necesidad de levantar los servicios cada que se prenda la computadora que lo contiene ya que esto resulta tedioso al usuario al momento de gestionar su sistema domótico.
- Para trabajos futuros se recomienda una optimización de los micrófonos utilizados en la plataforma domótica Rhasspy, así mismo se podría implementar dispositivos como reproductores de audio o cámaras que resultaría más interesante y llamativo al usuario haciendo del sistema domótico más fácil de manejar y gestionar.

## BIBLIOGRAFÍA

- [1] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez, y G. Belduma-Vacacela, «IoT Android Gateway for Monitoring and Control a WSN», en *Technology Trends*, vol. 798, M. Botto-Tobar, N. Esparza-Cruz, J. León-Acurio, N. Crespo-Torres, y M. Beltrán-Mora, Eds. Cham: Springer International Publishing, 2018, pp. 18-32.
- [2] M. Li, W. Gu, W. Chen, Y. He, Y. Wu, y Y. Zhang, «Smart Home: Architecture, Technologies and Systems», *Procedia Comput. Sci.*, vol. 131, pp. 393-400, ene. 2018, doi: 10.1016/j.procs.2018.04.219.
- [3] D. L. Hernández-Rojas, T. M. Fernández-Caramés, P. Fraga-Lamas, y C. J. Escudero, «Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications», *Sensors*, vol. 18, n.º 1, Art. n.º 1, ene. 2018, doi: 10.3390/s18010057.
- [4] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo, y A. Pan, «SCADA System Based on IoT for Intelligent Control of Banana Crop Irrigation», *Applied Technologies. ICAT 2019. Communications in Computer and Information Science*, vol. 1193, pp. 243-256, 2020, doi: [https://doi.org/10.1007/978-3-030-42517-3\\_19](https://doi.org/10.1007/978-3-030-42517-3_19). [5] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, y A. Pan, «Rules engine and complex event processor in the context of internet of things for precision agriculture», *Comput. Electron. Agric.*, vol. 154, pp. 347-360, nov. 2018, doi: 10.1016/j.compag.2018.09.013.
- [6] J. J. C. Calva, D. L. H. Rojas, R. F. M. Román, y C. D. R. García, «Seguridad IoT: Principales amenazas en una taxonomía de activos», *HAMUT'AY*, vol. 7, n.º 3, Art. n.º 3, ene. 2021, doi: 10.21503/hamu.v7i3.2192.
- [7] L. N. Phangbertha, A. Fitri, I. Purnamasari, y Y. Muliono, «Smart Socket for Electricity Control in Home Environment», *Procedia Comput. Sci.*, vol. 157, pp. 465-472, ene. 2019, doi: 10.1016/j.procs.2019.09.002.
- [8] D. Hernandez Rojas, B. Mazon Olivo, J. Novillo Vicuña, C. Escudero Cascon, A. Pan Bermudez, y G. Belduma Vacacela, «IoT Android Gateway for Monitoring and Control a WSN | SpringerLink». [https://link.springer.com/chapter/10.1007/978-3-319-72727-1\\_2](https://link.springer.com/chapter/10.1007/978-3-319-72727-1_2) (accedido abr. 19, 2021).
- [9] S. Nikou, «Factors driving the adoption of smart home technology: An empirical assessment», *Telemat. Inform.*, vol. 45, p. 101283, dic. 2019, doi: 10.1016/j.tele.2019.101283.
- [10] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, y N. Ghani, «Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations», *IEEE Commun. Surv. Tutor.*, vol. 21, n.º 3, pp. 2702-2733, thirdquarter 2019, doi: 10.1109/COMST.2019.2910750.
- [11] S. N. Swamy y S. R. Kota, «An Empirical Study on System Level Aspects of Internet of Things (IoT)», *IEEE Access*, vol. 8, pp. 188082-188134, 2020, doi: 10.1109/ACCESS.2020.3029847.
- [12] W. A. Jabbar *et al.*, «Design and Fabrication of Smart Home With Internet of Things Enabled Automation System», *IEEE Access*, vol. 7, pp. 144059-144074, 2019, doi: 10.1109/ACCESS.2019.2942846.

- [13] M. W. Condry y C. B. Nelson, «Using Smart Edge IoT Devices for Safer, Rapid Response With Industry IoT Control Operations», *Proc. IEEE*, vol. 104, n.º 5, pp. 938-946, may 2016, doi: 10.1109/JPROC.2015.2513672.
- [14] F. Shi, Q. Li, T. Zhu, y H. Ning, «A Survey of Data Semantization in Internet of Things», *Sensors*, vol. 18, n.º 1, Art. n.º 1, ene. 2018, doi: 10.3390/s18010313.
- [15] S. Uma, R. Eswari, R. Bhuvanya, y G. S. Kumar, «IoT based Voice/Text Controlled Home Appliances», *Procedia Comput. Sci.*, vol. 165, pp. 232-238, ene. 2019, doi: 10.1016/j.procs.2020.01.085.
- [16] D. Lu, C. Ding, J. Xu, y S. Wang, «Hierarchical Discriminant Analysis», *Sensors*, vol. 18, n.º 1, Art. n.º 1, ene. 2018, doi: 10.3390/s18010279.
- [17] G. Pau, C. Chaudet, D. Zhao, y M. Collotta, «Next Generation Wireless Technologies for Internet of Things», *Sensors*, vol. 18, n.º 1, Art. n.º 1, ene. 2018, doi: 10.3390/s18010221.
- [18] L. G. O. Crespo y J. B. C. Mejía, «Análisis de las estrategias aplicadas en el desarrollo de sistemas domóticos de seguridad», *Dominio Las Cienc.*, vol. 6, n.º 3, pp. 342-363, 2020.
- [19] B. S. Panca y C. Wijaya, «Data Transmission Performance Analysis on the Usage of ZLib Compression over Secure Shell Protocol», *Int. J. Technol. Bus.*, vol. 3, n.º 1, Art. n.º 1, 2019.
- [20] P. McLaren, G. Russell, W. J. Buchanan, y Z. Tan, «Decrypting live SSH traffic in virtual environments», *Digit. Investig.*, vol. 29, pp. 109-117, jun. 2019, doi: 10.1016/j.diin.2019.03.010.
- [21] E. Al-Masri *et al.*, «Investigating Messaging Protocols for the Internet of Things (IoT)», *IEEE Access*, vol. 8, pp. 94880-94911, 2020, doi: 10.1109/ACCESS.2020.2993363.
- [22] S. M. M. Fattah, N.-M. Sung, I.-Y. Ahn, M. Ryu, y J. Yun, «Building IoT Services for Aging in Place Using Standard-Based IoT Platforms and Heterogeneous IoT Products», *Sensors*, vol. 17, n.º 10, Art. n.º 10, oct. 2017, doi: 10.3390/s17102311.
- [23] A. Stoica, T. Kadar, C. Lemnar, R. Potolea, y M. Dînşoreanu, «Intent Detection and Slot Filling with Capsule Net Architectures for a Romanian Home Assistant», *Sensors*, vol. 21, n.º 4, Art. n.º 4, ene. 2021, doi: 10.3390/s21041230.
- [24] G. K. Adam, «DALI LED Driver Control System for Lighting Operations Based on Raspberry Pi and Kernel Modules», *Electronics*, vol. 8, n.º 9, Art. n.º 9, sep. 2019, doi: 10.3390/electronics8091021.
- [25] Y. Mao, X. Li, Y. Jia, S. Zhao, y Y. Zhang, «Security Analysis of Smart Home Based on Life Cycle», en *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, ago. 2019, pp. 1444-1449, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00261.
- [26] N. Modrzyk, *Real-Time IoT Imaging with Deep Neural Networks: Using Java on the Raspberry Pi 4*. Berkeley, CA: Apress, 2020.
- [27] L. Xiao, X. Wan, X. Lu, Y. Zhang, y D. Wu, «IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?», *IEEE Signal Process. Mag.*, vol. 35, n.º 5, pp. 41-49, sep. 2018, doi: 10.1109/MSP.2018.2825478.

- [28] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, y P. Burnap, «A Supervised Intrusion Detection System for Smart Home IoT Devices», *IEEE Internet Things J.*, vol. 6, n.º 5, pp. 9042-9053, oct. 2019, doi: 10.1109/JIOT.2019.2926365.
- [29] A. A. Joshua, «Internet of Things (IoT): The Technology, Architecture and Applications – Prospects in Nigeria», *Internet Things Cloud Comput.*, vol. 8, n.º 4, Art. n.º 4, dic. 2020, doi: 10.11648/j.iotcc.20200804.11.
- [30] R. Karunamoorthi *et al.*, «Design and development of IoT based home computerization using Raspberry pi», *Mater. Today Proc.*, dic. 2020, doi: 10.1016/j.matpr.2020.10.673.
- [31] D. E. N. Ganesh, «Health Monitoring System using Raspberry Pi and IOT», *Orient. J. Comput. Sci. Technol.*, vol. 12, n.º 1, pp. 08-13, doi: 10.13005/ojcs12.01.03.
- [32] S. J. Johnston y S. J. Cox, «The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams», *Electronics*, vol. 6, n.º 3, Art. n.º 3, sep. 2017, doi: 10.3390/electronics6030051.
- [33] B. Sudharsan y M. Chockalingam, «A Microphone Array and Voice Algorithm based Smart Hearing Aid», *Int. J. Comput. Appl.*, vol. 178, n.º 41, pp. 1-6, ago. 2019, doi: 10.5120/ijca2019919295.
- [34] J. Novillo-Vicuña, D. H. Rojas, B. M. Olivo, J. M. Ríos, y O. C. Villavicencio, *Arduino y el Internet de las cosas*. 3Ciencias, 2018.
- [35] W. W. Gibbs, «Build your own Amazon Echo - Turn a PI into a voice controlled gadget [Resources\_Hands on]», *IEEE Spectr.*, vol. 54, n.º 5, pp. 20-21, may 2017, doi: 10.1109/MSPEC.2017.7906891.