



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

**Desarrollo e implementación de un sistema domótico utilizando el protocolo
standard matter.**

**PAMBI CRUZ KENNETH AARON
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**FAJARDO MUNZON MARCO ANDRES
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA
2023**



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo e implementación de un sistema domótico utilizando el protocolo standard matter.

**PAMBI CRUZ KENNETH AARON
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**FAJARDO MUNZON MARCO ANDRES
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA
2023**



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

PROPUESTAS TECNOLÓGICAS

Desarrollo e implementación de un sistema domótico utilizando el protocolo standard matter.

**PAMBI CRUZ KENNETH AARON
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**FAJARDO MUNZON MARCO ANDRES
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

HERNANDEZ ROJAS DIXYS LEONARDO

**MACHALA
2023**

Grupo10_Pambi Kenneth y
Fajardo
Marco_Turnitin_Proyecto_Titula
cion.pdf
por KENNETH PAMBI CRUZ

Fecha de entrega: 22-mar-2024 12:21p.m. (UTC-0500)

Identificador de la entrega: 2328031073

Nombre del archivo: Grupo10_Pambi_Kenneth_y_Fajardo_Marco_Turnitin_Proyecto_Titulacion.pdf (4.18M)

Total de palabras: 26659

Total de caracteres: 141237

Grupo10_Pambi Kenneth y Fajardo

Marco_Turnitin_Proyecto_Titulacion.pdf

INFORME DE ORIGINALIDAD

1 %	1 %	0 %	%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.utmachala.edu.ec Fuente de Internet	< 1 %
2	revistas.ups.edu.ec Fuente de Internet	< 1 %
3	sedici.unlp.edu.ar Fuente de Internet	< 1 %
4	www.coursehero.com Fuente de Internet	< 1 %
5	lebergolutions.com 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

Los que suscriben, PAMBI CRUZ KENNETH AARON y FAJARDO MUNZÓN MARCO ANDRÉS, en calidad de autores del siguiente trabajo escrito titulado Desarrollo e implementación de un sistema domótico utilizando el protocolo standard Matter, otorgan 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 tienen potestad para otorgar los derechos contenidos en esta licencia.

Los autores declaran 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.

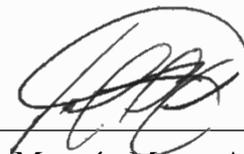
Los autores como garantes de la autoría de la obra y en relación a la misma, declaran que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asumen la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

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.



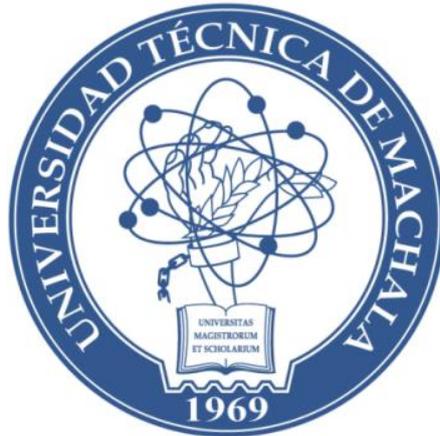
Pambi Cruz Kenneth Aaron

0705924868



Fajardo Munzón Marco Andrés

0706809456



UTMACH

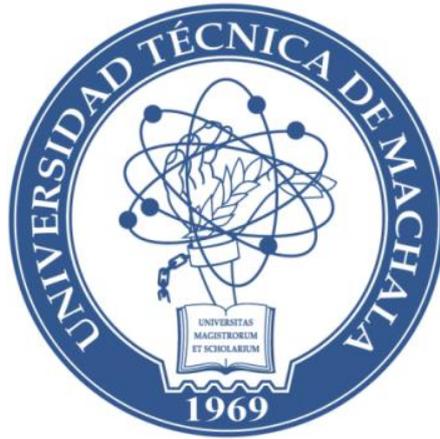
FACULTAD DE INGENIERÍA CIVIL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

PAMBICRUZ KENNETH AARON
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

FAJARDO MUNZÓN MARCO ANDRÉS
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

MACHALA

2024



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO
UTILIZANDO EL PROTOCOLO STANDARD MATTER.

PAMBI CRUZ KENNETH AARON

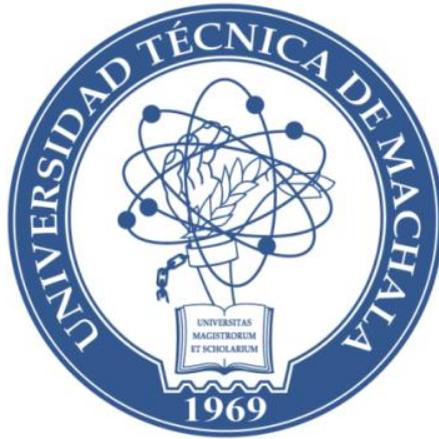
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

FAJARDO MUNZÓN MARCO ANDRÉS

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

MACHALA

2024



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO
UTILIZANDO EL PROTOCOLO STANDARD MATTER.

PAMBICRUZ KENNETH AARON

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

FAJARDO MUNZÓN MARCO ANDRÉS

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

ING. HERNÁNDEZ ROJAS DYXIS LEONARDO, PHD.

MACHALA

2024

DEDICATORIA

Dedico esta tesis a mis queridos padres, Fausto y Glenda, cuyo inquebrantable apoyo y sacrificio han sido fundamentales en mi camino académico. A mi padre, por su constante respaldo económico y por llevarme en su taxi a la universidad, y a mi madre, por sus sabios consejos y cuidado desde mi infancia. Su amor y apoyo han sido la fuerza que me ha impulsado a alcanzar este logro. Con profunda gratitud, les dedico este trabajo como expresión de mi eterno agradecimiento.

A mi querida tía Helen, quien ha sido mi segunda madre. Tu constante presencia y amor desinteresado han sido fundamentales en mi vida. Este trabajo es un homenaje a tu influencia positiva. Te dedico este trabajo con profunda admiración y cariño.

Pambi Cruz Kenneth Aaron

Quiero dedicar este trabajo a mi querida familia, a mis padres y hermana, quienes han sido mi mayor apoyo desde el principio. Su amor y apoyo incondicional han sido fundamentales en mi vida académica y personal. Agradezco a mis padres por su constante aliento y por guiarme en el camino correcto, y por siempre impartirme esos buenos consejos de nunca dejar el estudio. Les dedico este trabajo con todo mi amor y gratitud.

Marco Andrés Fajardo Munzón

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mis padres, Fausto y Glenda, por su apoyo incondicional y sacrificio, que han sido fundamentales en mi desarrollo académico y personal. Su amor y aliento han sido mi inspiración.

Asimismo, deseo reconocer a mi querido amigo Jeandry Mayon, quien durante siete años ha sido un compañero leal y un sostén invaluable. Su generosidad y apoyo económico han aliviado mis preocupaciones financieras y han hecho posible la adquisición de los materiales necesarios para mi formación universitaria. Su amistad ha sido un regalo preciado que valoro enormemente.

Agradezco especialmente a mi amigo Marco Fajardo, quien ha sido un pilar constante a lo largo de todos los semestres de nuestra carrera. Su amistad sincera y disposición para ayudarme en cualquier momento han sido fundamentales para superar los desafíos académicos. Estoy profundamente agradecido por su apoyo incondicional.

Finalmente, agradezco a Daniel Villano, quien ha sido un amigo constante y me ha ayudado en tareas de programación que no comprendía. Su amistad y asistencia académica han sido de gran valor para mí. Estoy eternamente agradecido por su apoyo.

Pambi Cruz Kenneth Aaron

Agradezco de corazón a mis padres y a mi hermana por estar siempre presentes y apoyarme en los momentos en que más lo he necesitado. Su ayuda y compañía han sido fundamentales para mí. Su amor incondicional y su apoyo constante me han dado fuerzas para enfrentar los desafíos de la vida. Estoy profundamente agradecido por tenerlos a mi lado.

De igual manera deseo integrar a este agradecimiento a mis amigos Villano Daniel, y Jeandry Mayon por su gran amistad y conocimientos a lo largo de carrera que me han brindado para superarme cada día más y a mi mejor amigo y compañero de Tesis Pambi Kenneth, que, gracias a su capacidad para resolver los problemas y continua motivación en todos estos años de conocerlo, se logró forjar un lazo de amistad en cual agradezco infinitamente.

Fajardo Munzón Marco Andrés

RESUMEN

La tecnología IoT, se ha extendido a diferentes dominios de aplicación, por ejemplo, la domótica, la cual permite automatizar objetos cotidianos, ya sea en un ámbito empresarial o en el hogar. En la actualidad existen diferentes protocolos de comunicación de dispositivos IoT, que se tiene en un hogar, pero en los últimos años la creciente adopción de la tecnología Matter y su potencial para facilitar la interoperabilidad y el control de dispositivos domésticos inteligentes, ha dado paso al desarrollo de un sistema domótico utilizando este protocolo. Para resolver el problema de la falta de estándares comunes en la domótica, permitiendo la integración fluida de dispositivos de diferentes fabricantes y tecnologías. Se llevo a cabo en configuración e instalación de un Gateway basado en un ESP32 Border Router, aparte de esto usando dispositivos ESP32, con soporte Matter. Se incorporo dispositivos Matter comerciales, asistentes virtuales de diferentes marcas, para demostrar la interoperabilidad. Además, se integró el sistema con Home Assistant para el control centralizado de los dispositivos domóticos.

Los resultados dados incluyeron un sistema domótico funcional aplicándola de manera local que utilice el protocolo Matter para la comunicación entre dispositivos IoT, demostrando la interoperabilidad con los demás dispositivos comerciales IoT. También se logró una integración exitosa en la plataforma Home Assistant para facilitar el control y la gestión de los dispositivos.

En conclusión, esto se enfoca en superar las limitaciones en la domótica, particularmente en la falta de estándares comunes. Mediante la implementación de un sistema utilizando el protocolo estándar Matter y el hardware ESP32, el objetivo es facilitar la interoperabilidad y el control de dispositivos inteligentes, permitiendo la integración de equipos de diferentes fabricantes y tecnologías. La incorporación de dispositivos comerciales Matter y la integración con HASSIO contribuirá a la evaluación y control centralizado de los dispositivos en el hogar.

Palabras clave: IoT, protocolo matter, ESP32, protocolo thread, casa automatizada, gateway matter.

ABSTRACT

IoT technology has expanded to various application domains, such as home automation, which allows for the automation of everyday objects, whether in a business setting or at home. Currently, there are different communication protocols for IoT devices used in a household, but in recent years, the growing adoption of Matter technology and its potential to facilitate interoperability and control of smart home devices has led to the development of a home automation system using this protocol. This aims to solve the problem of lacking common standards in home automation, enabling seamless integration of devices from different manufacturers and technologies. Configuration and installation of an ESP32 Border Router-based Gateway were carried out, along with using ESP32 devices with Matter support. Commercial Matter devices and virtual assistants from different brands were incorporated to demonstrate interoperability. Additionally, the system was integrated with Home Assistant for centralized control of home automation devices.

The results included a functional home automation system using the Matter protocol for local communication between IoT devices, showcasing interoperability with other commercial IoT devices. Successful integration with the Home Assistant platform was also achieved to facilitate device control and management.

In conclusion, this focuses on overcoming limitations in home automation, particularly the lack of common standards. By implementing a system using the standard Matter protocol and ESP32 hardware, the goal is to facilitate interoperability and control of smart devices, allowing integration of equipment from different manufacturers and technologies. The inclusion of commercial Matter devices and integration with HASSIO will contribute to the assessment and centralized control of devices in the home.

Key Words: IoT, Matter protocol, ESP32, Thread protocol, automated home, Matter gateway.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	18
i. Declaración y formulación del Problema	19
Formulación del problema	20
Problema Principal:.....	20
ii. Objeto de estudio y Campo de acción.....	20
Objeto de estudio:.....	20
Campo de acción:	20
iii. Objetivos.....	20
Objetivo General.....	20
Objetivos específicos	20
iv. Hipótesis y variables o Preguntas de investigación	21
Hipótesis principal.....	21
Variables y dimensionamiento	21
v. Justificación	22
vi. Organización del documento.....	23
CAPÍTULO I. MARCO TEÓRICO	24
1.1. Antecedentes de la Investigación.....	24
1.2. Antecedentes históricos	30
1.2.1. Historia de la IoT.....	30
1.2.2. Historia de la Domótica.....	31
1.2.3. Historia de los Protocolos de Comunicación de IoT.....	32
1.3. Antecedentes Teóricos.....	33
1.3.1. Internet de las Cosas	33
1.3.2. Domótica	35
1.3.3. Protocolos de Conexión.....	35
1.3.4. Protocolos de Aplicación de IoT en domótica	38
1.3.5. Estándar Matter	40

1.3.6.	Herramientas para la construcción del sistema domótico con protocolo Matter.	49
1.4.	Antecedentes Contextuales	53
1.4.1.	Delimitación del contexto de Estudio	53
1.4.2.	Impacto Social del protocolo Matter.....	53
1.4.3.	Ámbito de aplicación	53
1.4.4.	Establecimiento de requerimientos.....	54
CAPÍTULO II. DESARROLLO DEL PROTOTIPO		56
2.1.	Definición del prototipo.....	56
2.2.	Metodología de desarrollo del prototipo.....	57
2.2.1.	Enfoque, alcance y diseño de investigación.....	57
2.2.2.	Unidades de análisis.....	58
2.2.3.	Técnicas e instrumentos de recopilación de datos.....	58
2.2.4.	Técnicas de procesamiento de datos para la obtención de resultados.....	58
2.2.5.	Metodología o métodos específicos.....	58
2.2.6.	Herramientas y/o Materiales	60
2.3.	Desarrollo del prototipo.....	60
2.3.1.	FASE 1: Planeación	61
2.3.2.	FASE 2: Diseño.....	63
2.3.3.	FASE 3: Desarrollo.....	66
2.3.4.	FASE 4: Pruebas.....	82
2.4.	Ejecución del prototipo.....	88
CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO		98
3.1.	Plan de evaluación	98
3.1.1.	Objetivo.....	98
3.1.2.	Cronograma.....	98
3.1.3.	Métodos de evaluación.....	99
REFERENCIAS BIBLIOGRÁFICAS.....		141
ANEXOS.....		145
	Anexo 1 – Dispositivos ESP32 Terminados.....	145

Anexo 2 - Observaciones de Evaluación Escenario 1	147
Anexo 3 – Plantilla de Evaluación por voz terminada (Escenario 1).....	148
Anexo 4 – Plantilla de Evaluación por Interfaz terminada (Escenario 1).....	149
Anexo 5 – Evaluación del rendimiento por Voz (Escenario 1).....	150
Anexo 6 – Evaluación del rendimiento por Interfaz (Escenario 1).....	151
Anexo 7 - Observaciones de Evaluación Escenario 2	152
Anexo 8 - Plantilla de Evaluación por voz terminada (Escenario 2)	153
Anexo 9 - Plantilla de Evaluación por Interfaz terminada (Escenario 2).....	154
Anexo 10 – Evaluación del rendimiento por Voz (Escenario 2).....	155
Anexo 11 - Evaluación del rendimiento por Interfaz (Escenario 2)	156

ÍNDICE DE FIGURAS

Figura 1. Árbol de problema.....	19
Figura 2. Estado del arte.....	24
Figura 3. Proceso y resultado de la búsqueda.....	27
Figura 4. Documentos publicados por año.....	28
Figura 5. Documentos encontrados en las diferentes bases de datos.....	28
Figura 6. Análisis de contenidos - VOSviewer.....	29
Figura 7. Historia de la IoT.....	30
Figura 8. Historia de la Domótica.....	31
Figura 9. Historia de los protocolos de comunicación de IoT.....	32
Figura 10. Matter protocolo universal.....	41
Figura 11. Arquitectura SDK de Espressif para Matter.....	42
Figura 12. Soluciones de Espressif con Matter.....	47
Figura 13. Plataformas de Materia Espressif.....	48
Figura 14. Módulo ESP32-S3 y Módulo ESP32 C3.....	50
Figura 15. Módulo ESP32 – H2 MINI.....	51
Figura 16. Definición de las capas del prototipo.....	56
Figura 17. Fases de la metodología XP.....	59
Figura 18. Diseño del prototipo del dispositivo esp32 para de sensor de temperatura y pulsador.....	63
Figura 19. Diseño del Prototipo del dispositivo led.....	64
Figura 20. Diagrama de Flujo del sistema domótico matter.....	65
Figura 21. Ventana de descarga de código matter.....	66
Figura 22. Código del protocolo Matter.....	67
Figura 23. Selección de la librería matter.....	67
Figura 24. Ventana de subida de protocolo matter.....	68
Figura 25. Carpeta que contiene el protocolo matter.....	68
Figura 26. Compilación del código de protocolo matter.....	68
Figura 27. Compilación completada y subida al esp32-S3.....	69
Figura 28. Activación de C++17 en el IDE Arduino.....	69
Figura 29. Importación de las librerías y configuración de los pines.....	70
Figura 30. Configuración de clúster y nodos.....	71
Figura 31. Código de inicio de dispositivo Matter.....	72
Figura 32. Código de la función loop.....	72
Figura 33. Codificación del sensor DHT y esp32.....	73
Figura 34. Establecimiento de funciones en para la interacción matter.....	74
Figura 35. Implementación del código para el inicio de los eventos y atributos.....	74

Figura 36. Codificación para el inicio y lectura de la temperatura del sensor	75
Figura 37. Resultado del proceso para la creación de código manual.	76
Figura 38. Codificación para la configuración del proveedor DAC.....	76
Figura 39. Codificación para el inicio del nodo Matter	76
Figura 40. Codificación para generar Código Qr	76
Figura 41. Resultado del proceso para la creación de código Qr	77
Figura 42. Codificación del repositorio espressif	77
Figura 43. Codificación para instalar la librería esp-idf.....	78
Figura 44. Codificación para implementar el proyecto de Gateway.....	78
Figura 45. Codificación para iniciar y proyecto y sus configuraciones.....	79
Figura 46. Configuración de Gateway	79
Figura 47. Configuración de Red del Gateway	80
Figura 48. Configuración de la clave de la Red thread	80
Figura 49. Codificación de la instalación del Firmware.....	81
Figura 50. Verificación de la conexión de la red.....	81
Figura 51. Resultado en consola de la instalación del Firmware del Gateway	81
Figura 52. Vinculación de código Matter.....	82
Figura 53. Emparejamiento de Dispositivo Matter.....	83
Figura 54. Error de vinculación proporcionado por Google Home.....	83
Figura 55. Emparejamiento desde Alexa app	84
Figura 56. Entorno del dispositivo matter vinculado.....	84
Figura 57. Home Assistant.....	85
Figura 58. Entorno de la aplicación Home Assistant en el celular	85
Figura 59. Búsqueda del protocolo de vinculación (Matter)	86
Figura 60. Instalación de servicio matter.....	86
Figura 61. Interfaz del ingreso de código matter.....	87
Figura 62. Conexión exitosa del dispositivo.....	87
Figura 63. Interfaz de encendido y apagado del dispositivo Matter	88
Figura 64. Resultado de la ejecución del código de foco led matter	88
Figura 65. Código Qr de vinculación de dispositivo matter	89
Figura 66. Proceso de emparejamiento de Alexa app	89
Figura 67. Selección del dispositivo propio matter	90
Figura 68. Proceso de Vinculación de Dispositivo con código.....	91
Figura 69. Vinculación exitosa de dispositivos propios.....	91
Figura 70. Encendido de led Matter.....	92
Figura 71. Prototipo con ESP32-S3 desarrollado y funcionando - Primera Parte.....	92
Figura 72. Prototipo con ESP32-S3 desarrollado y funcionando - Segunda Parte.....	93

Figura 73. Figura 59. Prototipo con ESP3-C3 como Sensor	93
Figura 74. Sistema domótico con los dispositivos personalizados.....	94
Figura 75. Sistema domótico con los dispositivos personalizados finalizados.....	94
Figura 76. Conexión de dispositivo Thread con Asistente Virtual.....	95
Figura 77. Conexión con el Gateway Thread con soporte Matter.....	96
Figura 78. Conexión Exitosa del dispositivo Thread con el Gateway	96
Figura 79. Prueba exitosa del uso del dispositivo Thread	97
Figura 80. Resultados de Tiempo de respuesta del comando Enciende Luz Matter (Escenario 1).....	107
Figura 81. Resultados de Tiempo de respuesta del comando Enciende Luz Thread (Escenario 1).	108
Figura 82. Resultados de Tiempo de respuesta del comando Decir Temperatura (Escenario 1).....	108
Figura 83. Resultados de Tiempo de respuesta Enciende foco matter comercial (Escenario 1).....	109
Figura 84. Resultados de Tiempo de respuesta del comando Enciende Switch (Escenario 1).	110
Figura 85. Resultados de Tiempo de respuesta del comando Apaga Luz Matter (Escenario 1).	110
Figura 86. Resultados de Tiempo de respuesta del comando Apaga Switch Matter (Escenario 1).....	111
Figura 87. Resultados de Tiempo de respuesta comando Apaga foco Matter Comercial (Escenario 1).112	
Figura 88. Resultados de Tiempo de respuesta del comando apaga Luz Thread (Escenario 1).	112
Figura 89. Resultados de Tiempo de respuesta del comando Encienda Sala (Escenario 1).	113
Figura 90. Resultados de Tiempo de respuesta del comando Enciende Sala de Cine (Escenario 1).....	114
Figura 91. Resultados de Tiempo de respuesta del comando Apagar Sala (Escenario 1).....	114
Figura 92. Resultados de Tiempo de respuesta del comando Apaga Sala de Cine (Escenario 1).....	115
Figura 93. Tiempo de respuesta del dispositivo ESP32 S3 (Escenario 1).....	117
Figura 94. Tiempo de respuesta del dispositivo ESP C3 Switch (Escenario 1).	117
Figura 95. Tiempo de respuesta del dispositivo ESP32 C3 Sensor (Escenario 1).....	118
Figura 96. Tiempo de respuesta del dispositivo ESP32 H2 (Escenario 1).	118
Figura 97. Tiempo de respuesta del dispositivo Matter comercial (Escenario 1).....	119
Figura 98. Resultados de Tiempo de respuesta del comando Enciende Luz Matter (Escenario 2).....	120
Figura 99. Resultados de Tiempo de respuesta del comando Enciende Luz Thread (Escenario 2).	121
Figura 100. Resultados de Tiempo de respuesta del comando Decir Temperatura (Escenario 2).	122
Figura 101. Resultados de Tiempo de respuesta Enciende foco matter comercial (Escenario 2).	123
Figura 102. Resultados de Tiempo de respuesta del comando Enciende Switch (Escenario 2).	124
Figura 103. Resultados de Tiempo de respuesta del comando Apaga Luz Matter (Escenario 2).....	125
Figura 104. Resultados de Tiempo de respuesta del comando Apaga Switch Matter (Escenario 2).....	126
Figura 105. Resultados de Tiempo de respuesta Apaga foco Matter Comercial (Escenario 2).....	127
Figura 106. Resultados de Tiempo de respuesta del comando apaga Luz Thread (Escenario 2).....	128
Figura 107. Resultados de Tiempo de respuesta del comando Encienda Sala (Escenario 2).	129
Figura 108. Resultados de Tiempo de respuesta del comando Enciende Sala de Cine (Escenario 2)....	130
Figura 109. Resultados de Tiempo de respuesta del comando Apagar Sala (Escenario 2).	131

Figura 110. Resultados de Tiempo de respuesta del comando Apaga Sala de Cine (Escenario 2). 132

Figura 111. Tiempo de respuesta del dispositivo ESP32 S3 (Escenario 2). 134

Figura 112. Tiempo de respuesta del dispositivo ESP32 C3 Switch (Escenario 2). 135

Figura 113. Tiempo de respuesta del dispositivo ESP32 C3 Sensor (Escenario 2). 136

Figura 114. Tiempo de respuesta del dispositivo ESP32 H2 (Escenario 2). 137

Figura 115. Tiempo de respuesta del dispositivo Matter comercial (Escenario 2). 138

ÍNDICE DE TABLAS

Tabla 1. Conceptualización de las variables	21
Tabla 2. Variables y dimensionamientos.....	22
Tabla 3. Preguntas de investigación.....	25
Tabla 4. Criterios de inclusión y exclusión en español	26
Tabla 5. Requerimientos del prototipo	55
Tabla 6. Técnicas e instrumentos de recopilación de datos	58
Tabla 7. Metodología de Desarrollo del sistema	59
Tabla 8. Herramientas o Materiales	60
Tabla 9. Roles del equipo de trabajo.....	61
Tabla 10. Resumen de historias de usuarios	62
Tabla 11. Plan de iteraciones.....	62
Tabla 12. Cronograma del plan de Evaluación.....	98
Tabla 13. Criterios de evaluación	99
Tabla 14. Pruebas de Aceptación.....	99
Tabla 15. Plantilla de Observaciones de la evaluación	103
Tabla 16. Plantilla de Evaluación por control de voz.....	104
Tabla 17. Plantilla de evaluación del rendimiento por voz.....	105
Tabla 18. Latencia de Dispositivos ESP32 con protocolo Matter (Escenario 1).....	116
Tabla 19. Latencia de Dispositivos esp32 con protocolo Matter (Escenario 2).	133

GLOSARIO

A

Automatización: Proceso de utilización de tecnología para realizar tareas o procesos sin intervención humana directa. A menudo se busca la automatización con el objetivo de incrementar la eficiencia, la precisión y/o eliminar tareas repetitivas.

E

Ethernet: Estándar de comunicación para redes locales (LAN) basado en el envío de paquetes. Se utiliza principalmente para establecer conexiones a nivel de enlace de datos y físico, y ha sido una base fundamental para la creación de redes en muchos entornos.

Estándar: Conjunto de normas o especificaciones técnicas que buscan garantizar que productos, procesos o servicios cumplan con ciertos requisitos de calidad o funcionamiento. Los estándares permiten la compatibilidad e interoperabilidad entre diferentes productos y sistemas.

I

Interoperabilidad: Capacidad de diferentes sistemas, aplicaciones o componentes para trabajar juntos y compartir información o funcionalidades de manera efectiva, sin importar las diferencias en sus diseños o implementaciones iniciales.

M

Metodología: Conjunto estructurado de métodos, procesos o técnicas que se siguen en una disciplina particular. Se utiliza para guiar el desarrollo, la investigación o la solución de problemas en un campo específico.

P

Protocolo: Conjunto de reglas y convenciones que determinan cómo se debe transmitir y recibir información entre dos o más entidades en un sistema de comunicación. Estas reglas garantizan que los dispositivos puedan entenderse y compartir información de manera efectiva.

R

Redundante: Característica de un sistema o componente que tiene componentes adicionales, copias o procesos para asegurar su funcionamiento en caso de fallo de uno de sus elementos. La redundancia se utiliza para mejorar la fiabilidad y disponibilidad de sistemas y redes.

INTRODUCCIÓN

En la era actual, donde la tecnología y la conectividad son esenciales para la vida cotidiana, la domótica ha emergido como una solución innovadora para mejorar la calidad de vida y la eficiencia en los hogares. Sin embargo, a pesar de los avances tecnológicos, la industria de la domótica enfrenta desafíos significativos, principalmente la falta de interoperabilidad entre dispositivos de diferentes fabricantes debido a la diversidad de protocolos de comunicación, como se dice en [1], habla sobre la importancia de la conectividad a Internet para el control remoto de diferentes dispositivos. Esta situación se ha llevado a un entorno fragmentado y complejo, donde los usuarios finales se ven obligados a elegir entre diferentes protocolos y plataformas, dificultando la creación de un sistema domótico integrado.

El protocolo Matter surge como una solución prometedora a este problema, ofreciendo un estándar común que busca unificar la comunicación entre dispositivos domóticos. En este contexto, el presente proyecto tiene como objetivo principal desarrollar e implementar un sistema domótico local basado en el protocolo Matter, que permita la integración y control centralizado de dispositivos domóticos comerciales y personalizados, se presenta Matter como un estándar de conectividad de código abierto y sin regalías desarrollado por la Connectivity Standards Alliance (CSA-IoT) [2]. A través de la implementación de un Gateway basado en ESP32 Border Router y ESP32 de diferentes modelos como dispositivos con protocolo Matter, y su integración con la plataforma Home Assistant, se busca brindar a los usuarios una experiencia de hogar inteligente más satisfactoria, eficiente y unificada.

La relevancia de este proyecto radica en su contribución al avance tecnológico en el campo de la domótica, demostrando la viabilidad y eficiencia del protocolo Matter en la creación de sistemas domóticos integrados. Además, se espera que los resultados obtenidos impulsen la adopción de esta tecnología en el mercado, beneficiando tanto a fabricantes como a usuarios finales.

i. Declaración y formulación del Problema

En el campo de la domótica, el Internet de las Cosas, una tecnología transformadora que conecta objetos inteligentes de manera única, integrando el mundo físico, biológico y digital en una nueva era [3]. Existe una diversidad de dispositivos y protocolos que dificultan la interoperabilidad y la comunicación fluida entre ellos, ya que cuentan con diferentes tipos de protocolos de comunicación, esto presenta un obstáculo para los usuarios finales que desean controlar y gestionar sus dispositivos domóticos de manera centralizada y eficiente.

Además, la falta de un estándar común para la comunicación entre dispositivos domóticos ha dado lugar a un entorno confuso y complejo. Los usuarios se ven obligados a elegir entre diferentes protocolos y plataformas, lo que dificulta la selección de los dispositivos adecuados y la creación de un sistema integrado, la ausencia de una solución estándar dificulta la escalabilidad y expansión del sistema domótico a medida que se agregan nuevos dispositivos y funcionalidades.

El Ecuador como país tercermundista, en los hogares estándares, no tienen conocimiento y familiaridad de los protocolos y tecnologías disponibles, para la utilización de sus productos inteligentes que tienen y que pueden mejorarlos. Existe un desconocimiento generalizado sobre las opciones existentes y las ventajas que puede ofrecer un estándar usual en términos de facilidad de uso, seguridad y flexibilidad.

En la **Figura 1**, se presenta el problema y a su alrededor las causas y efectos del problema.

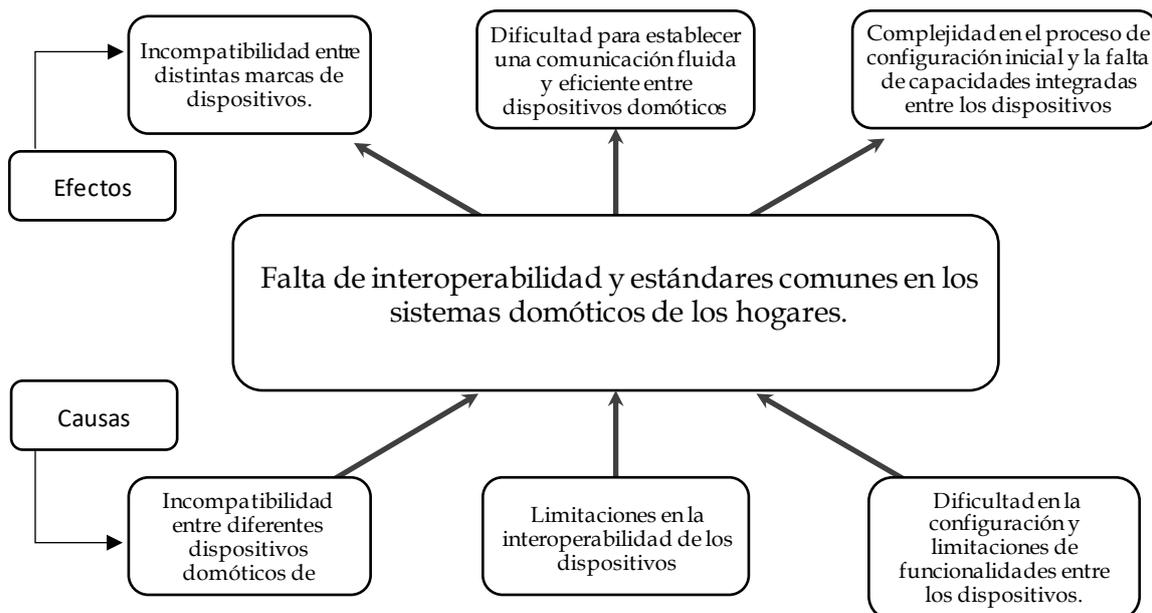


Figura 1. Árbol de problema

Formulación del problema

Problema Principal:

¿Cómo implementar un sistema domótico local para lograr una integración y control centralizados de dispositivos de un hogar inteligente, de manera local?

Problemas específicos:

- ¿Cuáles son los requisitos técnicos y de hardware necesarios para implementar el protocolo Matter en dispositivos personalizados ESP32?
- ¿Cómo se puede garantizar la interoperabilidad y la comunicación fluida entre los dispositivos Matter comerciales y dispositivos personalizados desarrollados?
- ¿Cuál es la mejor estrategia para integrar dispositivos Matter con la plataforma Home Assistant, asegurando un control completo y conveniente de los dispositivos domóticos del sistema?

ii. Objeto de estudio y Campo de acción

Objeto de estudio:

- Desarrollo e implementación de un sistema domótico de manera local, utilizando el protocolo estándar Matter.

Campo de acción:

- Tecnología domótica y protocolo estándar Matter.

iii. Objetivos

Objetivo General

Implementar un sistema domótico local basado en el protocolo estándar Matter para la integración y control centralizado de dispositivos comerciales y personalizados, ofreciendo a los usuarios un control eficiente y unificado.

Objetivos específicos

- Recopilar y analizar fuentes bibliográficas para la redacción de la fundamentación teórica del proyecto.

- Realizar un firmware y software necesario en los dispositivos ESP32 que serán los dispositivos Matter, y ESP32 Border Router que servirá como un Gateway para habilitar la funcionalidad del protocolo Matter.
- Aplicar estrategias para lograr la interoperabilidad y comunicación fluida entre los dispositivos Matter comerciales y los dispositivos personalizados de desarrollo propio.
- Integrar el sistema domótico basado en Matter con la plataforma Home Assistant, permitiendo un control completo y conveniente de los dispositivos domóticos.

iv. Hipótesis y variables o Preguntas de investigación

Hipótesis principal

Mediante la implementación de un sistema domótico local basado en el protocolo Matter y utilizando dispositivos ESP32, se logra una integración y control centralizados que facilitan una comunicación eficiente y una interoperabilidad optimizada entre todos los dispositivos. Con el ESP32 Border Router como puerta de enlace, se permite el control de dispositivos que utilizan el protocolo Thread. En este entorno domótico con aplicación del protocolo Matter, cualquier dispositivo ESP32 ofrece un tiempo de respuesta máximo de 3 segundos.

Variables y dimensionamiento

A continuación, se presenta una breve descripción conceptual de las variables independientes y dependientes en la **Tabla 1**. Por otro lado:

Tabla 1. Conceptualización de las variables

Variables	Conceptos
Variable independiente: Implementación de un sistema domótico con el estándar matter y dispositivos ESP32.	La implementación de dispositivos domóticos utilizando el protocolo Matter, se complementará en la adopción y configuración de dispositivos inteligentes a instalar en los ESP32 que cumplen con el estándar Matter para la comunicación en el sistema domótico. Esto implica la selección, instalación y configuración de dispositivos compatibles con Matter.
Variable dependiente: Mejora en la integración, comunicación e interoperabilidad de dispositivos en el sistema domótico.	La mejora en la integración y control de dispositivos en el sistema domótico se refiere a la optimización y eficiencia en la comunicación y coordinación de los dispositivos domóticos implementados (ESP32). Esto incluye la capacidad de interconectar y controlar de manera eficiente los dispositivos Matter, así como la integración con un sistema centralizado como Home Assistant para una gestión más completa y sencilla de los dispositivos.

En la **Tabla 2** se detallan las variables, categorías, indicadores y métodos que surgieron durante la elaboración de la hipótesis de este estudio.

Tabla 2. Variables y dimensionamientos

Variables	Categorías	Indicadores	Técnicas
Variable independiente: Implementación de un sistema domótico, con protocolo matter y dispositivos ESP32.	1. Instalación de dispositivos Matter en ESP32 2. Integración del sistema domótico	1. Configuración y puesta en marcha de los ESP32 para admitir la comunicación a través del protocolo Matter. 2. Instalación física y conexión de los dispositivos compatibles con Matter en los ESP32.	1. Revisión y análisis de la documentación técnica y especificaciones de los dispositivos domóticos compatibles con Matter. 2. Uso de herramientas de software y configuración para establecer la comunicación entre los dispositivos Matter y otros componentes del sistema domótico.
Variable dependiente: Mejora en la integración, comunicación e interoperabilidad de dispositivos en el sistema domótico.	1. Comunicación efectiva entre dispositivos 2. Control centralizado y gestión simplificada para usuario final.	1. Transmisión de datos entre los dispositivos en el sistema domótico. 2. Capacidad para gestionar y controlar múltiples dispositivos de forma unificada y sencilla. 3. Facilidad de uso y accesibilidad de la interfaz de control centralizado, como Home Assistant.	1. Pruebas de compatibilidad entre dispositivos domóticos de diferentes marcas y protocolos de comunicación para verificar su capacidad de trabajar juntos. 2. Evaluación de la interfaz y la usabilidad del sistema centralizado, como Home Assistant, a través de pruebas de usuario y encuestas.

v. Justificación

La importancia de la nueva tecnología Matter en lo que tiene que ver con el ámbito de la domótica y su creciente aceptación en la sociedad actual. Existe una necesidad de dar paso al desarrollo e implementación de un sistema domótico utilizando el protocolo estándar Matter debido a los escasos estándares comunes en la industria, el cual dificulta la interoperabilidad y la integración de dispositivos de diferentes fabricantes masivos de hoy en día que existe en la sociedad. El presente estudio se propone demostrar la eficiencia de Matter, y viabilidad en la creación de un sistema domótico integrado y controlado centralmente.

La solución que se ha planteado implica el desarrollo e implementación de un Gateway basado en un ESP32 Border Router, los cuales serán configurados con Matter para poder establecer una conexión con otros dispositivos domóticos, tanto comerciales como los de elaboración propia. Los

dispositivos que se implementara tanto propios como comerciales estarán equipados con Matter que será su protocolo principal, pero contarán con otros protocolos de comunicación en este caso usamos Thread, lo que permitirá una representación completa del sistema domótico y la interconexión.

En este proyecto, además, se ha utilizado un sistema en Home Assistant para el control centralizado de todos los dispositivos, incluyendo los ESP32 con Matter, que se desarrollan, permitiendo al usuario final tener un control completo sobre los dispositivos domóticos.

Los beneficios que se obtienen se extienden a diferentes niveles, a nivel científico, el cual se contribuirá al conocimiento sobre la tecnología Matter, su historia, la arquitectura y su implementación en la domótica. A nivel tecnológico, el cual se podrá demostrar la interoperabilidad, la integración de dispositivos utilizando Matter y otros protocolos de comunicación. A nivel Económico, el cual se impulsará la adopción de la tecnología Matter en el mercado de la domótica, brindando a los usuarios finales una solución eficiente y compatibilidad con diferentes dispositivos; y, por último, a nivel social, el cual se mejorará la experiencia del usuario final al proporcionar un control centralizado y simplificado de sus dispositivos domóticos.

vi. Organización del documento

La disposición de este trabajo investigativo se ha estructurado en capítulos, los cuales se describen de la siguiente manera:

Capítulo I: En esta sección inicial, se abordan los fundamentos teóricos y contextuales relacionados con la temática en cuestión, proporcionando una base sólida para la comprensión del proyecto.

Capítulo II: Este capítulo se centra en el diseño y desarrollo del sistema domótico, detallando las metodologías adoptadas, las herramientas utilizadas y la implementación del protocolo Matter en dispositivos específicos.

Capítulo III: Aquí se presenta el proceso de evaluación del sistema, destacando la interoperabilidad y comunicación entre dispositivos. Además, se discuten los resultados obtenidos y se comparan con las expectativas iniciales.

CAPÍTULO I. MARCO TEÓRICO

1.1. Antecedentes de la Investigación

A continuación, se usará la Metodología de Revisión Sistemática de la Literatura (SRL: Systematic Review of the Literature), esta metodología se compone de diferentes etapas, incluyendo la formulación de preguntas de investigación, la identificación de palabras claves y la creación de una cadena de búsqueda, la aplicación de criterios de inclusión y exclusión, y por último el proceso y los resultados de la búsqueda [4].

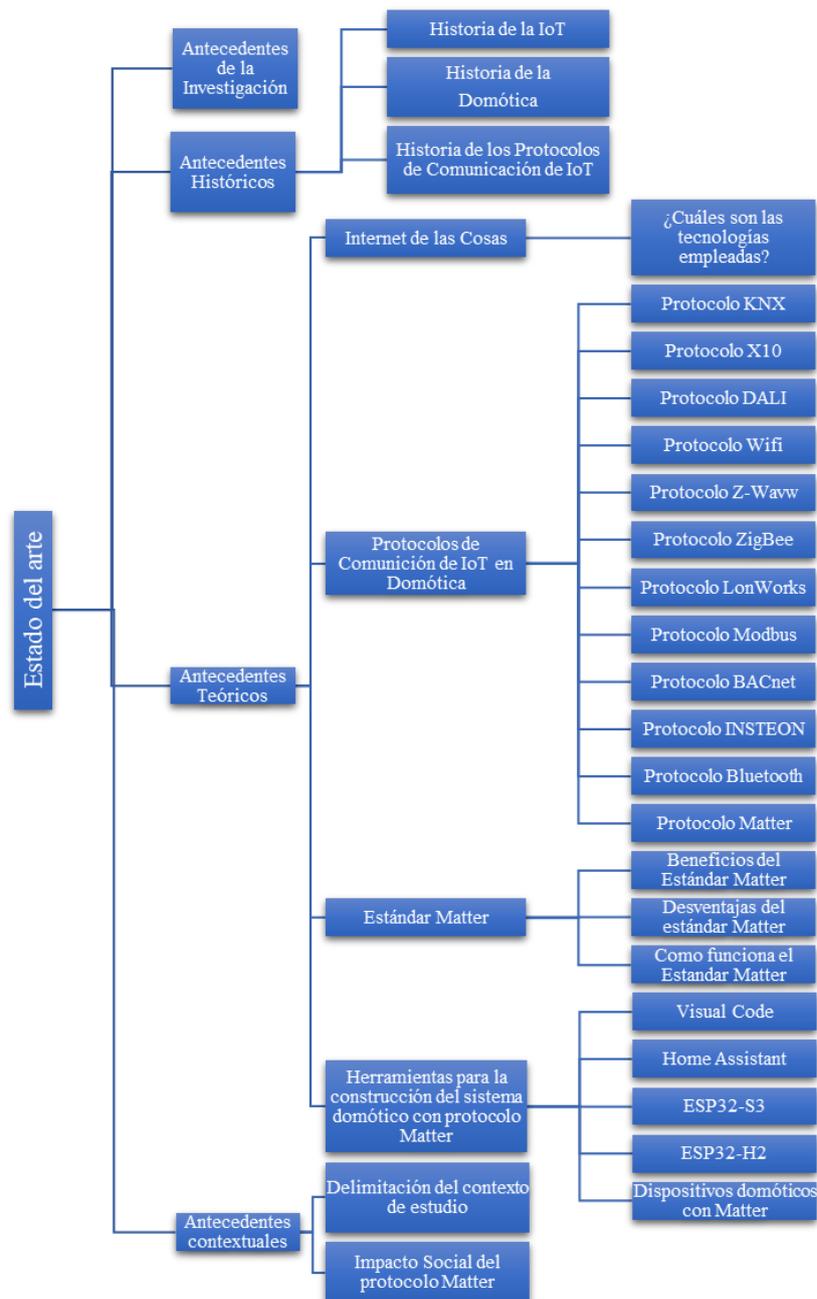


Figura 2. Estado del arte

a) Preguntas de investigación

en la **Tabla 3**, se presentarán las preguntas de investigación:

Tabla 3. Preguntas de investigación

Pregunta de Investigación	Descripción y motivación
RQ1. ¿Cuáles son los beneficios y ventajas de utilizar el protocolo Matter en un sistema domótico?	El objetivo de esta pregunta es explorar y comprender los beneficios y ventajas que ofrece el protocolo Matter en un sistema domótico.
RQ2. ¿Cómo se va a realizar la integración y conexión entre dispositivos Matter comerciales y dispositivos ESP32 con Matter que serán implementados de manera manual?	La motivación de esta pregunta es indagar sobre el proceso y la tecnología que se utilizara para lograr la integración y conexión exitosa entre todos los dispositivos, permitiendo la interoperabilidad en sistema domótico.
RQ3. ¿Cuál es el papel del <i>Gateway</i> basado en ESP32 Border Router en el sistema domótico implementado con Matter?	El objetivo de esta pregunta es comprender el papel fundamental del Gateway basado en el ESP32 Border Router en el sistema domótico implementado con Matter.
RQ4. ¿Qué funcionalidades y capacidades ofrece la plataforma Home Assistant en la integración y control de dispositivos domóticos con tecnología Matter?	El propósito de esta pregunta surge para explorar las funcionalidades y capacidades específicas de la plataforma Home Assistant en relación con la integración y control de dispositivos domóticos utilizando el protocolo Matter.
RQ5. ¿Cuáles son las implicaciones prácticas y beneficios para los usuarios finales al utilizar un sistema domótico basado en el protocolo Matter?	Esta pregunta busca identificar y analizar los beneficios prácticos para los usuarios de un sistema domótico basado en el protocolo Matter, como mejoras en la comodidad, eficiencia energética, seguridad y facilidad de uso.
RQ6. ¿Cuáles son los beneficios y ventajas de utilizar el protocolo Matter en la comunicación entre dispositivos electrónicos y el sistema domótico de un hogar inteligente?	El objetivo de esta pregunta surge para el proceso de investigación y comprensión de los beneficios y ventajas específicas que ofrece el protocolo Matter en términos de comunicación entre los diferentes dispositivos y el sistema domótico de un hogar.
RQ7. ¿Cuáles son los protocolos adicionales de comunicación utilizados en conjunto con Matter para mejorar la cobertura y funcionalidad del sistema domótico?	Esta pregunta tiene como objetivo explorar los protocolos adicionales de comunicación que se utilizan hoy en día además de Matter para mejorar la cobertura y funcionalidad del sistema domótico.
RQ8. ¿Qué desafíos o problemas se pueden enfrentar al implementar un sistema domótico con el protocolo Matter?	El propósito de esta pregunta es identificar y analizar los desafíos o problemas que pueden surgir al implementar un sistema domótico con el protocolo Matter.

b) Palabras claves y Cadena(s) de búsqueda

La estrategia de búsqueda que nos permitió implementar una búsqueda satisfactoria fue la utilización de términos técnicos clave (sistema domótico, protocolos de comunicación IoT, protocolo Matter, Matter y Home Assistant, Arquitectura IoT), que se utilizan comúnmente en el ámbito del Internet de las Cosas, las bases de datos digitales usadas para la consulta bibliográfica fueron Google académico, ScienceDirect, SpringerLink, MDPI, IEEE Xplore y Scopus.

Para refinar la cadena de búsqueda se especificó considerando los términos principales de las tecnologías bajo la investigación (“Protocolo Matter”, “Arquitectura IoT”, “Sistema Domótico”,

“Protocolos de comunicación IoT”, “Protocolo Matter en ESP32”, “Sistema Domótico con Home Assistant”, “Home Assistant y desarrollo Matter”), además se emplearon palabras clave en inglés relacionadas con el tema de interés (Internet of Things, IoT Communication Protocol, Matter Protocol).

En las diferentes cadenas de búsqueda, pudimos recalcar, que el protocolo Matter vinculado con el desarrollo de dispositivos no da muchos resultados, ya que es un protocolo nuevo, que aún se encuentra en desarrollo, por tal motivo se trató de encontrar artículos relevantes a nuestro tema, definiendo así cadenas de búsqueda como: (“Sistema IoT con protocolo Matter” o “Protocolo Matter y Home Assistant” o “Protocolos de Comunicación IoT” o “Sistema Domótico implementado con Home Assistant y el Protocolo Matter” , “Ventajas y Desventajas del uso del protocolo Matter” o “Problemas usuales en el uso del Protocolo Matter”)

c) Criterios de inclusión y exclusión

Para dar paso al proceso de inclusión, se procedió a realizar una verificación y análisis de revistas de carácter científico y otras fuentes complementarias como libros, artículos, conferencias y tesis de postgrado que se relacionan con sistemas Domóticos con protocolos Matter, Protocolos de comunicación IoT, uso del Internet de las Cosas y Protocolo Matter. Para llevar a cabo este proceso se han tomado varios puntos, como el impacto del artículo, el rango de tiempo en que se publicó (2019 - 2023). En cambio, para el proceso de exclusión se consideró los artículos que no forman parte del rango de tiempo de publicación, temas que no forman parte de la investigación, protocolos de comunicación obsoletos, entre otros, en la siguiente **Tabla 4** se puede denotar los criterios de inclusión y exclusión.

Tabla 4. Criterios de inclusión y exclusión en español

#	Criterio de inclusión
1	Estudios primarios
2	Estudios que abordan en los protocolos de Comunicación IoT y Estándar Matter
3	Estudios publicados entre el año 2019 hasta junio de 2023
4	Estudios que relacionan protocolo Matter, Uso de Home Assistant, Sistema Domótico
5	Estudios sobre los beneficios del Protocolo Matter
#	Criterio de exclusión
1	Estudios secundarios
2	Artículos cortos (≤ 4 páginas)
3	Estudios duplicados
4	Artículos escritos en otros idiomas que no sean en inglés o español.
5	Estudios según el tipo de documento (documento de sesión y capítulo de libro)
6	Estudios poco relevantes al tema de investigación
7	Publicaciones cuyo texto completo no estaba disponible para revisión y/o análisis

d) Proceso y resultados de la búsqueda

En la **Figura 2** se presenta un mapa conceptual en el cual se puede ver el proceso y resultado de la búsqueda bibliográfica que se ha realizado:

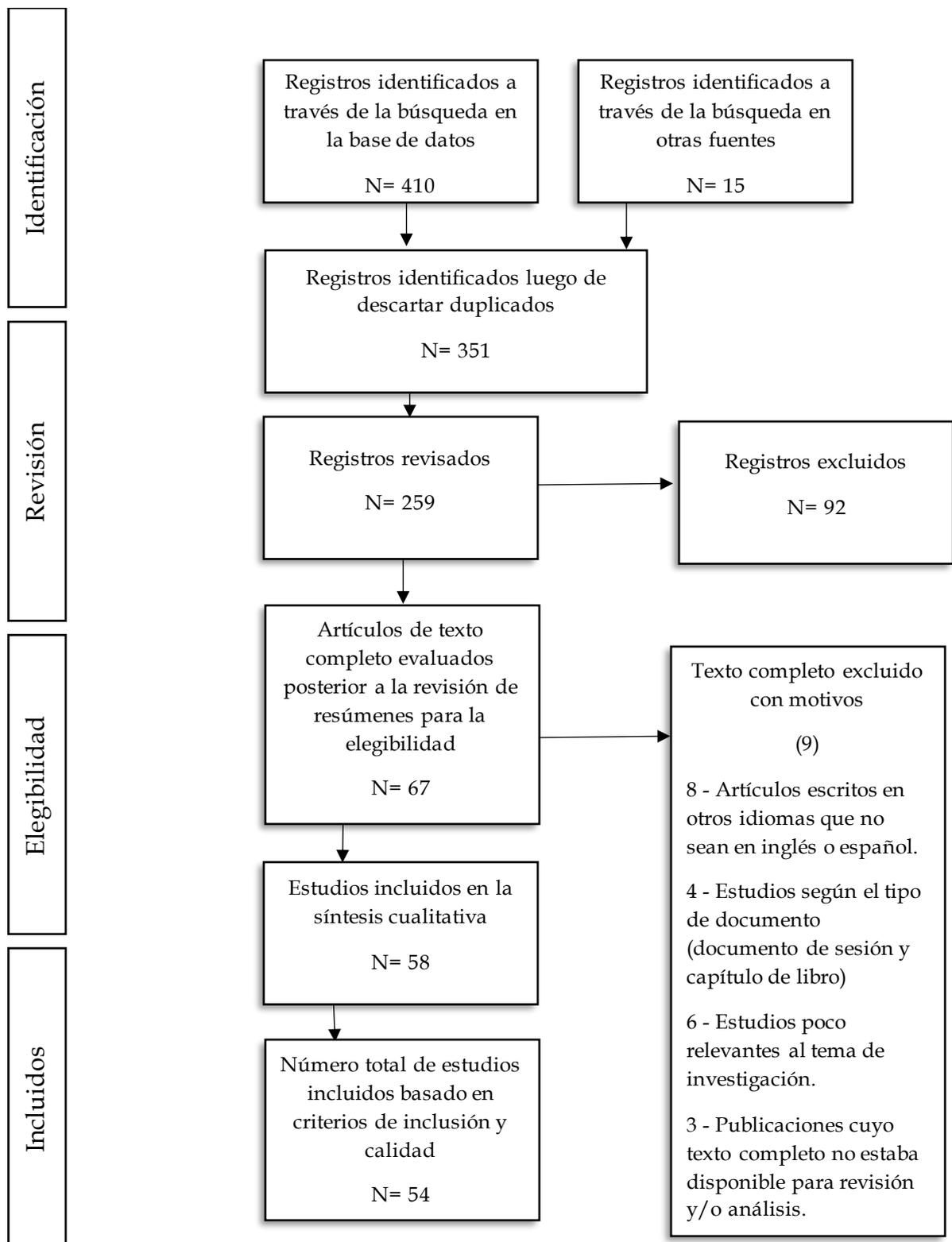


Figura 3. Proceso y resultado de la búsqueda

Resultados de búsqueda

En la **Figura 4**, se presenta un gráfico lineal que ilustra los hallazgos de nuestro proceso de búsqueda en diversas bases de datos bibliográficas y repositorios digitales. Logramos recolectar artículos de investigación anualmente desde 2019 hasta 2023.

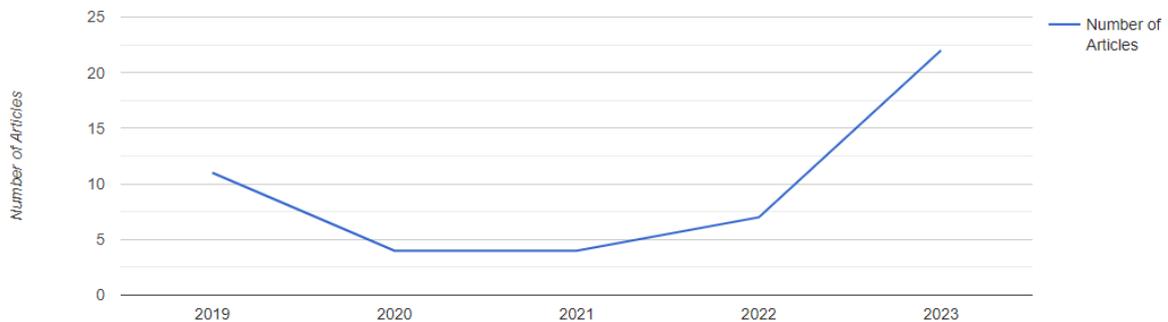


Figura 4. Documentos publicados por año

Fuente: Scopus

La **Figura 5** ilustra de manera clara y concisa un gráfico de pastel. En él, se desglosan las diversas bases de datos de artículos utilizadas. Cada segmento del gráfico representa un porcentaje específico, reflejando la proporción en la que se han realizado búsquedas de artículos en cada una de estas fuentes. Esta representación visual nos permite comprender rápidamente la predominancia de ciertas bases de datos sobre otras en términos de consultas realizadas.

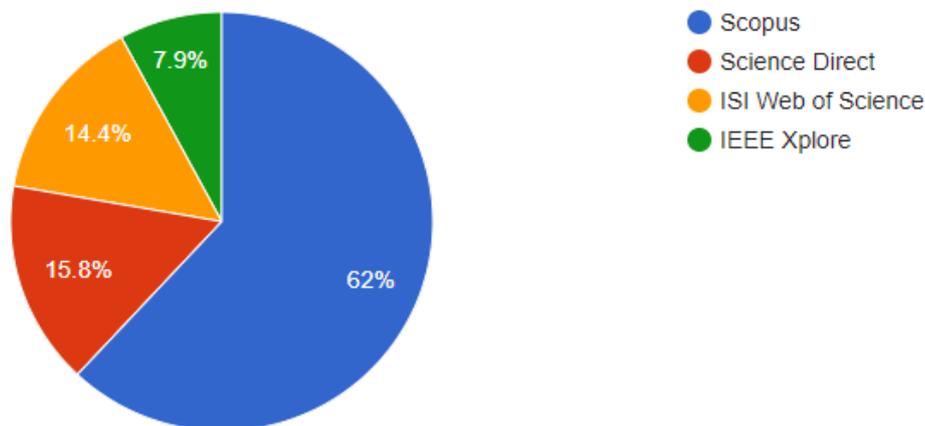


Figura 5. Documentos encontrados en las diferentes bases de datos

La **Figura 6** presenta un análisis bibliométrico de las referencias empleadas en este estudio. Para ello, se hizo uso de la herramienta VOSviewer, evidenciando así la existencia de información y estudios vinculados al tema tratado.

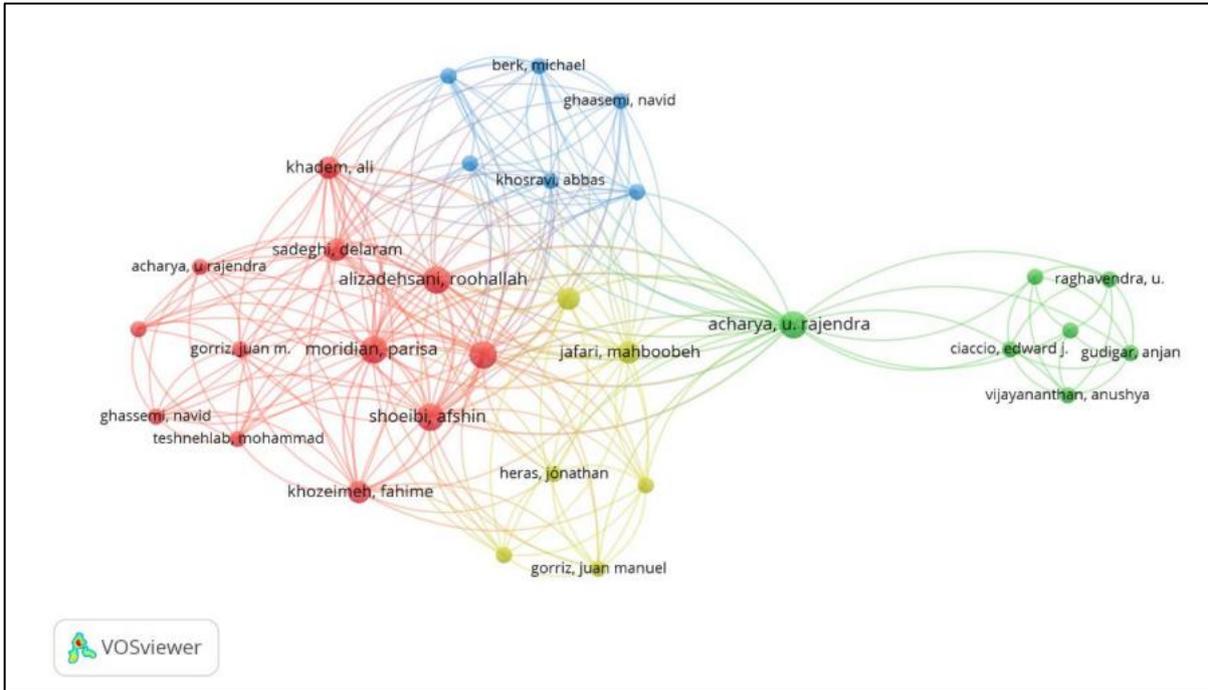


Figura 6. Análisis de contenidos - VOSviewer

1.2. Antecedentes históricos

1.2.1. Historia de la IoT

La línea de tiempo presentada en la **Figura 7** relata la historia de la evolución del Internet de las Cosas (IoT), destacando los eventos clave desde sus inicios hasta la actualidad.

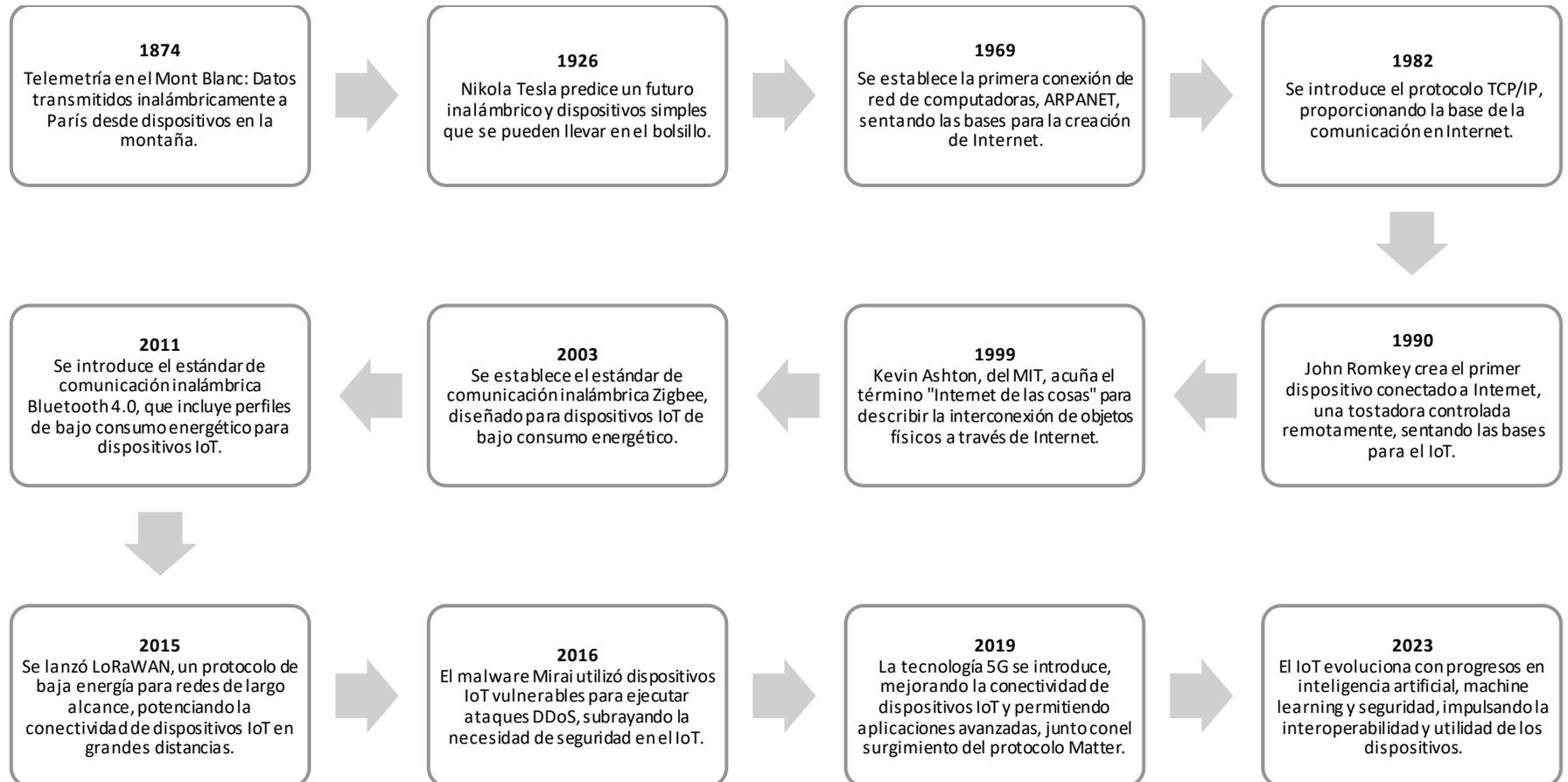


Figura 7. Historia de la IoT

1.2.2. Historia de la Domótica

En la **Figura 8**, se traza una línea temporal que recorre la historia de la domótica, detallando los momentos más importantes que han marcado su evolución desde sus inicios hasta el presente.

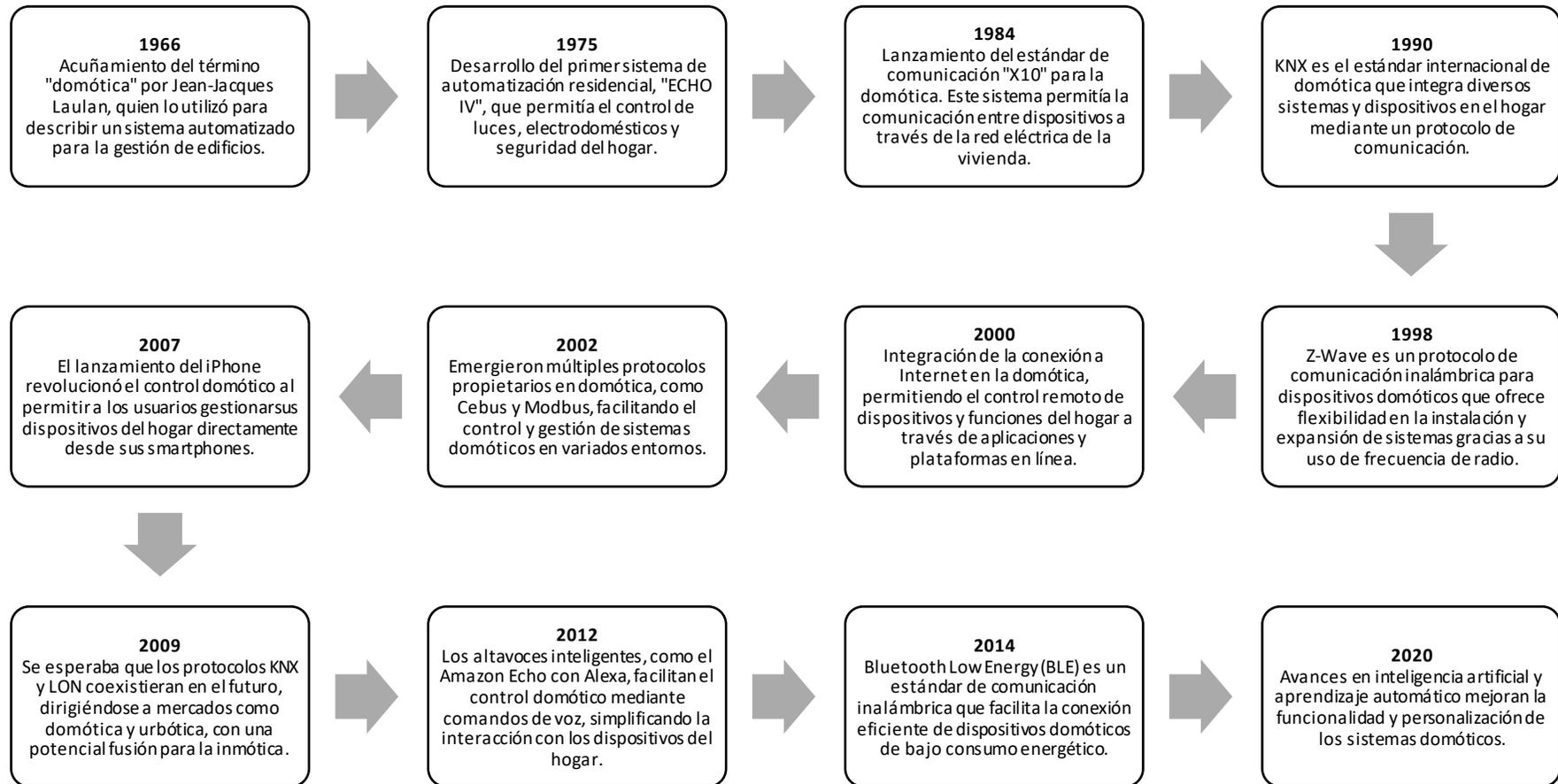


Figura 8. Historia de la Domótica

1.2.3. Historia de los Protocolos de Comunicación de IoT

La **Figura 9** engloba los eventos más destacados que conforman la historia de los protocolos de comunicación para el Internet de las Cosas (IoT) y su trayectoria de desarrollo.

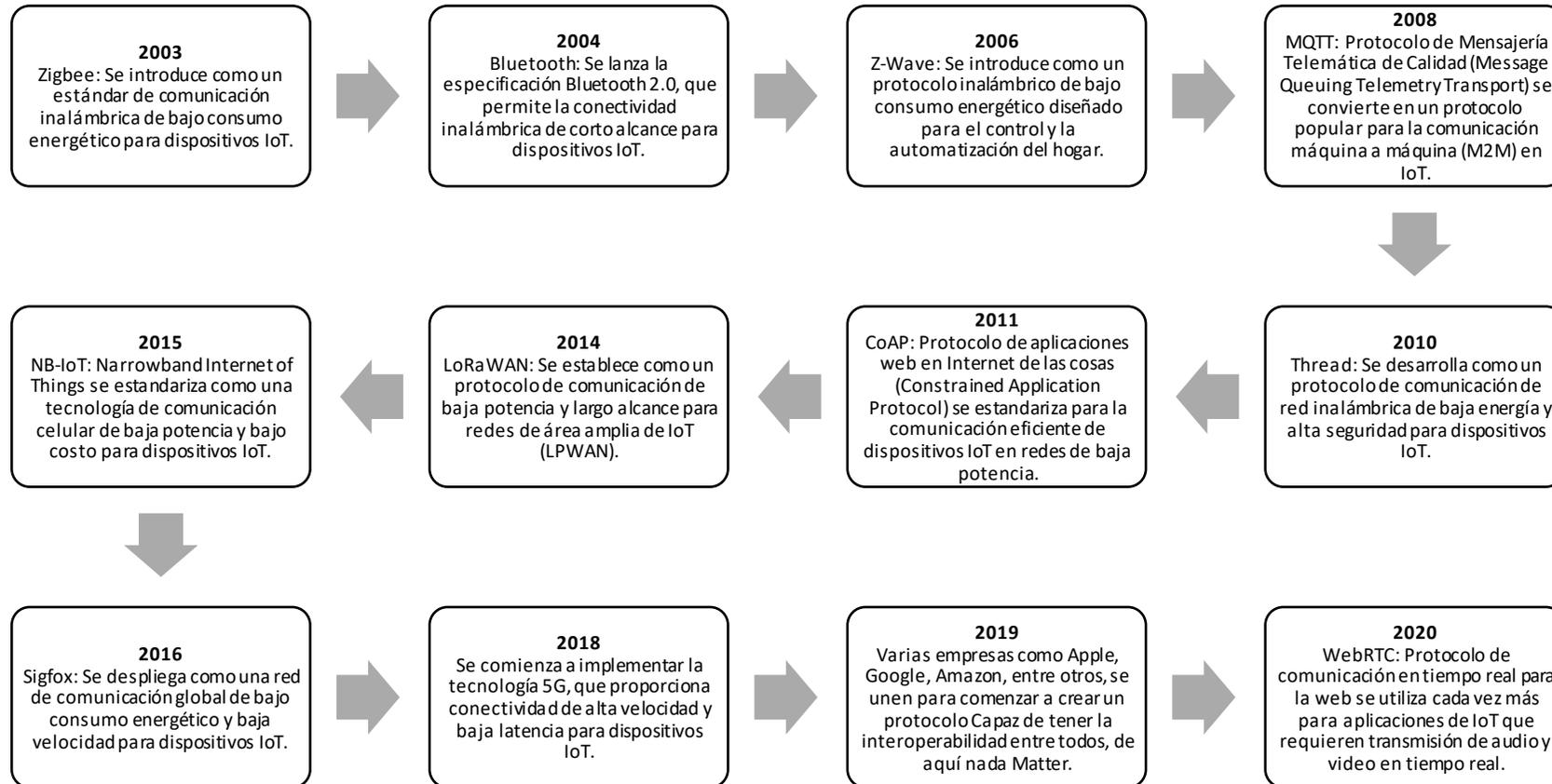


Figura 9. Historia de los protocolos de comunicación de IoT

1.3. Antecedentes Teóricos

1.3.1. Internet de las Cosas

De acuerdo a [5], el internet de las Cosas (IoT) es una revolucionaria red que está interconectada de dispositivos físicos, electrodomésticos, vehículos y otros objetos que utilizan sensores, software y conectividad para recolectar, transmitir y procesar datos a través de Internet. Para este trabajo se encontró varios artículos los cuales permite tener idea de cómo el IoT afecta otros hogares de distintos países en donde este estudio [6], se centra en la adopción y uso de tecnologías para hogares inteligentes en Noruega, un país tecnológicamente avanzado, y se identifica como es el impacto de la IoT.

Además de potenciar el mañana, busca mejorar el entorno y la manera en cómo el trabajo propone un sistema electrónico de bajo costo basado en IoT para monitorear variables relacionadas con el consumo de energía eléctrica en redes de baja tensión. La eficiencia energética se ha vuelto relevante debido a la demanda mundial de electricidad, generando un impacto negativo en el planeta [7].

Esto permite la interacción y toma de decisiones autónomas de los dispositivos, mejorando la eficiencia, comodidad y calidad de vida en diferentes ámbitos como hogares inteligentes, ciudades inteligentes, salud, agricultura y transporte, ofrece una panorámica del Internet de las cosas (IoT), abordando aspectos como aplicaciones, arquitecturas, protocolos y tecnologías según la literatura reciente[8].

Gracias al IoT, se puede comprender que al incluir aplicaciones, un esquema y diferentes parámetros se puede potenciar esta rama, en [9], se puede denotar en el cual destaca la colaboración de sensores inteligentes para ejecutar casos de uso sin intervención humana, y se sugiere que integrar la inteligencia en el IoT permitirá aplicaciones más potentes, se destaca la importancia del Internet de las cosas industrial (IIoT) en el contexto de la Industria 4.0 y cómo los dispositivos IIoT inteligentes y seguros son fundamentales en este entorno [10].

El informe destaca la importancia de identificar y desarrollar aplicaciones, concluyendo con recomendaciones específicas sobre modelización de sistemas IIoT, protocolos en tiempo real y computación en niebla (fog computing).

Este artículo [11], se centra en los desafíos de garantizar la seguridad en las Redes de Área Amplia de Baja Potencia (LPWAN), que son tecnologías de comunicación inalámbrica utilizadas para la transferencia eficiente de datos entre dispositivos de baja potencia. En el contexto de Internet de

las cosas (IoT), donde estas redes se utilizan cada vez más, se destaca la necesidad de abordar los desafíos de seguridad y eficiencia inherentes a estas tecnologías.

Dada la expansión continua de las tecnologías IoT, la seguridad y la privacidad se han vuelto desafíos más complejos, especialmente con el creciente número de dispositivos híbridos en grandes organizaciones. La seguridad es crucial para proteger el hardware, los aspectos de red de los dispositivos y el acceso a la información contra entidades no autorizadas[12].

El IoT brinda oportunidades sin precedentes para acceder a datos y servicios especializados en diversos campos como la educación, seguridad, asistencia y transporte, transformando significativamente la forma en que vivimos, además de mejorar nuestra calidad de vida en general [13].

¿Cuáles son las tecnologías empleadas?

El internet de las Cosas (IoT) utiliza tecnologías y protocolos para la comunicación entre dispositivos. Algunos dispositivos se conectan directamente a internet, mientras que otros requieren adaptadores o protocolos para su integración. La diversidad de protocolos dificulta la interoperabilidad entre dispositivos IoT, por tal motivo surge el protocolo Matter, como solución para establecer un estándar unificado que promueve la interoperabilidad entre diferentes dispositivos.

Matter permite a los fabricantes facilita la integración y mejora la experiencia del usuario en los sistemas domóticos, como se menciona [14], nos habla de la necesidad de una plataforma general de IoT, independiente de fabricantes, debido a la dispersión generada por sistemas tradicionales y soluciones propietarias, abogando por el uso de plataformas de código abierto en infraestructuras existentes para evitar dependencias y limitaciones.

En un futuro más automatizado, el Internet de las Cosas (IoT) se presenta como una herramienta clave. Diversos trabajos están utilizando un protocolo que emplea funciones físicamente no clonales, biometría y contratos inteligentes de Ethereum para prevenir ataques de repetición, suplantación y clonación, se puede denotar que un excelente protocolo y comunicación entre dispositivos inteligentes en nuestro entorno como se ve en las encuestas de [15], la revisión subraya la importancia de la conectividad inteligente y el cómputo contextual en un entorno IoT exitoso, destacando las demandas esenciales, como la comprensión de los usuarios y sus dispositivos, redes de comunicación ubicuas, arquitecturas de software y herramientas analíticas.

Estos enfoques demuestran eficiencia de recursos y robustez frente a ciberataques, destacando la propuesta de un protocolo de autenticación computacionalmente económico. Esta propuesta utiliza blockchain y fog computing para garantizar la seguridad y privacidad en las aplicaciones de salud basadas en Ambient Intelligence (AmI)-IoT. En conjunto, estas innovaciones apuntan a un avance significativo en la capacidad del IoT para ofrecer un mundo más seguro y eficiente [16].

1.3.2. Domótica

Cuando hablamos de domótica nos estamos refiriendo a la integración de tecnología en los hogares para la automatización y el control de diversas funciones y dispositivos. La automatización en este contexto implica la implementación de tecnologías y sistemas que permiten controlar y gestionar diversas funciones dentro de una casa de manera inteligente y eficiente. Esto puede incluir aspectos como la iluminación, el control de temperatura, la seguridad, la gestión de energía, entre otros [17].

Es una disciplina que combina el uso de datos, de comunicaciones y protocolos para facilitar la automatización de diferentes actividades diarias en el hogar, y además el estudio proporciona una visión integral de cómo el IoT puede contribuir a la sostenibilidad del transporte ferroviario, ofreciendo beneficios operativos y señalando áreas clave para investigaciones futuras [18].

Esta no solo busca mejorar la comodidad y conveniencia del usuario final, sino que también tiene un impacto positivo en el medio ambiente al permitir un control más eficiente de la energía eléctrica en el hogar [19].

Al tener uso de sistemas automatizados y sensores, la domótica tiene la posibilidad de optimizar el uso de la energía, reducir el consumo innecesario y contribuir a la sostenibilidad para automatizar y controlar funciones en el hogar [20].

1.3.3. Protocolos de Conexión

El Internet de las Cosas (IoT) desempeña un papel crucial al conectar dispositivos en redes amplias capaces de pensar, procesar datos y comunicarse mediante diversos protocolos. La domótica, como una aplicación específica del IoT, se beneficia de la interconexión de dispositivos para mejorar la automatización y el control del hogar. Tiempos atrás se usaba LoRaWAN en donde parecía una buena opción para ese entonces, inclusive en el trabajo [21], nos habla Los resultados demuestran la eficacia y eficiencia del esquema de protección propuesto para detectar y aislar líneas defectuosas en redes de transmisión MT-HVDC, junto con una técnica de restauración rápida.

Se debe abordar la importancia de elegir protocolos de comunicación adecuados, considerando las limitaciones de hardware de los dispositivos pequeños en la red IoT, lo cual es especialmente relevante en el ámbito de la domótica [22].

Estos protocolos están diseñados para establecer conexiones entre dispositivos y transportar datos entre ellos:

Protocolo Bluetooth

Es un protocolo inalámbrico de corto alcance para la comunicación entre dispositivos electrónicos. Permite la transferencia de datos y voz de forma segura y eficiente. Es conocido por su bajo consumo de energía y amplia compatibilidad con diferentes dispositivos. Utiliza ondas de radio en la banda de 2.4 GHz y admite varios perfiles de comunicación, como audio, transferencia de archivos y control remoto.

La prevalencia de Bluetooth Low Energy (BLE) como una tecnología inalámbrica ubicua en dispositivos electrónicos de consumo, destacando su éxito en el Internet de las Cosas (IoT) y la electrónica de consumo. Aunque BLE ofrece ventajas en comunicaciones inalámbricas de corto alcance y bajo consumo, también presenta desafíos de seguridad y privacidad. Es ampliamente utilizado en dispositivos móviles, auriculares, altavoces y otros dispositivos electrónicos. Proporciona una conexión inalámbrica fácil y conveniente para la transferencia de datos entre dispositivos cercanos [23].

Protocolo Z-Wave

El protocolo Z-Wave es una tecnología inalámbrica propietaria desarrollada por Silicon Lab. Este ofrece comunicaciones fiables y de baja latencia, utilizando un cifrado avanzado (cifrado S2) para garantizar la seguridad de los datos transmitidos. Lo que tiene de interesante es que permite la conexión hasta de 232 dispositivos en una red, y su alcance puede llegar hasta los 30 metros, con la capacidad de retransmisión para extender aún más la cobertura.

Z-Wave, utilizada en la automatización de hogares y oficinas, carece de detalles de implementación que informen adecuadamente a los consumidores sobre la seguridad de sus sistemas de sensores y actuadores. A través de ingeniería inversa en una red Z-Wave real, se desvelan aspectos del protocolo de enrutamiento, revelando vulnerabilidades en la integridad de la fuente y los datos [24].

Protocolo Wifi

El protocolo Wifi es ampliamente utilizado en IoT, para lo que es la comunicación inalámbrica entre dispositivos, también conocido como 802.11. Las ventajas incluyen disponibilidad y compatibilidad amplias, velocidades rápidas, conexión a Internet y soporte.

El siguiente trabajo [25], revisa de manera integral la tecnología de comunicación, el diseño arquitectónico, las aplicaciones avanzadas y los protocolos de los sistemas de Smart Grid con integración Wifi, asistidos por IoT. Además, se identifican preocupaciones clave, desafíos futuros y brechas en la investigación de estos sistemas, proporcionando una base para futuras direcciones de investigación en este campo en constante evolución.

La investigación [26], se centra en analizar el rendimiento del "sniffing" bajo diferentes entornos inalámbricos utilizando productos disponibles comercialmente. El objetivo es identificar factores como la configuración del canal y la configuración del punto de acceso que afectan el rendimiento del "sniffing" utilizando protocolo de Wifi.

Protocolo Zigbee

Este es un sistema de protocolo abierto diseñado específicamente para la domótica, Los dispositivos Zigbee establecen una red inalámbrica en malla, lo que significa que no necesitan estar individualmente conectados al enrutador, sino que se comunican a través de otros dispositivos en la red [27].

En contraste, las redes Zigbee puro se caracterizan por la conexión directa entre dispositivos finales, lo que limita el tipo de tráfico de red generado. Este artículo [28], propone que las variables obtenidas directamente desde Zigbee puro en un entorno distribuido pueden ser utilizadas para la identificación de intrusiones, centrándose en los dispositivos finales en lugar de la red en su conjunto, unas de las ventajas es que Zigbee habilita sensores inalámbricos y mapas basados en IoT para incorporar algún sistema de control de objetos. En [29], el estudio simula el rendimiento del vehículo en condiciones predefinidas mediante código de firmware integrado en C.

Zigbee es un protocolo de comunicación inalámbrica diseñado específicamente para aplicaciones de baja potencia y bajo consumo energético, como las que se encuentran en el Internet de las Cosas (IoT) y sistemas de redes de sensores. Utiliza frecuencias de radio de corto alcance y opera en la banda de 2.4 GHz, ofreciendo un equilibrio óptimo entre eficiencia de energía y velocidad de transmisión de datos.

En el trabajo [30], el protocolo Zigbee se destaca como la elección clave para establecer vínculos de comunicación eficientes entre los Vehículos Aéreos No Tripulados (UAVs) y el Centro de Control en Tierra (GCC) en el contexto de un sistema multi-UAV para monitorear áreas de protección ambiental. Zigbee se elige por su capacidad para facilitar la comunicación en redes ad-hoc voladoras (FANETs), ofreciendo ventajas significativas en términos de alcance, eficiencia y costos.

1.3.4. Protocolos de Aplicación de IoT en domótica

Se tiene claro que, en la domótica, los protocolos de comunicación de IoT son fundamentales para interconectar dispositivos en el hogar. Estos protocolos abordan desafíos como la eficiencia energética, el direccionamiento y la movilidad. Las redes no IP, como Bluetooth, RFID y NFC, ofrecen un menor consumo de energía, pero con un alcance limitado. Las organizaciones de estándares establecen una visión común de arquitectura y protocolos. En resumen, los protocolos de comunicación de IoT en la domótica permiten la interconexión eficiente de dispositivos en el hogar, abandonando desafíos específicos de interoperabilidad y eficiencia energética [31].

Estos protocolos están diseñados para especificar cómo los dispositivos deben comunicarse entre sí en términos de formato de datos, timing, secuencia, etc. Estos protocolos suelen operar en las capas superiores del modelo OSI [32], la evolución de los protocolos de la capa de aplicación en el contexto del Internet de las cosas (IoT). Dado el crecimiento del IoT en los últimos años, se destaca la importancia de garantizar una comunicación adecuada entre diversas aplicaciones y dispositivos, como se puede ver en [33].

Se toma como propósito la arquitectura de interoperabilidad entre dispositivos que trabajan con el protocolo Modbus y una plataforma de IoT implementada según los estándares de oneM2M, como lo puede mostrar en [34].

Para superar las barreras de interoperabilidad entre cosas heterogéneas en el IoT, [35] el artículo propone una solución basada en la síntesis automatizada de mediadores de protocolo que respaldan la interconexión, este enfoque sistemático se apoya en el modelo de conector Data eXchange (DeX), que abstrae y representa de manera integral protocolos de middleware existentes.

Existen resultados experimentales que muestran que el TPRUDF logra un alto nivel de precisión en la utilización de recursos, mejora significativamente el rendimiento de la utilización de recursos, aumenta la disponibilidad de los recursos y disminuye el consumo de energía. Este marco

demuestra ser prometedor para abordar los desafíos actuales en la gestión eficiente de recursos en entornos IoT, particularmente en el contexto de ciudades inteligentes [36].

Protocolo KNX

El protocolo KNX es un estándar de comunicación de red basado en OSI para la domótica. Se creó mediante la convergencia de tres estándares anteriores: EHS, EIB y BatiBUS. KNX es gestionado por la Asociación KNX y se ha convertido en un estándar abierto desde 2016 [27]. Este protocolo permite la interconexión y control de dispositivos domóticos en el hogar, ofreciendo una solución integrada y compatible para la automatización de diferentes sistemas, como iluminación, climatización y seguridad. KNX proporciona flexibilidad, eficiencia y confiabilidad en la comunicación entre los dispositivos, lo que lo hace ampliamente utilizado en el campo de la domótica.

Protocolo X10

El protocolo X10 es utilizado para el control remoto de dispositivos eléctricos en el hogar, transmitiendo señales a través de la línea eléctrica. Aunque tiene limitaciones en cuanto a ancho de banda y número de dispositivos controlados, algunos dispositivos X10 más avanzados incorporan protocolos extendidos que permiten funcionalidades adicionales como la comunicación bidireccional y la verificación de la transmisión de datos. El uso del protocolo X10 presenta ventajas como la falta de necesidad de cableado adicional, ahorro de energía eléctrica y monitoreo del hogar, y su amplia adopción en Europa y Estados Unidos. Sin embargo, una desventaja es que el control es centralizado, dependiendo de un controlador para ejecutar las órdenes, lo que puede ocasionar problema.

Protocolo DALI

Este protocolo, llamado DALI (Digital Addressable Lighting Interface/ Interfaz de iluminación direccionable digital) es un estándar de comunicación utilizado en sistemas de control de iluminación, a diferencia del protocolo X10, que se basa en la línea eléctrica, DALI utiliza un bus de datos independiente para transmitir señales de control entre los dispositivos de iluminación. El protocolo DALI permite el control individual y preciso de cada luminaria, así como la configuración de escenas y la programación.

Protocolo LonWorks

El Protocolo LonWorks es un estándar de comunicación diseñado para sistemas de automatización y control en edificios. Esta se basa en una arquitectura de red descentralizada, donde los dispositivos se comunican entre sí utilizando mensajes cortos. LonWorks utiliza un protocolo

propietario y es compatible con diferentes medios de comunicación, como cables eléctricos, fibra óptica o líneas de señal. Entre las características destacadas se encuentran su alta confiabilidad, capacidad de autodetección de dispositivos, bajo consumo de energía y escalabilidad [37].

Protocolo Modbus

Cuando hablamos del Protocolo Modbus, se entiende que es un estándar de comunicación utilizado en la industria para el intercambio de información entre dispositivos. Se basa en una arquitectura maestro-esclavo y utiliza una estructura de mensajes simple [34]. Es independiente de cualquier medio de comunicación y se puede implementar en redes como RS-485, TCP/IP o Ethernet. Modbus es ampliamente utilizado en aplicaciones de control y monitoreo en sistemas industriales, es conocido por su fácil implementación, bajo costo y alta interoperabilidad.

Protocolo BACnet

BACnet es un protocolo estándar para la comunicación en sistemas de automatización de edificios, además que permite la interoperabilidad entre dispositivos de diferentes fabricantes. Utiliza una estructura de datos jerárquica basada en objetos para controlar elementos como iluminación, climatización y seguridad. BACnet es altamente escalable y admite diferentes medios de comunicación, se puede encontrar en, que BACnet es un estándar para ser utilizado en grandes industrias que estén automatizadas [38].

Es ampliamente utilizado en edificios comerciales y residenciales para lograr una gestión eficiente y proporciona flexibilidad y seguridad en la integración de sistemas de control y gestión de edificios [39].

Protocolo INSTEON

Es un protocolo de comunicación mejorado del X10, más fiable y flexible. Combina radiofrecuencia y cableado eléctrico para transferir datos. Su red redundante y de doble malla asegura la integridad de la información. Se destaca por su capacidad de confirmación de mensajes y minimización de errores. Ofrece una solución eficaz y confiable para la automatización del hogar.

1.3.5. Estándar Matter

Matter, es un protocolo de comunicación IoT, que surgió en el 2019 el cual se considera hoy en día como un protocolo universal de conexión diseñado para dispositivos inteligentes en el contexto del Internet de las Cosas (IoT), el cual permite la interoperabilidad entre diferentes plataformas, eliminando la necesidad de preocuparse por la compatibilidad con diversos ecosistemas como Amazon Alexa, Google Home o Apple HomeKit. En la **Figura 10** se puede ver una pirámide en

donde el protocolo Matter se sitúa en la cúspide y bajo él se encuentran los demás protocolos de comunicación IoT que se han desarrollado antes de estos, se puede destacar que en [6], la creciente interconexión mundial, impulsada por la proliferación de dispositivos conectados, redes y sistemas intrincados, lo que ha generado oportunidades sin precedentes en automatización y transformación digital sofisticada, de tal forma Matter crece como protocolo ideal.

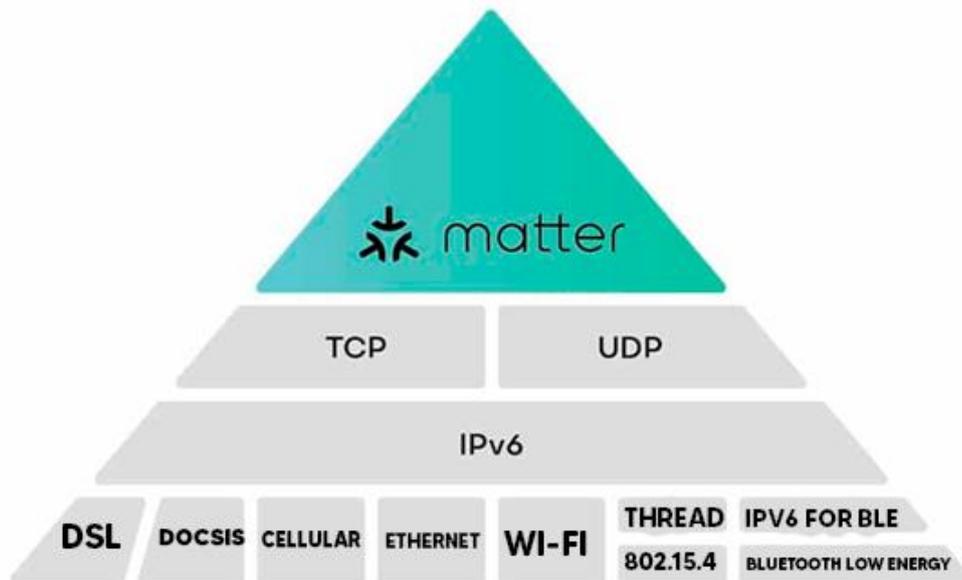


Figura 10. Matter protocolo universal

Beneficios del estándar Matter

- Al estar basado en la tecnología IP, Matter facilita la creación de un ecosistema conectado en la nube, pero también funciona de manera local sin depender de internet.
- Se integra con estándares existentes como Ethernet, Wifi, Bluetooth y Thread, este último basado en Zigbee para proporcionar una conectividad inalámbrica de baja energía. De esta manera, Matter ofrece flexibilidad y simplicidad en la interconexión de dispositivos inteligentes.
- Los dispositivos Matter se conectan entre sí de forma local a través de la red Wifi, Ethernet o Thread, para una respuesta más rápida y una mejor confiabilidad.
- Los dispositivos Matter se pueden usar en varias plataformas de una casa inteligente.

Este artículo [40], aborda el desafío de interoperabilidad entre plataformas heterogéneas de Internet de las cosas (IoT) al considerar que cada plataforma utiliza un identificador de recursos

(ID) diferente. La falta de estandarización en los identificadores de recursos dificulta la identificación de dispositivos y la utilización de servicios entre plataformas IoT diversas.

1.3.5.1. Desventajas del estándar Matter

Al ser un protocolo reciente, solo funcionará con una pequeña cantidad de categorías de dispositivos en el lanzamiento

SDK de Espressif para Matter

El SDK para Matter de Espressif está construido sobre el SDK de Matter de código abierto y proporciona API simplificadas, periféricos de uso común, herramientas y utilidades para seguridad, fabricación y producción, acompañados de documentación exhaustiva. Incluye ricas referencias de producción, destinadas a simplificar el proceso de desarrollo de los productos Matter y permitir a los usuarios pasar a producción en el menor tiempo posible, así como se puede observar en la **Figura 11**:

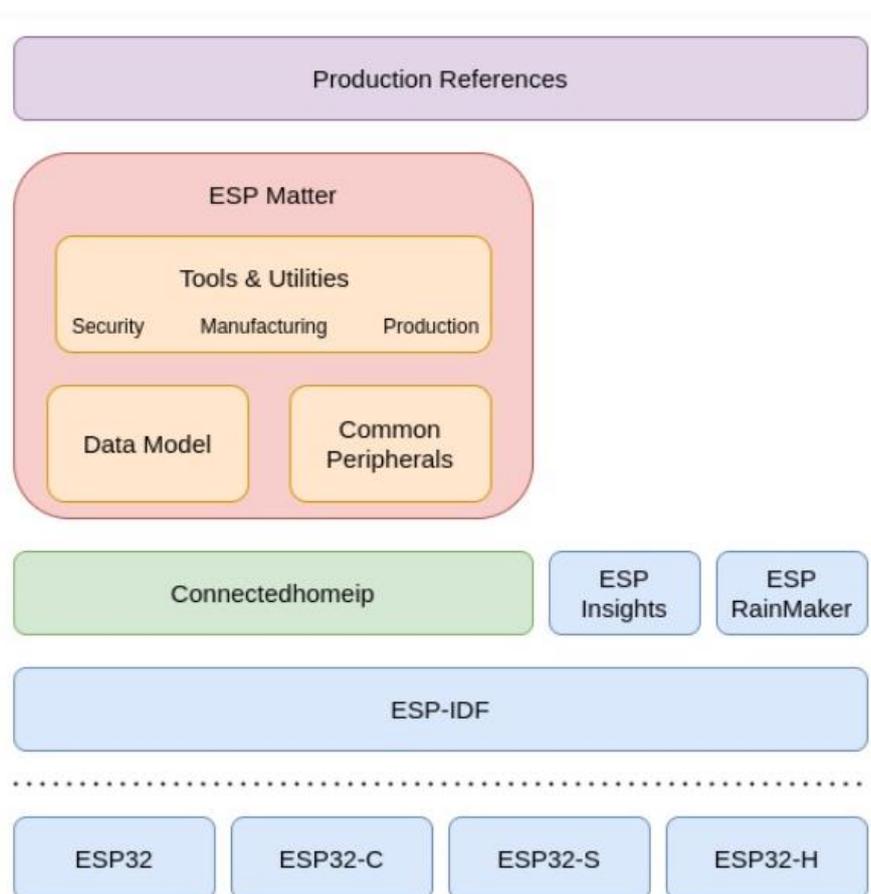


Figura 11. Arquitectura SDK de Espressif para Matter

Fuente: Tomado de [41]

¿Cómo funciona el estándar Matter?

Este protocolo utiliza la infraestructura de red local existente, como lo de Wifi, Ethernet, y Thread para permitir la comunicación entre los dispositivos compatibles. Al agregar un dispositivo que tenga el protocolo Matter a nuestro proveedor, sea Google Home, Alexa, o Apple, este se conecta a la misma red que el dispositivo, ya sea alguno de estos antes mencionados que actúa como el centro de control Matter.

Gestión Múltiple

La característica de gestión múltiple en el protocolo Matter facilita directamente la interoperabilidad entre dispositivos certificados por Matter. Gracias a esta funcionalidad, los dispositivos certificados pueden integrarse a cualquier entorno compatible con Matter, desde una aplicación individual hasta sistemas completos de hogares inteligentes como Google Home, Amazon Alexa, Apple HomeKit y Samsung SmartThings.

Por ejemplo, si alguien en un hogar inteligente tiene un timbre Google Nest, una bombilla Hue y un altavoz Amazon Echo, y todos están certificados por Matter, su comunicación es fluida. Si alguien pulsa el timbre Nest, la bombilla Hue podría encenderse y el altavoz Echo podría indicar que hay una visita. Sin Matter, lograr esta integración sería complicado para el usuario promedio.

La gestión múltiple de Matter permite a los usuarios tener más opciones en el mundo de los hogares inteligentes, fomentando una competencia sana entre dispositivos y rompiendo las barreras de un único ecosistema. Aquellos que gusten del software Google Home y el hardware Apple HomeKit pueden emplear dispositivos certificados por Matter para mejorar su experiencia en diferentes sistemas de forma local, segura y simultánea.

Multitransporte en Red

Matter combina las ventajas del Wifi, Thread y su base común de protocolo de Internet (IP) para establecer una red eficaz y fiable compatible con diversos ecosistemas y tecnologías. La tecnología de multitransporte en red posibilita que dispositivos IoT y de hogares inteligentes conectados a Wifi o ethernet interactúen directamente con servicios en la nube y entre ellos, hay que tener en cuenta que para escenarios de magnitudes industriales podríamos compararlo contra Modbus, dado que estos sistemas controlan infraestructuras esenciales como redes eléctricas, plantas de tratamiento de agua y redes de transporte, la seguridad se vuelve de suma importancia. Sin embargo, los métodos actuales para evaluar la seguridad de los ICS a menudo son ad-hoc y difíciles de formalizar en una metodología sistemática con resultados predecibles [42].

La tecnología Thread, una red inalámbrica de malla diseñada para dispositivos de hogar inteligente de bajo ancho de banda y larga distancia, también es compatible con Matter. Dispositivos eficientes en energía basados en Thread, como sensores, enchufes y bombillas, pueden conectarse a routers Thread certificados por Matter para aprovechar todo su potencial.

Con Matter, un router Thread puede enlazarse con un puente Thread y un router Wifi. Siempre que cada aparato en este conjunto esté certificado por Matter, todos los conectados a los routers Wifi y Thread podrán comunicarse entre sí. Los puentes compatibles con Matter se pueden fusionar con varios dispositivos, permitiendo que aquellos no certificados se unan e integren a la red de Matter.

La cohesión que proporciona Matter permite a los creadores diseñar dispositivos IoT que trabajen en conjunto, independientemente de la marca o el sistema de hogar inteligente, ofreciendo interfaces unificadas capaces de controlar múltiples dispositivos certificados por Matter.

Compatibilidad y dispositivos Matter

Matter es un estándar en dispositivos de domótica. Los productos compatibles muestran un logotipo específico. Philips Hue integrará Matter en bombillas y lámparas mediante una actualización del software del Hue Bridge. Otros dispositivos y marcas, como termostatos, sistemas de seguridad y WiZ, también serán compatibles. Apple introdujo soporte para Matter en iOS/iPadOS 16.1, permitiendo la interacción y control de estos dispositivos a través de sus plataformas.

Tomando en cuenta el uso de dispositivos esp32, se optó por trabajar con estos dispositivos por el excelente rendimiento de la familia ESP32, se puede destacar, [43] el trabajo en donde se usa un dispositivo de la misma marca, obteniendo un prototipo fina y útil, este trabajo se centra en evaluar la viabilidad de integrar placas de bajo costo, como Raspberry Pi y microcontroladores ESP8266, en una red descentralizada con equipos industriales de Controladores Lógicos Programables (PLC). El objetivo es utilizar estos dispositivos como nodos para facilitar el intercambio de datos entre el piso de manufactura y los servicios empresariales de una manera simple, fiable y económica.

Aparatos de Google que admiten Matter

Matter representa el nuevo patrón universal para hogares inteligentes, ideado para simplificar para el usuario la identificación de dispositivos compatibles para el hogar, el proceso de instalación de estos dispositivos y para garantizar interacciones más ágiles y eficientes.

Ventajas de los aparatos que admiten Matter

Cada dispositivo Matter es compatible con otros de su tipo y con Google Home. Proceso unificado y eficiente para instalar cualquier dispositivo del hogar que admita Matter. Los dispositivos Matter se vinculan directamente mediante la red Wifi, Ethernet o Thread presente en tu hogar, asegurando respuestas ágiles y una conexión más estable. Puedes usar los dispositivos Matter en múltiples sistemas de hogar inteligente (como Google Home, Apple HomeKit y otros) simultáneamente.

Funcionamiento de los aparatos que admiten Matter

Matter opera directamente usando la red Wifi, Ethernet y Thread de tu hogar. Al añadir un dispositivo que admita Matter a Google Home, este se conecta a la misma red que tu aparato de Google que sirve como el núcleo central de Matter. Así, puedes manejarlo tanto dentro como fuera de casa.

Por poner un caso, si le solicitas a tu Nest Mini que active una luz inteligente que es compatible con Google Home o cualquier otro dispositivo del hogar, usualmente, el Nest Mini manda ese requerimiento vía Internet a los servicios cloud de Nest, que después se envía al cloud del socio fabricante de la luz, y finalmente se remite por Internet a tu hogar para encender la luz.

Para los dispositivos que admiten Matter, el proceso es más directo y por ende más veloz, incluso si no te encuentras en tu hogar. Al hacer el mismo pedido, Google Home lo recogerá a través de tu dispositivo Google que funge como núcleo central de Matter. Este núcleo instruirá a la luz para que se active directamente a través de la red Wifi, Ethernet o Thread en tu hogar, evitando la necesidad de pasar tu instrucción por Google Cloud, luego al cloud del fabricante y finalmente por Internet a tu hogar y de nuevo a la luz.

Forma en que Google Home enlaza dispositivos Matter mediante Thread

Thread es un estándar de conexión inalámbrica de tipo malla, semejante al Wifi. Sin embargo, se concibió específicamente para la interconexión de dispositivos de hogar inteligente, ofreciendo rapidez, confiabilidad y eficacia superiores.

No todos los aparatos inteligentes para el hogar que admiten Matter y provienen de terceros fabricantes poseen una radio Thread. Es posible que algunos dispositivos Matter se enlacen a Google Home solo a través de la conexión Ethernet o Wifi de tu hogar. Si deseas aprovechar al máximo las capacidades de Matter en Thread junto con Google Home, requerirás un equipo de Google que esté equipado con un router de borde Thread.

Este router de borde Thread presente en tu equipo de Google une la red de dispositivos inteligentes que utilizan Thread con los demás dispositivos en la red Wifi o Ethernet. Esto asegura que todos interactúen de manera unificada en Google Home.

Los siguientes equipos de Google tienen la capacidad de actuar como routers de borde Thread en conjunto con Google Home:

- Monitores: Nest Hub (2ª generación), Nest Hub Max
- Routers Wifi: Nest Wifi Pro (Wifi 6E)

¿Qué equipos y sistemas son compatibles con Matter?

Amazon, Google y Apple, como fundadores de Matter, son las tres principales corporaciones que apoyan este estándar. También se mencionará los dispositivos de Samsung compatibles con Matter, ya que la empresa también ha entrado en el mundo del hogar inteligente.

Compatibilidad de Amazon Alexa con Matter

Aunque los productos inteligentes de Amazon suelen necesitar un hub específico para su comunicación, con Matter pueden conectarse directamente sin necesidad de uno. Los dispositivos de Amazon que cuentan con la certificación Matter incluyen una capa de aplicación específica que asegura una gestión fluida del hogar inteligente por parte del usuario.

Los usuarios vinculan sus dispositivos al altavoz Amazon Echo, que actúa como un controlador de Matter, y gestionan todo el sistema de hogar inteligente con la app Alexa. Por el momento, Amazon Alexa soporta la conexión con Matter a través de Wifi en dispositivos como Echo, Echo Show, Echo Studio, Echo Plus, Echo Dot y Echo Flex.

Compatibilidad de Google Nest y Google Home con Matter

Google Home permite integrar cualquier dispositivo con certificación Matter en el hogar inteligente. Los usuarios pueden manejar su sistema mediante un equipo Google que funciona como un hub de Matter. Además, Google ofrece la posibilidad de gestionar estos dispositivos incluso estando fuera de casa.

Google ha actualizado varios productos para que puedan actuar como centros de Matter, incluyendo Nest Wifi Pro, Google Home, Google Home Mini, Nest Audio, Nest Mini, Nest Hub (primera y segunda generación) y Nest Hub Max.

Compatibilidad de Apple Home con Matter

Los dispositivos certificados por Matter de Apple pueden ser controlados usando la aplicación Home. Actualmente, Apple soporta dispositivos como interruptores, aires acondicionados, cortinas, luces, cerraduras, termostatos y varios sensores en su integración con Matter. Apple también ha lanzado un modo para desarrolladores compatible con Matter para iOS e iPadOS 16.

Compatibilidad de Samsung SmartThings con Matter

Samsung planea activar la compatibilidad con Matter para los hubs SmartThings v2 y v3 en 2023 a través de actualizaciones OTA. Estos actuarán como controladores y routers de borde Thread. Los usuarios podrán vincular dispositivos como frigoríficos, televisores y monitores inteligentes de Samsung que funcionen con Matter.

CSA tiene previsto integrar Matter en electrodomésticos como lavadoras y aspiradoras, así como en cámaras, detectores de humo y dispositivos de calidad del aire hacia finales de año. Esto ampliará el abanico de opciones para los fabricantes de hogares inteligentes, permitiéndoles ofrecer una mayor variedad de productos a sus clientes.

Soluciones de Espressif con Matter

En la **Figura 12**, se puede ver una estructura de las soluciones de Espressif proporciona en el estándar matter:

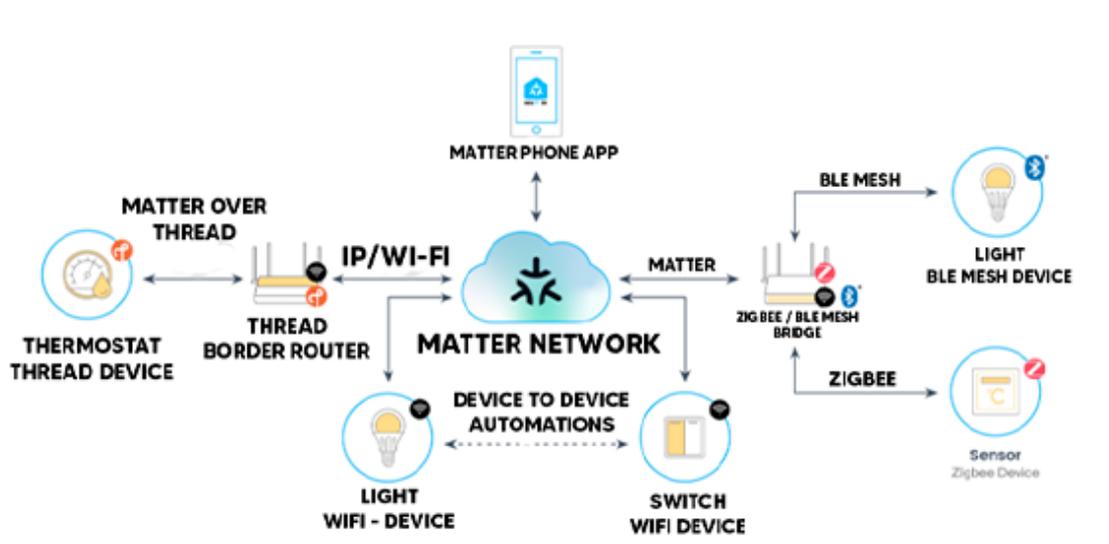


Figura 12. Soluciones de Espressif con Matter

Fuente: Tomado de [41]

Las Soluciones Matter de Espressif consisten en:

- Un espectro completo de plataformas de dispositivos Matter
- SDK listo para producción para Matter
- Integración de Matter y ESP RainMaker

Plataformas de Materia Espressif

Las soluciones de la plataforma Espressif son las que se muestran a continuación, como se ve en la **Figura 13**:

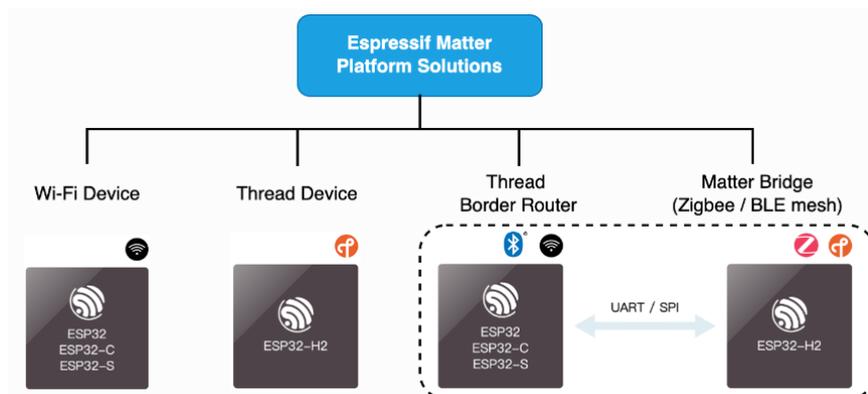


Figura 13. Plataformas de Materia Espressif

Fuente: Tomado de [41]

- Los SoC y módulos habilitados para Wifi, como las series ESP32, ESP32-C y ESP32-S, se pueden utilizar para construir dispositivos Matter Wifi.
- Los SoC y módulos ESP32-H con 802.15.4 se pueden utilizar para construir dispositivos Matter Thread.
- Al combinar eficientemente ESP32-H y nuestros SoC Wifi, se puede construir un enrutador Thread Border para conectar la red Thread con la red Wifi. Proporcionamos kits de desarrollo de hardware, diseños de referencia y SDK listo para producción, que admite la última característica Thread 1.3 para Matter.
- También se proporcionan soluciones de puente Matter-Zigbee y Matter-BLE Mesh que permiten que dispositivos que no son Matter basados en Zigbee, Bluetooth LE Mesh y otros protocolos se conecten al ecosistema Matter. Un Matter-Zigbee Bridge utiliza ESP32-H y otro SoC Wifi, mientras que un Matter-BLE Mesh Bridge se puede realizar en un solo SoC con interfaces Wifi y Bluetooth LE.

1.3.6. Herramientas para la construcción del sistema domótico con protocolo Matter

Visual Code

Este es un editor de código ligero y multiplataforma, el cual es ideal para programar en varios lenguajes como JavaScript, TypeScript y Node.js, C++, C# y ofrece la posibilidad de agregar extensiones para otros lenguajes populares [44].

Home Assistant

Es una plataforma de domótica de código abierto que permite controlar y dar una automatización de dispositivos inteligentes en el hogar. Proporciona una interfaz centralizada para administrar y dar supervisión de diferentes actividades del hogar básicas, como crear escenas, programar tareas y crear reglas personalizadas, esto mediante dispositivos como luces, termostatos, cámaras, cerraduras y otros dispositivos compatibles [45].

WIFI

Según Yan Li, [26] el Wifi es una tecnología inalámbrica fundamental en nuestra era digital, permitiendo la conexión de dispositivos variados como smartphones, ordenadores portátiles, cámaras y más, a Internet. Esta conexión no solo facilita el acceso a la web, sino que también crea una red interconectada, posibilitando la comunicación y el intercambio fluido de información entre dispositivos. Esta interconexión ha transformado la forma en que se comparte y accede a la información, integrando aún más la tecnología en nuestra vida cotidiana.

IDE Arduino

El IDE de Arduino es un software libre y de código abierto diseñado para programar y subir "sketches" o programas a placas Arduino y otros microcontroladores similares. Proporciona un editor sencillo que destaca elementos del código y sugiere correcciones, facilitando el proceso de escritura.

Además, su aplicación es fácil de usar, los usuarios pueden compilar y transferir su código a las placas Arduino. Esta herramienta es fundamental para el desarrollo de diferentes tipos de proyectos del ámbito de la domótica, simplificando la tarea de programación y creación de prototipos en esta plataforma, el desarrollo de un nuevo dispositivo basado en Arduino IDE permite subir el firmware, [46] muestra un ejemplo del uso de IDE Arduino.

Lenguaje de programación C++

Con base a Akifumi Ohno [47], el lenguaje de programación C++ se ha consolidado como uno de los lenguajes más robustos y versátiles en el mundo de la programación. Dotado de un modelo de

memoria y computación avanzado, C++ es sinónimo de alto rendimiento y eficiencia. Pero su verdadero valor radica en su capacidad para ofrecer mecanismos de abstracción potentes y flexibles, permitiendo al desarrollador crear y adaptar tipos de objetos alineados con las especificidades de una aplicación o sistema. Aunque es un descendiente directo del lenguaje C, C++ amplía sus capacidades incorporando características como el desarrollo orientado a objetos. No solo permite a los programadores acceder y manipular directamente los recursos de hardware, sino que también fomenta estilos de programación de alto nivel que se asemejan más a la percepción humana de las tareas computacionales. Estos estilos, que engloban la abstracción de datos, desarrollo orientado a objetos y desarrollo genérico, refuerzan la versatilidad y adaptabilidad del C++, haciendo que siga siendo relevante y preferido por muchos desarrolladores en el ámbito contemporáneo.

ESP32-S3

En la **Figura 14** se presenta al módulo ESP32-S3 del lado izquierdo, y se especifican algunos de sus partes más importantes comparados con el básico de que se conoce más comúnmente.

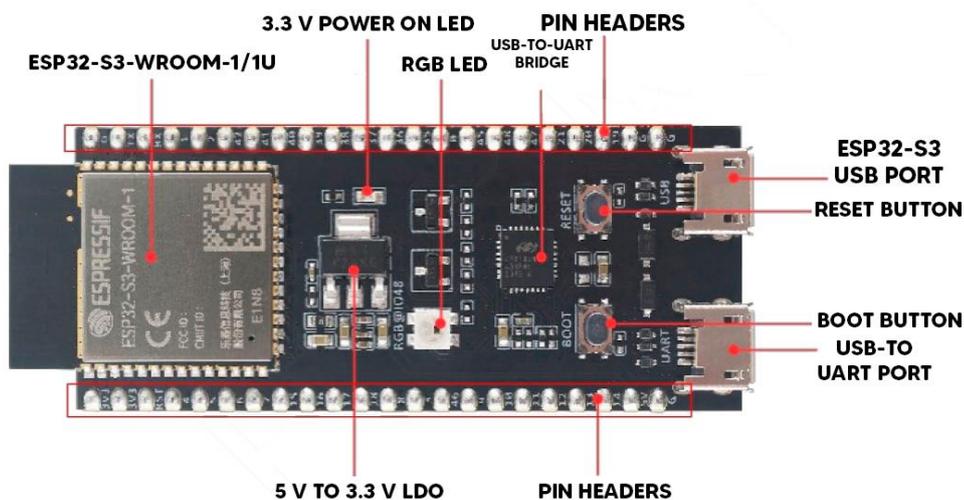


Figura 14. Módulo ESP32-S3

Fuente: Tomado de [48]

El ESP32-S3 es un módulo con microcontrolador que ofrece conectividad Wifi y Bluetooth LE. Cuenta con una amplia gama de periféricos e interfaces, como SPI, LCD, UART, I2C, entre otros. También incluye numerosos GPIOs y una interfaz USB de alta velocidad para la comunicación,

El ESP32-C3 es un microcontrolador diseñado para aplicaciones de IoT (Internet de las cosas) que requieren conectividad Wifi. Pertenece a la familia ESP32 de Espressif, conocida por su versatilidad y rendimiento en entornos de IoT. [49].

Características Principales del Módulo ESP32 S3:

- Tipo: MCU Wifi+Bluetooth
- Chip: ESP32-S3
- Modelo: ESP32-S3-WROOM-1-N8R8
- Tipo de montaje: SMD
- Número de pines: 41
- CPU: Xtensa dual-core de 32 bits LX7 240 MHz
- Memoria Flash: 8 MB Quad SPI
- Memoria PSRAM: 8 MB Octal SPI

ESP32-H2 MINI

En la **Figura 15** se presenta al módulo ESP32-H2-MINI, en el cual se denotan sus partes más importantes y como se encuentra estructurado:

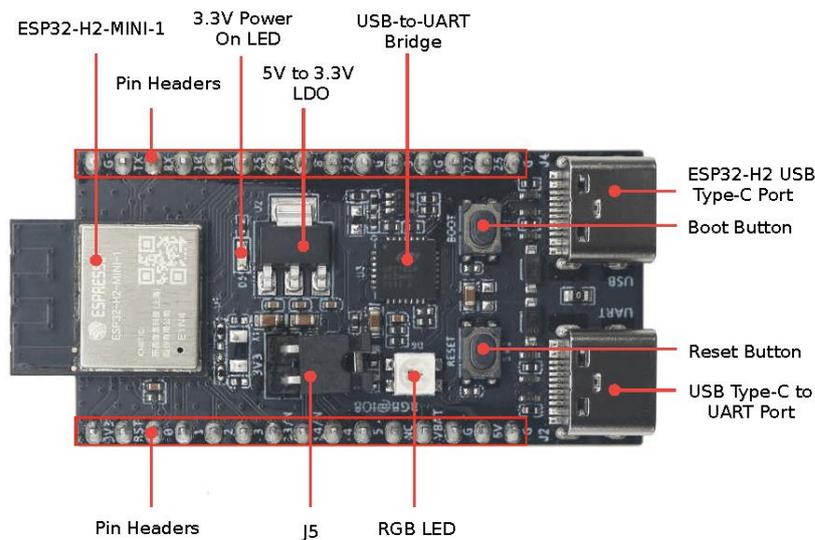


Figura 15. Módulo ESP32 – H2 MINI

Fuente: Tomado de [50]

El ESP32-H2 MINI es un microcontrolador de bajo consumo con conectividad IEEE 802.15.4 y Bluetooth 5. Ofrece memoria RAM y ROM, interfaces de comunicación y GPIO programables. Es seguro y eficiente para dispositivos IoT. Ideal para proyectos domóticos y automatización [51].

Características Principales:

- Frecuencia: 2.4 GHz
- Tipo de interfaz: SPI
- Voltaje de alimentación - Mín.: 3 V
- Voltaje de alimentación - Máx.: 3.6 V
- Temperatura de trabajo mínima: - 40 C
- Temperatura de trabajo máxima: + 105 C
- Dimensiones: 16.6 mm x 13.2 mm x 2.4 mm
- Antena: PCB Antena
- Velocidad de transmisión de datos: 250 kb/s
- Estilo de montaje: PCB Mount
- Corriente de suministro operativa: 500 mA

Dispositivos domóticos con Matter

Se van a utilizar 2 dispositivos Matter comerciales que hay en el mercado tecnológico, el cual va a ser un enchufe inteligente, con Matter y un foco RGB, con tecnología Matter también, y algunos asistentes virtuales como Alexa, Google Nets, y Siri.

a) Asistente Virtual Alexa:

La integración de Alexa es fundamental en este desarrollo del proyecto, precisamente, por la compatibilidad de Alexa con este protocolo es la que facilita que se pueda probar el desarrollo del proyecto. Esta característica de Alexa no solo asegura una comunicación estandarizada con otros dispositivos, sino que también es el eje central que permitirá la realización exitosa y eficiente de todo el desarrollo del proyecto.

b) Asistente Virtual Google Nets:

Al igual que Alexa, este asistente virtual facilitará la comunicación con el dispositivo Matter que estamos desarrollando, demostrando la interoperabilidad entre todos los dispositivos que se conectarán. Dado que Google Nest es parte de las marcas involucradas en el desarrollo del protocolo Matter, nos brindará la oportunidad de probar la conexión entre el asistente y el dispositivo durante el proceso de desarrollo. Esto nos permitirá confirmar que la creación del dispositivo Matter está siendo exitosa.

1.4. Antecedentes Contextuales

1.4.1. Delimitación del contexto de Estudio

Este proyecto tendrá énfasis en el protocolo matter, el cual se realizara la configuración en aplicación de software en dispositivos esp32 de diferentes modelos, llevando a cabo un sistema domótico, que se pueda comunicar entre estos dispositivos personalizados y el dispositivo comercial que se va a adquirir, además de la presencia de un border router el cual va a permitir que los dispositivos matter puedan comunicarse con otros dispositivos con otros protocolos, permitiendo que las conexiones sean plug and play sin tener preocupaciones que protocolo de comunicación usa.

Las limitaciones que se temen, vienen de la disponibilidad y soporte de las librerías matter y codificación, ya que al ser un protocolo que aún está en experimentación, no se tienen mucha información sobre cómo aplicar un claro desarrollo de configuración correcta en los diferentes dispositivos matter, además de la disponibilidad de algunos dispositivos esp32 que no se encuentran en el país.

1.4.2. Impacto Social del protocolo Matter

En la actualidad, la tecnología, la IoT, y las casas inteligentes o la domótica, está abordando nuestras vidas de manera que está transformando y mejorando la forma en que interactuamos con la tecnología en entornos cotidianos. Se debe tener en cuenta, que ya a la fecha esta tecnología existía, pero de una manera oculta que unas personas no estaban del todo seguras, pero al ir evolucionando estas tecnologías en el ámbito de la domótica, esta tecnología está tocando más impacto en todas nuestras vidas, y gracias al desarrollo de un protocolo conocido como Matter, estas tecnologías se podrán unificar varios enlaces de comunicación o marcas, dando paso a la interoperabilidad entre diferentes dispositivos, y así reciclar los dispositivos que se usan ya en casa y no descartarlos de este entorno.

1.4.3. Ámbito de aplicación

Con el auge tecnológico y la creciente interconexión de dispositivos, el Internet de las Cosas (IoT) ha revolucionado la forma en que interactuamos con nuestro entorno. La domótica, como subconjunto de IoT, busca optimizar y automatizar los sistemas y procesos dentro de los hogares y edificaciones. En países en desarrollo, como Ecuador, la adopción de estas tecnologías enfrenta desafíos particulares, desde la falta de estándares unificados hasta barreras de conocimiento y acceso, tomamos una visión del trabajo [52], en cuál nos relata de su enfoque en el estudio que

desarrolla un panel de control público de ThingSpeak y una interfaz gráfica de usuario Virtuino vinculada a un servidor IoT. Estas interfaces permiten a los usuarios acceder convenientemente a la información de calidad del aire en cualquier momento y desde cualquier lugar, y enfocamos un todo esto a Home Assistant.

Dentro de este contexto, surge la necesidad de implementar sistemas domóticos que permitan la integración y control centralizado de dispositivos, tanto comerciales como personalizados, bajo un protocolo estándar como Matter. Este proyecto se centra en la creación de un sistema domótico local que utiliza el protocolo Matter, facilitando la interoperabilidad entre diferentes dispositivos y ofreciendo a los usuarios una experiencia más eficiente y unificada.

El ámbito de aplicación se extiende a hogares y edificaciones en Ecuador que buscan modernizar sus sistemas y dispositivos domóticos. Gracias a la implementación de este sistema, los usuarios podrán gestionar y controlar de manera centralizada sus dispositivos, desde luces y electrodomésticos hasta sistemas de seguridad, aprovechando las ventajas del protocolo Matter y la plataforma Home Assistant. Esta solución no solo mejora la experiencia del usuario, sino que también abre puertas a futuras innovaciones y adaptaciones en el campo de la domótica en la región.

1.4.4. Establecimiento de requerimientos

Con el surgimiento y evolución constante del Internet de las Cosas (IoT) y la domótica, existe una creciente demanda de sistemas integrados que puedan comunicarse de forma eficiente y eficaz entre sí, independientemente del fabricante o tecnología subyacente. La falta de interoperabilidad y la necesidad de una transmisión de datos de alta velocidad en la era del big data industrial[53], ha sido una barrera significativa para la plena realización de los hogares inteligentes, y el protocolo Matter promete abordar este desafío.

El proyecto busca abordar el desafío de interoperabilidad en hogares inteligentes al implementar el protocolo Matter. La visión es ofrecer un sistema domótico integrado que pueda comunicarse fluidamente entre diversos dispositivos y tecnologías, eliminando las barreras actuales de protocolos existentes. Con ESP32 como base hardware y Home Assistant como interfaz de gestión, el objetivo es brindar una solución robusta, escalable y fácil de usar que mejore la experiencia del usuario en su hogar inteligente.

En la **Tabla 5** se detallan los requisitos y sus respectivas funcionalidades, que se esperan del prototipo del sistema domótico:

Tabla 5. Requerimientos del prototipo

Requerimientos	Funcionalidades
Protocolo Matter	<ul style="list-style-type: none"> - Integración con dispositivos domóticos. - Comunicación estandarizada entre dispositivos.
Plataforma Home Assistant	<ul style="list-style-type: none"> - Control centralizado de dispositivos. - Interfaz de usuario amigable. - Configuración personalizada.
Dispositivos ESP32 y ESP32 Border Router	<ul style="list-style-type: none"> - Implementación del firmware y software para Matter. - Funcionamiento como Gateway en el sistema.
Interoperabilidad	<ul style="list-style-type: none"> - Comunicación fluida entre dispositivos comerciales y personalizados. - Adaptabilidad a diferentes protocolos de comunicación.
Control Centralizado (Home Assistant)	<ul style="list-style-type: none"> - Acceso a todos los dispositivos desde una única plataforma. <ol style="list-style-type: none"> 1. Configuración y gestión unificada.
Adaptabilidad y Escalabilidad	<ul style="list-style-type: none"> - Capacidad para añadir nuevos dispositivos con facilidad. <ol style="list-style-type: none"> 2. Expansión del sistema según las necesidades del usuario.
Monitoreo en Tiempo Real	<ul style="list-style-type: none"> - Visualización del estado de los dispositivos. <ol style="list-style-type: none"> 3. Alertas y notificaciones en tiempo real.
Compatibilidad con Otros Protocolos	<ul style="list-style-type: none"> - Integración con dispositivos que usen Wifi, Thread. <ol style="list-style-type: none"> 4. Traducción de protocolos para la comunicación.

CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo

Según señalan Mazón B. y Pan A. [54], hay múltiples estructuras, paradigmas y esquemas teóricos para IoT que han sido sugeridos por entidades, la esfera académica y la industria.

En nuestro prototipo, se consideran varias capas esenciales, para el comienzo y fin de este prototipo de manera correcta, ya que contamos con un tema el cual está siendo desarrollado y actualizado día a día, para eso se ha destacado la resolución dividida en las siguientes capas, en la cual en la **Figura 16** se presentan los componentes que integran cada una de las capas correspondientes, y se los detallan más adelante, se ha destacado algunos modelos de como ver el desarrollo del modelo de capas del prototipo [54]:

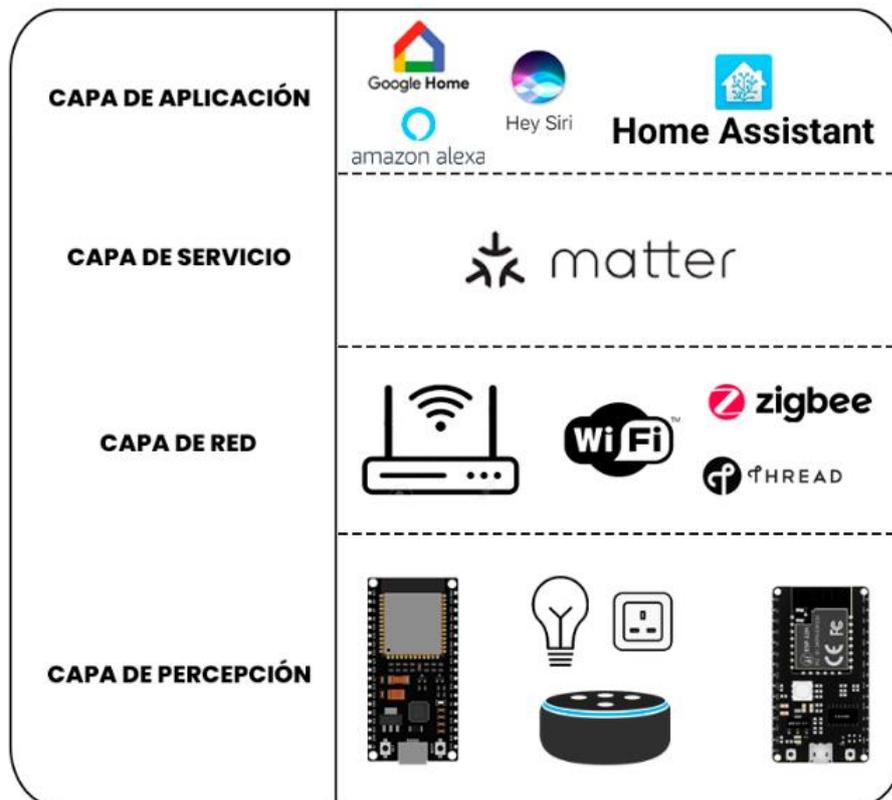


Figura 16. Definición de las capas del prototipo

- **Capa de Percepción:** Esta capa está compuesta por los dispositivos físicos que recopilan y envían datos. Incluye los módulos ESP32 H2 Mini, ESP32 C3, ESP32 S3, ESP32 Border Router, así como otros dispositivos comerciales equipados con protocolos como Matter y Thread.

- **Capa de Red:** En esta capa, se gestionan las comunicaciones entre los dispositivos y el sistema central. Se encarga de la transmisión de datos y garantiza una comunicación fluida y segura entre los dispositivos y el servidor o plataforma central.
- **Capa de Servicio:** Aquí se procesan y almacenan los datos recopilados. Se implementan servicios que permiten la interoperabilidad entre diferentes dispositivos y protocolos, asegurando que todos los dispositivos, independientemente del protocolo que utilicen, puedan comunicarse y trabajar juntos de manera eficiente, en esta parte se encuentran los enlaces de protocolos a usar, como es el Matter, Wifi, Zigbee.
- **Capa de Aplicación:** Esta capa proporciona una interfaz al usuario final a través de Home Assistant, permitiendo visualizar y gestionar los datos recopilados, así como controlar y configurar los dispositivos conectados. Home Assistant facilita la interacción del usuario con el sistema, ofreciendo una experiencia de usuario intuitiva y personalizable, además que tendrá la posibilidad del uso de las aplicaciones que ya están en el mercado como Alexa, Siri, Google Home.

El objetivo principal de este prototipo es demostrar la interoperabilidad de los dispositivos basados en el protocolo Matter con otros dispositivos y protocolos. Al adoptar un enfoque basado en capas, se puede garantizar que cada componente del sistema desempeñe su función de manera eficiente, mientras que, en conjunto, trabajan para lograr una integración y comunicación fluida entre dispositivos de diferentes fabricantes y protocolos.

2.2. Metodología de desarrollo del prototipo

2.2.1. Enfoque, alcance y diseño de investigación

El enfoque de nuestra investigación será principalmente cuantitativo, ya que se recopilarán datos y se medirán variables relacionadas con la implementación del sistema domótico utilizando el protocolo Matter en dispositivos ESP32. Se realizará una evaluación de su eficiencia y rendimiento en términos de conectividad y control de dispositivos, también se destaca que se pretendía escoger la metodología SI4IoT, [55] el cual se presenta como una solución que utiliza la Ingeniería Dirigida por Modelos para simplificar el desarrollo de aplicaciones IoT sofisticadas, demostrando su aplicabilidad mediante un caso de uso en un entorno de Smart Home, pero se complementa con algo más propio y acorde al nuevo protocolo que se está desarrollando.

El alcance de investigación abarcará el desarrollo e implementación del sistema domótico utilizando el protocolo Matter en dispositivos ESP32. Se incluirá la configuración de un Gateway en ESP32 Border Router para conectar los dispositivos Matter, así como la integración con Home

Assistant para el control de los dispositivos domóticos. Además, se evaluará la interoperabilidad entre los dispositivos Matter comerciales y los desarrollados por el equipo de investigación.

El diseño de investigación será descriptivo, ya que se describirán las características y funcionalidades del sistema domótico implementado con Matter en dispositivos ESP32. También se utilizará un enfoque experimental para realizar mediciones y pruebas de rendimiento, como distancia de cobertura, velocidad de transferencia y latencia. Los resultados obtenidos permitirán evaluar la eficacia y el potencial del sistema en términos de mejora de la comunicación y control de dispositivos en un entorno doméstico.

2.2.2. Unidades de análisis

Población (universo)

La población corresponderá a las mediciones de datos realizadas para cada una de las métricas definidas (Interoperabilidad entre dispositivos, Compatibilidad y Conectividad, Escalabilidad y por último consumo de energía).

Muestra

No se va a sacar la muestra debido a que se trabajara con el total de la población.

2.2.3. Técnicas e instrumentos de recopilación de datos

Tabla 6. Técnicas e instrumentos de recopilación de datos

Técnica	Instrumentos
Pruebas de Latencia	Recolección de datos de latencia que se generan en el sistema domótico, mediante la ejecución de comandos, en los dispositivos esp32.

2.2.4. Técnicas de procesamiento de datos para la obtención de resultados

Se aplicará estadística descriptiva:

- Gráfico de Barras.

2.2.5. Metodología o métodos específicos

Para decidir qué metodología se va a implementar, se ha realizado un análisis de tres enfoques distintos: Scrum, Programación Extrema (XP) y Cascada, la cual se ha escogido la Programación Extrema[56].

En la **Tabla 8**, se han registrado las principales explicaciones y las etapas o tareas propuestas por la metodología Programación Extrema (XP).

Tabla 7. Metodología de Desarrollo del sistema

Metodología	Definiciones	Etapas o Fases
Programación Extrema	Programación Extrema es una metodología ágil que se centra en la calidad del código y en la participación del cliente para lograr una entrega rápida y continua de software.	<ul style="list-style-type: none"> - Planificación del proyecto - Diseño - Desarrollo de características - Pruebas - Integración

Después de examinar las definiciones y las etapas propuestas, se llevó a cabo un análisis considerando el objetivo y la problemática del proyecto para seleccionar la que mejor se ajustara. Como resultado, se optó por la metodología de Programación Extrema (XP) [57]. Esta elección se fundamentó en que XP abarca completamente el proceso de desarrollo del trabajo de tesis, lo cual es esencial, y en la simplicidad de sus actividades en comparación con las metodologías Scrum y Cascada. La Programación Extrema (XP) es una estrategia ágil bien establecida en la gestión de proyectos. Se personalizó para este proyecto, permitiendo ajustes y validaciones a lo largo del desarrollo. La estrategia incluye las siguientes fases, como se puede evidenciar en la **Figura 17**:

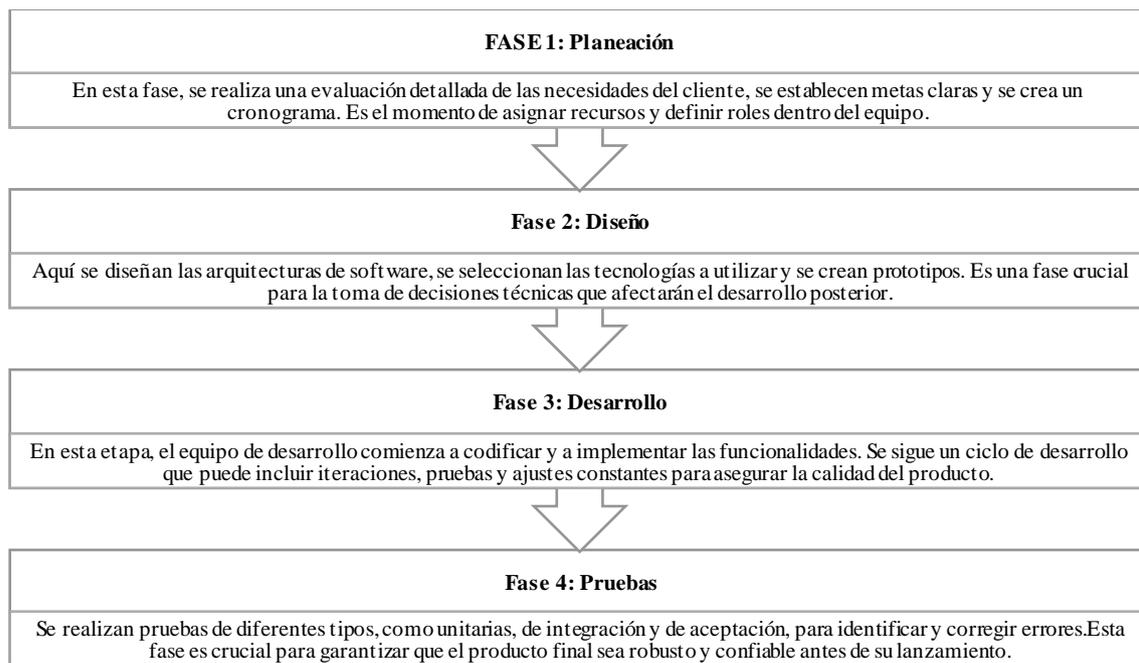


Figura 17. Fases de la metodología XP

Según en el trabajo de [58], destacan que la Programación Extrema (XP) se ha destacado recientemente como un enfoque ágil prominente en el ámbito del desarrollo de software.

2.2.5.1. Métodos

- **Método hipotético-deductivo:** Este método se basa en la formulación de una hipótesis y luego en la realización de pruebas o experimentos para confirmar o refutar dicha hipótesis. Es un enfoque lógico y deductivo para validar o descartar suposiciones.
- **Método analítico:** Se emplea para descomponer un objeto de estudio en partes individuales y analizarlas minuciosamente. Permite comprender cada componente por separado y cómo se relacionan con el todo, lo que facilita la obtención de información detallada.
- **Método estadístico:** Este método se utiliza para recopilar, organizar y analizar datos numéricos. Permite tomar decisiones basadas en la interpretación de los resultados estadísticos, lo que proporciona una comprensión cuantitativa de los fenómenos estudiados.
- **Método bibliográfico:** Se basa en la revisión y análisis de diversas fuentes bibliográficas, como libros, artículos, tesis, y otros documentos científicos. Permite obtener información relevante y fundamentar el estudio en la investigación previa realizada por otros.
- **Método propositivo:** Este enfoque se centra en la formulación de propuestas o soluciones a problemas identificados en el estudio. Se utiliza para plantear alternativas y desarrollar estrategias que aborden las cuestiones investigadas.

2.2.6. Herramientas y/o Materiales

La información que establecimos en la **Tabla 8** detallamos las herramientas de software y hardware que se usaron para llevar a cabo el presente estudio:

Tabla 8. Herramientas o Materiales

Herramientas de Software	Herramientas de Hardware
Visual Studio Code	ESP32 – S3 – C3
Home Assistant	ESP32 – H2 MINI U1
Plataforma IDE Arduino	ESP32 - Border Router
Lenguaje C++	Dispositivos comerciales con Protocolo Matter
Protocolo Thread	Router

2.3. Desarrollo del prototipo

Para la estructuración de este sistema domótico que utiliza el protocolo Matter se adoptó la metodología XP, orientada especialmente para soluciones IoT. Esta adaptación encajará perfectamente con el desarrollo del sistema y su integración con Home Assistant para la supervisión y administración de los dispositivos domóticos que se desarrollarán.

2.3.1. FASE 1: Planeación

Esta fase inicial es crucial en nuestra versión adaptada de la metodología XP. Aquí se precisan las iteraciones y se configuran los lineamientos esenciales que marcarán la evolución del proyecto.

Roles del Equipo XP

La metodología distingue las distintas funciones esenciales para mantener un equilibrio eficiente en el trabajo del equipo. Es posible que algunos miembros desempeñen varias funciones, y un individuo puede asumir varios roles. La asignación de estas responsabilidades se especifica en la **Tabla 9** a continuación:

Tabla 9. Roles del equipo de trabajo

Rol	Descripción	Encargado
Usuario	Representa al propietario o habitante de la vivienda con domótica, asumiendo la responsabilidad de definir y priorizar los requerimientos y funciones del sistema domótico basado en el protocolo Matter.	Usuario de casa domótica
Desarrollador	Es el encargado de redactar y mantener el código para la implementación del protocolo Matter, garantizando su eficiencia mediante pruebas unitarias y optimización del código.	Pambi Kenneth Fajardo Marco
Verificador	Se centra en confirmar la operatividad y calidad del sistema, realizando pruebas manuales y automáticas, y suministrando retroalimentación al equipo acerca de la funcionalidad del sistema domótico.	Pambi Kenneth Fajardo Marco
Gestor de Proyecto	Se encarga de gestionar el listado de requerimientos y funciones del sistema domótico, asegurando que las tareas esenciales se efectúen en el periodo definido.	Fajardo Marco Pambi Kenneth
Entrenador	Su función es asistir al equipo en potenciar su eficacia y calidad, a través de formación y asesoría en las técnicas y prácticas de XP adaptado al protocolo Matter.	Dixys Hernández

Historia de Usuarios:

La **Tabla 10** presenta las historias de usuarios, destacando los progresos realizados en cada fase y señalando a los participantes responsables en cada etapa:

Tabla 10. Resumen de historias de usuarios

ID	Historia de Usuario	Descripción	Involucrados	Prioridad
HU1	Integración con Dispositivos	Implementar el protocolo Matter para lograr la comunicación entre dispositivos domésticos variados.	Fajardo Marco Pambi Kenneth	Alta
HU2	Interfaz de Control	Desarrollo de una interfaz amigable que permita al usuario doméstico controlar y monitorear sus dispositivos, dentro de Home Assistant	Pambi Kenneth	Alta
HU3	Privacidad	Reforzar la seguridad de las comunicaciones a través de cifrado y autenticación, según los estándares del protocolo Matter.	Pambi Kenneth	Media
HU4	Optimización del Rendimiento	Evaluar y mejorar la eficiencia y velocidad de respuesta del sistema, asegurando la fluidez de la experiencia del usuario.	Fajardo Marco Pambi Kenneth	Baja
HU5	Capacidades de Diagnóstico	Incorporar herramientas que permitan identificar y solucionar problemas relacionados con la conectividad o el funcionamiento de los dispositivos.	Pambi Kenneth Fajardo Marco	Baja
HU6	Interoperabilidad	Garantizar que diferentes dispositivos, independientemente de la marca o el fabricante, puedan interactuar sin problemas a través del protocolo Matter.	Fajardo Marco Pambi Kenneth	Alta
HU7	Desarrollo de los dispositivos Matter	Desarrollar de manera correcta los dispositivos ESP32, con su correspondiente protocolo Matter.	Pambi Kenneth Fajardo Marco	Alta

Plan de Iteraciones

Basándonos en la tabla previa, la **Tabla 11** ilustra el esquema de iteraciones conforme a la importancia de cada historia de usuario:

Tabla 11. Plan de iteraciones

Iteración	Orden de las historias de usuario	Duración
1re iteración	HU7, HU6, HU1	5 semanas
2da iteración	HU3, HU2	4 semanas
3ra iteración	HU4, HU5	3 semanas

Durante la primera iteración, se establece el desarrollo de las historias de usuario HU7, HU6 y HU1 con una duración prevista de 5 semanas. Para la segunda iteración, se ha programado trabajar en las historias HU3 y HU2, con un periodo estimado de 4 semanas. Finalmente, en la tercera iteración, las historias HU4 y HU5 se han planificado para ser llevadas a cabo en un lapso de 3 semanas. Es evidente que la organización de las historias de usuario en distintas iteraciones se basa

en criterios de prioridad y niveles de complejidad. Estas iteraciones han sido estructuradas con duraciones que varían entre 3 y 5 semanas, promoviendo así entregas regulares y permitiendo una constante retroalimentación entre los miembros del equipo.

2.3.2. FASE 2: Diseño

En la **Figura 18** Se visualiza un esquema sencillo del diseño del dispositivo de temperatura del ESP32, con un sensor de temperatura DHT11 en el lado izquierdo y en el lado derecho se puede ver como se estableció el diseño del pulsador con el Esp32 -C3, en el cual se procedió a realizar su debida conexión de cada uno, los dos están hechos con un Esp32 -C3:

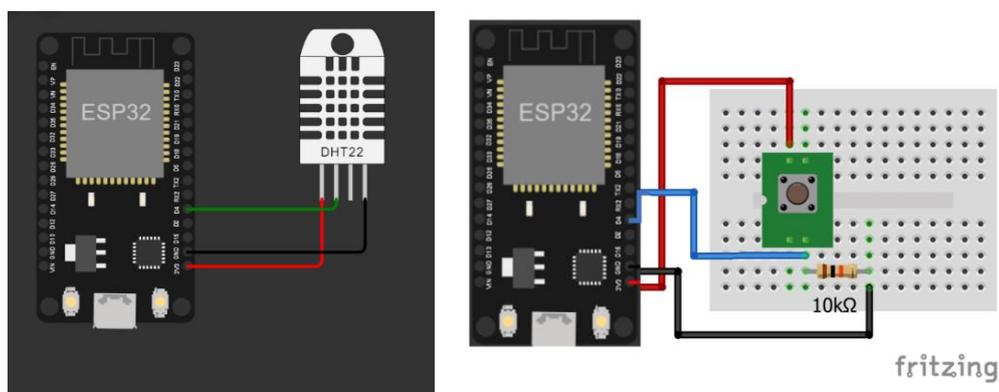


Figura 18. Diseño del prototipo del dispositivo esp32 para de sensor de temperatura y pulsador

En la **Figura 19** se visualiza el esquema de conexiones del ESP32 S3. Este módulo central está conectado a un foco LED y a un pulsador. Ambos componentes se conectan al ESP32 S3 a través de pines específicos. El foco LED, representado como (1), se encarga de la señalización luminosa, mientras que el pulsador, indicado como (2), permite la interacción del usuario. El ESP32 S3, señalado en el diagrama como (3), es el corazón del sistema donde se instalará el protocolo Matter, facilitando la comunicación y control de los dispositivos conectado.

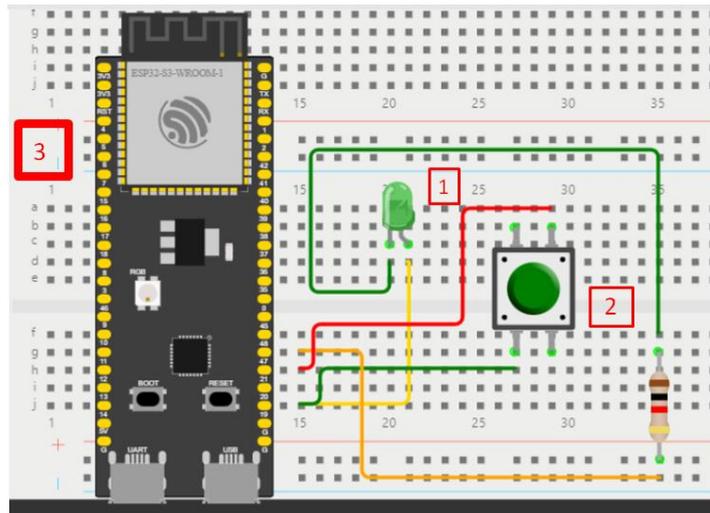


Figura 19. Diseño del Prototipo del dispositivo led

En la **Figura 20** se presenta el diseño de nuestro sistema domótico basado en el protocolo Matter. La primera capa muestra dispositivos que utilizan tanto el protocolo Matter como un protocolo externo, en este caso, Thread. Dentro de esta capa, encontramos el ESP32-S3, que tiene incorporado el protocolo Matter, y un Gateway que emplea Thread. Es importante destacar que también se incluye un dispositivo certificado por Matter para evaluar la interoperabilidad entre diferentes dispositivos. En la segunda capa, se evidencia la conexión mediante red Wifi, que permite la interacción con asistentes virtuales. Esta red también facilita la conexión con Home Assistant, que proporciona una interfaz adecuada para gestionar eficientemente los dispositivos, evidenciando así la interoperabilidad entre dispositivos Matter y diversas marcas.

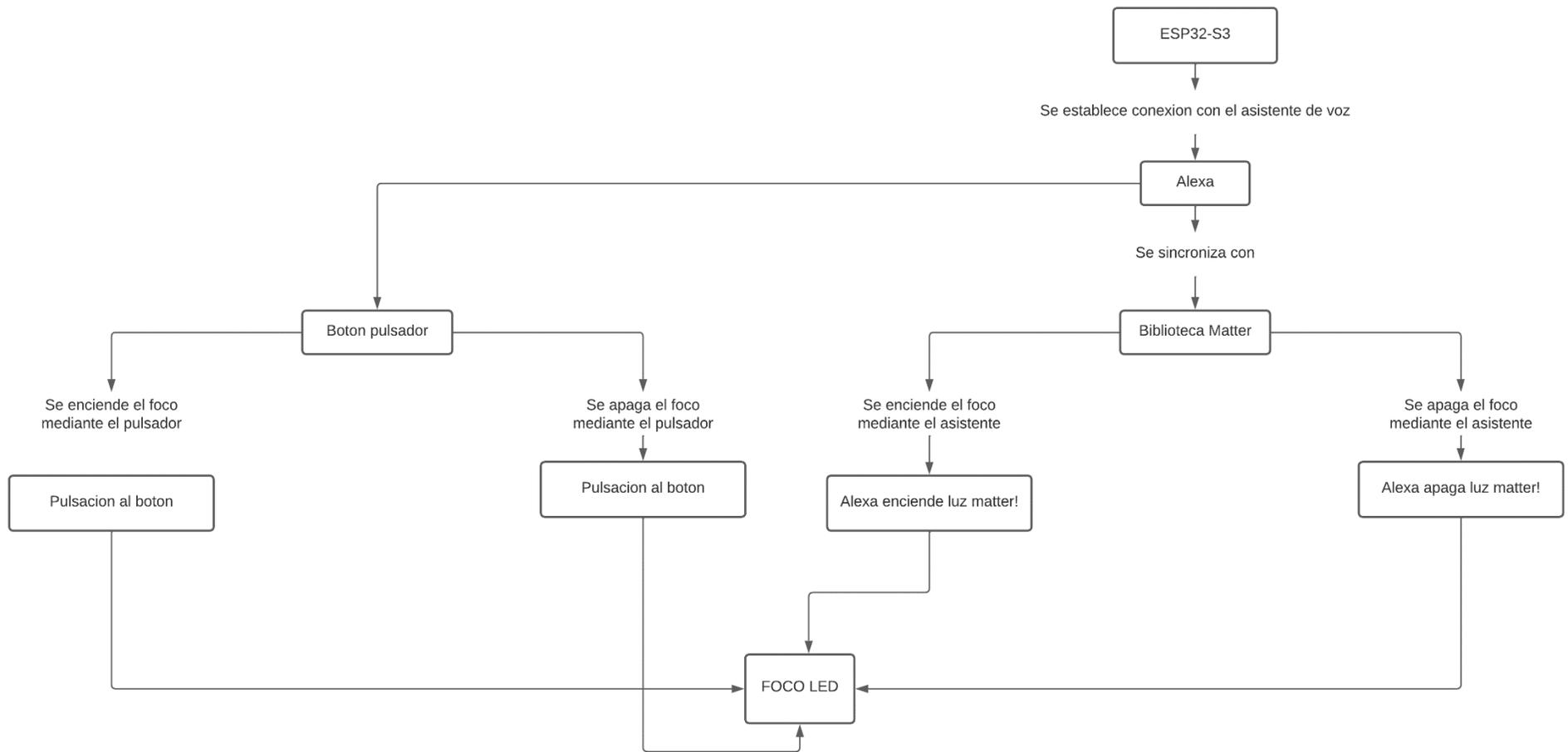


Figura 20. Diagrama de Flujo del sistema domótico matter

2.3.3. FASE 3: Desarrollo

Programación del protocolo Matter en el ESP32-S3

A continuación, se denotará el uso de la librería correspondiente para la instalación del protocolo Matter en cuál, se encuentra disponible su codificación en el repositorio de GitHub: (<https://github.com/Yacubane/esp32-arduino-matter/releases>), en cuál contiene todo lo necesario para que nuestro esp32 S3, sea reconocido como dispositivo Matter, claro se debe tomar en cuenta los pasos que se debe tomar para la instalación de este protocolo y su perfecto desarrollo, para el cual usamos la plataforma de IDE Arduino y comenzamos su instalación, primero conectamos nuestro esp32 S3 a nuestro pc, y se procede a abrir nuestro IDE Arduino, y buscamos entre nuestras opciones como se ve en la **Figura 21**:

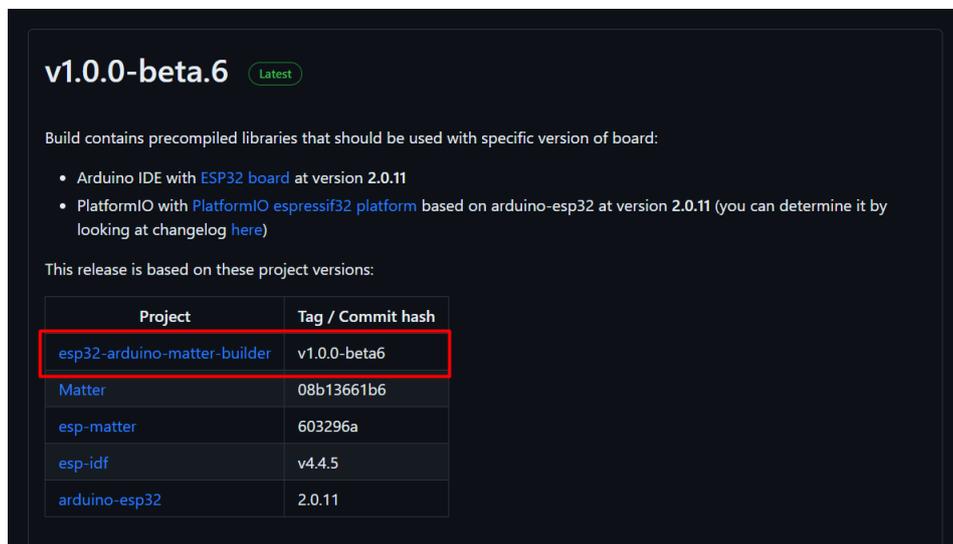


Figura 21. Ventana de descarga de código matter

Se comenzó descargando la codificación del protocolo Matter, herramienta esencial en proyectos domóticos. Una vez obtenido el código, este se integró en el entorno de desarrollo IDE Arduino. Con la codificación adecuadamente configurada en el IDE, se procedió a cargarla en el hardware correspondiente, marcando así un avance significativo en la implementación del sistema.

En la **Figura 22**, se puede ver cómo está estructurado el código Matter.

```
LUZMATTER.ino  Matter.h  x
1  #pragma once
2
3  #define CHIP_HAVE_CONFIG_H true
4  #define CHIP_ADDRESS_RESOLVE_IMPL_INCLUDE_HEADER <lib/address_resolve/AddressResolve_DefaultImpl.h>
5
6  // undef ESP32 while importing Matter
7  // ESP32 define makes some defines in Matter headers not working
8  #undef ESP32
9  // import sdkconfig from exported Matter project
10 #include "sdkconfig_matter.h"
11 #include "esp_matter.h"
12 #include "esp_matter_console.h"
13 #include "esp_matter_ota.h"
14 #include "esp_matter_providers.h"
15
16 #include "esp32-arduino-matter_patches_matter.h"
17 #define ESP32
18 #include "esp32-arduino-matter_patches_arduino.h"
19
```

Figura 22. Código del protocolo Matter

En la **Figura 23**, **Figura 24** y **Figura 25**, se puede observar la subida del protocolo matter el cual nos permitirá indagar en su estructura, proporcionarlo a nuestros dispositivos esp32:

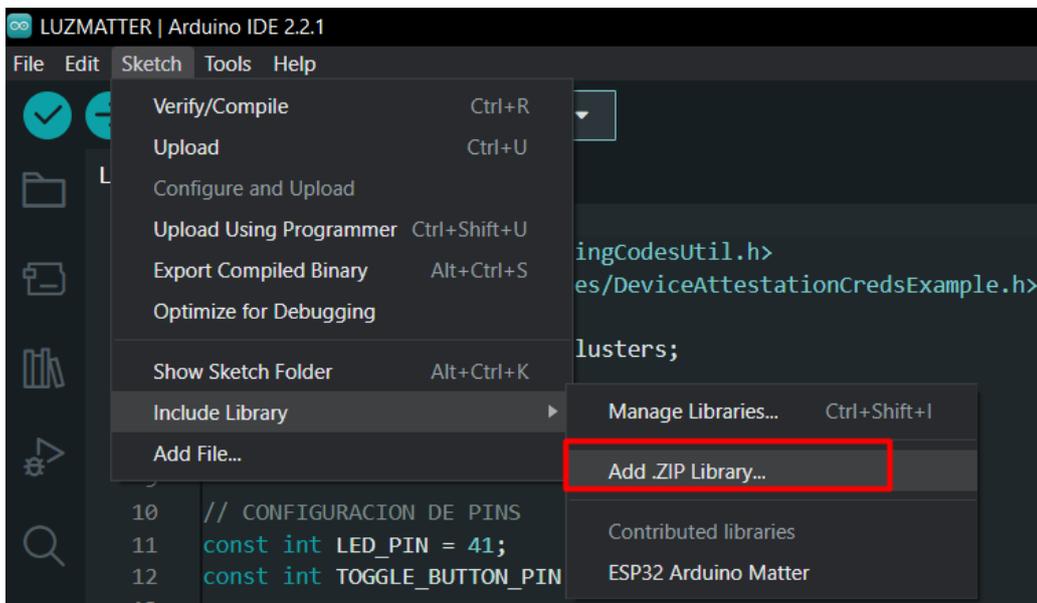


Figura 23. Selección de la librería matter

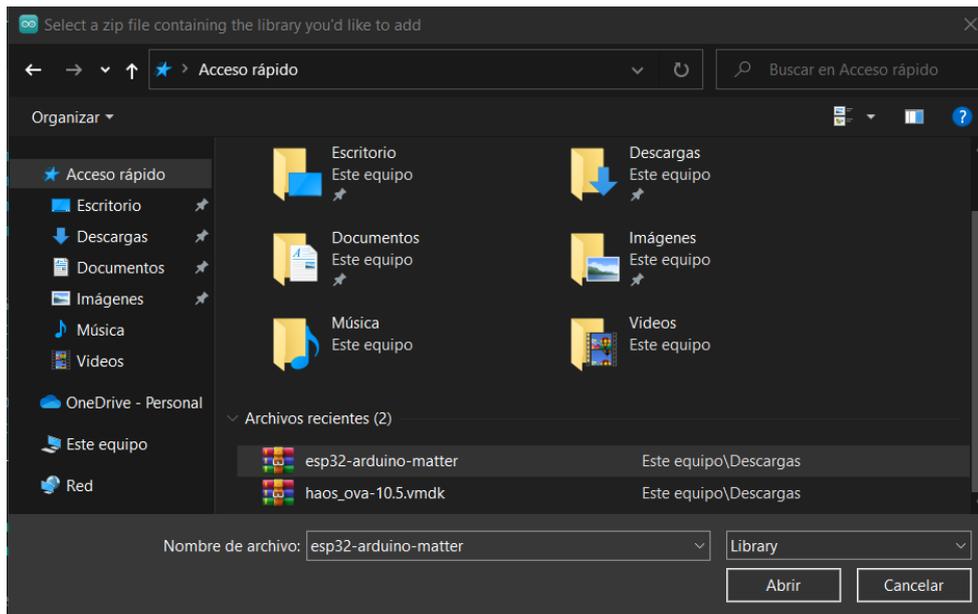


Figura 24. Ventana de subida de protocolo matter

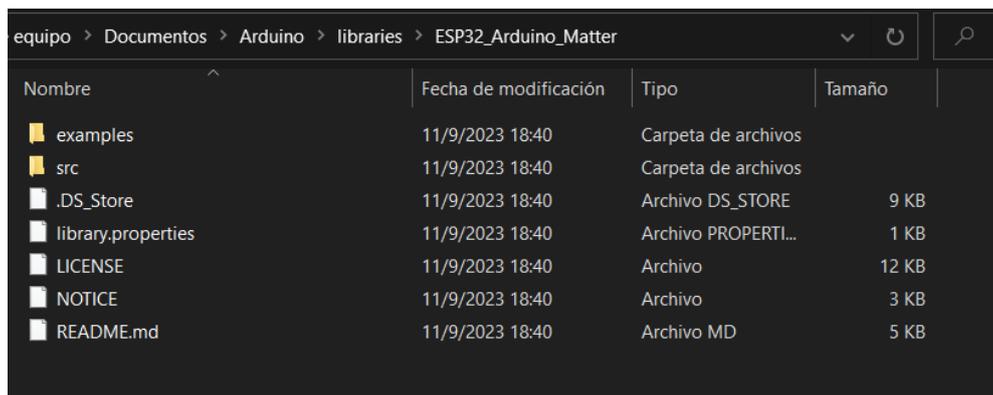


Figura 25. Carpeta que contiene el protocolo matter

Se procedió a conectar el ESP32-S3 al entorno de desarrollo. Una vez establecida la conexión, se cargó el protocolo Matter en el ESP32-S3. Tras esta acción, la compilación se llevó a cabo de manera exitosa, confirmándose así la correcta integración del protocolo en el dispositivo.

En la **Figura 26** se ve la compilación de nuestro código Matter en el dispositivo ESP32-S3.

```

Output
Library ESP32 Arduino Matter has been declared precompiled:
Using precompiled library in C:\Users\jeand\Documents\Arduino\libraries\ESP32_Arduino_Matter\src\esp32s3

```

Figura 26. Compilación del código de protocolo matter

En la **Figura 27**, se ve como nuestro código matter se compiló de manera correcta.

```
Output
Writing at 0x00154af1... (89 %)
Writing at 0x00159f32... (90 %)
Writing at 0x00160155... (92 %)
Writing at 0x0016883d... (94 %)
Writing at 0x00170caa... (96 %)
Writing at 0x00176aa4... (98 %)
Writing at 0x0017c0f5... (100 %)
Wrote 1510752 bytes (897744 compressed) at 0x00010000 in 9.1 seconds (effective 1322.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin... Done
```

Figura 27. Compilación completada y subida al esp32-S3

Programación del ESP32-S3 con el Protocolo Matter para la Iniciación y Control de Iluminación LED

Primeramente, para tener todo en orden con nuestro entorno de trabajo IDE Arduino, vamos a activar el C++17 para el perfecto funcionamiento de este, a continuación, en la **Figura 28**, se puede observar que se debe reemplazar una parte de su estructura que está identificada como “std=gnu”, y según su versión, se debe reemplazarla con esta cadena de palabras “std=gnu++17”:

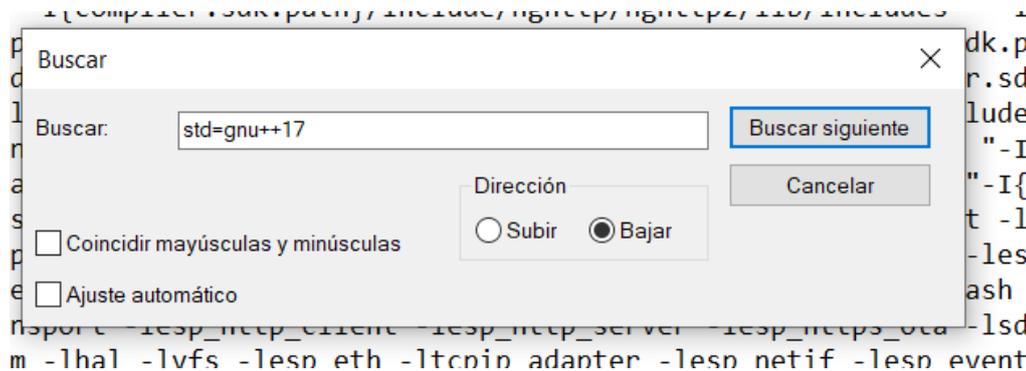


Figura 28. Activación de C++17 en el IDE Arduino

Para la correcta programación y funcionamiento del ESP32-S3 en el contexto del control de iluminación LED mediante el protocolo Matter, se inició con la importación de las bibliotecas necesarias. Estas bibliotecas proporcionan las funciones y herramientas esenciales para trabajar con el dispositivo y el protocolo mencionado. Posteriormente, se llevó a cabo la configuración de los pines del ESP32-S3, como se puede ver en la **Figura 29**, en la cual se ha definido los roles específicos para cada uno, asegurando así una comunicación y control eficientes del foco LED:

Figura 30, se puede ver fragmento del código al que nos referimos

```
// Cuando se solicita una actualización, se verifica la ruta (punto final, clúster y atributo)
// si coincide con el atributo de luz. En caso afirmativo, el LED cambia de estado a uno nuevo.
static esp_err_t on_attribute_update(attribute::callback_type_t type, uint16_t endpoint_id, uint32_t cluster_id,
                                   uint32_t attribute_id, esp_matter_attr_val_t *val, void *priv_data) {
    if (type == attribute::PRE_UPDATE && endpoint_id == light_endpoint_id &&
        cluster_id == CLUSTER_ID && attribute_id == ATTRIBUTE_ID) {
        // We got an light on/off attribute update!
        bool new_state = val->val.b;
        digitalWrite(LED_PIN, new_state);
    }
    return ESP_OK;
}

void setup() {
    Serial.begin(115200);
    pinMode(LED_PIN, OUTPUT);
    pinMode(TOGGLE_BUTTON_PIN, INPUT_PULLUP);

    // Habilitar el registro de depuración
    esp_log_level_set("*", ESP_LOG_DEBUG);

    // Setup Matter node
    node::config_t node_config;
    node_t *node = node::create(&node_config, on_attribute_update, on_identification);

    // Configurar punto final/clúster/atributos de Light con valores predeterminados
    on_off_light::config_t light_config;
    light_config.on_off.on_off = false;
    light_config.on_off.lighting.start_up_on_off = false;
    endpoint_t *endpoint = on_off_light::create(node, &light_config, ENDPOINT_FLAG_NONE, NULL);
}
```

Figura 30. Configuración de clúster y nodos

Además, se guarda una referencia al atributo de encendido/apagado, que se utiliza más adelante para leer y actualizar el valor del atributo. También se configura y se inicia el dispositivo Matter con información de emparejamiento. Finalmente, se proporcionan funciones para leer y establecer el valor del atributo de encendido/apagado de la luz, permitiendo así una interacción efectiva entre los componentes del sistema domótico, en la siguiente **Figura 31**, se puede encontrar fragmento del código de inicio del dispositivo Matter y los métodos de acción de recibir datos del encendido y apagado.

```

// Guardar referencia de atributo activado/desactivado. Se utilizará para leer el valor del atributo más adelante.
attribute_ref = attribute::get(cluster::get(endpoint, CLUSTER_ID), ATTRIBUTE_ID);

// Guardar ID de punto final generado
light_endpoint_id = endpoint::get_id(endpoint);

// Configurar DAC (este es un buen lugar para configurar también datos de comisiones personalizados, códigos de acceso, etc.)
esp_matter::set_custom_dac_provider(chip::Credentials::Examples::GetExampleDACProvider());

// Iniciar dispositivo Matter
esp_matter::start(on_device_event);

// Código de pareamiento
PrintOnboardingCodes(chip::RendezvousInformationFlags(chip::RendezvousInformationFlag::kBLE));
}

// Lee el valor del atributo de encendido/apagado de la luz.
esp_matter_attr_val_t get_onoff_attribute_value() {
    esp_matter_attr_val_t onoff_value = esp_matter_invalid(NULL);
    attribute::get_val(attribute_ref, &onoff_value);
    return onoff_value;
}

// Establece el valor del atributo de encendido/apagado de la luz
void set_onoff_attribute_value(esp_matter_attr_val_t* onoff_value) {
    attribute::update(light_endpoint_id, CLUSTER_ID, ATTRIBUTE_ID, onoff_value);
}

```

Figura 31. Código de inicio de dispositivo Matter

A continuación, la función loop() se encarga de monitorizar constantemente el estado del botón. Se utiliza un retardo de rebote (DEBOUNCE_DELAY) para evitar lecturas erróneas o repetitivas del botón. Si el botón es presionado, se lee el valor actual del atributo de encendido/apagado del LED, se invierte dicho valor y, posteriormente, se establece el nuevo estado. Esta acción permite que el botón actúe como un interruptor de encendido/apagado para el LED, y también puede ser controlado mediante un asistente de voz a través de la lógica implementada en el protocolo Matter, así como se ve en la **Figura 32**:

```

// Cuando se presiona el botón de alternancia de luz (con antirrebote),
// se cambia el valor del atributo de luz
void loop() {
    if ((millis() - last_toggle) > DEBOUNCE_DELAY) {
        if (!digitalRead(TOGGLE_BUTTON_PIN)) {
            last_toggle = millis();
            // Read actual on/off value, invert it and set
            esp_matter_attr_val_t onoff_value = get_onoff_attribute_value();
            onoff_value.val.b = !onoff_value.val.b;
            set_onoff_attribute_value(&onoff_value);
        }
    }
}

```

Figura 32. Código de la función loop

Adicional a este proceso se debe aumentar, que para el otro dispositivo matter se estableció un avance de desarrollo de un dispositivo matter de temperatura, el cual se procedió a desarrollar el siguiente código, en el cual se puede ver en la **Figura 33**, que este fragmento de código, que incluye bibliotecas del protocolo matter y configura un sensor DHT de temperatura y humedad en el pin 4, además define identificadores para el clúster y

atributos del sensor, crea referencias a estos atributos y establece una función de devolución de llamada para manejar eventos del dispositivo en el contexto del protocolo matter:

```
#include <DHT.h>
#include "Matter.h"
#include <app/server/OnboardingCodesUtil.h>
#include <credentials/examples/DeviceAttestationCredsExample.h>
using namespace chip;
using namespace chip::app::Clusters;
using namespace esp_matter;
using namespace esp_matter::endpoint;

#define DHT_PIN 4
#define DHT_TYPE DHT11

// Setup DHT sensor
DHT dht(DHT_PIN, DHT_TYPE);

// Cluster and attribute ID used by the temperature and humidity sensor
const uint32_t CLUSTER_ID = TemperatureMeasurement::Id;
const uint32_t ATTRIBUTE_ID_TEMPERATURE = TemperatureMeasurement::Attributes::MeasuredValue::Id;
const uint32_t ATTRIBUTE_ID_HUMIDITY = RelativeHumidityMeasurement::Attributes::MeasuredValue::Id;

// Endpoint and attribute references that will be assigned to the temperature and humidity sensor
uint16_t sensor_endpoint_id = 0;
attribute_t *temperature_attribute_ref;
attribute_t *humidity_attribute_ref;

// Device event callback
static void on_device_event(const ChipDeviceEvent *event, intptr_t arg) {}
```

Figura 33. Codificación del sensor DHT y esp32

En la **Figura 34**, se implementa el código de dos funciones de devolución de llamada en el dispositivo. La función **on_identification** maneja la identificación del dispositivo, y la función **on_attribute** se encarga de las actualizaciones de atributos. En el caso de una identificación, simplemente confirma que se realizó con éxito. Para las actualizaciones de atributos, verifica si el clúster y el atributo corresponden a la temperatura o la humedad del dispositivo y actualiza los valores apropiados. En pocas palabras, este código permite que el dispositivo se comunique y actualice sus atributos correctamente en una red Matter:

```

// Identification callback
static esp_err_t on_identification(identification::callback_type_t type, uint16_t endpoint_id,
                                  uint8_t effect_id, uint8_t effect_variant, void *priv_data) {
    return ESP_OK;
}

// Attribute update callback
static esp_err_t on_attribute_update(attribute::callback_type_t type, uint16_t endpoint_id, uint32_t cluster_id,
                                    uint32_t attribute_id, esp_matter_attr_val_t *val, void *priv_data) {
    if (type == attribute::PRE_UPDATE && endpoint_id == sensor_endpoint_id) {
        if (cluster_id == CLUSTER_ID && attribute_id == ATTRIBUTE_ID_TEMPERATURE) {
            // Update temperature attribute
            esp_matter::attribute_t temperature_attribute_ref;
        } else if (cluster_id == CLUSTER_ID && attribute_id == ATTRIBUTE_ID_HUMIDITY) {
            // Update humidity attribute
            esp_matter::attribute_t humidity_attribute_ref;
        }
    }
    return ESP_OK;
}

```

Figura 34. Establecimiento de funciones en para la interacción matter

En la **Figura 35**, se puede denotar que el código en la función **setup ()** configura un dispositivo para operar en una red matter. Se inicia la comunicación serial, se habilita el registro de mensajes de depuración y se crea un nodo matter con funciones de devolución de llamada para manejar eventos y actualizaciones de atributos. Se configura un sensor de temperatura y humedad, se establecen referencias a sus atributos, y se configura un proveedor de datos de acceso directo personalizado. Finalmente, se inicia el dispositivo matter, listo para unirse a la red y gestionar eventos:

```

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

    // Enable debug logging
    esp_log_level_set("", ESP_LOG_DEBUG);

    // Setup Matter node
    node::config_t node_config;
    node_t *node = node::create(&node_config, on_attribute_update, on_identification);

    // Setup temperature and humidity sensor endpoint/cluster/attributes with default values
    temperature_sensor::config_t sensor_config;
    sensor_config.temperature_measurement.measured_value = 0.0;
    endpoint_t *endpoint = temperature_sensor::create(node, &sensor_config, ENDPOINT_FLAG_NONE, NULL);

    // Save temperature and humidity attribute references
    temperature_attribute_ref = attribute::get(cluster::get(endpoint, CLUSTER_ID), ATTRIBUTE_ID_TEMPERATURE);
    humidity_attribute_ref = attribute::get(cluster::get(endpoint, CLUSTER_ID), ATTRIBUTE_ID_HUMIDITY);

    // Save generated endpoint id
    sensor_endpoint_id = endpoint::get_id(endpoint);

    // Setup DAC (this is a good place to also set custom commission data, passcodes, etc.)
    esp_matter::set_custom_dac_provider(chip::Credentials::Examples::GetExampleDACProvider());

    // Start Matter device
    esp_matter::start(on_device_event);
}

```

Figura 35. Implementación del código para el inicio de los eventos y atributos

Ahora, por último, en la **Figura 36**, se establece el código que inicia el sensor de temperatura y humedad DHT, lee sus valores, actualiza los atributos de temperatura y humedad en un dispositivo Matter y espera 5 segundos antes de repetir el proceso. También imprime códigos de incorporación del dispositivo Matter:

```
// Print onboarding codes
PrintOnboardingCodes(chip::RendezvousInformationFlags(chip::RendezvousInformationFlag::kBLE));

// Initialize DHT sensor
dht.begin();
}

void loop() {
// Read temperature and humidity from DHT sensor
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();

// Update temperature and humidity attributes
esp_matter_attr_val_t temperature_value;
temperature_value.val.f = temperature;
attribute::update(sensor_endpoint_id, CLUSTER_ID, ATTRIBUTE_ID_TEMPERATURE, &temperature_value);

esp_matter_attr_val_t humidity_value;
humidity_value.val.f = humidity;
attribute::update(sensor_endpoint_id, CLUSTER_ID, ATTRIBUTE_ID_HUMIDITY, &humidity_value);

delay(5000);
}
```

Figura 36. Codificación para el inicio y lectura de la temperatura del sensor

A continuación, en el proceso de configuración del dispositivo Matter, se establecen códigos de emparejamiento y un código QR para facilitar su integración con otros dispositivos y aplicaciones. Durante la inicialización en la función “setup ()”, se configura un proveedor DAC personalizado mediante `esp_matter::set_custom_dac_provider()`, aprovechando el proveedor de ejemplos `chip::Credentials::Examples::GetExampleDACProvider()`.

Posteriormente, el dispositivo Matter se inicia con `esp_matter::start(on_device_event)`. Como parte final de la configuración, se invoca la función `PrintOnboardingCodes(chip::RendezvousInformationFlags(chip::RendezvousInformationFlag::kBLE))`, que genera e imprime códigos de emparejamiento, como se puede ver en la **Figura 37** y un código QR que se puede identificar en la **Figura 38**. Estos elementos son esenciales para permitir una conexión segura y sencilla del dispositivo con controladores Matter, como aplicaciones móviles u otros sistemas.

```
Output Serial Monitor x
Message (Enter to send message to 'ESP32S3 Dev Module' on 'COM3')
I (1281) chip[DL]: WIFI_EVENT_STA_START
E (1281) esp_matter_core: Failed to open the node nvs_namespace
I (1284) chip[DL]: Done driving station state, nothing else to do...
I (1294) chip[SVR]: SetupQRCode: [MT:Y.K9042C00KA0648G00]
I (1297) chip[DL]: Configuring CHIPoBLE advertising (interval 25 ms, connectable)
I (1308) chip[DL]: Device already advertising, stop active advertisement and restart
I (1301) chip[SVR]: Copy/paste the below URL in a browser to see the QR Code:
I (1323) chip[SVR]: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3AY.K9042C00KA0648G00
I (1332) chip[SVR]: Manual pairing code: [34970112332]
I (31070) chip[DL]: Configuring CHIPoBLE advertising (interval 500 ms, connectable)
I (31071) chip[DL]: Device already advertising, stop active advertisement and restart
```

Figura 37. Resultado del proceso para la creación de código manual.

Ahora, explicando cómo se genera el Qr Matter, se puede ver en la **Figura 38** la función que se utiliza para configurar el proveedor DAC (Device Attestation Credentials) personalizado. DAC es una parte crítica del proceso de incorporación de Matter y se utiliza para garantizar que los dispositivos sean autenticados de manera segura durante el proceso de incorporación. En este caso, se utiliza un proveedor de credenciales de ejemplo proporcionado por Matter (GetExampleDACProvider).

```
esp_matter::set_custom_dac_provider(chip::Credentials::Examples::GetExampleDACProvider());
```

Figura 38. Codificación para la configuración del proveedor DAC

Luego, como se puede ver en la **Figura 39**, esta función inicia el dispositivo Matter en el ESP32 e inicia el nodo Matter y realiza la configuración necesaria para que el dispositivo funcione en la red Matter.

```
esp_matter::start(on_device_event);
```

Figura 39. Codificación para el inicio del nodo Matter

En la **Figura 40**, se identifica la función que se utiliza para imprimir los códigos necesarios para configurar el dispositivo Matter. En este caso, se utiliza `chip::RendezvousInformationFlag::kBLE` para especificar que se generen códigos QR para configuración a través de Bluetooth Low Energy (BLE).

```
PrintOnboardingCodes(chip::RendezvousInformationFlags(chip::RendezvousInformationFlag::kBLE));
```

Figura 40. Codificación para generar Código Qr

La generación del código QR único se lleva a cabo en el fondo como parte de la implementación interna de Matter. El código QR contiene información necesaria para que los asistentes inteligentes

puedan descubrir y configurar el dispositivo en la red Matter, como se puede ver en la **Figura 41**:

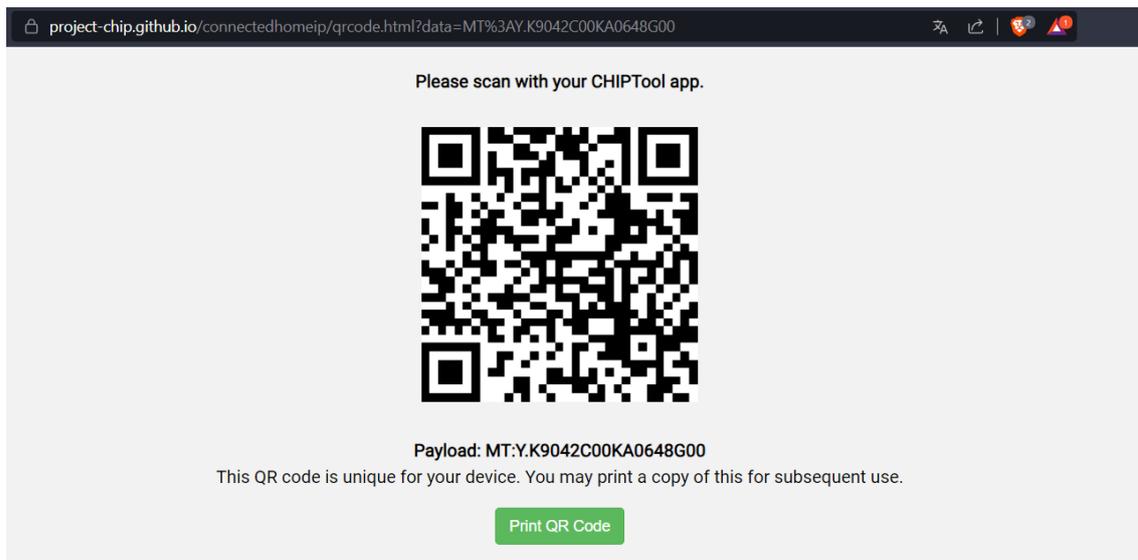


Figura 41. Resultado del proceso para la creación de código Qr

Programación del ESP32 Border Router como Gateway para el protocolo Thread con soporte Matter

Para implementar este desarrollo, se eligió instalar Ubuntu en una partición específica del disco de la laptop debido a su naturaleza de código abierto, activa comunidad y compatibilidad con herramientas de desarrollo para IoT. La instalación se realizó de manera guiada, asignando un espacio adecuado en la partición para el sistema operativo y futuras expansiones. La interfaz gráfica de Ubuntu facilitó la configuración de opciones de red y seguridad. Con Ubuntu instalado, se dispone de un entorno de trabajo estable y flexible, brindando la plataforma necesaria para el desarrollo eficiente de soluciones IoT basadas en el estándar Matter y Thread.

Ahora en la consola se procede a realizar la clonación del repositorio espressif/esp-idf, utilizando el comando (`git clone -b v5.1.1 --recursive https://github.com/espressif/esp-idf.git`). Esta acción específica es crucial para adquirir la versión precisa del código fuente y sus dependencias, estableciendo así una sólida base de desarrollo para el Gateway Thread en el contexto de Matter, así como se puede ver en la **Figura 42**

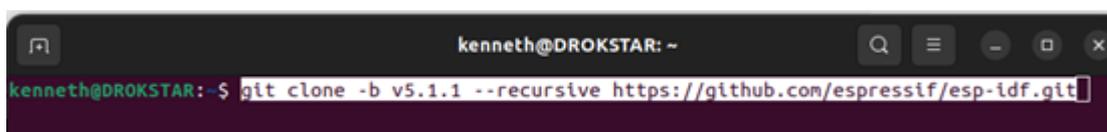
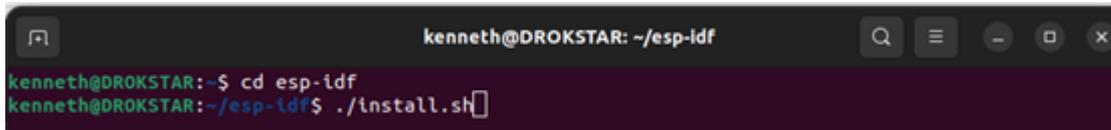


Figura 42. Codificación del repositorio espressif

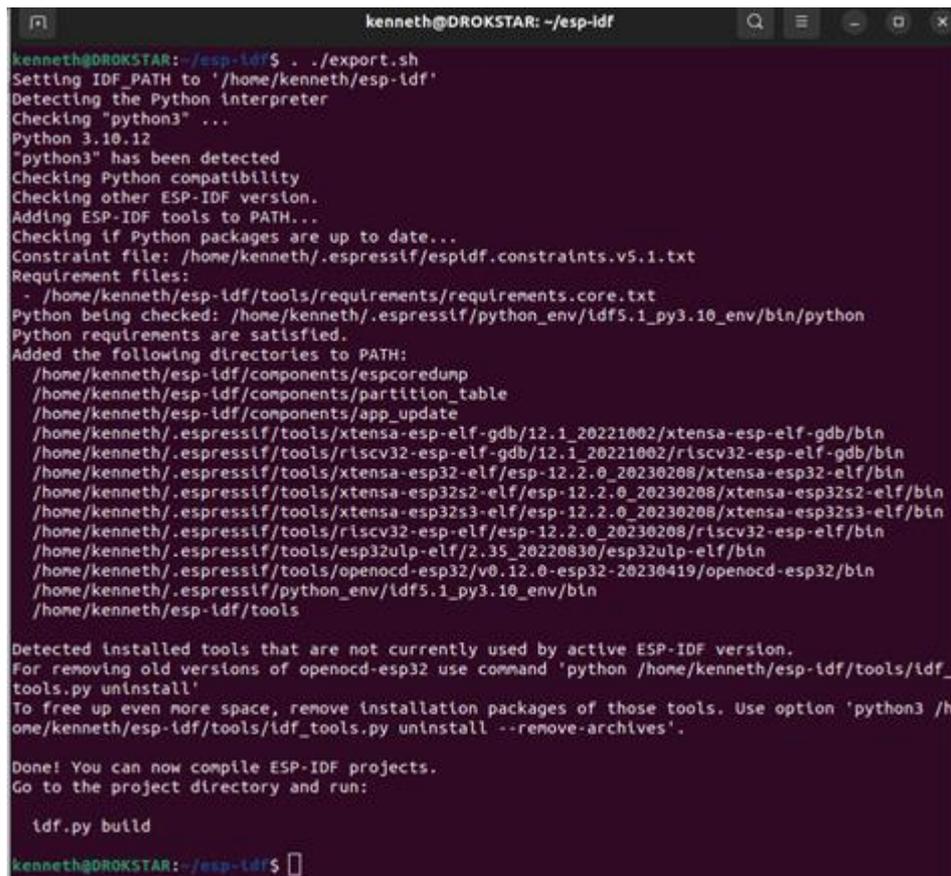
En el siguiente paso, se dirigió a la carpeta esp-idf y se procede a la instalación con el comando `./install.sh`. Este paso es crucial para configurar el entorno de desarrollo en el sistema Ubuntu para la implementación del Gateway Thread a Matter, así como se ve en la **Figura 43**:



```
kenneth@DROKSTAR: ~/esp-idf
kenneth@DROKSTAR:~$ cd esp-idf
kenneth@DROKSTAR:~/esp-idf$ ./install.sh
```

Figura 43. Codificación para instalar la librería esp-idf

Luego de ejecutar el script con `./export.sh`, la consola muestra la configuración exitosa del entorno. Se detecta la ruta de ESP-IDF, se verifica la compatibilidad de Python, y se añaden las herramientas de ESP-IDF al PATH, como se puede ver en la **Figura 44**. Además, se confirman las versiones de herramientas instaladas y se sugiere eliminar versiones antiguas no utilizadas. Ahora, el entorno está listo para compilar proyectos ESP-IDF. Solo faltaría ir al proyecto y ejecutarlo `idf.py build`, pero hay cosas que se deben modificar antes:



```
kenneth@DROKSTAR:~/esp-idf$ ./export.sh
Setting IDF_PATH to '/home/kenneth/esp-idf'
Detecting the Python interpreter
Checking "python3" ...
Python 3.10.12
"python3" has been detected
Checking Python compatibility
Checking other ESP-IDF version.
Adding ESP-IDF tools to PATH...
Checking if Python packages are up to date...
Constraint file: /home/kenneth/.espressif/esp-idf.constraints.v5.1.txt
Requirement files:
- /home/kenneth/esp-idf/tools/requirements/requirements.core.txt
Python being checked: /home/kenneth/.espressif/python_env/idf5.1_py3.10_env/bin/python
Python requirements are satisfied.
Added the following directories to PATH:
/home/kenneth/esp-idf/components/espcoredump
/home/kenneth/esp-idf/components/partition_table
/home/kenneth/esp-idf/components/app_update
/home/kenneth/.espressif/tools/xtensa-esp-elf-gdb/12.1_20221002/xtensa-esp-elf-gdb/bin
/home/kenneth/.espressif/tools/riscv32-esp-elf-gdb/12.1_20221002/riscv32-esp-elf-gdb/bin
/home/kenneth/.espressif/tools/xtensa-esp32-elf/esp-12.2.0_20230208/xtensa-esp32-elf/bin
/home/kenneth/.espressif/tools/xtensa-esp32s2-elf/esp-12.2.0_20230208/xtensa-esp32s2-elf/bin
/home/kenneth/.espressif/tools/xtensa-esp32s3-elf/esp-12.2.0_20230208/xtensa-esp32s3-elf/bin
/home/kenneth/.espressif/tools/riscv32-esp-elf/esp-12.2.0_20230208/riscv32-esp-elf/bin
/home/kenneth/.espressif/tools/esp32ulp-elf/2.35_20220830/esp32ulp-elf/bin
/home/kenneth/.espressif/tools/openocd-esp32/v0.12.0-esp32-20230419/openocd-esp32/bin
/home/kenneth/.espressif/python_env/idf5.1_py3.10_env/bin
/home/kenneth/esp-idf/tools

Detected installed tools that are not currently used by active ESP-IDF version.
For removing old versions of openocd-esp32 use command 'python /home/kenneth/esp-idf/tools/idf_tools.py uninstall'
To free up even more space, remove installation packages of those tools. Use option 'python3 /home/kenneth/esp-idf/tools/idf_tools.py uninstall --remove-archives'.

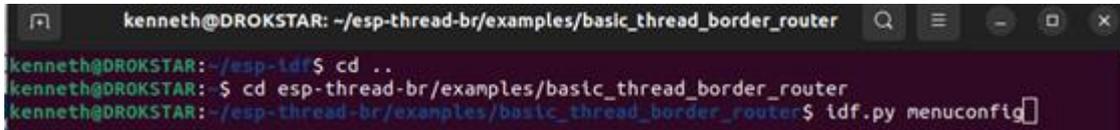
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build
kenneth@DROKSTAR:~/esp-idf$
```

Figura 44. Codificación para implementar el proyecto de Gateway

Después de cambiar al directorio que alberga el proyecto del enrutador de borde Thread, se ejecuta el menú de configuración de este con el siguiente comando en la consola: `(cd ~/esp-thread-`

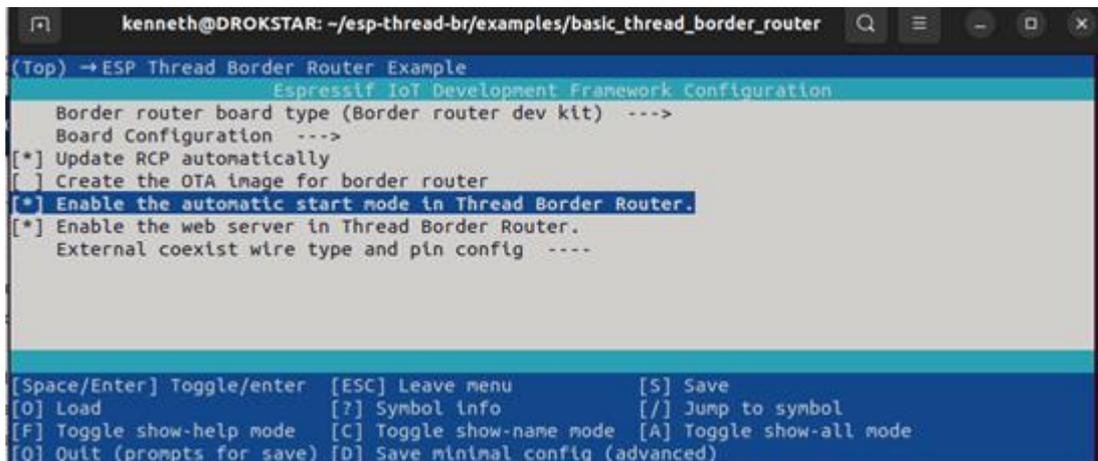
br/examples/basic_thread_border_router - idf.py menuconfig), como se puede ver en la **Figura 45**. Esto permite acceder y ajustar las configuraciones específicas del proyecto, facilitando la personalización de parámetros relevantes para el enrutador de borde Thread en el entorno ESP-IDF:



```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router
kenneth@DROKSTAR: ~/esp-idf$ cd ..
kenneth@DROKSTAR: $ cd esp-thread-br/examples/basic_thread_border_router
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router$ idf.py menuconfig
```

Figura 45. Codificación para iniciar y proyecto y sus configuraciones

Ahora, una vez que se inicializa, la interfaz de configuración del proyecto, ubicada en el directorio del enrutador de borde Thread, se destaca la opción para iniciar automáticamente el Border Router como se ve en la **Figura 46**. En la consola, se observa la configuración donde se marca la casilla de "Enable the automatic start mode in Thread Border Router". Esta selección activa la función de inicio automático del enrutador de borde Thread, proporcionando una visualización clara de las opciones disponibles y permitiendo la personalización de la configuración para adaptarse a las necesidades específicas del proyecto:



```
(Top) → ESP Thread Border Router Example
Espressif IoT Development Framework Configuration
Border router board type (Border router dev kit) --->
Board Configuration --->
[*] Update RCP automatically
[] Create the OTA image for border router
[*] Enable the automatic start mode in Thread Border Router.
[*] Enable the web server in Thread Border Router.
External coexist wire type and pin config ----

[Space/Enter] Toggle/enter  [ESC] Leave menu      [S] Save
[O] Load                    [?] Symbol info      [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Qult (prompts for save) [D] Save minimal config (advanced)
```

Figura 46. Configuración de Gateway

En el proceso de configuración, se accede a la sección de configuración de la conexión, donde se establecen los parámetros para la red Wifi doméstica en el Border Router, como se ve en la **Figura 47**. En la consola, se visualiza la configuración con la opción de proporcionar el SSID y la contraseña directamente o recibirlos desde la entrada estándar. Se destaca la entrada para el SSID (JOSE1980) y la contraseña. Estas configuraciones permiten adaptar el Border Router a la red Wifi específica de la casa, ya en el trabajo [59], muestra categorizando los Gateway en IoT en dos grupos principales: Gateway básicos y Smart Gateway. Luego, los Smart Gateway se subdividen en tres categorías: Gateway pasivos, semiautomatizados y totalmente automatizados, asegurando

una conexión adecuada para el proyecto en cuestión:

```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router
(Top) → Example Connection Configuration
Espressif IoT Development Framework Configuration
[*] connect using WiFi interface
[ ] Get ssid and password from stdin
[*] Provide wifi connect commands
(TEL DAY) WiFi SSID
(JOSE1980) WiFi Password
(6) Maximum retry
WiFi Scan Method (All Channel) --->
WiFi Scan threshold --->
WiFi Connect AP Sort Method (Signal) --->
[ ] connect using Ethernet interface
[*] Obtain IPv6 address
Preferred IPv6 Type (Local Link Address) --->

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Figura 47. Configuración de Red del Gateway

En la consola, configuramos el nombre y la clave de la red Thread, junto con otros parámetros clave del conjunto operativo de OpenThread. Destacamos la entrada del nombre de la red Thread como "matter" y otros detalles como el canal, identificador PAN, y claves necesarias, así como se muestra en la **Figura 48**. Finalmente, guardamos estos ajustes pulsando "S". Este proceso asegura la configuración adecuada de la red Thread del Border Router para cumplir con los requisitos del proyecto Matter:

```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router
(Top) → Component config → OpenThread → Thread Operational Dataset
Espressif IoT Development Framework Configuration
(matter) OpenThread network name
(15) OpenThread network channel
(0x1234) OpenThread network pan id
(dead00beef00cafe) OpenThread extended pan id
(00112233445566778899aabbccddeeff) OpenThread network key
(104810e2315100afd6bc9215a6bfac53) OpenThread pre-shared commissioner key

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Figura 48. Configuración de la clave de la Red thread

Para cargar los ajustes en nuestra placa ESP32, utilizamos dos comandos específicos, **Figura 49**. El primero envía el firmware creado al ESP32:

```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router
kenneth@DROKSTAR:~/esp-thread-br/examples/basic_thread_border_router$ idf.py -p /dev/ttyACM0 flash
```

Figura 49. Codificación de la instalación del Firmware

El segundo comando abre un modo monitor para verificar la conexión a nuestra red y asegurarse de que todo funcione correctamente, **Figura 50**:

```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thread_border_router
kenneth@DROKSTAR:~/esp-thread-br/examples/basic_thread_border_router$ idf.py -p /dev/ttyACM0 monitor
```

Figura 50. Verificación de la conexión de la red

Estos pasos son cruciales para implementar los cambios y garantizar el correcto funcionamiento del Border Router en la red Thread según nuestras configuraciones.

Una vez completados los pasos anteriores con éxito, utilizamos el comando `state` para verificar la conexión del Thread Border Router y asegurarnos de que esté operando correctamente. Este comando proporciona información sobre el estado actual de la conexión y confirma si la implementación ha sido exitosa.

Como se puede ver en la **Figura 51**, al ejecutar el comando `state`, se observa la salida del Thread Border Router indicando que la conexión está establecida con éxito. Los mensajes `I(6476)` a `I(13266)` proporcionan detalles sobre la configuración, incluyendo la asignación de identificadores de router, el estado Mle y la transición a líder, confirmando así que el Thread Border Router está operativo y conectado a la red Thread de manera exitosa.

```
kenneth@DROKSTAR: ~/esp-thread-br/examples/basic_thre...
I (6476) OT_STATE: netif up
I (6476) OPENTHREAD: NAT64 ready
I (6616) wifi:<ba-add>idx:1 (lfx:0, 14:eb:b6:14:b6:49), tid:6, ssn:12, winSize:6
4
I(6626) OPENTHREAD:[N] Mle-----: Attach attempt 1, AnyPartition reattachin
g with Active Dataset
> state
detached
Done
> I(13246) OPENTHREAD:[N] RouterTable---: Allocate router id 33
I(13246) OPENTHREAD:[N] Mle-----: RLOC16 fffe -> 8400
I(13256) OPENTHREAD:[N] Mle-----: Role detached -> leader
I(13256) OPENTHREAD:[N] Mle-----: Partition ID 0x5b379ae8
I (13266) OPENTHREAD: Platform UDP bound to p1 (15556) OPENTHREAD: Platform UDP
bound to port 53535
> stateI(25546) OPENTHREAD: Received PIO
I (27516) example_connect: Got IPv6 event: Interface "example_netif_sta" address
: f0de:ad00:beef:cafe:4a27:e2ff:fe14:41d8, type: ESP_IP6_ADDR_IS_UNIQUE_LOCAL
> state
Leader
Done
>
```

Figura 51. Resultado en consola de la instalación del Firmware del Gateway

2.3.4. FASE 4: Pruebas

Al abordar la fase de pruebas, se puede empezar con la ejecución de la codificación que controla el encendido y apagado del LED en nuestro ESP32-S3, el cual opera bajo el protocolo Matter. Este proceso generará un código QR y, adicionalmente, un código para iniciar de forma manual en caso de que sea requerido. Tal como se puede observar en la **Figura 52**, estos códigos se presentan claramente:

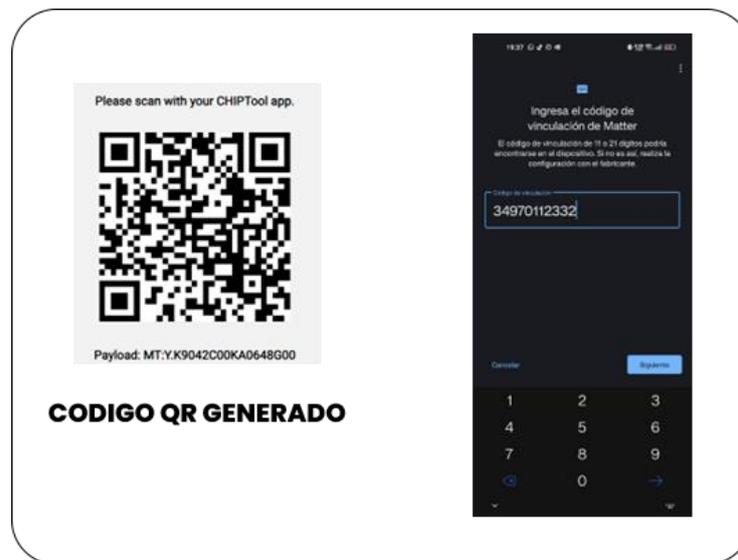


Figura 52. Vinculación de código Matter

Inicialmente, se estableció una conexión con un asistente virtual que respalda ciertas certificaciones de dispositivos bajo este protocolo. Sin embargo, durante el proceso de emparejamiento, como se puede ver en la **Figura 53**, nos encontramos con un obstáculo: un error de conexión. Este desafío, siendo esencial en la fase de desarrollo, se originó debido a que el dispositivo Matter no contaba con la certificación legal adecuada.

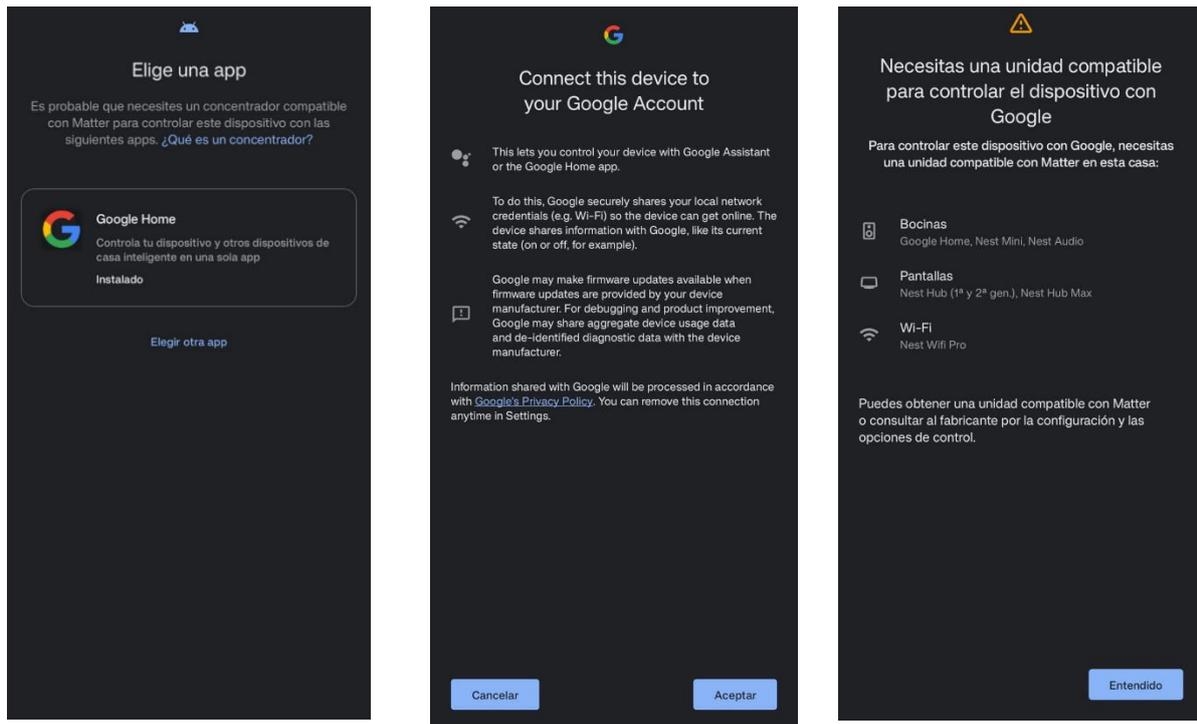


Figura 53. Emparejamiento de Dispositivo Matter

El error específico fue proporcionado por Google Home. En la **Figura 54**, se puede ver este mensaje:

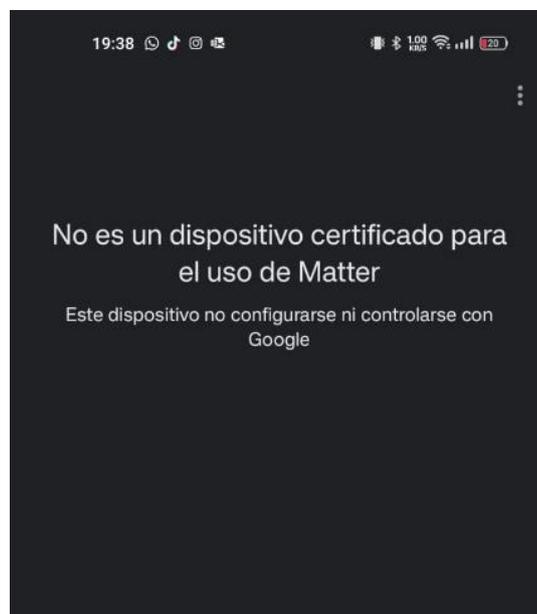


Figura 54. Error de vinculación proporcionado por Google Home

Después del inconveniente mencionado, se procedió a intentar la conexión del dispositivo con otro asistente virtual disponible, el cual demostró ser compatible sin requerir la certificación previamente mencionada. Para este propósito, se seleccionó "Alexa". Al realizar el emparejamiento, como se evidencia en la **Figura 55**, el proceso resultó exitoso:

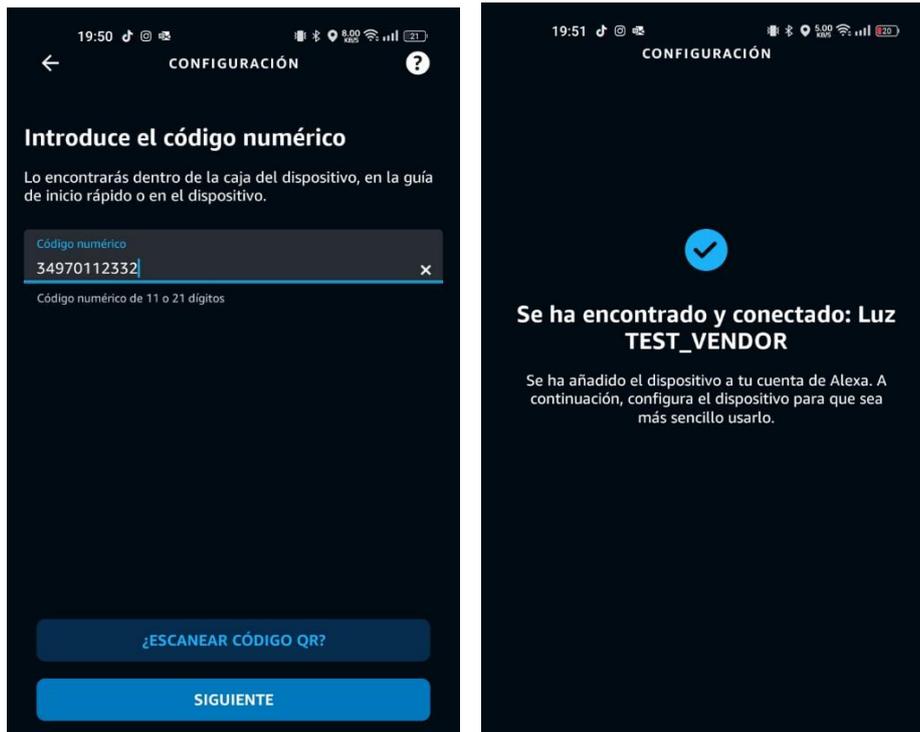


Figura 55. Emparejamiento desde Alexa app

En la **Figura 56**, se puede evidenciar ya el uso que se le puede dar a nuestro dispositivo matter:

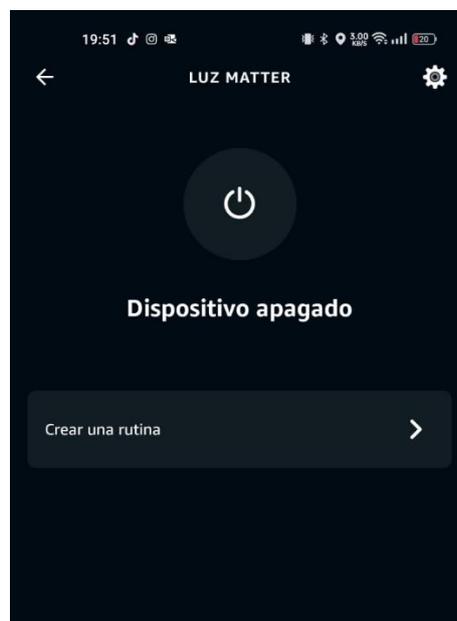


Figura 56. Entorno del dispositivo matter vinculado

Como se observa en la descripción anterior, el dispositivo se vinculó exitosamente con el asistente "Alexa" a través del protocolo Matter, ejecutando la codificación para el encendido y apagado del LED. Este éxito marca una etapa crucial en el desarrollo del sistema domótico basado en el protocolo Matter. Posteriormente, se optó por integrar el sistema de automatización de dispositivos

IoT "Home Assistant". En las siguientes secciones se evidencia cómo este sistema permitió incorporar el dispositivo Matter en su entorno operativo. Posteriormente, se accede a la dirección IP proporcionada por "Home Assistant" en la máquina virtual en la que opera, facilitando así el acceso a la interfaz de "Home Assistant", como se ve en la **Figura 57**:

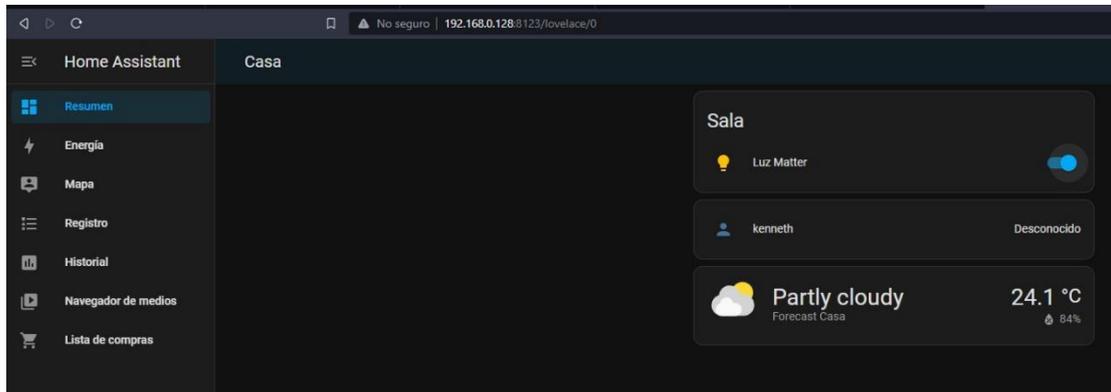


Figura 57. Home Assistant

Luego se procede a realizar el mismo paso que se realizó previamente, que trata sobre la vinculación del dispositivo Matter, que en este caso es el ESP32-S3, tal cual como se puede ver en la **Figura 58**, dando paso al correcto uso del dispositivo Matter dentro de la interfaz de Home Assistant:

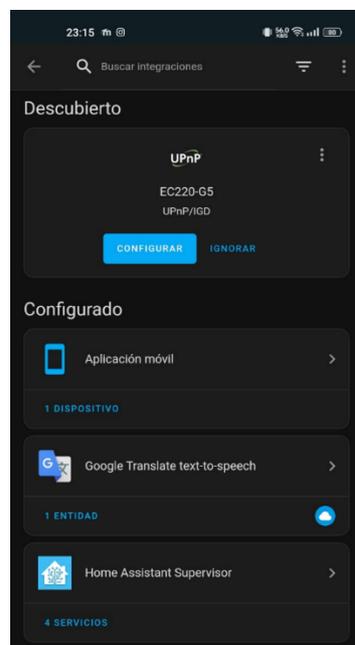


Figura 58. Entorno de la aplicación Home Assistant en el celular

En la **Figura 59**, se muestra como buscamos nuestro protocolo de comunicación que es Matter:

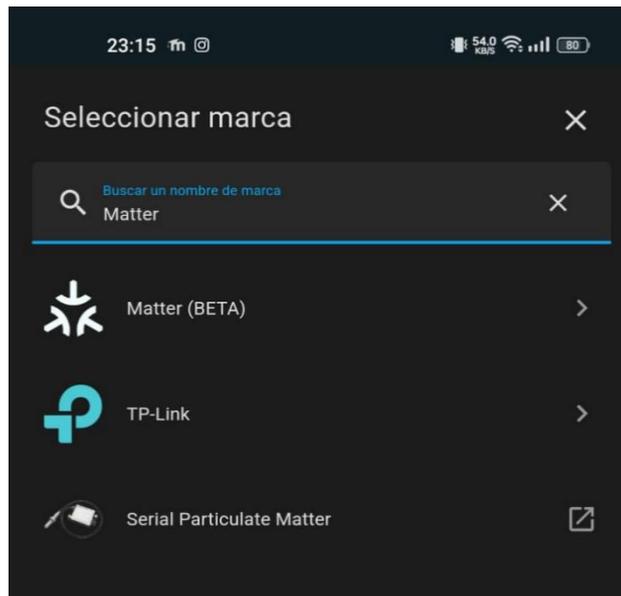


Figura 59. Búsqueda del protocolo de vinculación (Matter)

A continuación, se procede a instalar los componentes necesarios para la vinculación del dispositivo Matter, como se puede ver en la **Figura 60**:

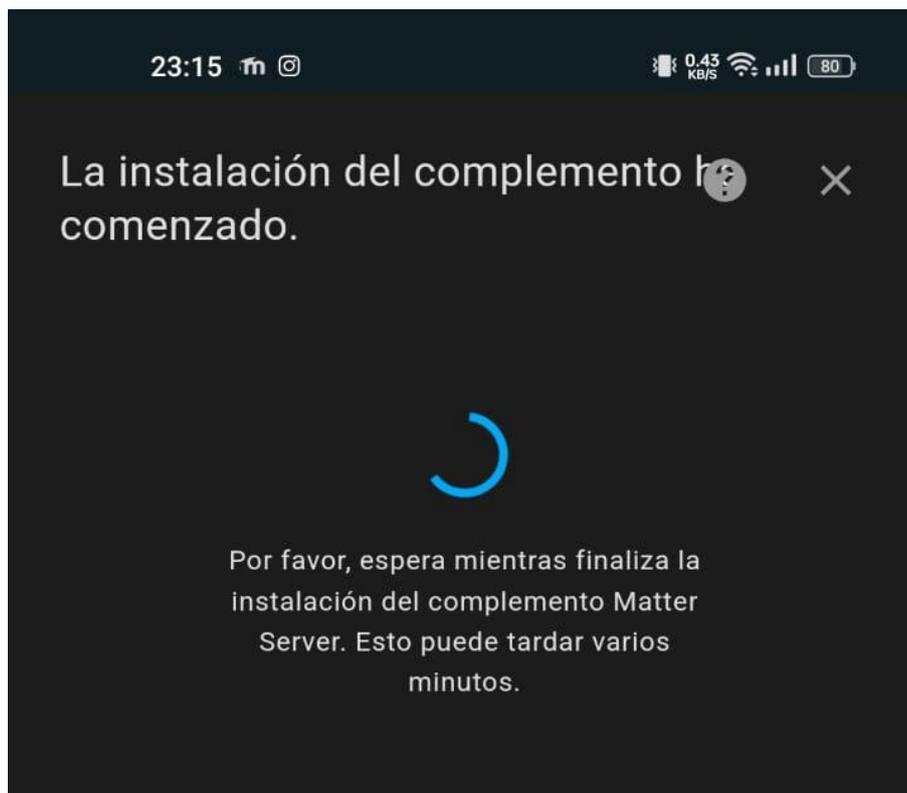


Figura 60. Instalación de servicio matter

Luego de esto se procede a colocar el código de reconocimiento de dispositivo matter y luego de una exitosa vinculación se puede ya utilizar nuestro dispositivo, esta colocación del código se puede ver en la **Figura 61**:

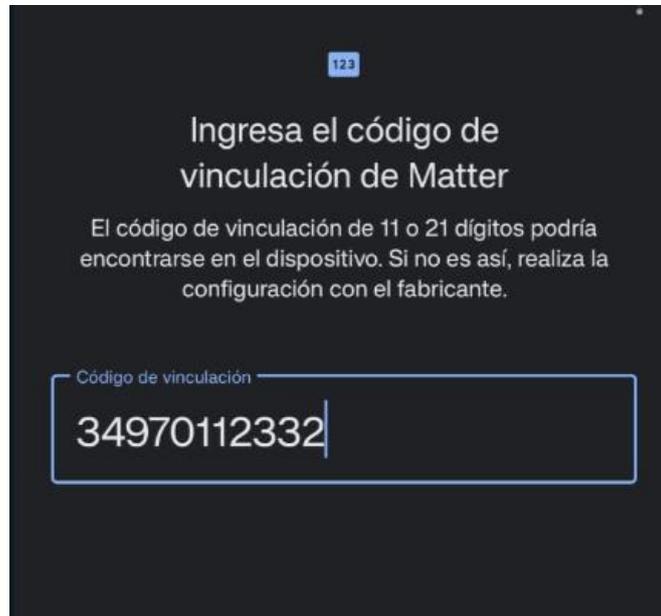


Figura 61. Interfaz del ingreso de código matter

A continuación, se ve en la **Figura 62**, que el dispositivo matter propio se ha unido de manera correcta, permitiendo así la interoperabilidad entre dos marcas por ahora Alexa, y Home Assistant:

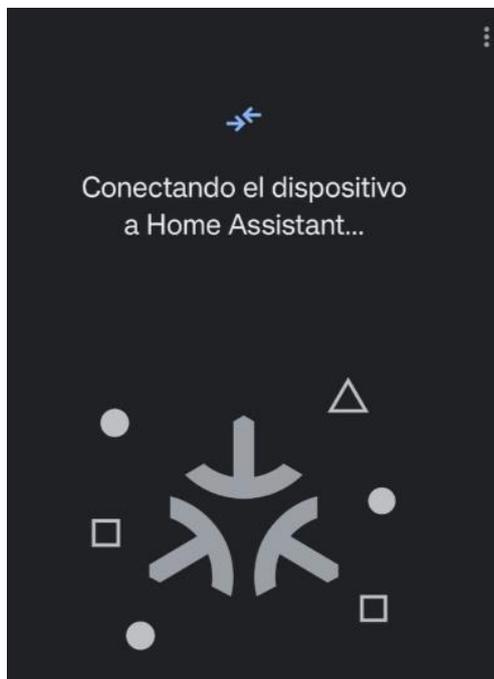


Figura 62. Conexión exitosa del dispositivo

Y luego se puede manejar nuestro dispositivo según su interfaz predeterminada que en este caso nos da Home Assistant, como se ve en la **Figura 63**:

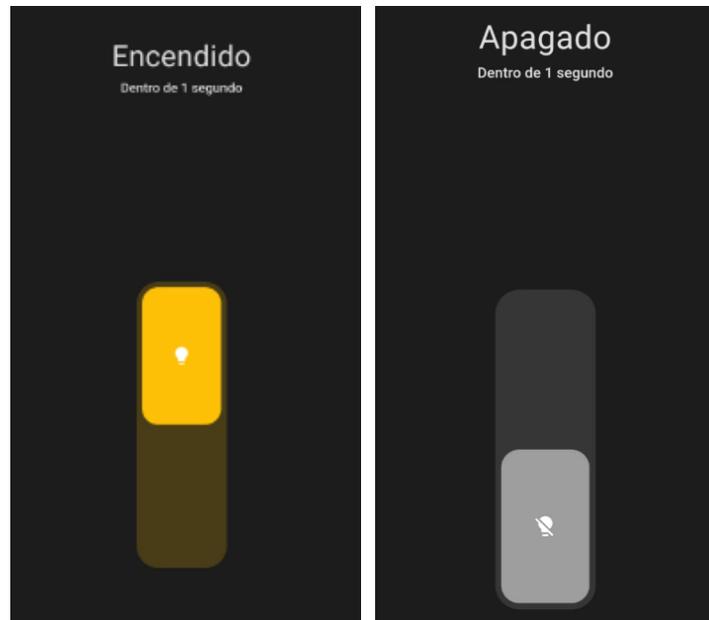


Figura 63. Interfaz de encendido y apagado del dispositivo Matter

2.4. Ejecución del prototipo

Antes de comenzar con cualquier proceso, es esencial preparar y programar el ESP32 para que funcione como un foco LED inteligente. Para ello, cargamos el firmware específico que transforma nuestro microcontrolador en un dispositivo domótico matter. Asegurando de tener todas las bibliotecas necesarias y de que el código esté libre de errores para garantizar una ejecución sin problema, se procederá a ejecutarlo dando una respuesta en nuestra salida como la que se ve en la **Figura 64**:

```
Output Serial Monitor x
Message (Enter to send message to 'ESP32S3 Dev Module' on 'COM3')
I (1281) chip[DL]: WIFI_EVENT_STA_START
E (1281) esp_matter_core: Failed to open the node nvs_namespace
I (1284) chip[DL]: Done driving station state, nothing else to do...
I (1294) chip[SVR]: SetupQRCode: [MT:Y.R9042C00KA0648G00]
I (1297) chip[DL]: Configuring CHIPoBLE advertising (interval 25 ms, connectable)
I (1308) chip[DL]: Device already advertising, stop active advertisement and restart
I (1301) chip[SVR]: Copy/paste the below URL in a browser to see the QR Code:
I (1323) chip[SVR]: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3AY.R9042C00KA0648G00
I (1332) chip[SVR]: Manual pairing code: [34970112332]
I (31070) chip[DL]: Configuring CHIPoBLE advertising (interval 500 ms, connectable)
I (31071) chip[DL]: Device already advertising, stop active advertisement and restart
```

Figura 64. Resultado de la ejecución del código de foco led matter

Una vez que el ESP32 está configurado, debería generar un código QR único y una clave o código también. Este código es esencial para el proceso de emparejamiento y garantiza una conexión segura entre el dispositivo y cualquier plataforma de control domótico. Es recomendable verificar

la claridad y legibilidad del código QR como se ve en la **Figura 65** para evitar problemas durante el escaneo:

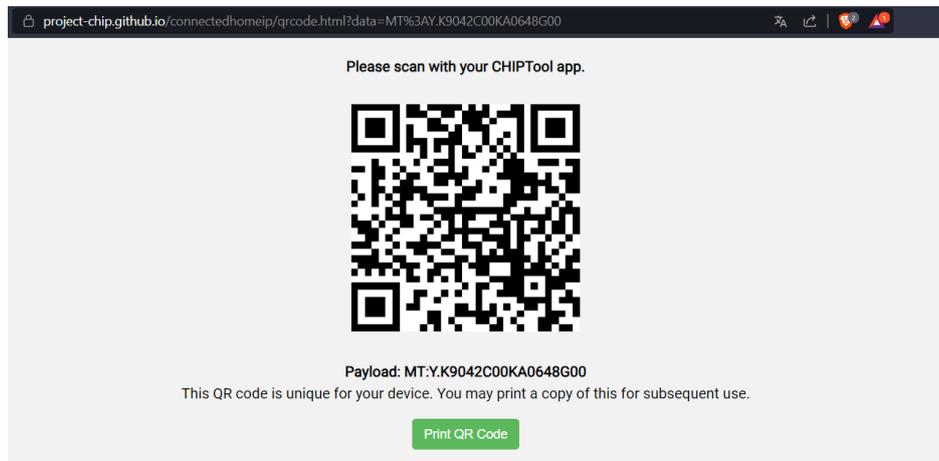


Figura 65. Código Qr de vinculación de dispositivo matter

Con la aplicación Alexa en el dispositivo móvil, se procede a buscar la sección de dispositivos en la opción para añadir un nuevo gadget. Se procede a asegurar de tener la última versión de la aplicación y de estar conectado a la misma red Wifi que tú ESP32 para facilitar el proceso de emparejamiento del dispositivo matter, como se ve en la **Figura 66**:

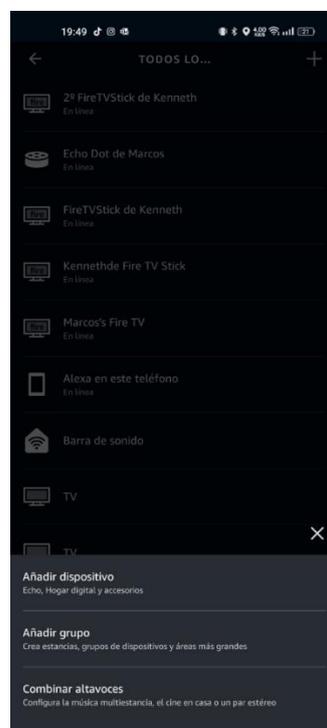


Figura 66. Proceso de emparejamiento de Alexa app

En la lista de dispositivos disponibles, buscamos y seleccionamos la opción que dice "Otro dispositivo". Esto abrirá una nueva pantalla donde se podrá ver diferentes protocolos y marcas. Se localiza y se selecciona el logo de Matter, que es el protocolo que nuestro ESP32 está utilizando, como se ve en la **Figura 67**:

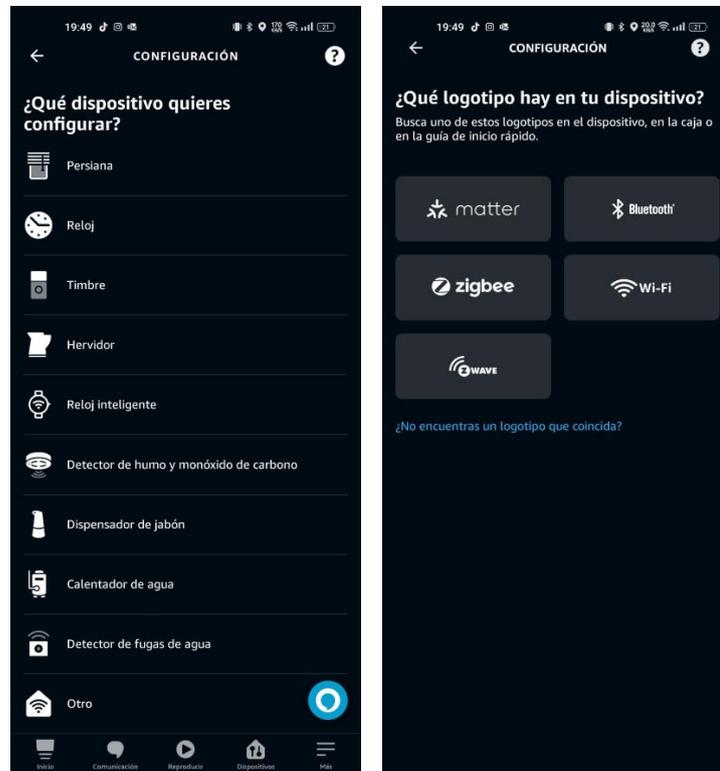


Figura 67. Selección del dispositivo propio matter

Con el código QR que generó nuestro ESP32, se escanea o ingresa manualmente el código de vinculación en la aplicación. Una vez hecho esto, la aplicación comenzará el proceso de emparejamiento. Durante este tiempo, es crucial no interrumpir la conexión ni cerrar la aplicación, tal cual como se ve en la **Figura 68**. Una vez que el dispositivo esté enlazado, se recibirá una notificación de confirmación, como se puede evidenciar en la **Figura 69**:

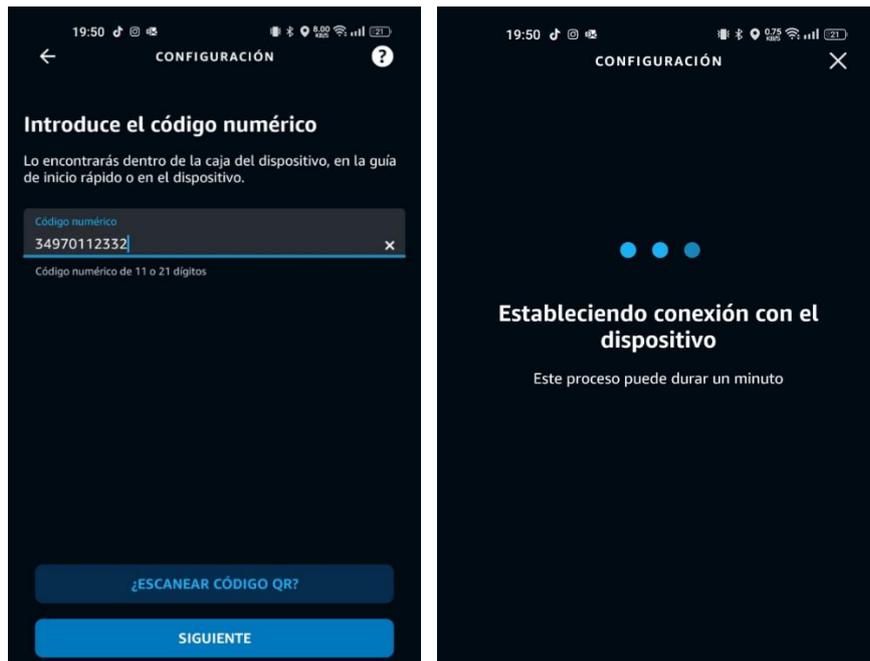


Figura 68. Proceso de Vinculación de Dispositivo con código

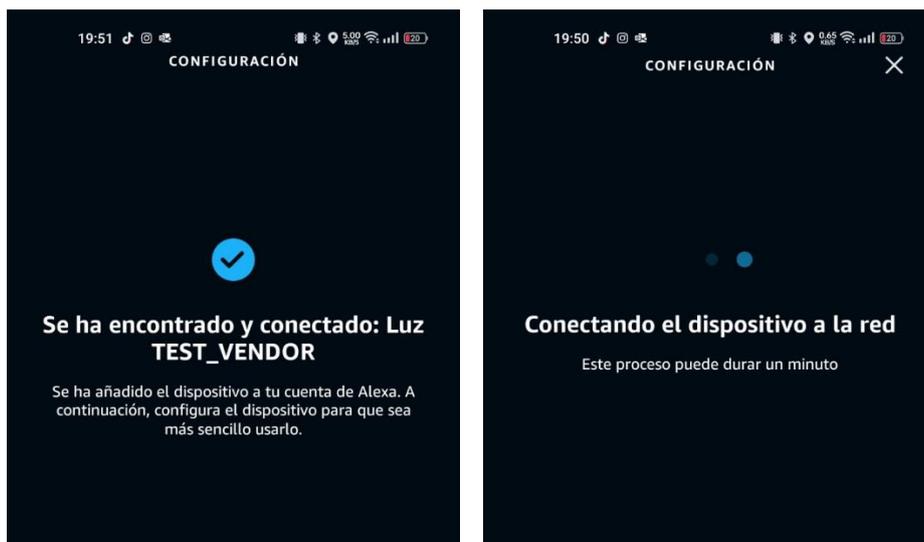


Figura 69. Vinculación exitosa de dispositivos propios

Tras haber emparejado exitosamente el ESP32 con la aplicación, es el momento de probar su funcionamiento. Se encenderá y apagará el foco LED a través de la app para asegurarte de que la comunicación es fluida y que el protocolo Matter pudo permitir esta comunicación, así como se ve en la **Figura 70**. Si todo funciona como se espera, se tendrá un foco LED inteligente integrado al sistema domótico:

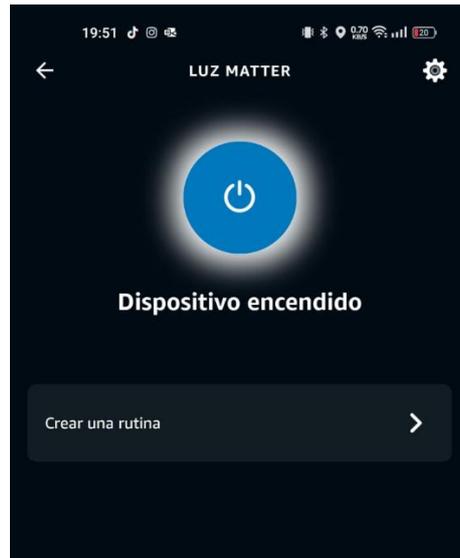


Figura 70. Encendido de led Matter

En la **Figura 71** se puede observar el encendido que este género al pulsar el botón desde la aplicación Alexa:

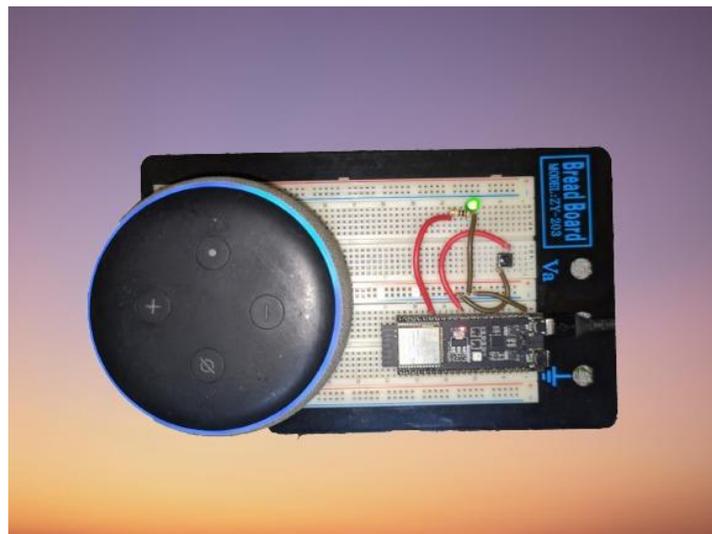
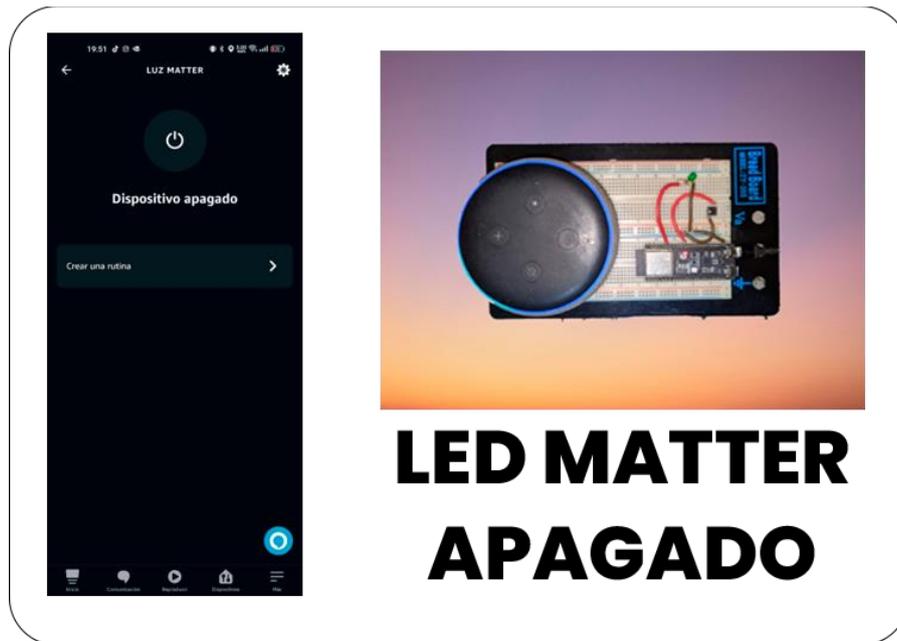


Figura 71. Prototipo con ESP32-S3 desarrollado y funcionando - Primera Parte

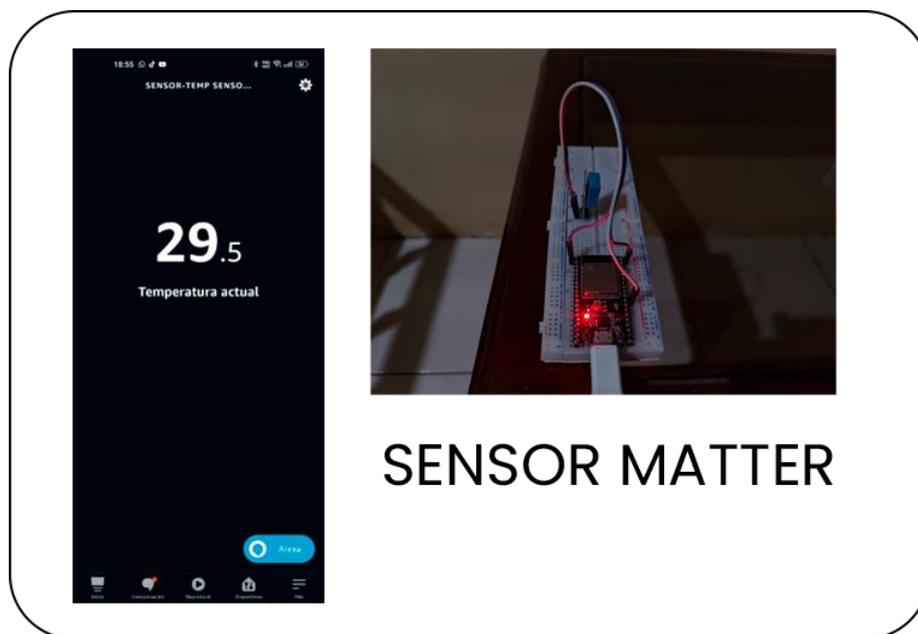
A continuación, se puede observar, en cambio, que al apagar este dispositivo matter propio ejecuto de manera correcta, como se ve en la **Figura 72**, todo se podrá evidenciar personalmente en la sustentación del proyecto mismo:



LED MATTER APAGADO

Figura 72. Prototipo con ESP32-S3 desarrollado y funcionando - Segunda Parte

Ahora para el otro dispositivo matter es prácticamente lo mismo, en este caso nuestro sensor, en el cual nos enseñara nuestra temperatura ambiente, para esto en la **Figura 73** se puede ver establecido en nuestra aplicación de Alexa:



SENSOR MATTER

Figura 73. Figura59. Prototipo con ESP32-C3 como Sensor

A continuación, en la **Figura 74** se ve nuestro sistema domótico armado, conectado y funcionando para la primera entrega y en la **Figura 75** se puede encontrar el sistema ya terminado, en el **Anexo 1**, se puede encontrar los dispositivos esp32 terminados de cada uno de nuestros dispositivos con protocolo matter, thread, nuestro Gateway ESP32 Border Router y el dispositivo comercial:

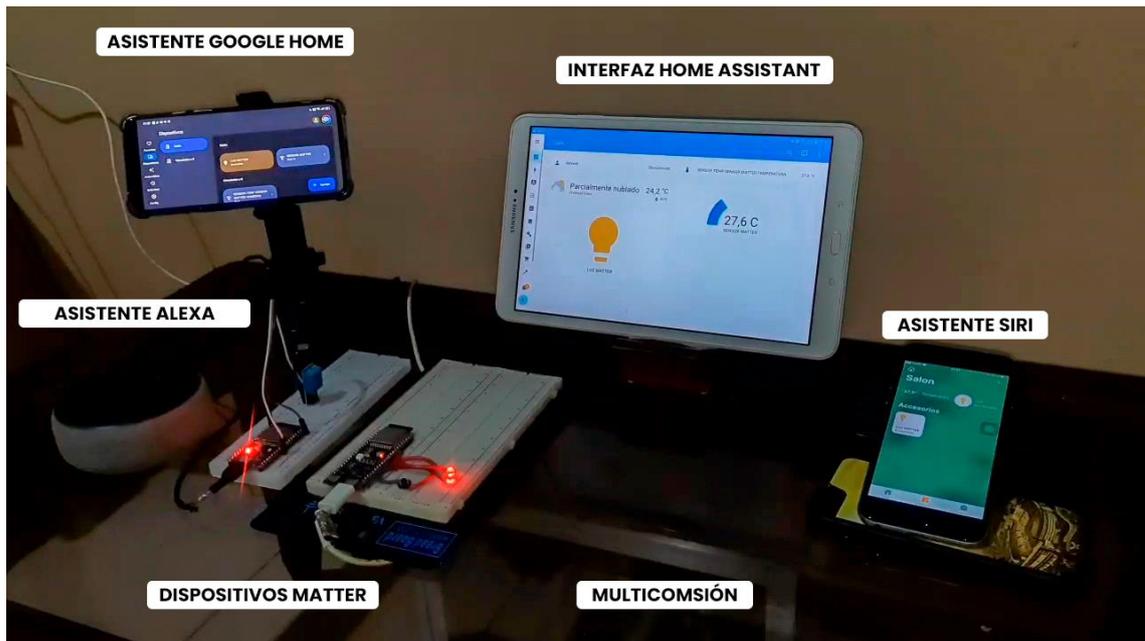


Figura 74. Sistema domótico con los dispositivos personalizados

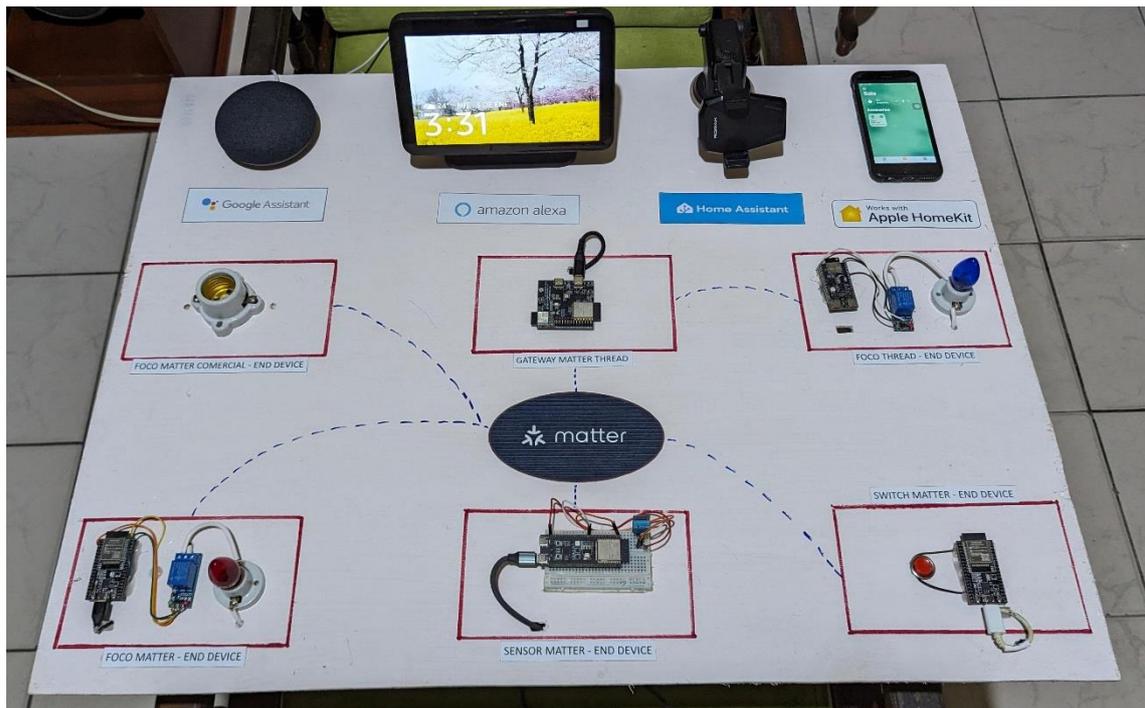


Figura 75. Sistema domótico con los dispositivos personalizados finalizados

Ahora se procedió a realizar la conexión de nuestro Gateway Thread con soporte Matter, con el dispositivo Thread que previamente ya se tenía hecho, para conectar un dispositivo Thread al Gateway, simplemente escaneamos el código QR del dispositivo Thread Matter con el asistente de voz. Posteriormente, se espera que el dispositivo se conecte automáticamente a la red Thread configurada por el Thread Border Router. Este método simplificado agiliza la integración de dispositivos Thread mediante un proceso guiado por el asistente, como se ve en la **Figura 76**:

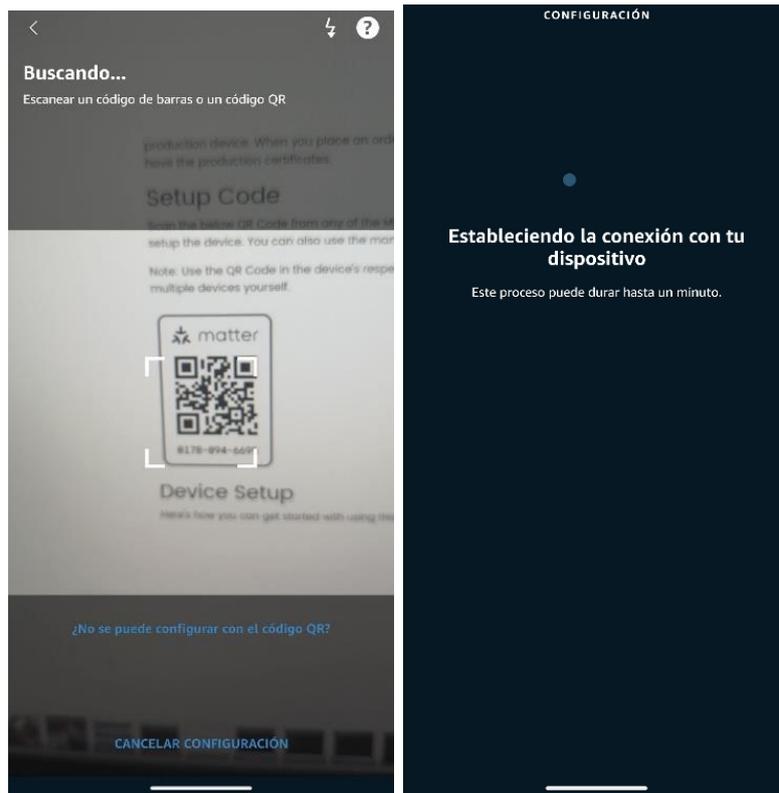


Figura 76. Conexión de dispositivo Thread con Asistente Virtual

Se tiene en cuenta que el dispositivo Gateway ya este encendido para su posterior conexión en este proceso, luego en este paso, elegimos nuestra red Thread y luego introducimos la clave de cifrado correspondiente que ya habíamos configurado en nuestro proceso de programación. Este proceso es esencial para asegurar una conexión segura y exitosa del dispositivo al Thread Border Router a través de la red Thread, como se ve en la **Figura 77**:

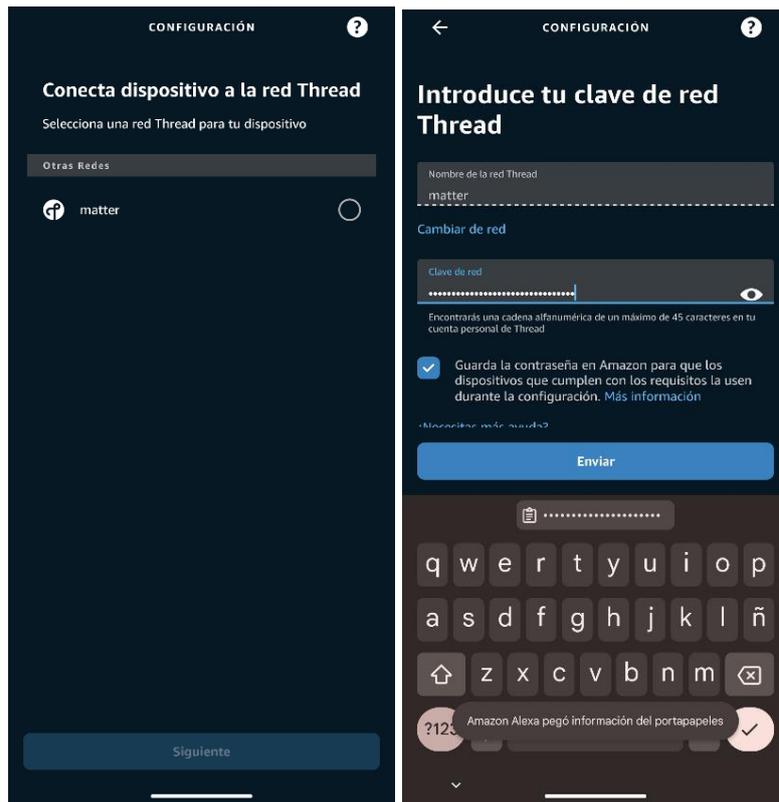


Figura 77. Conexión con el Gateway Thread con soporte Matter

En esta etapa, solo resta esperar a que el dispositivo se conecte automáticamente a la red Thread Matter. Este proceso puede llevar un momento, asegúrate, de permitir que el dispositivo complete la conexión de manera exitosa. Una vez conectado, el dispositivo debería formar parte de la red Thread gestionada por el Thread Border Router, como se ve en la **Figura 78**:

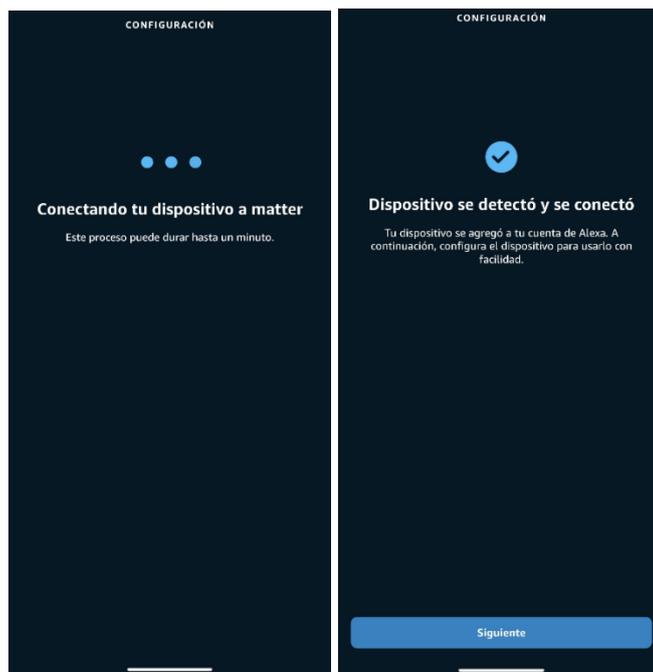


Figura 78. Conexión Exitosa del dispositivo Thread con el Gateway

Después de agregar el dispositivo a nuestro asistente, se puede verificar su funcionamiento. Esto implica confirmar que el dispositivo Thread Matter se ha integrado correctamente en la red Thread y que responde adecuadamente a las interacciones y comandos dentro del entorno del asistente, así como se ve en la **Figura 79**:



Figura 79. Prueba exitosa del uso del dispositivo Thread

CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

3.1.1. Objetivo

Evaluar el rendimiento, tiempo de respuesta y control por voz del prototipo de sistema domótico basado en el protocolo Matter, en dos diferentes escenarios, en donde se tenga un internet estable y en el otro no.

Específicamente, se evaluará lo siguiente:

- **Rendimiento:**
 - Capacidad de manejar un número determinado de dispositivos domóticos sin afectar el rendimiento.
 - Capacidad de responder a las solicitudes de los usuarios de forma rápida y eficiente.
 - Capacidad de responder de un dispositivo por medio del Gateway Thread con soporte matter.
- **Tiempo de respuesta:**
 - Tiempo de respuesta para encender/apagar un dispositivo domótico.
 - Tiempo de respuesta entre un dispositivo comercial y el personalizado.
 - Tiempo de respuesta de un dispositivo domótico Thread por medio del Gateway con soporte matter.
- **Control por voz:**
 - Capacidad de seguir instrucciones básicas del asistente de voz para controlar los dispositivos domóticos con soporte Matter.

3.1.2. Cronograma

En la **Tabla 12** se puede ver el cronograma de plan de evaluación que se llevara a cabo.

Tabla 12. Cronograma del plan de Evaluación

Actividades	Segundo Hemisemestre														
	Semana 11					Semana 12					Semana 13				
	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V
Diseño de las pruebas que se realizaran, teniendo en cuenta los objetivos.															
Montaje del prototipo, teniendo todos los dispositivos ya configurados y listos para llevar a cabo la evaluación, en diferentes escenarios.															
Realizar las pruebas diseñadas, recolectando datos de estos.															
Pruebas de Interoperabilidad y Compatibilidad															
Análisis de Datos y Preparación de Resultados Finales															
Presentación Final y Documentación															

3.1.3. Métodos de evaluación

Para evaluar los requisitos mencionados anteriormente, se utilizarán los siguientes métodos:

- **Pruebas funcionales:** Se realizarán pruebas funcionales para verificar que el prototipo puede realizar las tareas esperadas.

A continuación, para la prueba de funcionalidad se debe tener en cuenta que esta etapa es muy importante, dado que posibilita verificar si se ajusta a los propósitos y condiciones definidos durante la fase de diseño. En relación con este modelo de sistema de automatización del hogar con tecnología Matter, resulta fundamental llevar a cabo esta evaluación con el fin de garantizar que los usuarios puedan operar el sistema de manera agradable y que cumpla los objetivos establecidos.

Para realizar la evaluación de la funcionalidad he realizado los siguientes pasos:

Establecer los criterios de evaluación: Con base en los objetivos y requisitos establecidos en la etapa de diseño, en la **Tabla 13** se puede ver los criterios a evaluar:

Tabla 13. Criterios de evaluación

Criterio de evaluación	Pregunta de Evaluación	Descripción
Compatibilidad	¿Se ha implementado de manera efectiva el protocolo Matter para permitir la comunicación entre dispositivos domóticos diversos?	Verificar la interoperabilidad con dispositivos comerciales y personalizados.
Usabilidad	¿La interfaz de control desarrollada es intuitiva y amigable para que los usuarios puedan controlar y monitorear sus dispositivos eficientemente dentro de Home Assistant?	Evaluar la experiencia del usuario en términos de usabilidad.
Optimización	¿Cómo se mide y mejora continuamente la eficiencia y velocidad de respuesta del sistema para proporcionar una experiencia de usuario fluida?	Evaluar la optimización del rendimiento del sistema.
Diagnóstico y Resolución	¿Existen herramientas incorporadas que facilitan la identificación y resolución de problemas relacionados con la conectividad o el funcionamiento de los dispositivos?	Verificar la presencia de capacidades de diagnóstico en el sistema.

Establecimiento de Pruebas de aceptación: Definiendo los criterios de aceptación del sistema domótico, preparando el entorno de prueba y recogiendo los resultados, en la **Tabla 14** se puede ver los establecimientos de pruebas de aceptación: n/a: No aplicable.  : Cumple.  : Lo ha cumplido, pero no en la primera prueba.  : No lo ha cumplido/ ha habido algún impedimento.

Tabla 14. Pruebas de Aceptación

Criterios de Aceptación	Preguntas de Verificación	Sí	Si/ retraso	No
Funcionalidad	¿Se han probado todos los dispositivos del sistema domótico y están funcionando correctamente?			
Accesibilidad	¿La interfaz gestual dispone de iconos amplios?			
Intuitividad	¿Es fácil y sencillo el uso del prototipo, incluso para aquellos que no tienen conocimientos técnicos del protocolo matter?			
Fiabilidad	¿El prototipo funciona correctamente y sin interrupciones durante un período prolongado de tiempo?			

Criterios de Aceptación	Preguntas de Verificación	Sí	Si/ retraso	No
Seguridad	¿Se han implementado medidas de seguridad, como cifrado y autenticación, de acuerdo con los estándares del protocolo Matter para garantizar la seguridad de las comunicaciones?			
Interoperabilidad	¿Los dispositivos Matter se comunican eficientemente entre sí y con otros dispositivos comerciales, independientemente de la marca o el fabricante?			
Eficiencia	¿El sistema demuestra una gestión eficiente de los dispositivos personalizados y un rendimiento óptimo?			

El entorno de prueba para evaluar el prototipo Matter se diseñará considerando la simulación de un ambiente doméstico real, proporcionando condiciones óptimas para realizar pruebas de funcionalidad y accesibilidad. Las tareas se centrarán en acciones comunes del hogar, permitiendo a los usuarios interactuar con el sistema de manera natural. Aquí se detalla el entorno y las tareas específicas:

○ **Ambiente Doméstico Simulado:**

- **Descripción:** El entorno de prueba simula una vivienda tranquila, libre de ruido externo, con dispositivos incorporados como controles de temperatura y lámparas, además que en el otro escenario se usa un internet el cual sea muy deficiente y con pérdidas de señal en varios momentos, esto asegura una experiencia lo más cercana posible al uso real del prototipo, en una casa común con internet estable y otra que simule un entorno empresarial en el cual tenga un internet inestable.

○ **Selección Diversificada de Usuarios:**

- **Descripción:** Se tomará en cuenta a personas que tengan conocimiento de la tecnología IoT, personas sin conocimiento de la tecnología IoT y el protocolo Matter, además de personas que ya cuenten con uso de Asistentes virtuales en casa y personas que no han usado nunca. Esto garantizará una evaluación integral del prototipo.

○ **Pruebas de Funcionabilidad y Accesibilidad:**

- **Descripción:** Los usuarios realizarán tareas específicas para evaluar la funcionalidad y accesibilidad del prototipo. Las tareas se dividirán en dos categorías principales:
- **Tareas con Tecnología de Asistencia por Voz o Gestos Táctiles:**

- Enciende Luz matter
 - Enciende Luz Thread
 - Dame la Temperatura Matter
 - Enciende Foco Matter Comercial
 - Enciende Switch Matter
 - Apaga Luz Matter
 - Apaga Switch Matter
 - Apaga Foco Matter Comercial
 - Apaga Luz Thread
 - Enciende la Sala (Luz Matter Personalizada - Foco Thread)
 - Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)
 - Apaga la Sala (Luz Matter Personalizada - Foco Thread)
 - Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)
- **Evaluación de la Facilidad de Uso:**
 - **Descripción:** Los usuarios serán observados mientras realizan las tareas asignadas. Se recopilarán datos sobre la facilidad de uso, la eficiencia en la ejecución de tareas y cualquier retroalimentación proporcionada por los usuarios.
 - **Recopilación de Datos y Retroalimentación:**
 - **Descripción:** Se registrarán datos cuantitativos, como el tiempo tomado para completar cada tarea, así como datos cualitativos a través de la retroalimentación verbal de los usuarios. Estos datos serán fundamentales para evaluar la experiencia del usuario y realizar mejoras iterativas en el prototipo.
 - **Pruebas de rendimiento:** Se realizarán pruebas de rendimiento para determinar el rendimiento del prototipo en condiciones de carga.
 - **Interoperabilidad:**
 - **Prueba:** Realizar comunicación entre dispositivos domóticos diversos utilizando el protocolo Matter.
 - **Método:** Enviar comandos desde dispositivos Matter comerciales y personalizados para verificar la interoperabilidad y la correcta recepción de instrucciones.

- **Accesibilidad - Interfaz de Control:**
 - **Prueba:** Evaluar la eficiencia de la interfaz de control dentro de Home Assistant.
 - **Método:** Medir el tiempo que los usuarios tardan en realizar tareas comunes, como encender/apagar dispositivos, ajustar configuraciones, y recopilar comentarios sobre la experiencia de uso.
- **Eficiencia - Optimización del Rendimiento:**
 - **Prueba:** Medir la eficiencia y velocidad de respuesta del sistema en diversas condiciones.
 - **Método:** Realizar tareas variadas, desde simples hasta complejas, y medir los tiempos de respuesta. Analizar y ajustar el rendimiento para optimizar la experiencia del usuario.
- **Accesibilidad - Herramientas de Diagnóstico:**
 - **Prueba:** Verificar la presencia y efectividad de herramientas de diagnóstico.
 - **Método:** Simular problemas de conectividad o funcionamiento de dispositivos y utilizar las herramientas incorporadas para identificar y resolver los problemas.
- **Interoperabilidad - Comunicación sin Problemas:**
 - **Prueba:** Garantizar la comunicación sin problemas entre dispositivos de diferentes marcas a través del protocolo Matter.
 - **Método:** Enviar comandos desde dispositivos de diferentes fabricantes y verificar que el sistema responde adecuadamente, demostrando la interoperabilidad efectiva.
- **Desarrollo de los Dispositivos Matter:**
 - **Prueba:** Verificar la correcta implementación de los dispositivos ESP32 como dispositivos Matter.
 - **Método:** Enviar comandos específicos a los dispositivos ESP32 y verificar su respuesta, asegurando la compatibilidad con el sistema domótico y la correcta integración con el protocolo Matter.

Se tendrá en cuenta que también habrá un cuadro de observaciones en los que se especificaran anomalías, división del tiempo de respuesta, observaciones de demoras y otros elementos del sistema al efectuar las tareas dadas, y que futuramente sea necesario resolver o tener en cuenta, esta plantilla se puede ver en la **Tabla 15**, plantilla que sirve para las observaciones de la evaluación dada:

Tabla 15. Plantilla de Observaciones de la evaluación

Observaciones de la evaluación					
	Tareas de evaluación por control por voz	Criterios de evaluación	Criterios de aceptación		
		Observaciones de tiempo de Respuesta	Funcionalidad	Interoperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Luz Thread				
	Dame la Temperatura Matter				
	Enciende Foco Matter Comercial				
	Enciende Switch Matter				
	Apaga Luz Matter				
	Apaga Switch Matter				
	Apaga Foco Matter Comercial				
	Apaga Luz Thread				
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	El tiempo de respuesta fue más rápido en Thread con una diferencia de 2 segundos	Hubo un retraso de 3 segundos en la primera interacción	Problemas con el dispositivo matter	No se encontró ningún problema
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)				
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)				
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)				

Las plantillas que usar serán las siguientes, elaborada para control, por voz y por interfaz que se utilizara también, como se ve en la **Tabla 16**:

Tabla 16. Plantilla de Evaluación por control de voz

Evaluación por control por voz / Interfaz											
	Tareas de evaluación por control por voz	Criterios de evaluación				Criterios de aceptación					
		Compatibilidad	Usabilidad	Optimización	Diagnóstico y Resolución	Funcionalidad	Accesibilidad	Intuitividad	Fiabilidad	Interoperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	✓	✓	3 segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende Luz Thread	✓	✓	3 segundos	✓	✓	✓	✓	✓	✓	✓
	Dame la Temperatura Matter	✓	✓	5 segundos	✓	✓	n/a	✓	✓	✓	✓
	Enciende Foco Matter Comercial	✓	✓	3 segundos	✓	✓	✓	✓	✗	✓	✓
	Enciende Switch Matter	✓	✓	4 segundos	✓	n/a	✓	✓	✓	✓	✓
	Apaga Luz Matter	✓	✓	6 segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Switch Matter	✓	✓	5 segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Foco Matter Comercial										
	Apaga Luz Thread										
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)										
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)										
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)										
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)										

Para la evaluación de Rendimiento se establecieron las siguientes tablas, en la **Tabla 17** se ve la plantilla de evaluación del rendimiento por voz y por interfaz que también se la puede utilizar:

Tabla 17. Plantilla de evaluación del rendimiento por voz

Evaluación de rendimiento por voz/ Interfaz						
Asistente Virtual	Tareas de evaluación por control por voz	Velocidad de respuesta	Precisión de reconocimiento de voz	Capacidad de procesamiento	Fiabilidad	Interoperabilidad
	Enciende Luz matter	2 segundos	Alta	No se puede solicitar más de una tarea simultánea	Alta	Si
	Enciende Luz Thread	2 segundos	Alta	No se puede solicitar más de una tarea simultánea	Alta	Si
	Dame la Temperatura Matter					
	Enciende Foco Matter Comercial					
	Enciende Switch Matter					
	Apaga Luz Matter					
	Apaga Switch Matter					
	Apaga Foco Matter Comercial					
	Apaga Luz Thread					
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	2 segundos	Alta	No se puede solicitar más de una tarea simultánea	Alta	Si
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	6 segundos	Alta	No se puede solicitar más de una tarea simultánea	Alta	Si
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)					
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)					

3.1.4. Proceso de evaluación

El proceso de evaluación se llevará a cabo de la siguiente manera:

1. Se instalará el prototipo en un entorno de prueba.
2. Se ejecutarán las pruebas funcionales, de rendimiento, de tiempo de respuesta y de control por voz.
3. Se plantearán diferentes tipos de escenarios para las pruebas.
4. Se analizarán los resultados de las pruebas para determinar si el prototipo cumple con los requisitos.

3.2. Resultados de la evaluación

A continuación, se llevó a cabo una serie de 100 interacciones en dos escenarios distintos, considerando diversas condiciones. En el primer escenario, se ejecutaron las pruebas en un entorno donde la conexión a Internet es estable y el sistema domótico está plenamente estabilizado.

En contraste, en el segundo escenario, se configuró el entorno en un lugar con conexión a Internet constantemente inestable, como podría ser el caso de una empresa con numerosos dispositivos conectados a la red y diversos factores externos, en el **Anexo 2**, se puede ver los criterios que tomamos al realizar las observaciones de la evaluación, para el primer escenario, y en el **Anexo 11** se puede ver los resultados, en cambio, del escenario 2.

Cabe destacar que en el primer escenario, en general, la mayoría de las interacciones tuvieron tiempos de respuesta rápidos, aunque se registraron algunos retrasos notables, especialmente en Hassio y Siri, pero en particular, se destacan dos interacciones prolongadas de 5 segundos en Hassio, también se observaron tiempos de respuesta positivos en diferentes otros comandos y dispositivos, aunque algunos incidentes específicos, como un retraso de 9 segundos en la respuesta de Siri, merecen atención.

Además, se mencionan interacciones con tiempos de respuesta de 4, 5, 6 y 9 segundos en diversas interacciones. En algunos casos, se experimentaron retrasos más significativos, como 8 segundos en Google y 15 segundos a través de la interfaz de Google, y también se menciona una interacción de 13 segundos con Alexa, relacionada con la ejecución de dos dispositivos matter, uno comercial y el otro personal.

A continuación, vamos a ver las interacciones que se realizaron según un gráfico de barras y como se van derivando de los tiempos de cada uno del comando en cómo se estableció el prototipo, además de las interacciones divididas por segundos de retraso y por asistentes, cabe recalcar que se examinarán los resultados provenientes por las tablas de interacción por Voz y por Interfaz, a

continuación, veremos los resultados de los comandos y sus interacciones por voz que es el más importante:

- **Enciende Luz matter:**

En la **Figura 80**, se encontrarán los resultados la ejecución del comando resaltado por los diferentes asistentes virtuales, en el cual se puede decir en términos generales, no se detectaron problemas significativos, ya que la mayoría de las respuestas se produjeron en intervalos rápidos, variando entre 1 y 2 segundos, con un tiempo máximo de 3 segundos. No obstante, se observaron dos instancias específicas de interacciones más prolongadas, cada una con una duración de 5 segundos, provenientes de Hassio. Esta prolongación en los tiempos de respuesta, aunque limitada a casos específicos, podría generar inquietud o requerir atención para garantizar un rendimiento óptimo en todas las interacciones.

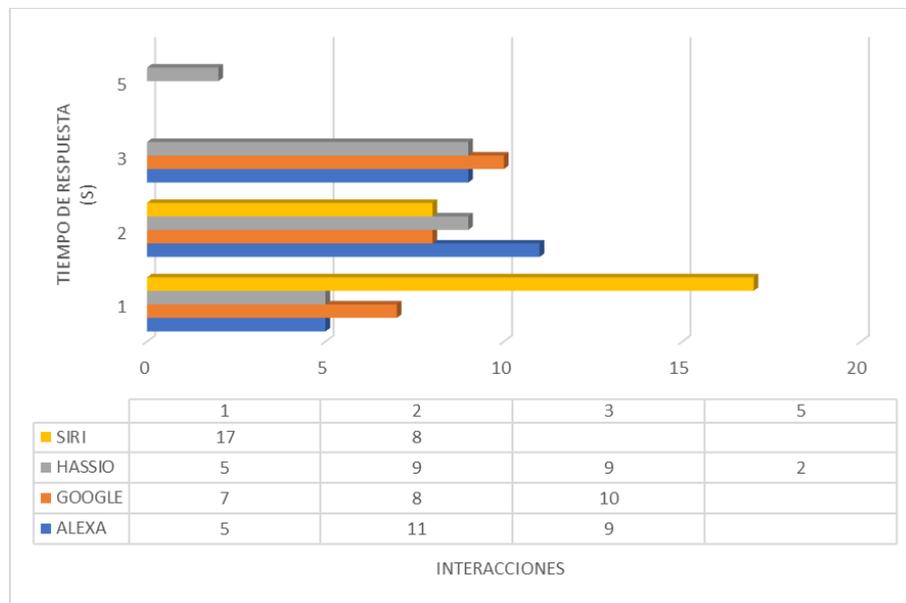


Figura 80. Resultados de Tiempo de respuesta del comando Enciende Luz Matter (Escenario 1).

- **Enciende Luz Thread:**

En la **Figura 81**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en la mayoría de las interacciones, se notó que los tiempos de respuesta alcanzaron un máximo de 3 segundos. Aunque algunas instancias registraron tiempos de 4, 5 y 7 segundos, se mantuvo un límite de 2 interacciones por segundo, distribuidas entre todos los asistentes. Este patrón sugiere que los retrasos no son atribuibles a un asistente en particular, sino que varían en diferentes contextos.

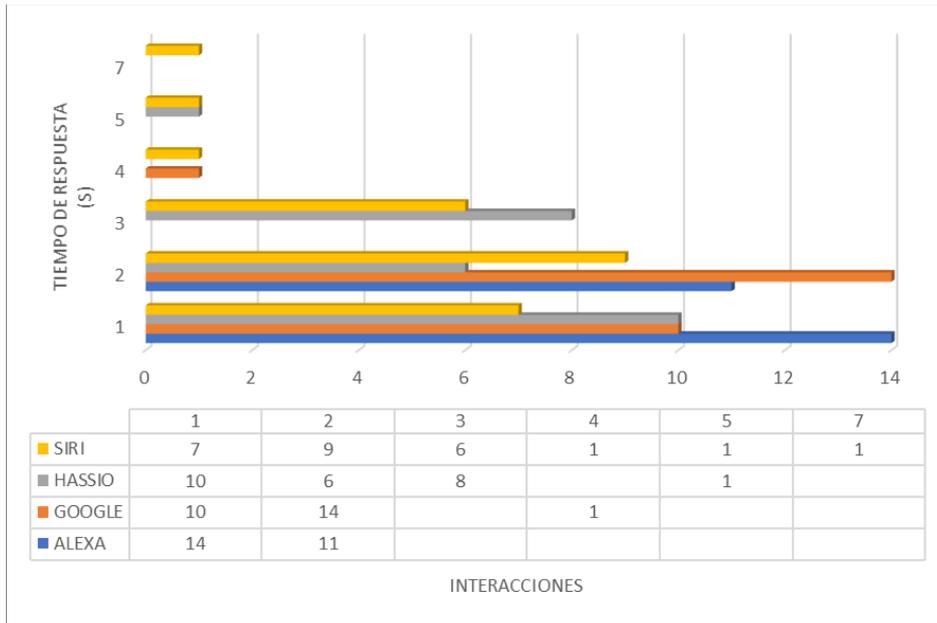


Figura 81. Resultados de Tiempo de respuesta del comando Enciende Luz Thread (Escenario 1).

- Dame la Temperatura Matter:

En la **Figura 82**, se encontrarán los resultados la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en el contexto general de las interacciones, se destacó un retraso específico de 9 segundos al recibir la respuesta del asistente virtual Siri en una situación específica. A pesar de este retraso, el análisis integral no reveló problemas sustanciales adicionales. Este incidente puntual parece ser una excepción y no indicador de problemas recurrentes o significativos en el desempeño del asistente.

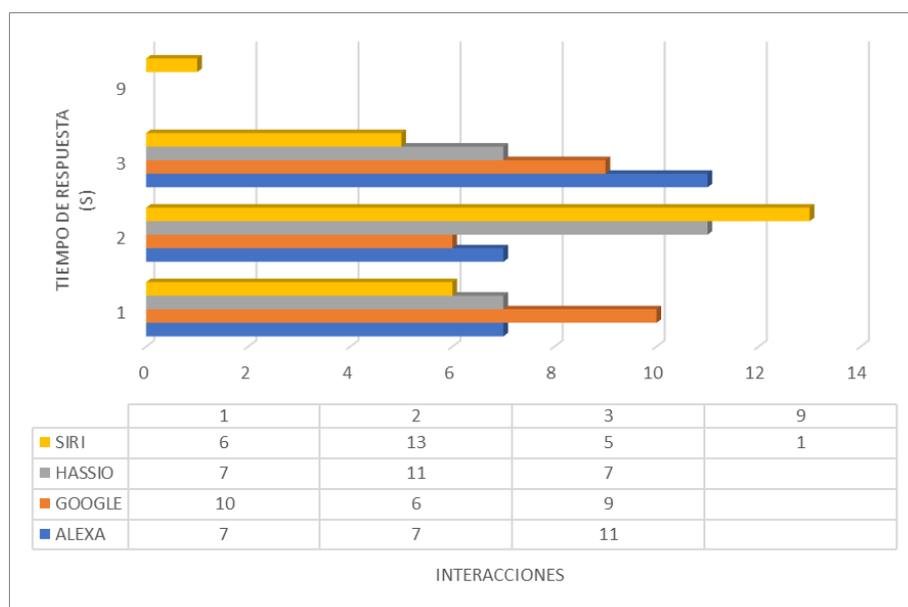


Figura 82. Resultados de Tiempo de respuesta del comando Decir Temperatura (Escenario 1).

- Enciende Foco Matter Comercial:

En la **Figura 83**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que en general, no se observaron problemas significativos, ya que la mayoría de las interacciones presentaron tiempos de respuesta de 1, 2 y 3 segundos. Sin embargo, se identificaron dos interacciones con Hassio y Siri que experimentaron un retraso de 4 segundos, una con Google con 8 segundos, y a través de la interfaz de Google se registró un retraso adicional de 15 segundos. Estos incidentes particulares resaltan variaciones en los tiempos de respuesta en contextos específicos.



Figura 83. Resultados de Tiempo de respuesta del comando Enciende foco matter comercial (Escenario 1).

- Enciende Switch Matter:

En la **Figura 84**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en este escenario, predominaron tiempos de respuesta positivos, aunque algunas interacciones experimentaron retrasos de 4, 5, 6 y 9 segundos. Cabe destacar que cada asistente tuvo solo una interacción con retraso, sin evidencia de una tendencia proporcional específica en los retrasos entre los distintos asistentes.

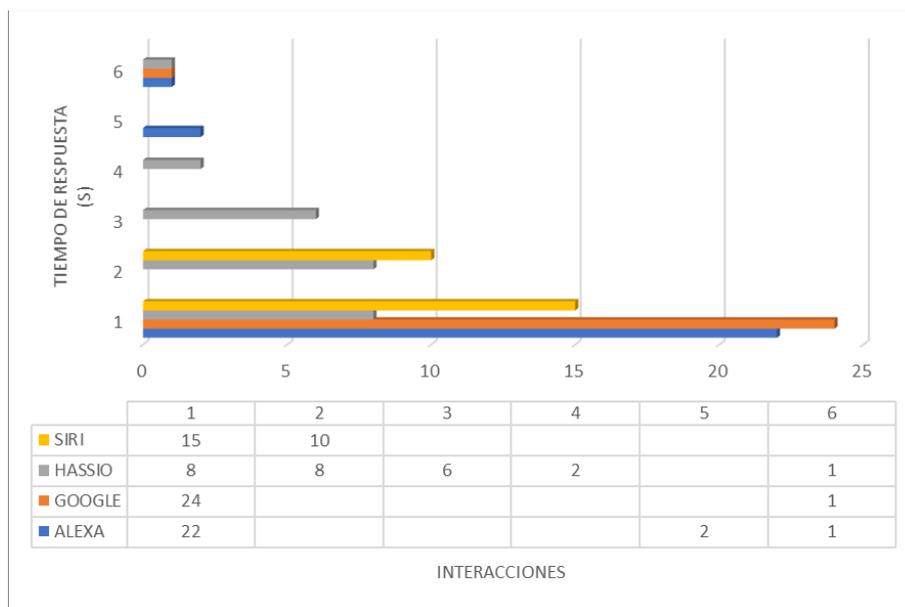


Figura 84. Resultados de Tiempo de respuesta del comando Enciende Switch (Escenario 1).

- Apaga Luz Matter:

En la **Figura 85**, se encontrarán los resultados la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en líneas generales, no se identificaron problemas, aunque se registró una interacción específica de 9 segundos en el tiempo de respuesta de Alexa en el modo de interfaz. Además, se observó otra interacción con Hassio, que presentó un retraso de 4 segundos. Estos incidentes particulares fueron las únicas excepciones en un contexto mayormente libre de problemas.

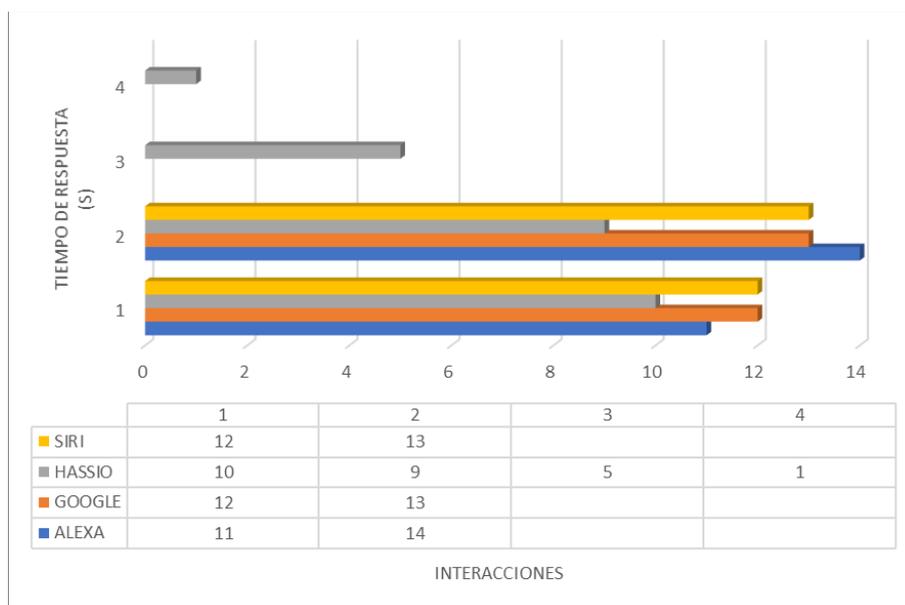


Figura 85. Resultados de Tiempo de respuesta del comando Apaga Luz Matter (Escenario 1).

- **Apaga Switch Matter:**

En la **Figura 86**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en el análisis general, no se identificaron problemas significativos. Sin embargo, se observaron dos interacciones con ligeros retrasos de 5 y 4 segundos, siendo estos los únicos incidentes notables en un conjunto de interacciones mayormente fluido y sin complicaciones.

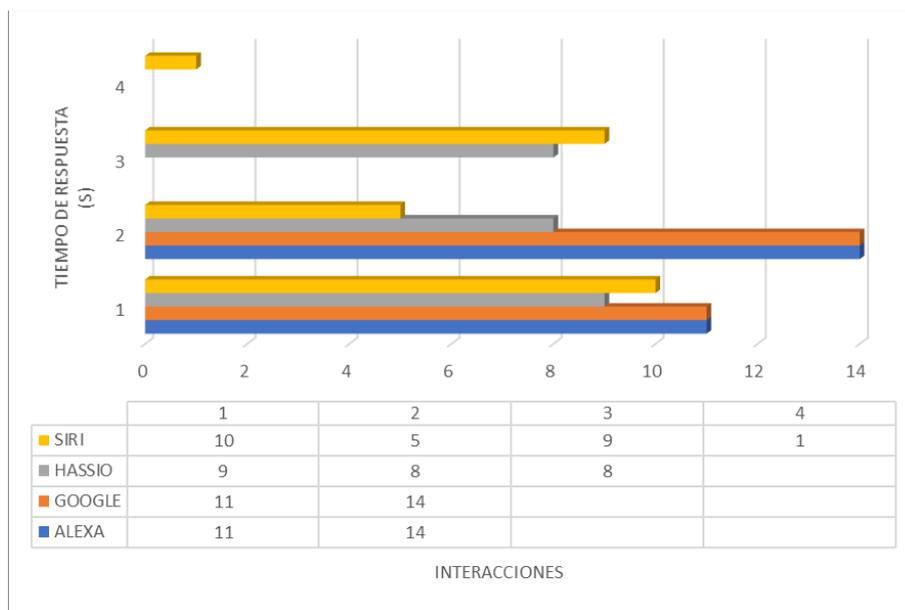


Figura 86. Resultados de Tiempo de respuesta del comando Apaga Switch Matter (Escenario 1).

- **Apaga Foco Matter Comercial:**

En la **Figura 87**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, en términos generales, no se encontraron problemas, excepto por una única interacción con un retraso de 5 segundos, proveniente de Google. No se registraron otras incidencias relevantes durante el análisis.

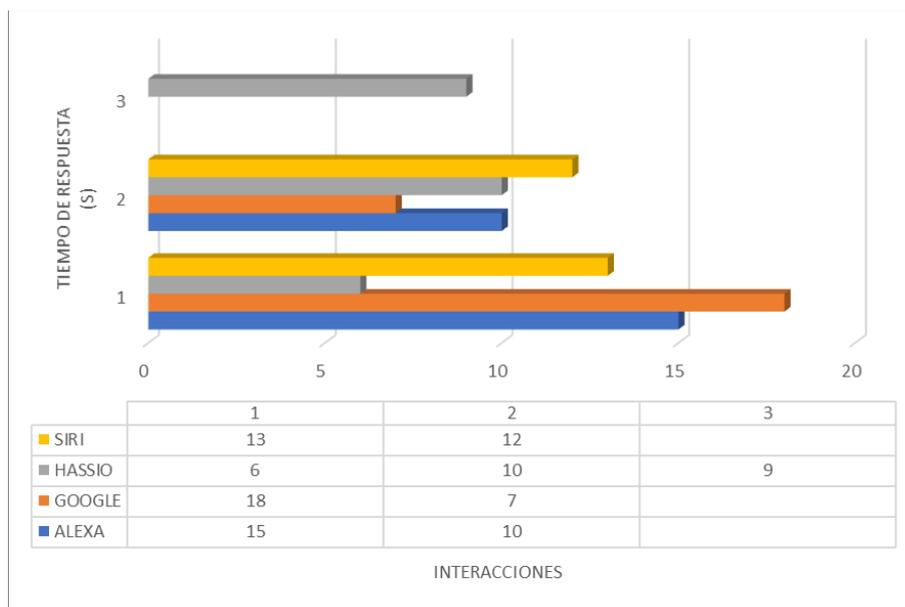


Figura 87. Resultados de Tiempo de respuesta del comando Apaga foco Matter Comercial (Escenario 1).

- Apaga Luz Thread:

En la Figura 88, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que en general, las interacciones se caracterizaron por tiempos de respuesta excelentes. No obstante, se observó una excepción: una única interacción prolongada de 7 segundos, específicamente relacionada con Google. A pesar de este incidente aislado, la mayoría de las interacciones mantuvieron un rendimiento óptimo.

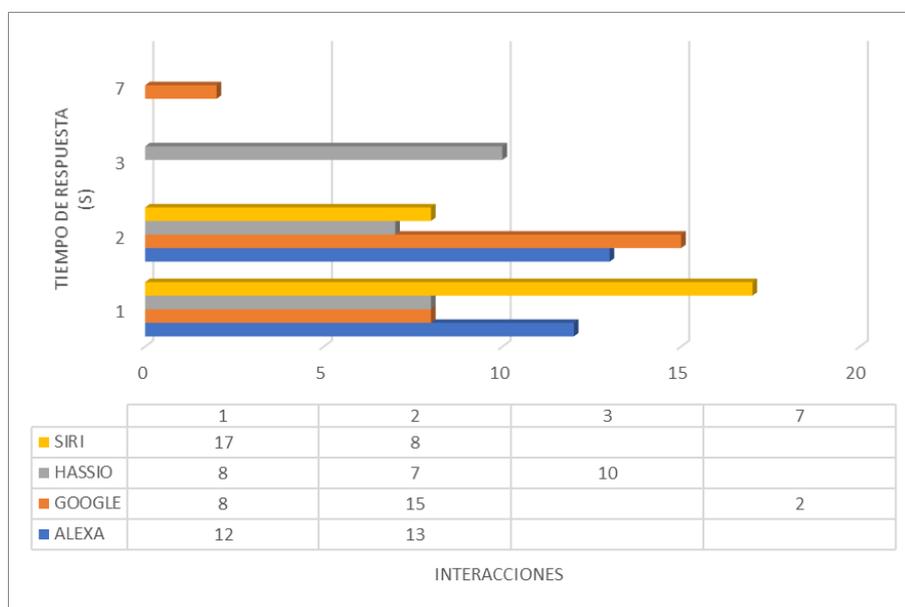


Figura 88. Resultados de Tiempo de respuesta del comando apaga LuzThread (Escenario 1).

- Enciende la Sala (Luz Matter Personalizada - Foco Thread):

En la Figura 89, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que se destacaron tiempos de respuesta excelentes

en ambos dispositivos. No obstante, se registraron algunos retrasos de 4 y 7 segundos, atribuidos a la secuencia de encendido de los dispositivos, con una separación de 2 o 3 segundos entre ellos.

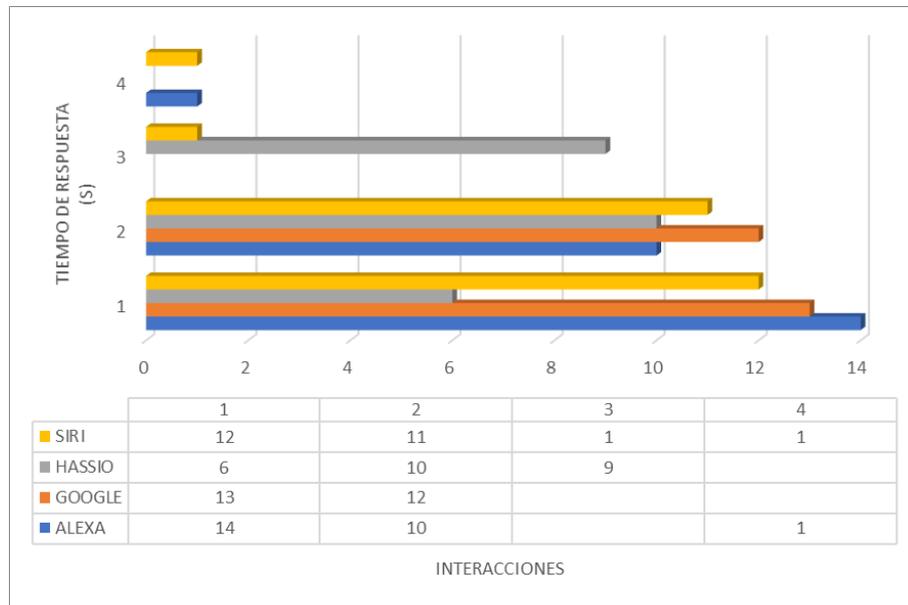


Figura 89. Resultados de Tiempo de respuesta del comando Encienda Sala (Escenario 1).

- Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial):

En la **Figura 90**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que, durante el análisis, no se detectaron problemas, y ambos servicios se iniciaron al mismo tiempo sin inconvenientes evidentes. Sin embargo, es relevante señalar que se observaron dos tiempos de respuesta particularmente notables, con una duración de 7 y 9 segundos, respectivamente. Estos retrasos específicos podrían requerir una evaluación adicional para comprender las condiciones que los ocasionaron.

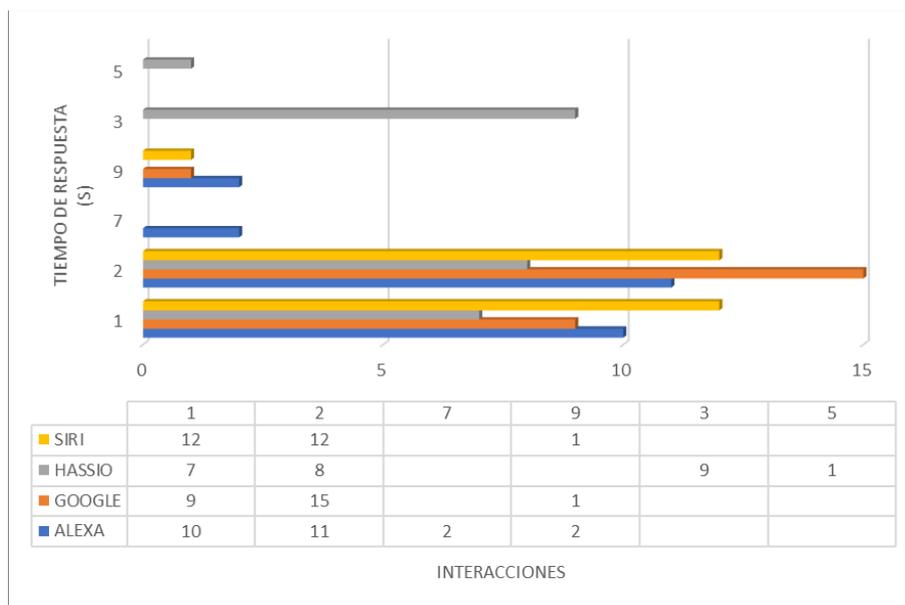


Figura 90. Resultados de Tiempo de respuesta del comando Enciende Sala de Cine (Escenario 1).

- **Apaga la Sala (Luz Matter Personalizada - Foco Thread):**

En la **Figura 91**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que es relevante destacar una prolongación de 13 segundos en el tiempo de respuesta proveniente de Alexa. Este retraso se generó al apagar la sala donde se encontraba el dispositivo Matter personalizado. Importante mencionar que la demora en encender el dispositivo Matter comercial influyó en esta prolongación. Este incidente subraya la conexión entre acciones específicas y sus impactos en los tiempos de respuesta.

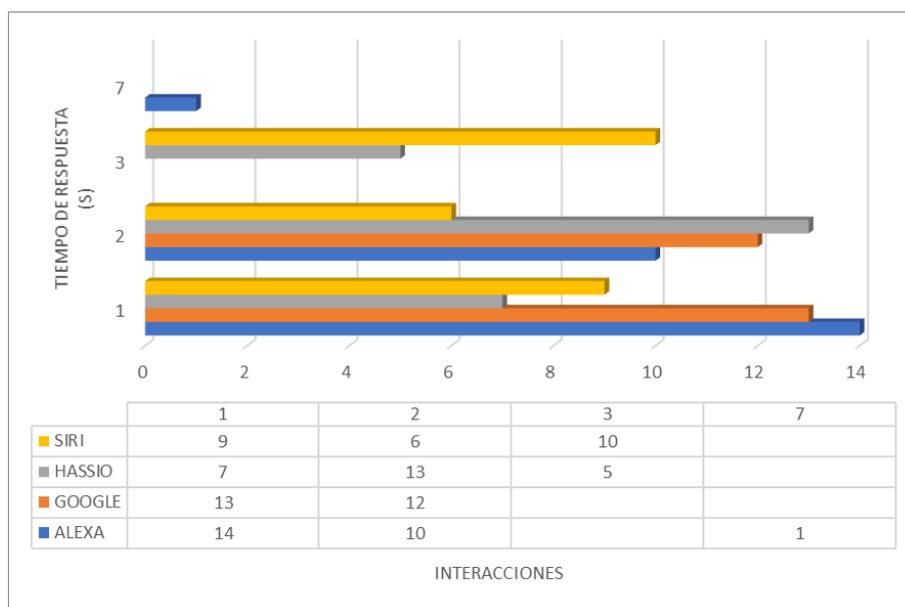


Figura 91. Resultados de Tiempo de respuesta del comando Apagar Sala (Escenario 1).

- **Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial):**

En la **Figura 92**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, se puede decir que es importante resaltar una prolongación de 12 segundos en el tiempo de respuesta proveniente de Alexa. Este retraso ocurrió al apagar la sala de cine que albergaba tanto el dispositivo Matter personalizado como el comercial. Es crucial señalar que la demora en encender el dispositivo Matter comercial podría haber contribuido a esta prolongación. Este incidente destaca la interdependencia de acciones específicas y sus efectos en los tiempos de respuesta.

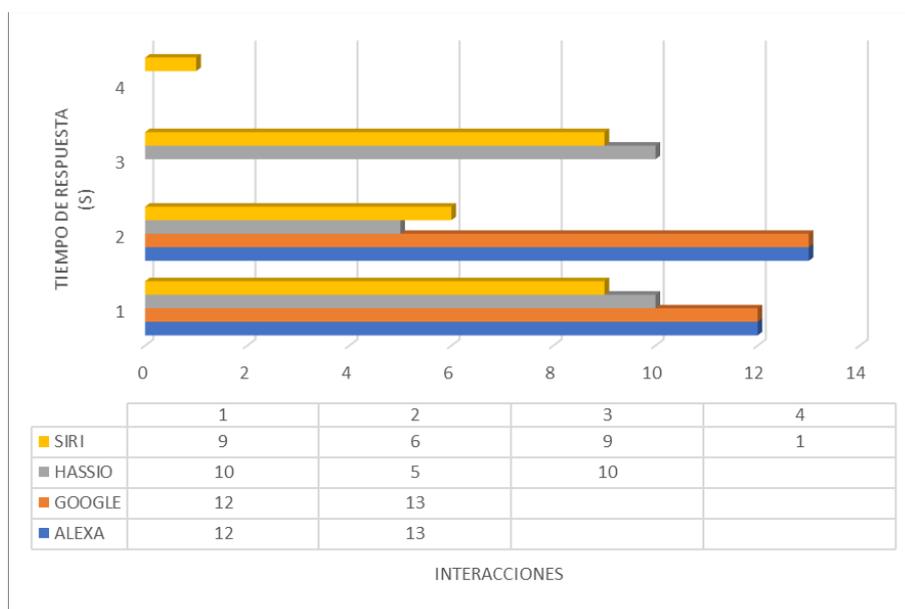


Figura 92. Resultados de Tiempo de respuesta del comando Apaga Sala de Cine (Escenario 1).

Teniendo en cuenta los planes de evaluación por control de voz e interfaz se tiene en cuenta que como se puede ver en el **Anexo 3** y en el **Anexo 4**, se tiene que los criterios de evaluación no sobrepasaron en su campo de optimización los 2 segundos de promedio y además que en sus criterios de aceptación todos sus parámetros nos dieron un resultado positivo, a excepción del parámetro de Intuitividad y la usabilidad nos dieron unos que otros inconvenientes, ya que por parte de las interacciones de público por medio de los comandos de voz se vieron más complicados que en el uso por interfaz, ya que los errores que se pudieron contar fueron referentes al reconocimiento de la voz a través del asistente, o por la falta de conocimiento de los comandos en específicos y correcta pronunciación del mismo, y viéndose que fue mucho más fácil el uso por interfaz.

Ahora referente a las pruebas de Rendimiento de Voz y de Interfaz como se ve en el **Anexo 5** y en el **Anexo 6** se pudo identificar que, el promedio de tiempo de respuesta no sobre pasa los 2 segundos y que en precisión de reconocimiento de voz como ya se mencionó antes se tenía en

cuenta al hablar de manera muy baja confundiendo los comandos y realizando otro que no era, a diferencia de lo que se obtuvo por medio de interfaz, el cual tampoco sobrepasaban los 2 segundos de tiempos de respuesta y teniendo una operación más fácil a ejecutarse cada comando diferente en cada asistente diferente.

Por último, en este escenario se realizó un promedio de la latencia de cada uno de nuestros dispositivos esp32 y el dispositivo comercial, en el cual se ha dividido por asistentes virtuales y un promedio de tiempos de respuestas de todas las operaciones que se llevaron a cabo en cada uno, así como se puede observar en la **Tabla 18**, a continuación:

Tabla 18. Latencia de Dispositivos ESP32 con protocolo Matter (Escenario 1).

Latencia Promedio				
Dispositivos	Asistentes			
	Alexa	Google	Hassio	Siri
ESP32 - S3 / Foco Led (Protocolo Matter)	1.85	1.71	2.02	1.82
ESP32 - C3 / Switch (Protocolo Matter)	1.51	1.43	1.84	1.60
ESP32 - C3 / Sensor de Temperatura (Protocolo Matter)	2.38	2.26	2.08	2.20
ESP32 - H2 / Foco Led (Protocolo Thread)	1.66	1.70	1.97	1.81
Foco Matter Comercial (Protocolo Matter)	1.73	1.90	1.95	1.74

En la cual se puede describir que el dispositivo de sensor de temperatura tiene un promedio más alto de tiempo de respuesta en cualquiera de nuestros asistentes virtuales superando los 2 segundos , pero teniendo en cuenta que este hace un proceso de toma de datos y ejecución del comando con estos mismos datos se puede asumir que son aceptables, ahora si verificamos los tiempos de respuestas de los dispositivos matter personalizado, contra el promedio del dispositivo matter comercial, se encuentra en un nivel similar, dando a entender que estos dos están a similitud en un entorno donde el internet es estable y no existe problemáticas de algún tipo referente a la red.

En la **Figura 93**, se puede observar como el dispositivo Esp32 – S3 conlleva una serie de diferentes comandos, pero respondiendo de manera positiva a todos estos con tiempo de respuesta de entre 1 y 3 segundos, teniendo en cuenta algunas derivaciones de tiempos prolongados no significativos de hasta 9 segundos, pero como se observó, eran interacciones que se encontraron solo un par de veces en los diferentes asistentes, durante todo este proceso.

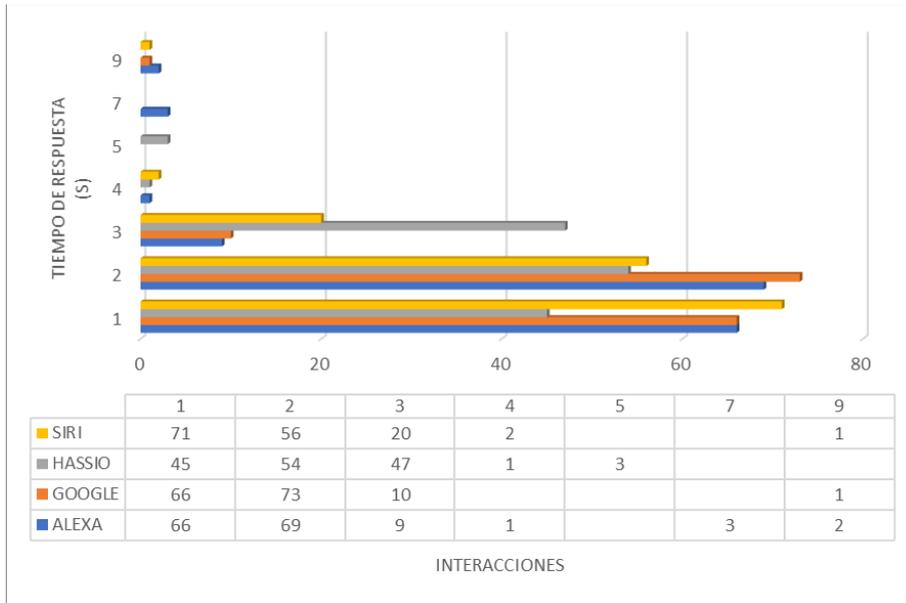


Figura 93. Tiempo de respuesta del dispositivo ESP32 S3 (Escenario 1).

En la **Figura 94**, se puede ver ahora el tiempo de respuesta que se obtuvo de todas las órdenes dada al asistente virtual, para ejecutar en el ESP32 – C3, teniendo, así como en la figura anterior una respuesta de entre 1 a 3 segundos máximo, pero tomando en cuenta que los 3 segundos que necesitaban era la plataforma de HASSIO, y Siri, y los tiempos de respuestas prolongados hasta 6 segundos provenientes de los diferentes asistentes, pero siendo interacciones no significativas.

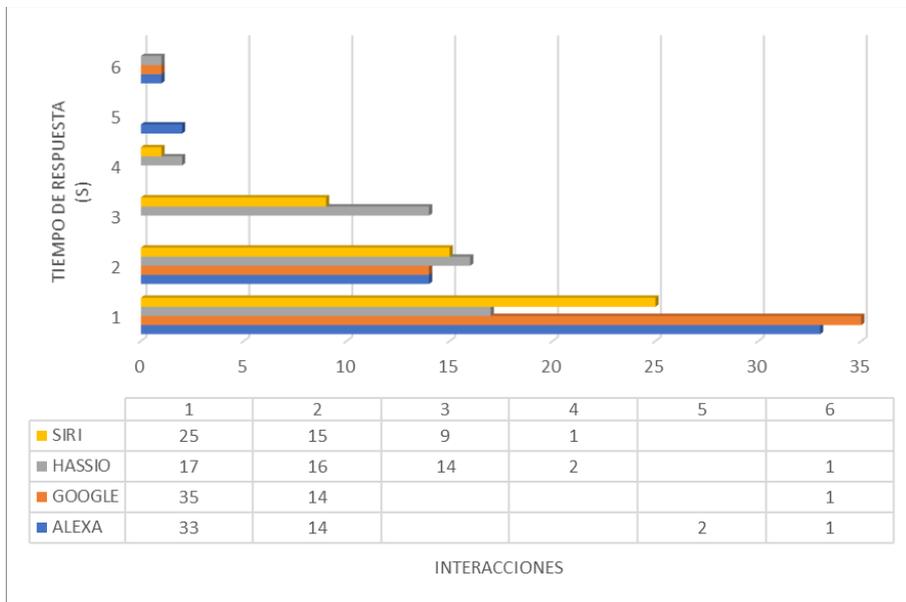


Figura 94. Tiempo de respuesta del dispositivo ESP C3 Switch (Escenario 1).

En la **Figura 95**, se puede ver el trabajo que realizó de manera exitosa el sensor de Temperatura, el cual fue de entre 1 y 3 segundos, y solo teniendo una interacción prologada de 9 segundos al responder cuando se ejecutó el comando proveniente de Siri.

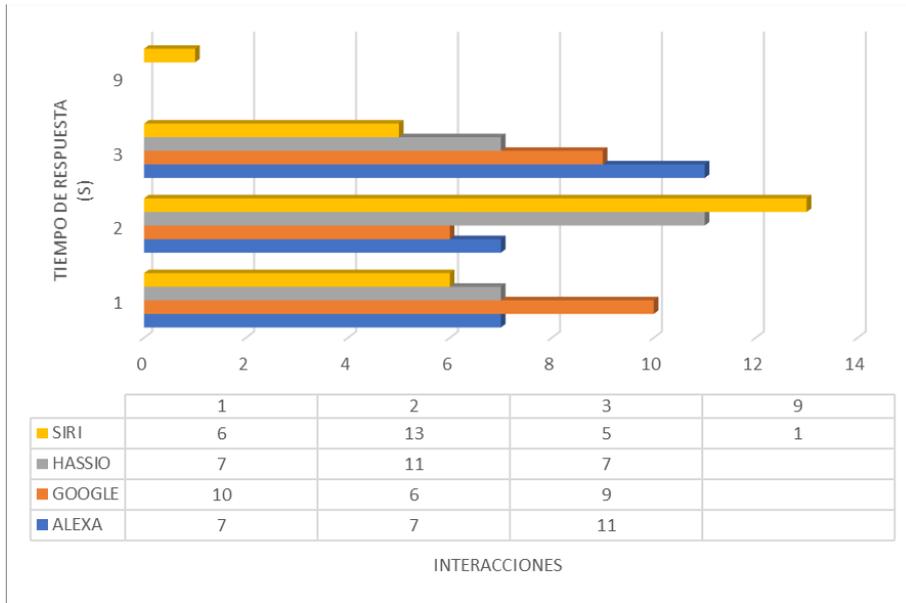


Figura 95. Tiempo de respuesta del dispositivo ESP32 C3 Sensor (Escenario 1).

Para el dispositivo ESP32 – H2 Mini, al realizar las diferentes tareas, se puede tener uso del Gateway Thread con soporte Matter, ya que el dispositivo esp32 h2, cuenta con un solo protocolo que es el Thread, poniendo al fin en práctica el uso de Gateway, en donde pudimos denotar que se lo pudo usar y tuvo tiempos de respuestas al igual que los otros de entre un máximo de 3 segundos promedios, y que funcionaba de manera correcta al ejecutar nuestro dispositivo Thread, dando solo algunos tiempos de retraso de entre 4, 5 y 7 segundos provenientes de algunos asistentes virtuales, como se puede ver en la **Figura 96:**

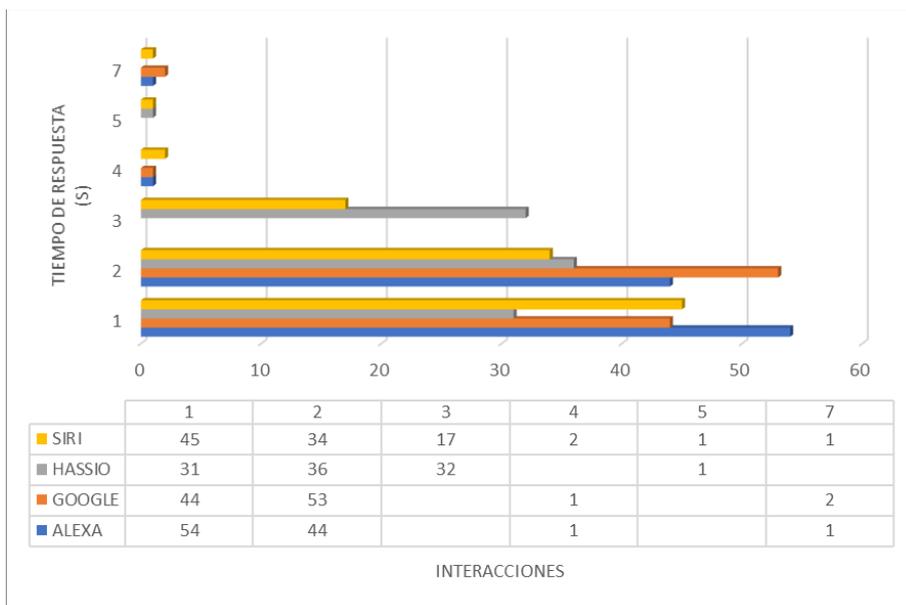


Figura 96. Tiempo de respuesta del dispositivo ESP32 H2 (Escenario 1).

Y por último se tiene el dispositivo comercial matter comercial el cual dio gran sorpresa dando algunos tiempos de retraso de entre 4 y 9 segundos, derivando de los diferentes asistentes virtuales que se encontraban en el prototipo, pero que de igual manera sus tiempos de respuestas habituales eran de entre 1 a 3 segundos, como se ve en la **Figura 97**:

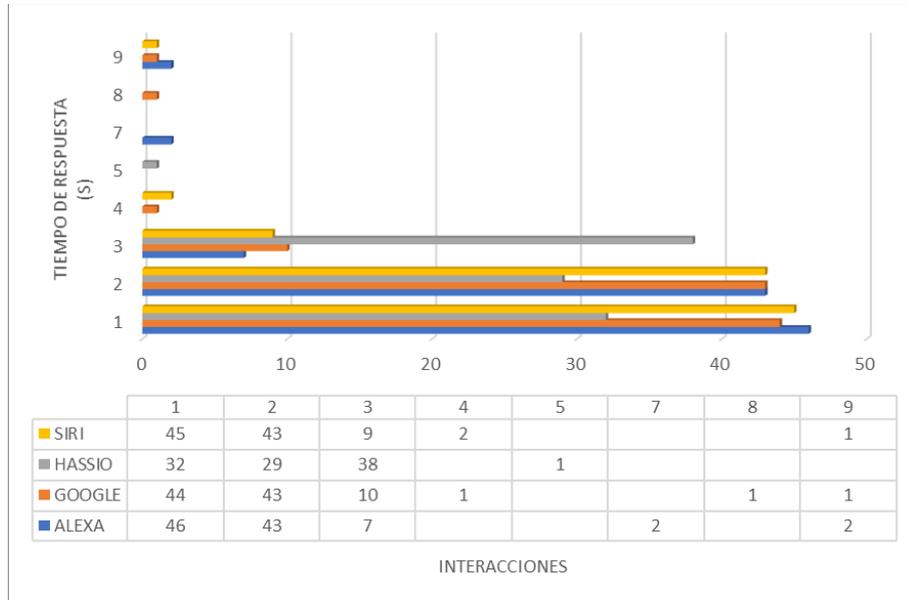


Figura 97. Tiempo de respuesta del dispositivo Matter comercial (Escenario 1).

Ahora se procede a analizar el segundo escenario:

En donde, se va a ver las interacciones que se realizaron según un gráfico de barras y como se van derivando de los tiempos de cada uno del comando en cómo se estableció el prototipo, además de las interacciones divididas por segundos de retraso y por asistentes, cabe recalcar que se examinarán los resultados provenientes por las tablas de interacción por Voz y en el **Anexo 7**, se podrá ver las interacciones divididas por asistentes y como se desarrolló por interfaz

- **Enciende Luz matter**

En la **Figura 98**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, en esta fase de interacciones, debido a la inestabilidad de la red, se observaron tiempos de retraso que variaron entre 5 y 12 segundos. Sin embargo, se registraron retrasos más extensos, que oscilaron entre 14 y 43 segundos. Es destacable que los retrasos más significativos fueron de 41, 43 y 53 segundos, siendo la mayoría atribuible a Hassio. Es importante señalar que estos retrasos se limitaron a una única interacción en cada intervalo de tiempo mencionado.

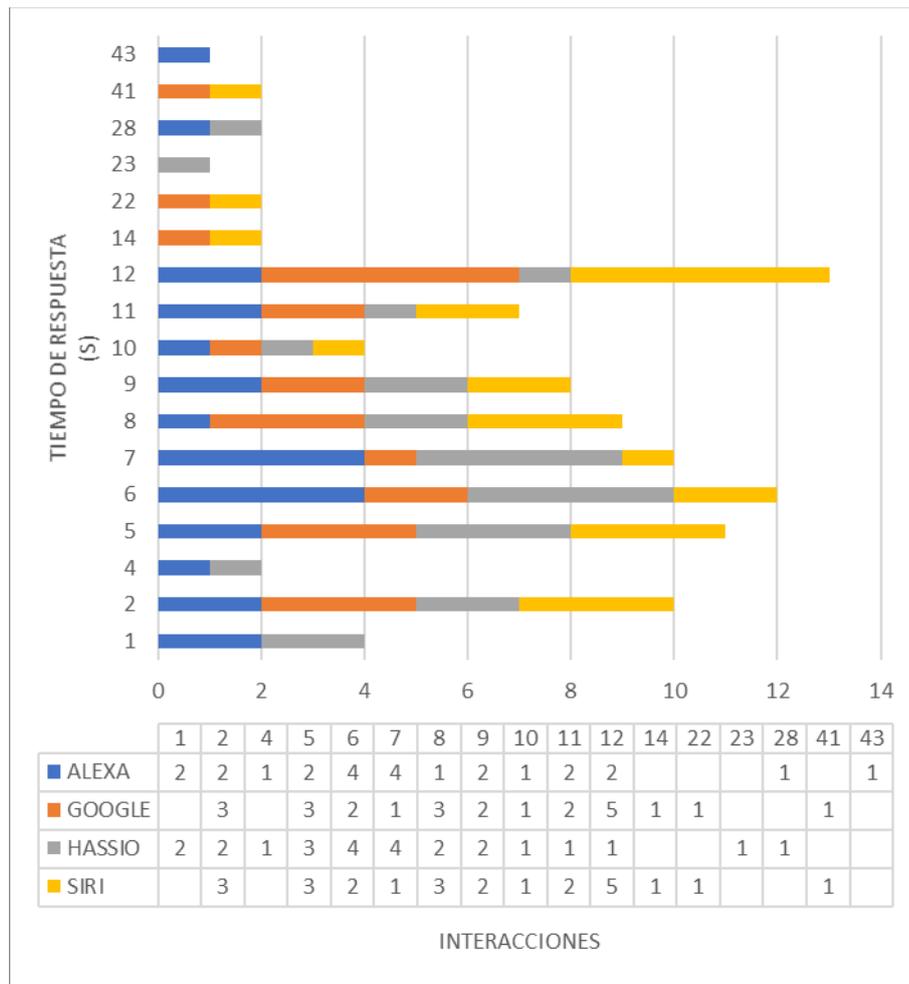


Figura 98. Resultados de Tiempo de respuesta del comando Enciende Luz Matter (Escenario 2).

- **Enciende Luz Thread**

En la **Figura 99**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, sus tiempos de respuesta, al igual que los demás, no representan un obstáculo para realizar la tarea solicitada. No obstante, también se registraron respuestas que superaron los 44 segundos, con un tiempo de respuesta de 68 segundos, atribuido exclusivamente a Hassio.

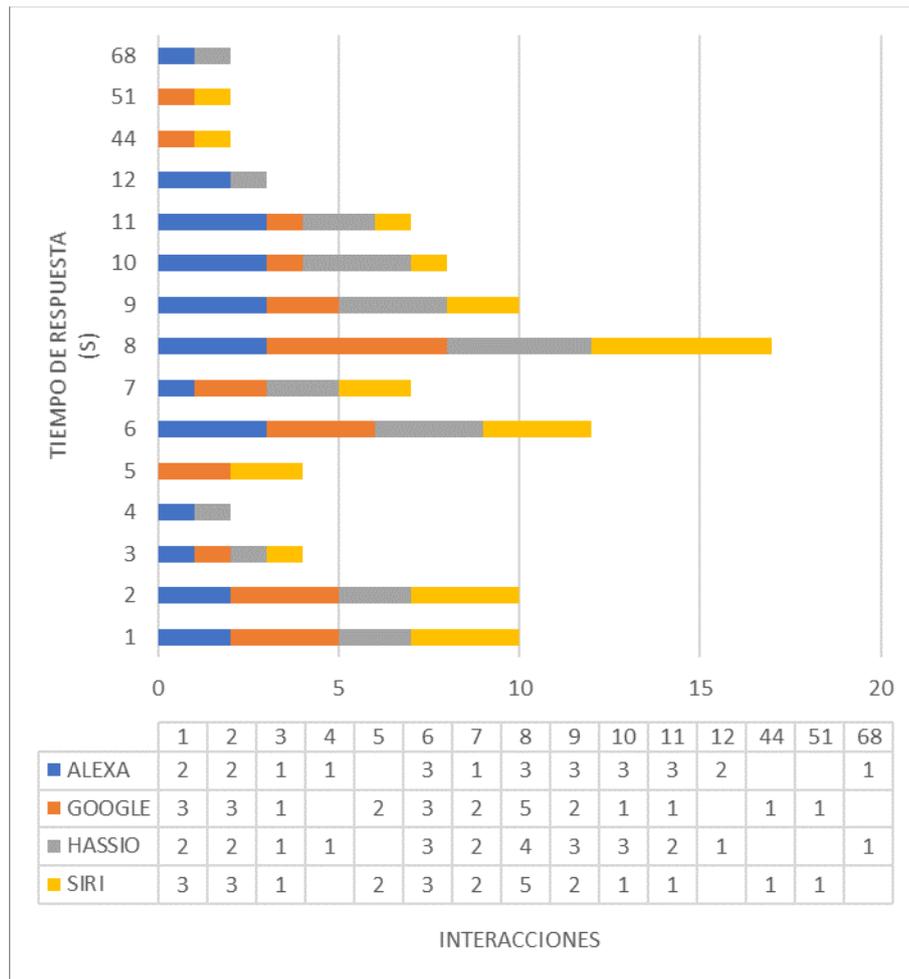


Figura 99. Resultados de Tiempo de respuesta del comando Enciende Luz Thread (Escenario 2).

- **Dame la Temperatura Matter**

En la **Figura 100**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, al ejecutar este comando y llevar a cabo estas interacciones, se observó que la mayoría de las respuestas oscilaron entre 5 y 12 segundos.

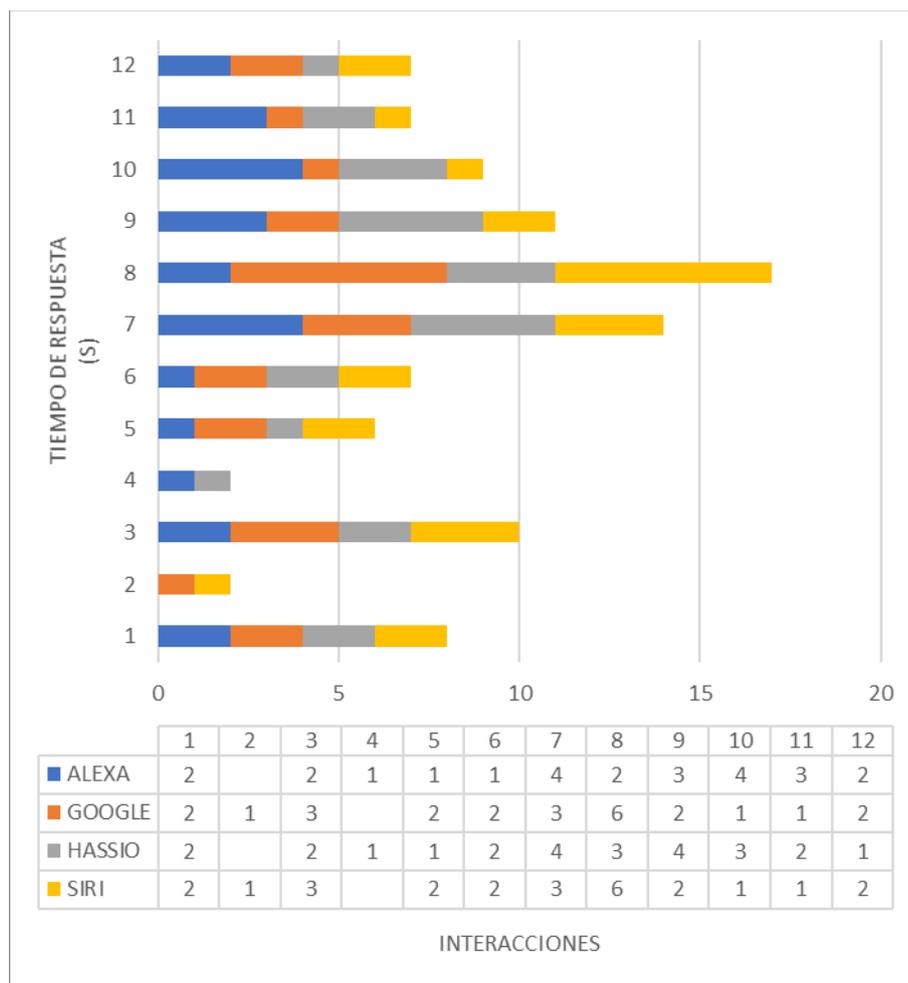


Figura 100. Resultados de Tiempo de respuesta del comando Decir Temperatura (Escenario 2).

- **Enciende Foco Matter Comercial**

En la **Figura 101**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, los retrasos obtenidos al apagar el dispositivo se limitaron a un máximo de 12 segundos de respuesta. Sin embargo, se registraron interacciones que superaron el minuto, destacando un retraso de 66 segundos por parte de Hassio y de Siri. Estos tiempos prolongados podrían requerir una evaluación adicional para comprender las posibles causas subyacentes.

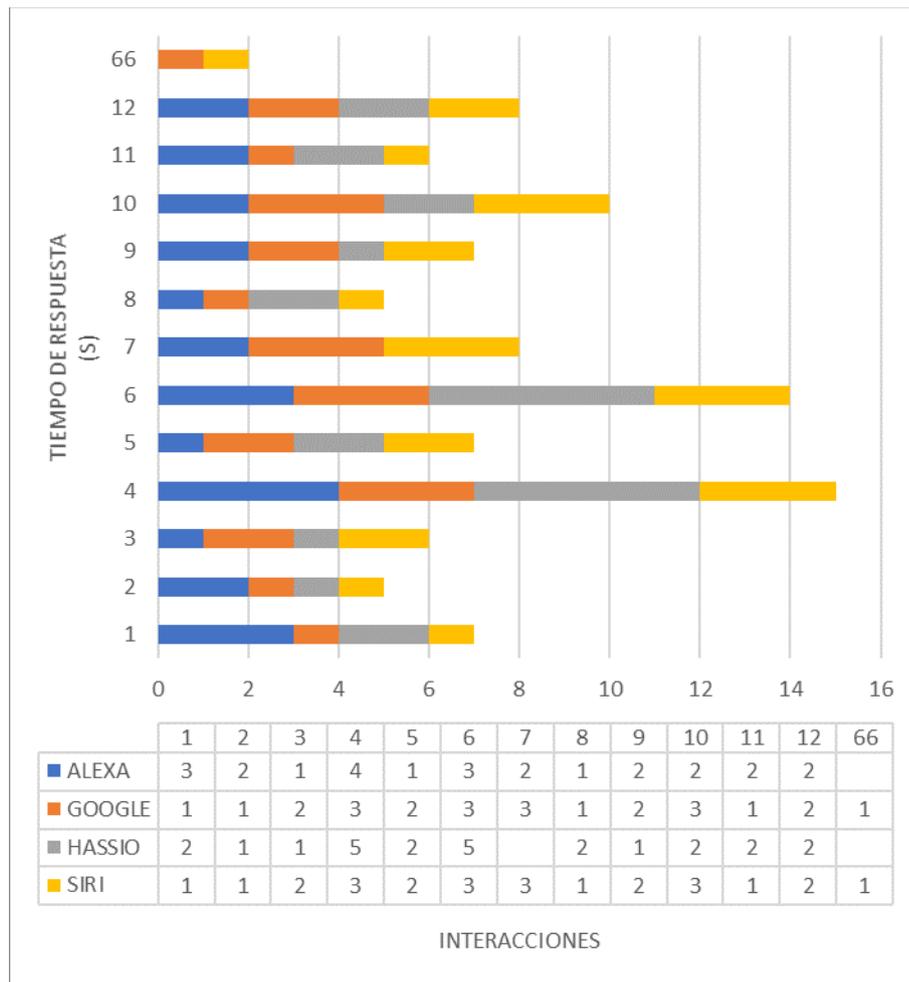


Figura 101. Resultados de Tiempo de respuesta del comando Enciende foco matter comercial (Escenario 2).

- **Enciende Switch Matter**

En la **Figura 102**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, las interacciones se mantuvieron por debajo de los 13 segundos, permitiendo una ejecución eficiente de la tarea, incluso en condiciones de conexión a la red más bajas. No obstante, se registraron un par de retrasos notables al ejecutar el comando por voz, con un tiempo de 51 segundos tanto por parte de Google como de Siri. Estos retrasos pueden influir en la experiencia general de usuario y podrían requerir una revisión detallada.

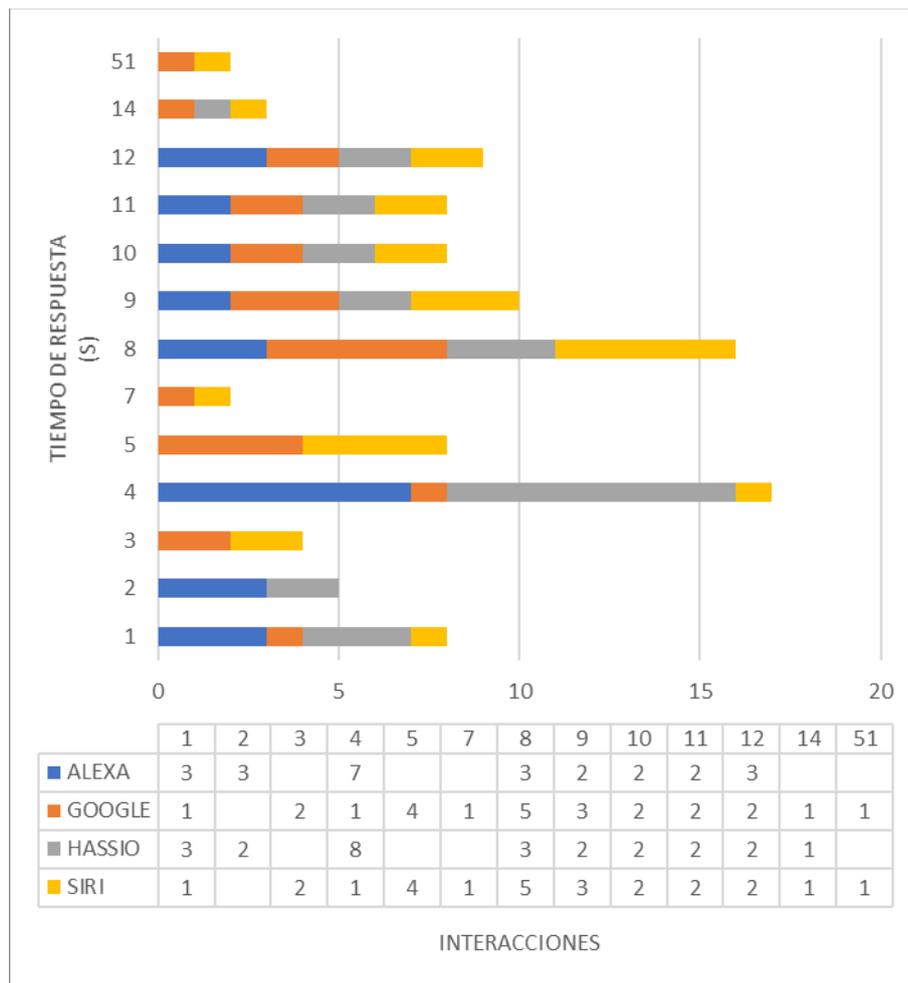


Figura 102. Resultados de Tiempo de respuesta del comando Enciende Switch (Escenario 2).

- **Apaga Luz Matter**

En la **Figura 103**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, no se experimentaron problemas, ya que los tiempos de respuesta se mantuvieron dentro de un máximo de 12 segundos. Sin embargo, se registró una respuesta excepcionalmente larga de 66 segundos, lo cual podría requerir una revisión detallada para comprender las circunstancias particulares de este evento.

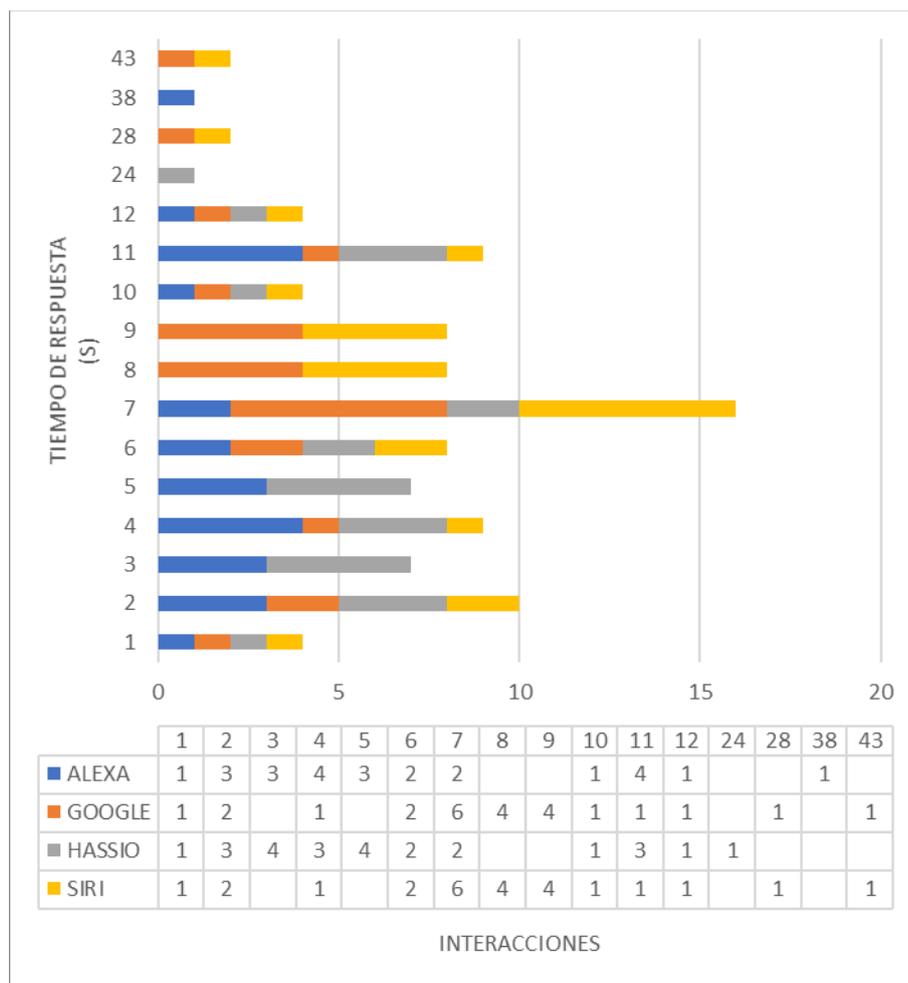


Figura 103. Resultados de Tiempo de respuesta del comando Apaga Luz Matter (Escenario 2).

- **Apaga Switch Matter**

En la **Figura 104**, se encontrarán los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, los retrasos variaron entre 3 y 10 segundos en las distintas interacciones. No obstante, se detectaron dos interacciones prolongadas que superaron el minuto de espera.

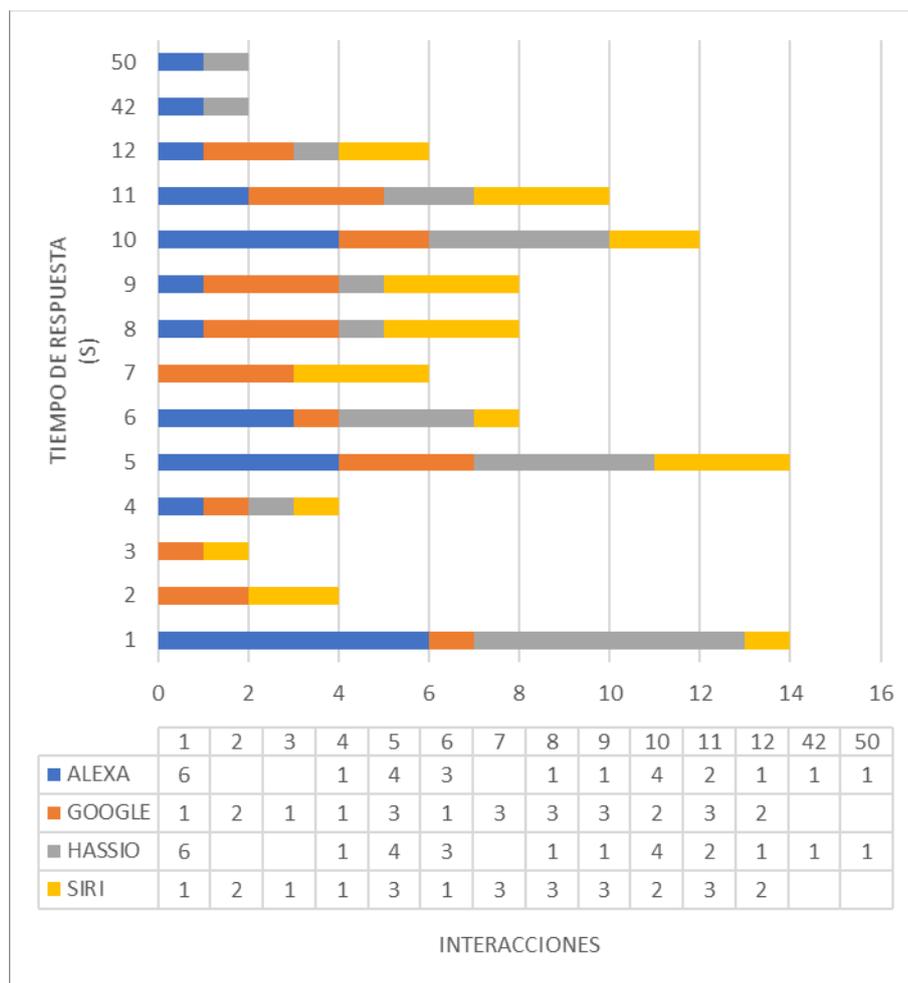


Figura 104. Resultados de Tiempo de respuesta del comando Apaga Switch Matter (Escenario 2).

- Apaga Foco Matter Comercial

En la **Figura 105**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, los tiempos de respuesta se mantuvieron mayormente por debajo de los 12 segundos, con algunas respuestas rápidas de 1 segundo o menos en ocasiones. No obstante, se destacó una interacción excepcional que superó el minuto, con un tiempo de respuesta de 61 segundos. Este evento particular podría requerir una evaluación adicional para comprender las circunstancias detrás de esta demora significativa.

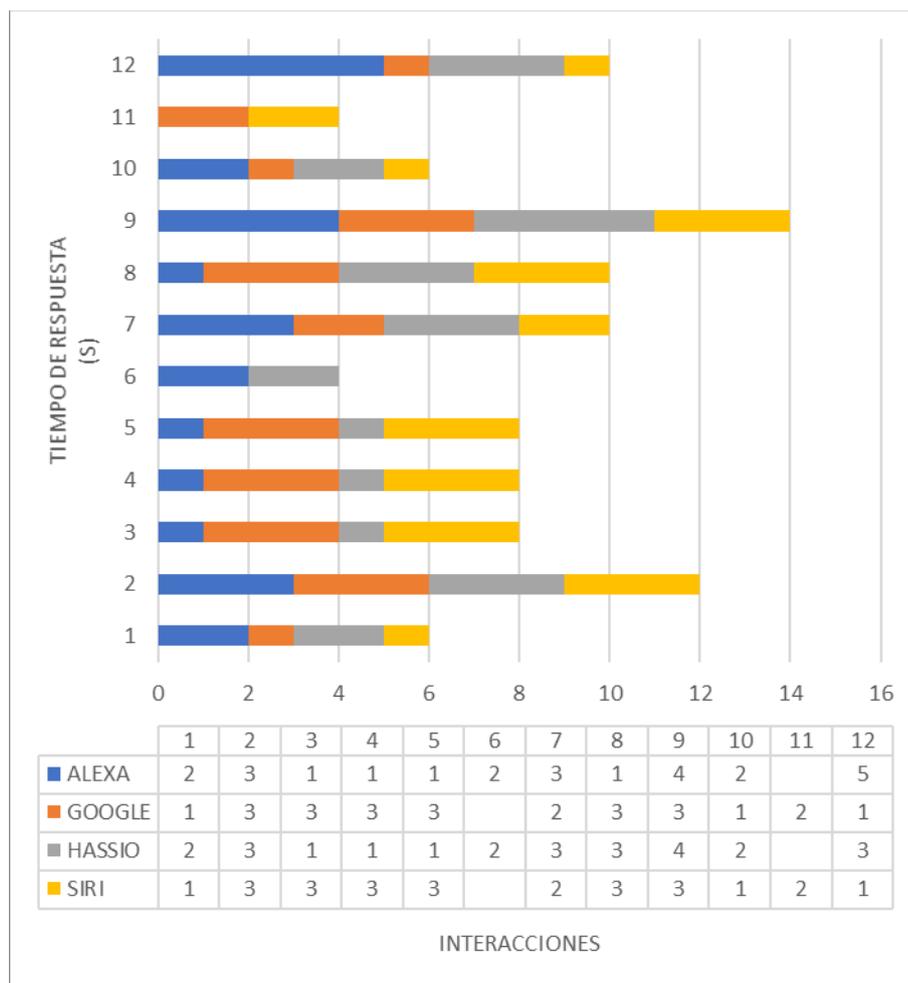


Figura 105. Resultados de Tiempo de respuesta del comando Apaga foco Matter Comercial (Escenario 2).

- **Apaga Luz Thread**

En la **Figura 106**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, no se identificaron problemas, aunque se observaron retrasos que variaron entre 1 y 12 segundos. Sin embargo, cabe destacar que se experimentaron retrasos en algunas interacciones que oscilaban entre los 30 segundos, llegando incluso a superar el minuto, especialmente en los asistentes de Google y Siri.

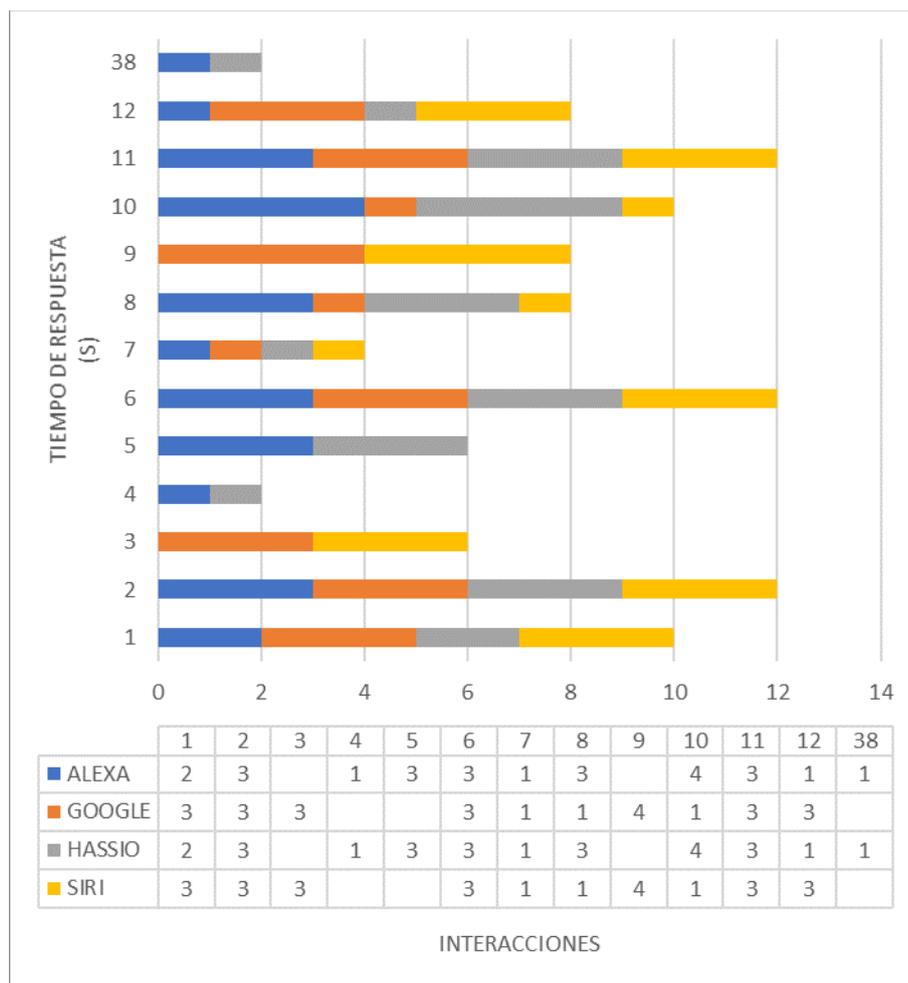


Figura 106. Resultados de Tiempo de respuesta del comando apaga Luz Thread (Escenario 2).

- **Enciende la Sala (Luz Matter Personalizada - Foco Thread)**

En la **Figura 107**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, hubo cargas de tiempo de retraso que oscilaron entre 26 segundos y más de 1 minuto al utilizar comandos por medio de voz. Estos retrasos específicos fueron de 32, 43, 57, 63 y 69 segundos respectivamente, siendo una única interacción por cada asistente en los diversos casos.

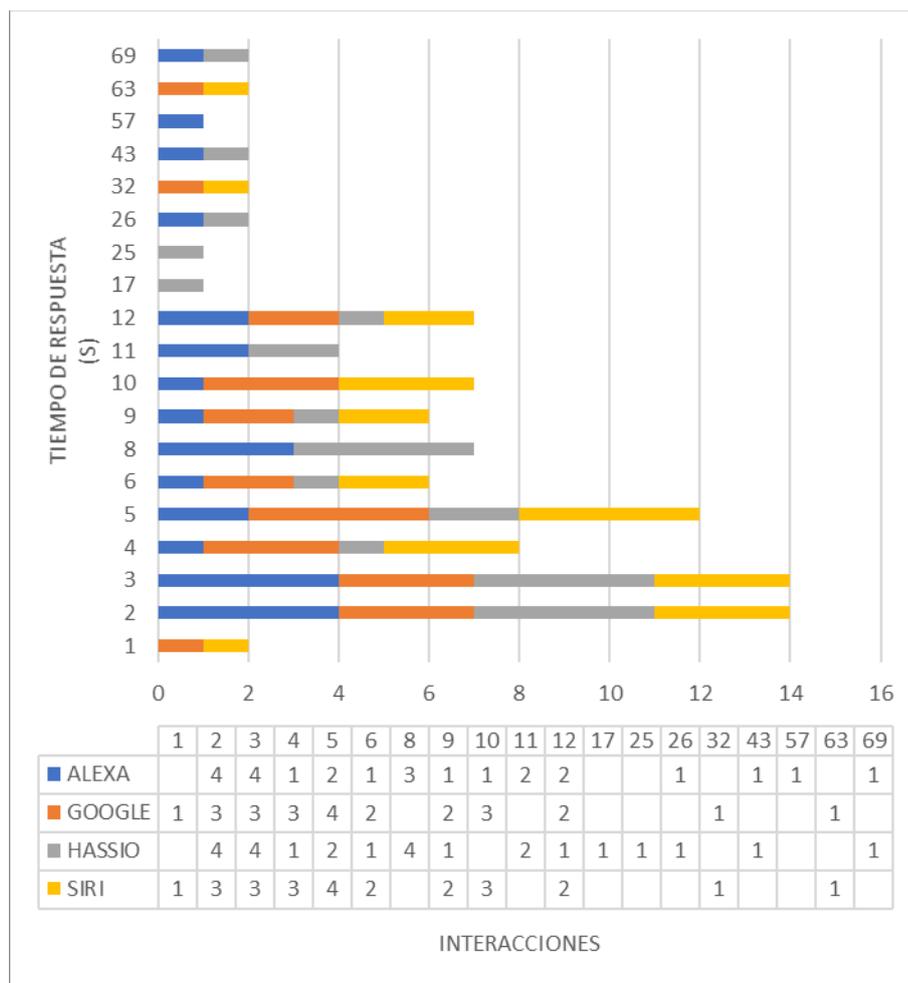


Figura 107. Resultados de Tiempo de respuesta del comando Enciende Sala (Escenario 2).

- **Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)**

En la **Figura 108**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, las interacciones se mantuvieron dentro de un límite de 12 segundos de retraso. Sin embargo, se identificaron muy pocas interacciones provenientes de todos los asistentes virtuales, destacando un tiempo de retraso de 58 segundos y otro de 61 segundos. Estos eventos particulares podrían requerir una atención adicional para comprender las razones detrás de los tiempos de respuesta significativamente prolongados.

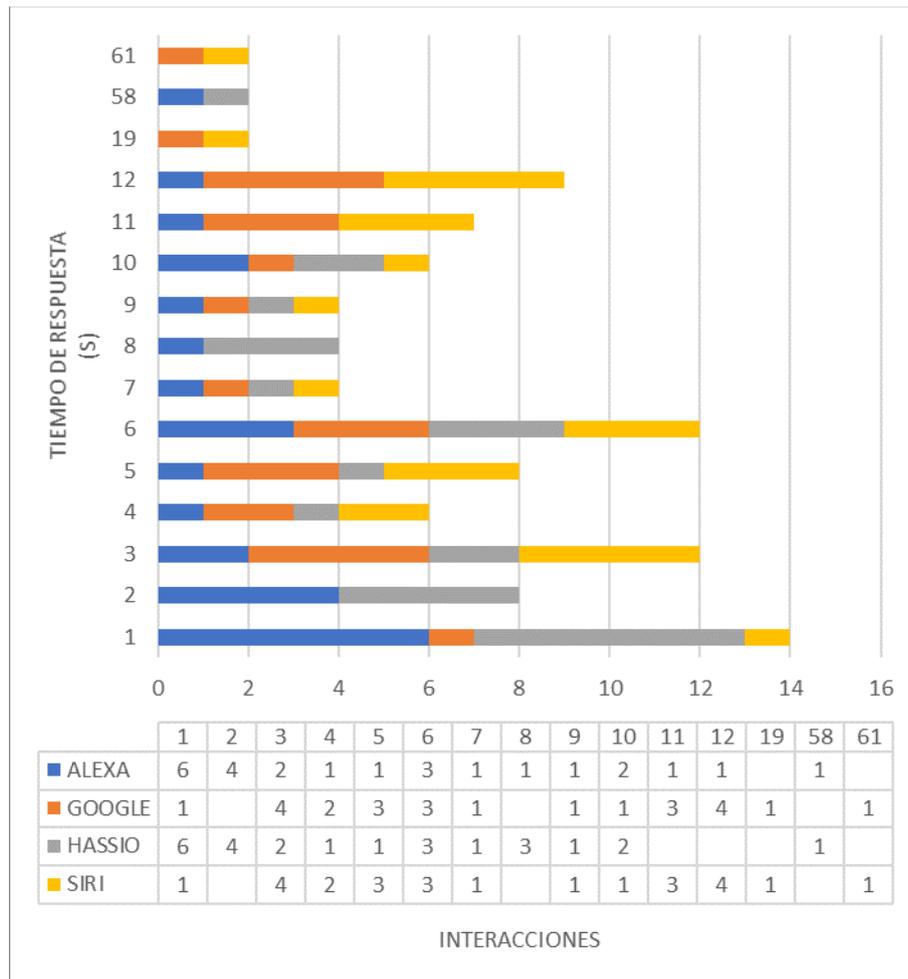


Figura 108. Resultados de Tiempo de respuesta del comando Enciende Sala de Cine (Escenario 2).

- **Apaga la Sala (Luz Matter Personalizada - Foco Thread)**

En la **Figura 109**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, en este escenario, los tiempos prolongados se mantenían por debajo de los 12 segundos hasta que se obtuvieron algunas interacciones con un tiempo de retraso de 33 segundos y 56 segundos, provenientes específicamente de las interacciones realizadas con Hassio. Estos retrasos particulares podrían requerir una revisión más detallada para comprender las circunstancias detrás de estas demoras.

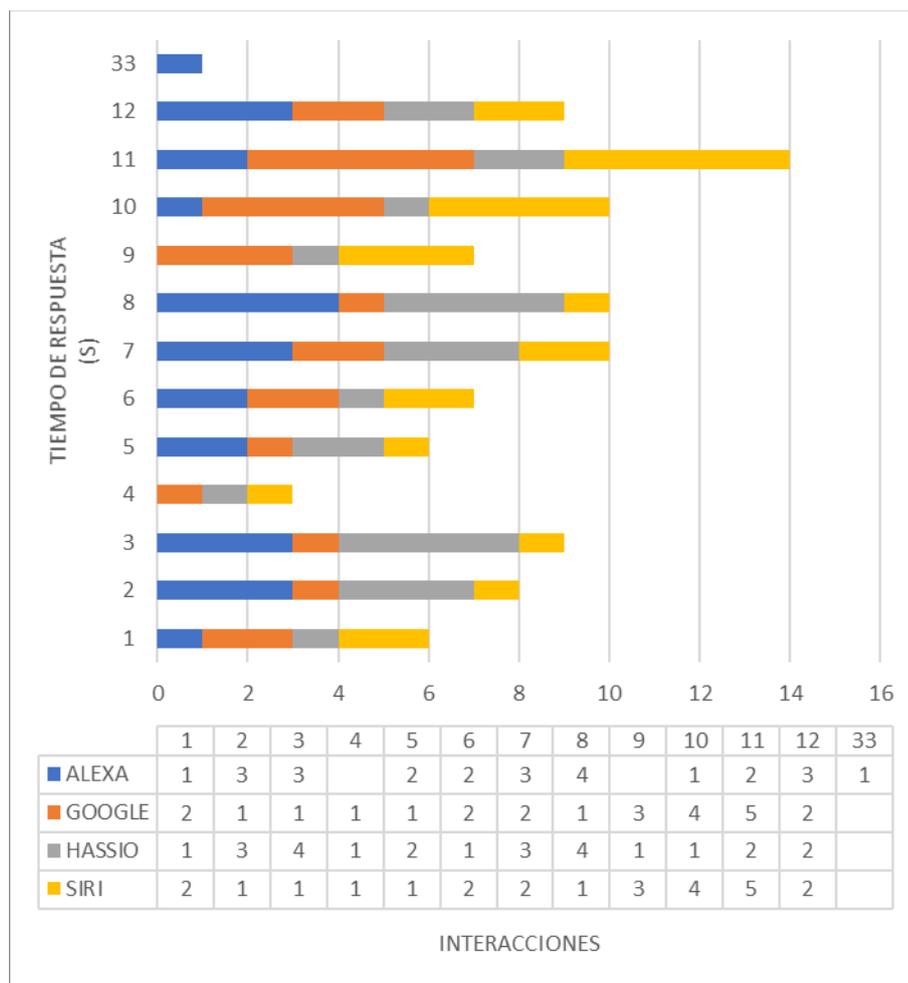


Figura 109. Resultados de Tiempo de respuesta del comando Apagar Sala (Escenario 2).

- **Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)**

En la **Figura 110**, se encuentran los resultados, la ejecución del comando resaltado por los diferentes asistentes virtuales, los tiempos de respuesta se ubicaron en un rango total de 1 a 12 segundos, siendo escasos los tiempos de respuesta de 1 segundo y más frecuentes los de 6 a 10 segundos.

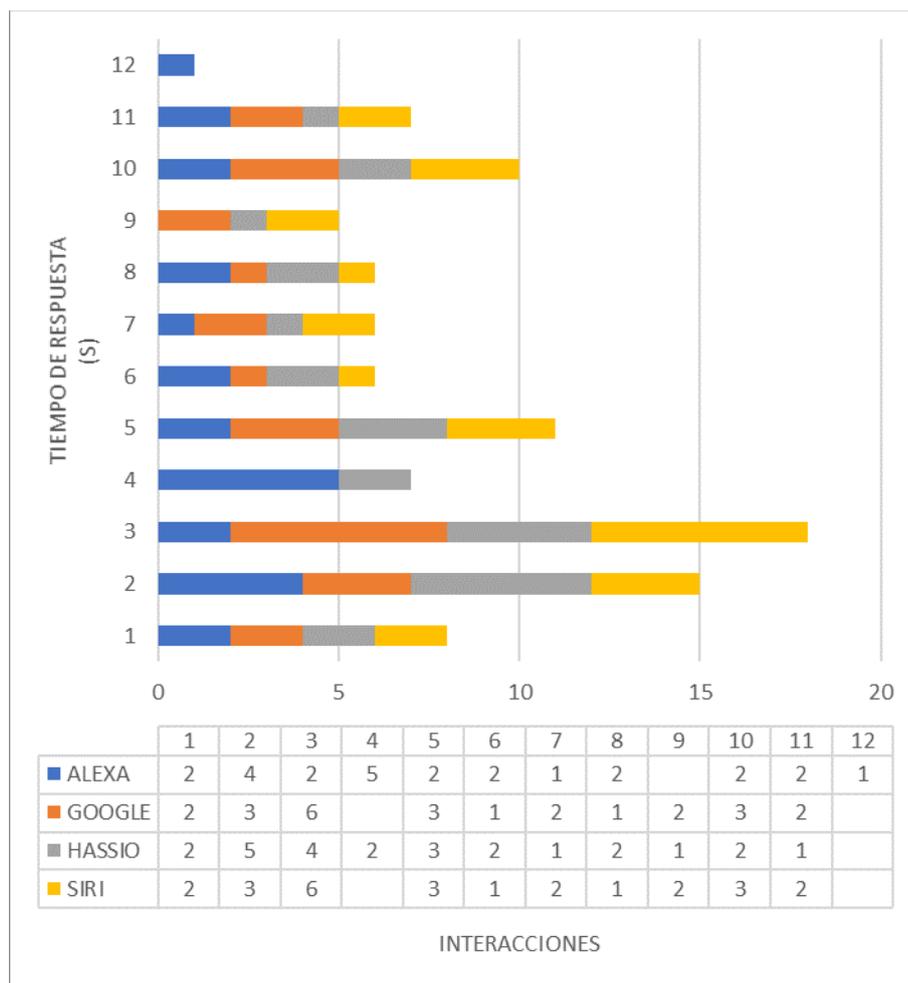


Figura 110. Resultados de Tiempo de respuesta del comando Apaga Sala de Cine (Escenario 2).

Teniendo en cuenta los planes de evaluación por control de voz e interfaz, como se detalla en el **Anexo 8** y en el **Anexo 9**, se observa que los criterios de evaluación no sobrepasaron los 10 segundos de promedio en su campo de optimización. Sin embargo, en los criterios de aceptación, todos los parámetros arrojaron resultados intermedios, a excepción de Intuitividad y Usabilidad, que presentaron algunos inconvenientes, así como en el primer escenario.

Las interacciones del público mediante comandos de voz se tornaron más complicadas que el uso por interfaz, especialmente en un entorno inestable. Los errores se relacionaron con el reconocimiento de voz por parte del asistente y la falta de conocimiento de comandos específicos, así como la correcta pronunciación. Por ende, el uso por interfaz resultó más fácil en este escenario.

En relación con las pruebas de Rendimiento de Voz y de Interfaz, detalladas en el **Anexo 10** y en el **Anexo 11**, se identificó que el promedio de tiempo de respuesta no superó los 9 segundos de la mayoría, y en otros se tuvo un promedio de 11 en un comando en tiempo de respuesta. La precisión del reconocimiento de voz se vio afectada al hablar de manera muy baja, generando confusiones

en los comandos. A diferencia de esto, el uso de interfaz mantuvo tiempos de respuesta por debajo de los 10 segundos, con una operación más sencilla para ejecutar cada comando en cada asistente. La inestabilidad del entorno podría haber contribuido al aumento en los tiempos de respuesta promedio, aunque se logró mantener la eficiencia en ambas modalidades de interacción, y se ejecutaron de manera correcta los comandos.

Por último, en este escenario se realizó un promedio de la latencia de cada uno de nuestros dispositivos esp32 y el dispositivo comercial, en el cual se ha dividido por asistentes virtuales y un promedio de tiempos de respuestas de todas las operaciones que se llevaron a cabo en cada uno, así como se puede observar en la **Tabla 19**, a continuación:

Tabla 19. Latencia de Dispositivos esp32 con protocolo Matter (Escenario 2).

Latencia Promedio				
Dispositivos	Asistentes			
	Alexa	Google	Hassio	Siri
ESP32 - S3 / Foco Led (Protocolo Matter)	8.03	8.08	7.66	8.09
ESP32 - C3 / Switch (Protocolo Matter)	7.64	7.53	8.18	7.94
ESP32 - C3 / Sensor de Temperatura (Protocolo Matter)	6.78	7.20	6.46	7.20
ESP32 - H2 / Foco Led (Protocolo Thread)	8.78	8.14	8.31	8.14
Foco Matter Comercial (Protocolo Matter)	6.98	7.15	7.04	7.17

Analizando, los totales de esta tabla se puede observar como los tiempos promedios de cada uno de los dispositivos matter personalizados no sobrepasan un promedio de respuesta de 7 u 8 segundos, dando para a que se puedan comparar entre un dispositivo matter personalizado, y un dispositivo matter comercial, ya que el comercial solo se mantiene en 7 segundos y que los personalizados suelen subir hasta más de 8 segundos, a continuación, se puede ver reflejado cada una de las interacciones que se realizaron en nuestros dispositivos.

Para nuestro dispositivo matter esp32 s3, se pudo visualizar que los tiempos de respuestas varían entre 1 a 12 segundos, pero también se vieron caídas de entre 14 a 69 segundos de responder a nuestro comando, es decir existían tareas que sobrepasaban el minuto dado a la inestabilidad de la red, los resultados se pueden ver en la **Figura 111**, a continuación:

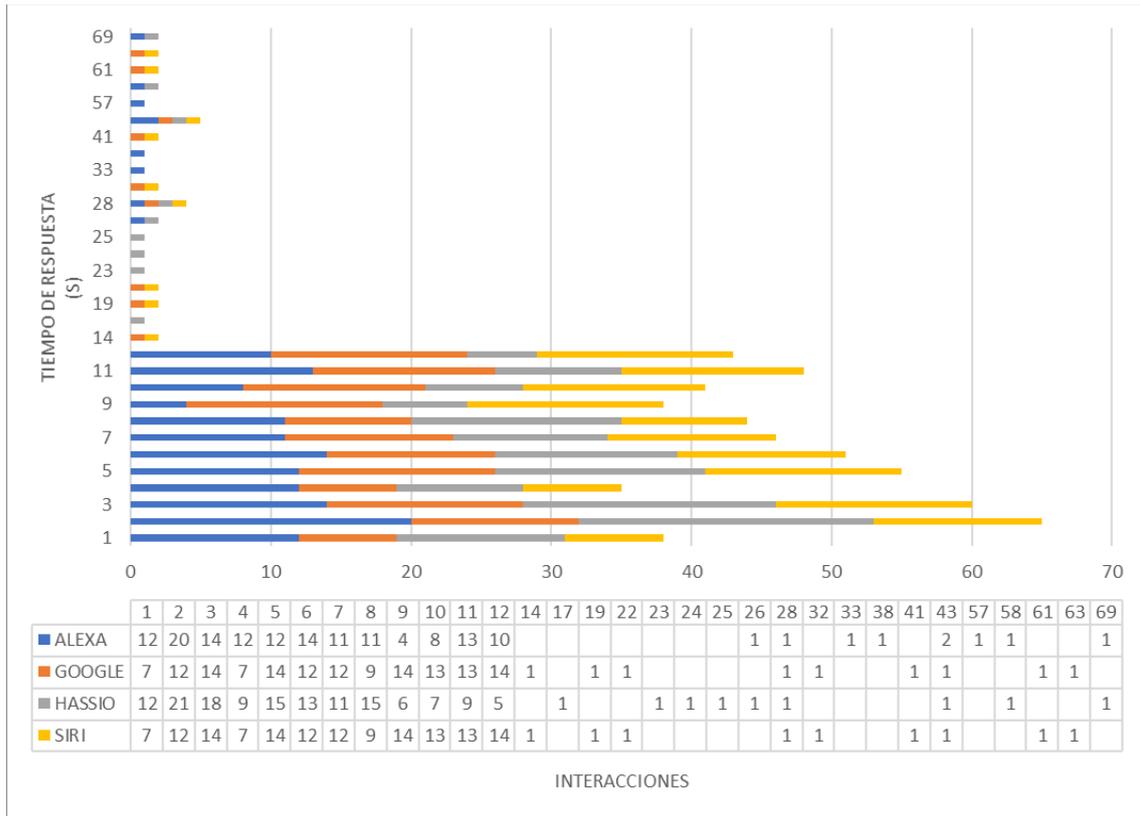


Figura 111. Tiempo de respuesta del dispositivo ESP32 S3 (Escenario 2).

Para nuestro dispositivo esp32 C3, encontramos que los tiempos de caídas, fueron muy pocos de entre 14 y 51 segundos, teniendo en cuenta que las interacciones fueron poquísimas las que estaban dentro de este rango, así como se ve en la **Figura 112**, a continuación:

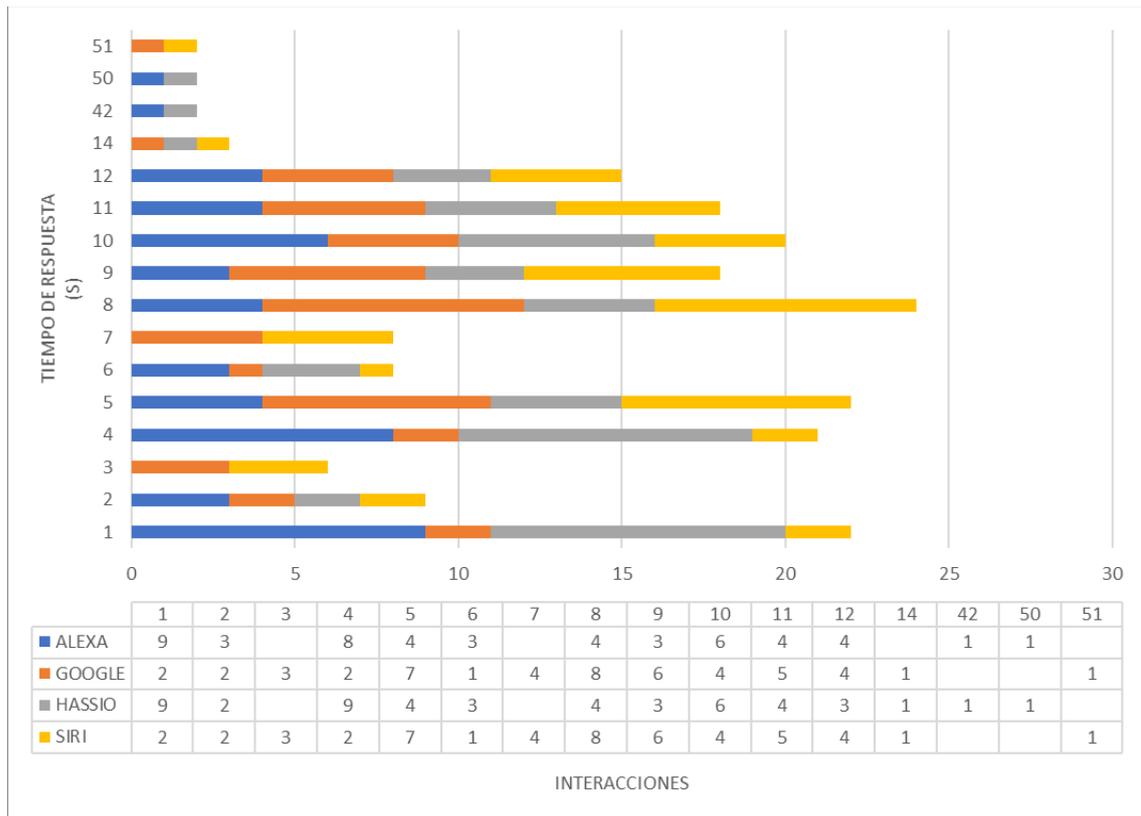


Figura 112. Tiempo de respuesta del dispositivo ESP32 C3 Switch (Escenario 2).

En nuestro dispositivo Esp32 C3 para el sensor de temperatura se tiene una respuesta en tiempo récord de entre 1 a 12 segundos, los cuales no sobrepasaron esos tiempos de respuesta de 12 segundos, en la **Figura 113**, se puede ver el resultado de las interacciones de nuestro sensor de temperatura:

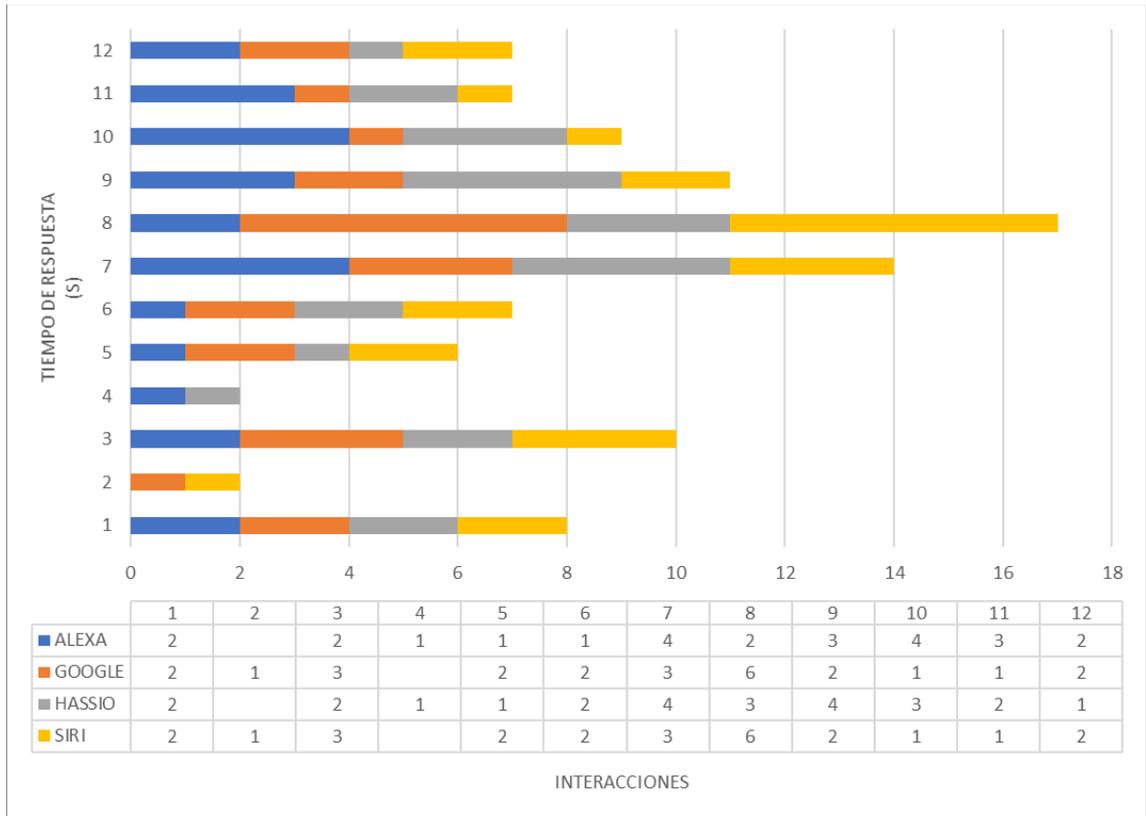


Figura 113. Tiempo de respuesta del dispositivo ESP32 C3 Sensor (Escenario 2).

Para nuestro dispositivo Thread, teniendo en cuenta que este comando primero debía pasar por nuestro Gateway para que se pudiera comunicar con el dispositivo Thread, se tiene de igual manera un resultado de tiempo de demora consideramos entre 26 y 69 segundos de retraso, así como se puede ver en la **Figura 114**:

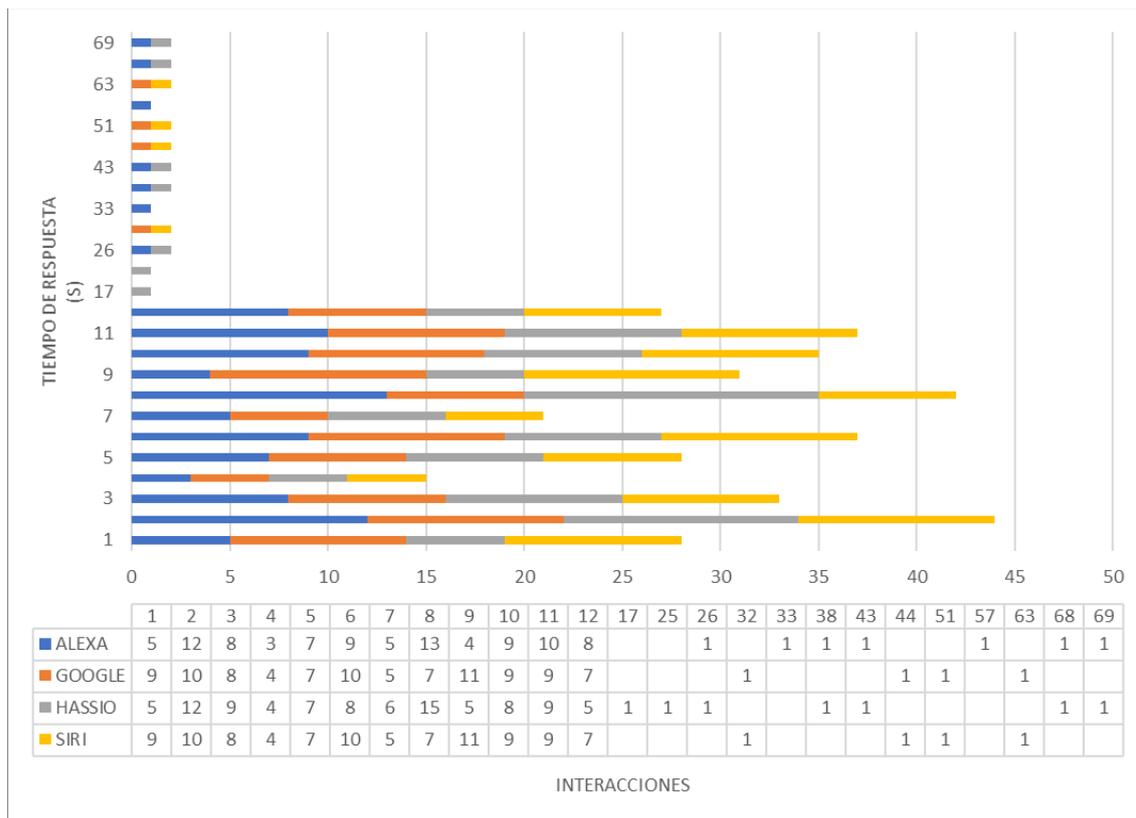


Figura 114. Tiempo de respuesta del dispositivo ESP32 H2 (Escenario 2).

Para concluir con la **Figura 115**, se puede ver que los tiempos de respuesta se vieron diversificadas como los demás dispositivos, siendo este un dispositivo comercial, y se puede concluir que los tiempos de respuestas más prolongados también sobrepasaron el minuto al igual que todos los demás dispositivos, en resumen se puede ver que la diferencia de respuesta en un escenario con una inestabilidad de internet es igual en la mayoría de dispositivos, ya sean personalizados y comerciales, y que los asistentes que tienen un poco más de promedio de tiempos prolongados fueron Hassio y Siri, pero no de una manera significativa, pero se debe tomar en cuenta este punto.

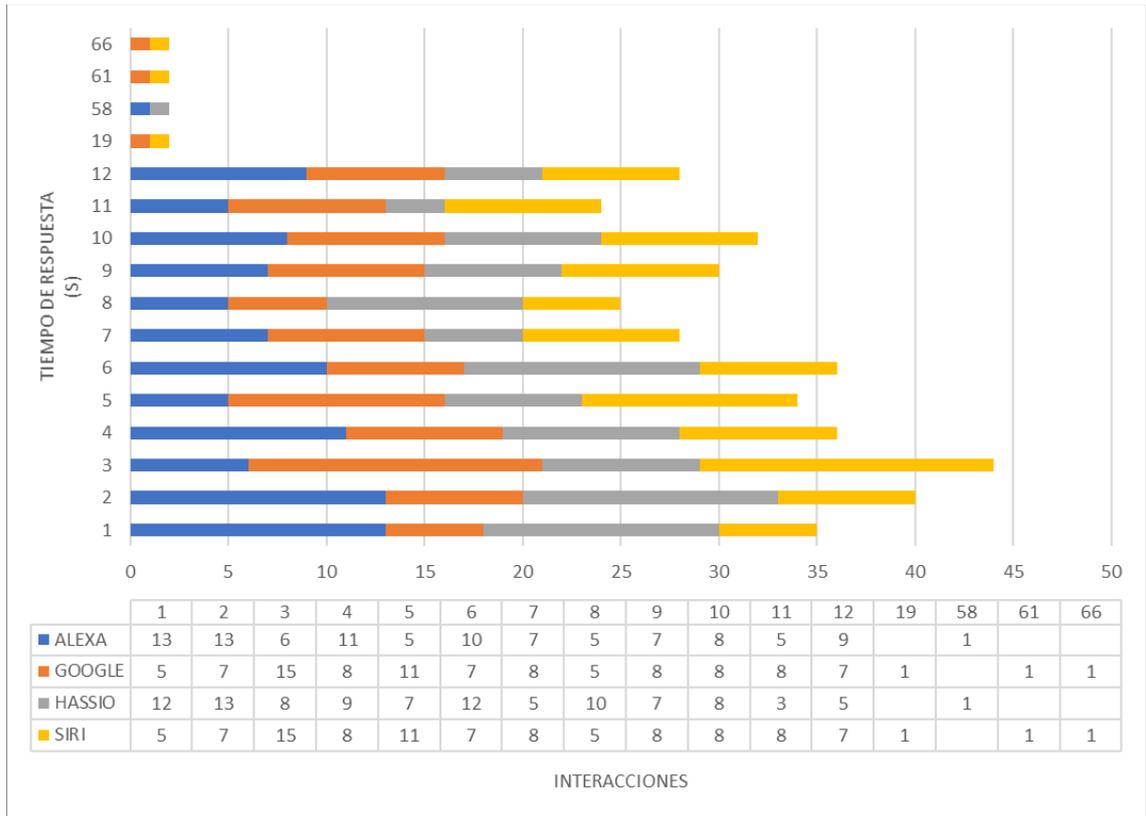


Figura 115. Tiempo de respuesta del dispositivo Matter comercial (Escenario 2).

CONCLUSIONES

La implementación del sistema domótico basado en el protocolo Matter ha demostrado ser factible y prometedora para lograr la integración y control centralizado de dispositivos comerciales y personalizados, en este caso se utilizó los dispositivos Esp32 los cuales dieron resultados positivos al ejecutarse el protocolo de comunicación asignado, además de que el Gateway funcionara de manera correcta cumpliendo el objetivo del dispositivo que era transmitir tarea ejecutada en un dispositivo matter para un dispositivo Thread, se obtuvo un tiempo de respuesta mejorado de 2 segundos en comparación con los 3 segundos previamente estimados.

La revisión bibliográfica proporcionó una comprensión profunda de los desafíos actuales en la domótica, y dio paso para la elección del protocolo Matter como un estándar potencial para superar estas limitaciones, teniendo en cuenta que la mayoría de información al ser un protocolo aún en desarrollo, se tiende a recoger información directa de los desarrolladores más no de artículos.

La implementación exitosa del firmware para dispositivos ESP32-S3, ESP32-H2 MINI U y ESP32-Border Router S3 – C3 ha sentado las bases tecnológicas necesarias para la funcionalidad Matter y teniendo en cuenta que el dispositivo H2 se usó para la implementación del protocolo Thread y se demostró la correcta interoperabilidad entre dispositivos.

Las estrategias aplicadas han resultado en una efectiva comunicación entre dispositivos comerciales, personalizados y dispositivos que cuentan con otro protocolo de comunicación, mejorando la cohesión del sistema domótico, luego de identificar los dos escenarios en donde el internet es inestable y estable, se identificó que los dispositivos cumplieron con sus objetivos y no se encontraron fallas ni respuestas nulas de ninguno.

La integración exitosa con Home Assistant ha mejorado significativamente el control y la conveniencia del sistema domótico. La plataforma sirve como un centro centralizado para gestionar dispositivos y automatizaciones, de todos nuestros dispositivos.

RECOMENDACIONES

Se toma en cuenta que en el escenario 2 dada la influencia de la inestabilidad de la red en los tiempos de respuesta, se obtuvieron interacciones mínimas, por tal razón se sugiere un enfoque continuo en la optimización del sistema para manejar variaciones en la calidad de la conexión, estrategias como la gestión de la retransmisión de paquetes pueden mejorar la robustez del sistema.

Implementar un sistema eficiente de actualizaciones de firmware y software a futuro, para los dispositivos Matter, ya que hasta la fecha se usó las librerías disponibles, pero se tiene conocimiento que estas quedaran obsoletas a futuro incluyendo más funcionalidades. Mantener el software actualizado garantiza no solo la corrección de posibles vulnerabilidades, sino también la incorporación de nuevas funcionalidades y mejoras de rendimiento.

Realizar evaluaciones periódicas del estado y la evolución del protocolo Matter. Dado que los estándares evolucionan, es fundamental adaptarse a las actualizaciones para garantizar la compatibilidad y aprovechar nuevas características.

Continuar realizando análisis de pruebas en diversos escenarios, especialmente aquellos que simulan condiciones del mundo real, como por ejemplo una empresa en donde los dispositivos no solo sean 4, sino que sean más de 50 o 100 dispositivos, incluyendo diferentes protocolos, además de Matter. Esto puede revelar áreas específicas que requieren atención y mejoras.

REFERENCIAS BIBLIOGRÁFICAS

- [1] L.-D. Liao *et al.*, «Design and Validation of a Multifunctional Android-Based Smart Home Control and Monitoring System», *IEEE Access*, vol. 7, pp. 163313-163322, 2019, doi: 10.1109/ACCESS.2019.2950684.
- [2] D. Belli, P. Barsocchi, y F. Palumbo, «Connectivity Standards Alliance Matter: State of the art and opportunities», *Internet Things*, vol. 25, p. 101005, abr. 2024, doi: 10.1016/j.iot.2023.101005.
- [3] A. Celik, I. Romdhane, G. Kaddoum, y A. M. Eltawil, «A Top-Down Survey on Optical Wireless Communications for the Internet of Things», *IEEE Commun. Surv. Tutor.*, vol. 25, n.º 1, pp. 1-45, 2023, doi: 10.1109/COMST.2022.3220504.
- [4] J. Du, K. F. Hew, y L. Liu, «What can online traces tell us about students' self-regulated learning? A systematic review of online trace data analysis», *Comput. Educ.*, vol. 201, 2023, doi: 10.1016/j.compedu.2023.104828.
- [5] O. Said, Y. Albagory, M. Nofal, y F. Al Raddady, «IoT-RTP and IoT-RTCP: Adaptive Protocols for Multimedia Transmission over Internet of Things Environments», *IEEE Access*, vol. 5, pp. 16757-16773, 2017, doi: 10.1109/ACCESS.2017.2726902.
- [6] S. Gøthesen, M. Haddara, y K. N. Kumar, «Empowering homes with intelligence: An investigation of smart home technology adoption and usage», *Internet Things*, vol. 24, p. 100944, dic. 2023, doi: 10.1016/j.iot.2023.100944.
- [7] N. E. Guevara, Y. H. Bolaños, J. P. Diago, y J. M. Segura, «Development of a low-cost IoT system based on LoRaWAN for monitoring variables related to electrical energy consumption in low voltage networks», *HardwareX*, vol. 12, p. e00330, oct. 2022, doi: 10.1016/j.ohx.2022.e00330.
- [8] C. Braghin, M. Lilli, y E. Riccobene, «A model-based approach for vulnerability analysis of IoT security protocols: The Z-Wave case study», *Comput. Secur.*, vol. 127, p. 103037, abr. 2023, doi: 10.1016/j.cose.2022.103037.
- [9] L. González, O. Sofía, D. Laguía, E. Gesto, y K. Hallar, «Internet del Futuro – Estudio de tecnologías IoT», *Inf. Científicos Téc. - UNPA*, vol. 12, n.º 3, Art. n.º 3, dic. 2020, doi: 10.22305/ict-unpa.v12.n3.744.
- [10] F. Khan, J. J. P. C. Rodrigues, y M. A. Jan, «Guest editorial: Network architectures and communication protocols for smart industrial IoT applications», *Digit. Commun. Netw.*, vol. 9, n.º 2, pp. 293-295, abr. 2023, doi: 10.1016/j.dcan.2023.03.004.
- [11] A. Feijoo-Añazco, D. Garcia-Carrillo, J. Sanchez-Gomez, y R. Marin-Perez, «Innovative security and compression for constrained IoT networks», *Internet Things*, vol. 24, p. 100899, dic. 2023, doi: 10.1016/j.iot.2023.100899.
- [12] H. U. Khan, M. Sohail, F. Ali, S. Nazir, Y. Y. Ghadi, y I. Ullah, «Prioritizing the multi-criterial features based on comparative approaches for enhancing security of IoT devices», *Phys. Commun.*, vol. 59, p. 102084, ago. 2023, doi: 10.1016/j.phycom.2023.102084.
- [13] 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.
- [14] T. Domínguez-Bolaño, O. Campos, V. Barral, C. J. Escudero, y J. A. García-Naya, «An overview of IoT architectures, technologies, and existing open-source projects», *Internet Things*, vol. 20, p. 100626, nov. 2022, doi: 10.1016/j.iot.2022.100626.
- [15] W. Kassab y K. A. Darabkh, «A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations», *J. Netw. Comput. Appl.*, vol. 163, p. 102663, ago. 2020, doi: 10.1016/j.jnca.2020.102663.

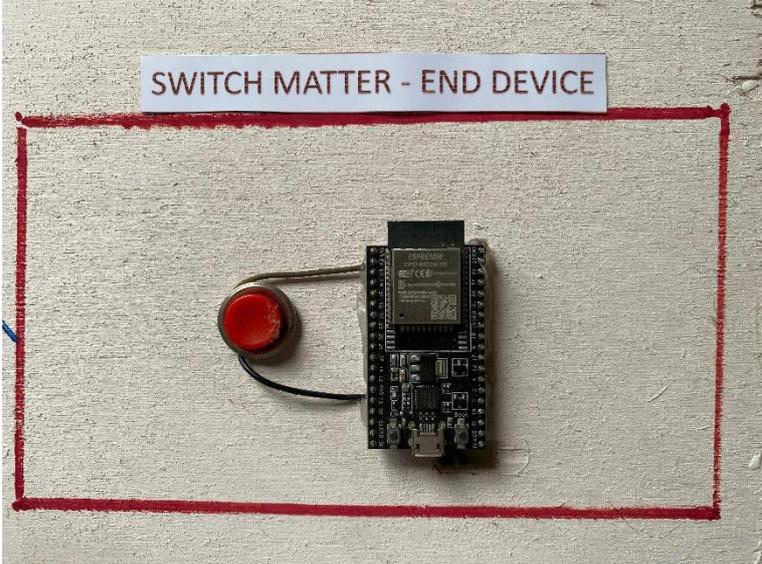
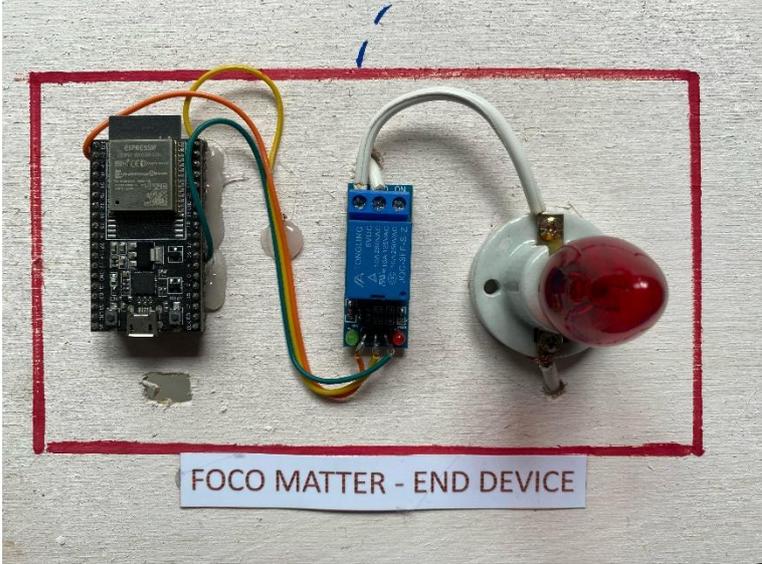
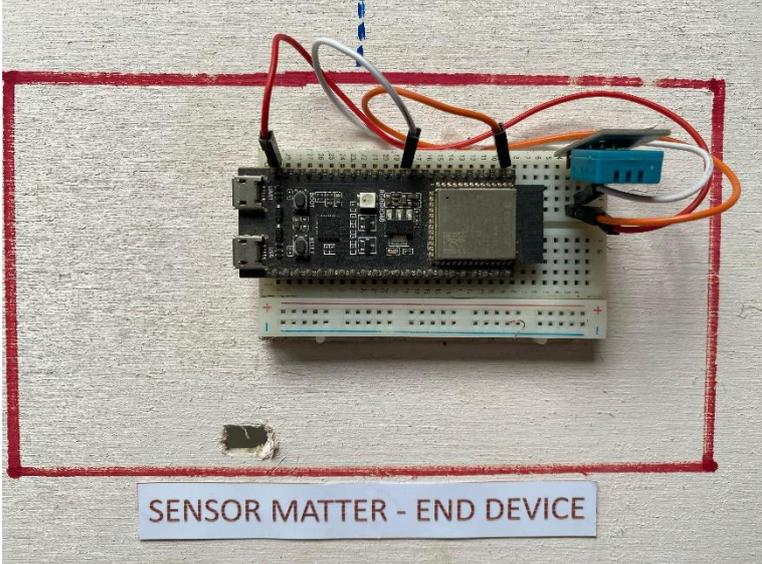
- [16] M. Masud, G. S. Gaba, P. Kumar, y A. Gurto, «A user-centric privacy-preserving authentication protocol for IoT-AMI environments», *Comput. Commun.*, vol. 196, pp. 45-54, dic. 2022, doi: 10.1016/j.comcom.2022.09.021.
- [17] T. S. Sandoval Ocando y B. A. Ordoñez, «Sistema de Control para la Automatización de una Vivienda Inteligente», *Télématique Rev. Electrónica Estud. Telemáticos*, vol. 18, n.º 2, pp. 40-56, 2019, Accedido: 6 de febrero de 2024. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=7889219>
- [18] P. Singh, Z. Elmi, V. Krishna Meriga, J. Pasha, y M. A. Dulebenets, «Internet of Things for sustainable railway transportation: Past, present, and future», *Clean. Logist. Supply Chain*, vol. 4, p. 100065, jul. 2022, doi: 10.1016/j.clscn.2022.100065.
- [19] A. Barua, M. A. Al Alamin, Md. S. Hossain, y E. Hossain, «Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey», *IEEE Open J. Commun. Soc.*, vol. 3, pp. 251-281, 2022, doi: 10.1109/OJCOMS.2022.3149732.
- [20] C. K. Nkuba, S. Woo, H. Lee, y S. Dietrich, «ZMAD: Lightweight Model-Based Anomaly Detection for the Structured Z-Wave Protocol», *IEEE Access*, vol. 11, pp. 60562-60577, 2023, doi: 10.1109/ACCESS.2023.3285476.
- [21] A. A. Eladl, M. A. Saeed, B. E. Sedhom, A. Y. Abdelaziz, y V. Bureš, «LoRaWAN-based IoT protocol for three levels central protection scheme in MT-HVDC networks with ANFIS-PSO restoration algorithm», *Energy Rep.*, vol. 10, pp. 4556-4573, nov. 2023, doi: 10.1016/j.egy.2023.11.013.
- [22] C. Bayılmış, M. A. Ebleme, Ü. Çavuşoğlu, K. Küçük, y A. Sevin, «A survey on communication protocols and performance evaluations for Internet of Things», *Digit. Commun. Netw.*, vol. 8, n.º 6, pp. 1094-1104, dic. 2022, doi: 10.1016/j.dcan.2022.03.013.
- [23] P. Locatelli, M. Perri, D. M. Jimenez Gutierrez, A. Lacava, y F. Cuomo, «Device discovery and tracing in the Bluetooth Low Energy domain», *Comput. Commun.*, vol. 202, pp. 42-56, mar. 2023, doi: 10.1016/j.comcom.2023.02.008.
- [24] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, y L. O. Mailloux, «The Z-Wave routing protocol and its security implications», *Comput. Secur.*, vol. 68, pp. 112-129, jul. 2017, doi: 10.1016/j.cose.2017.04.004.
- [25] Md. O. Qays, I. Ahmad, A. Abu-Siada, Md. L. Hossain, y F. Yasmin, «Key communication technologies, applications, protocols and future guides for IoT-assisted smart grid systems: A review», *Energy Rep.*, vol. 9, pp. 2440-2452, dic. 2023, doi: 10.1016/j.egy.2023.01.085.
- [26] Y. Li, J. Barthelemy, S. Sun, P. Perez, y B. Moran, «A Case Study of Wifi Sniffing Performance Evaluation», *IEEE Access*, vol. 8, pp. 129224-129235, 2020, doi: 10.1109/ACCESS.2020.3008533.
- [27] A. F. Ergan y K. Sarioglu, «Design and Application of a Novel Dimmable Lighting Control Unit with Active Power Measurement via KNX Protocol», presentado en 2021 13th International Conference on Electrical and Electronics Engineering, ELECO 2021, 2021, pp. 555-559. doi: 10.23919/ELECO54474.2021.9677882.
- [28] E. A. Pola Jiménez, M. López Sánchez, J. G. González Serna, N. González Franco, D. Mújica Vargas, y G. Santamaría Bonfil, «Análisis comparativo de variables utilizadas en redes distribuidas Zigbee para detección de intrusiones desde dispositivos finales en una red de vehículos autónomos no tripulados», *Rev. DYNA*, vol. 97, n.º 6, pp. 606-612, 2022, doi: <https://doi.org/10.6036/10620>.
- [29] A. Palaniappan, R. Muthiah, y M. Tiruchi Sundaram, «Zigbee enabled IoT based intelligent lane control system for autonomous agricultural electric vehicle application», *SoftwareX*, vol. 23, p. 101512, jul. 2023, doi: 10.1016/j.softx.2023.101512.
- [30] D. S. Pereira *et al.*, «Zigbee Protocol-Based Communication Network for Multi-Unmanned Aerial Vehicle Networks», *IEEE Access*, vol. 8, pp. 57762-57771, 2020, doi: 10.1109/ACCESS.2020.2982402.

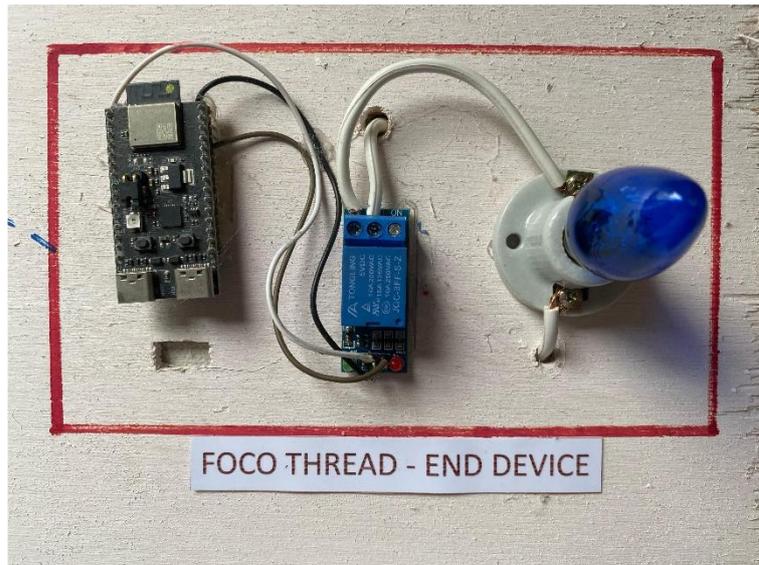
- [31] D. Gualpa Yumi, «Diseño e implementación de un sistema basado en equipos de tecnología LonWorks para la docencia de la Domótica», 2021, Accedido: 7 de febrero de 2024. [En línea]. Disponible en: <https://idus.us.es/handle/11441/115421>
- [32] A. Amjad, F. Azam, M. W. Anwar, y W. H. Butt, «A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT», *IEEE Access*, vol. 9, pp. 96528-96545, 2021, doi: 10.1109/ACCESS.2021.3094763.
- [33] P. K. Donta, S. N. Srirama, T. Amgoth, y C. S. R. Annavarapu, «Survey on recent advances in IoT application layer protocols and machine learning scope for research directions», *Digit. Commun. Netw.*, vol. 8, n.º 5, pp. 727-744, oct. 2022, doi: 10.1016/j.dcan.2021.10.004.
- [34] S. Elamanov, H. Son, B. Flynn, S. K. Yoo, N. Dilshad, y J. Song, «Interworking between Modbus and internet of things platform for industrial services», *Digit. Commun. Netw.*, oct. 2022, doi: 10.1016/j.dcan.2022.09.013.
- [35] G. Bouloukakis, N. Georgantas, P. Ntumba, y V. Issarny, «Automated synthesis of mediators for middleware-layer protocol interoperability in the IoT», *Future Gener. Comput. Syst.*, vol. 101, pp. 1271-1294, dic. 2019, doi: 10.1016/j.future.2019.05.064.
- [36] D. Fawzy, S. M. Moussa, y N. L. Badr, «An IoT-based resource utilization framework using data fusion for smart environments», *Internet Things*, vol. 21, p. 100645, abr. 2023, doi: 10.1016/j.iot.2022.100645.
- [37] K. W. Song y G. S. Choi, «Fieldbus based distributed servo control using LonWorks/IP gateway/web servers», *Mechatronics*, vol. 20, n.º 3, pp. 415-423, abr. 2010, doi: 10.1016/j.mechatronics.2010.02.008.
- [38] V. Graveto, T. Cruz, y P. Simões, «Security of Building Automation and Control Systems: Survey and future research directions», *Comput. Secur.*, vol. 112, p. 102527, ene. 2022, doi: 10.1016/j.cose.2021.102527.
- [39] H. Esquivel-Vargas, M. Caselli, y A. Peter, «BACGraph: Automatic Extraction of Object Relationships in the BACnet Protocol», en *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, jun. 2021, pp. 45-48. doi: 10.1109/DSN-S52858.2021.00029.
- [40] J. Koo y Y.-G. Kim, «Resource identifier interoperability among heterogeneous IoT platforms», *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, n.º 7, pp. 4191-4208, jul. 2022, doi: 10.1016/j.jksuci.2022.05.003.
- [41] «1. Introduction - ESP32-H2 - — Espressif's SDK for Matter latest documentation». Accedido: 13 de septiembre de 2023. [En línea]. Disponible en: <https://docs.espressif.com/projects/esp-matter/en/latest/esp32h2/introduction.html>
- [42] R. J. Rodríguez, S. Marrone, I. Marcos, y G. Porzio, «MOSTO: A toolkit to facilitate security auditing of ICS devices using Modbus/TCP», *Comput. Secur.*, vol. 132, p. 103373, sep. 2023, doi: 10.1016/j.cose.2023.103373.
- [43] J. T. Ventura, A. H. R. Puente, y J. R. H. García, «Rendimiento para la interoperabilidad entre Raspberry pi, ESP8266 y PLC con Node-RED para el IIoT», *Ingenius*, n.º 29, Art. n.º 29, ene. 2023, doi: 10.17163/ings.n29.2023.08.
- [44] J. S. Gómez Meza, S. V. Matute Arias, y T. J. Negrete Peña, «Diseño de un prototipo IoT para el monitoreo de material particulado en espacios reducidos utilizando ESP32 con servidor hospedado en la nube», *Sinerg. Académica*, vol. 4, n.º 3, pp. 1-18, 2021.
- [45] S. Raff, S. Rose, y T. Huynh, «Perceived creepiness in response to smart home assistants: A multi-method study», *Int. J. Inf. Manag.*, vol. 74, p. 102720, feb. 2024, doi: 10.1016/j.ijinfomgt.2023.102720.
- [46] W. Gao, X. Luo, Y. Liu, Y. Zhao, y Cui, «Development of an arduino-based integrated system for sensing of hydrogen peroxide», *Sens. Actuators Rep.*, vol. 3, p. 100045, nov. 2021, doi: 10.1016/j.snr.2021.100045.

- [47] A. Ohno, S. Matsumoto, M. Ohshita, y K. Kaida, «A learning support system of C programming language for novices as a Platform for Learning Analytics», en *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)*, sep. 2020, pp. 270-273. doi: 10.1109/IIAI-AAI50415.2020.00060.
- [48] «ESP32-S3 Development Kits - Espressif Systems | Mouser». Accedido: 12 de septiembre de 2023. [En línea]. Disponible en: <https://www.mouser.ec/new/espressif/espressif-esp32-s3-dev-kits/>
- [49] «esp32-s3_datasheet_en.pdf». Accedido: 26 de agosto de 2023. [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf
- [50] F. Souza, «ESP32-H2-DevKitM-1: Primeira placa da Espressif sem Wifi», Embarcados - Sua fonte de informações sobre Sistemas Embarcados. Accedido: 12 de septiembre de 2023. [En línea]. Disponible en: <https://embarcados.com.br/esp32-h2-devkitm-1-primeira-placa-da-espressif-sem-Wifi/>
- [51] «esp32-h2_datasheet_en.pdf». Accedido: 26 de agosto de 2023. [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32-h2_datasheet_en.pdf
- [52] W. A. Jabbar, T. Subramaniam, A. E. Ong, M. I. Shu'Ib, W. Wu, y M. A. de Oliveira, «LoRaWAN-Based IoT System Implementation for Long-Range Outdoor Air Quality Monitoring», *Internet Things*, vol. 19, p. 100540, ago. 2022, doi: 10.1016/j.iot.2022.100540.
- [53] Y. Yuanyuan y C. Meng, «The design of adaptive communication frame supporting high-speed transmission based on ModBus protocol», *Procedia Comput. Sci.*, vol. 183, pp. 551-556, ene. 2021, doi: 10.1016/j.procs.2021.02.096.
- [54] B. Mazon-Olivo y A. Pan, «Internet of Things: State-of-the-art, Computing Paradigms and Reference Architectures», *IEEE Lat. Am. Trans.*, vol. 20, n.º 1, pp. 49-63, ene. 2022, doi: 10.1109/TLA.2022.9662173.
- [55] D. Alulema, J. Criado, L. Iribarne, A. J. Fernández-García, y R. Ayala, «SI4IoT: A methodology based on models and services for the integration of IoT systems», *Future Gener. Comput. Syst.*, vol. 143, pp. 132-151, jun. 2023, doi: 10.1016/j.future.2023.01.023.
- [56] S. Wood, G. Michaelides, y C. Thomson, «Successful extreme programming: Fidelity to the methodology or good teamworking?», *Inf. Softw. Technol.*, vol. 55, n.º 4, pp. 660-672, abr. 2013, doi: 10.1016/j.infsof.2012.10.002.
- [57] V. Jovanovic, T. Murphy, y A. Greca, «Use of extreme programming (XP) in teaching introductory programming», en *32nd Annual Frontiers in Education*, nov. 2002, pp. F1G-. doi: 10.1109/FIE.2002.1158141.
- [58] P. Sharma y N. Hasteer, «Analysis of linear sequential and extreme programming development methodology for a gaming application», en *2016 International Conference on Communication and Signal Processing (ICCSP)*, abr. 2016, pp. 1916-1920. doi: 10.1109/ICCSP.2016.7754505.
- [59] G. Beniwal y A. Singhrova, «“A systematic literature review on IoT gateways”», *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, n.º 10, Part B, pp. 9541-9563, nov. 2022, doi: 10.1016/j.jksuci.2021.11.007.

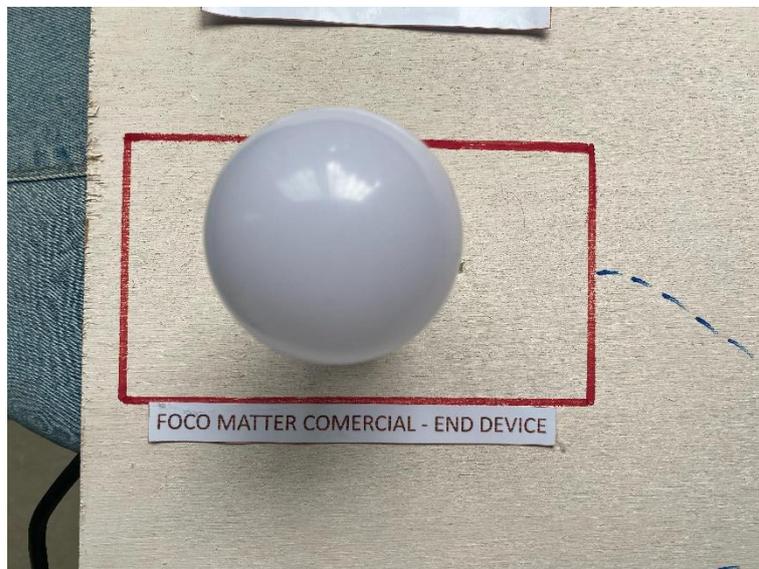
ANEXOS

Anexo 1 – Dispositivos ESP32 Terminados

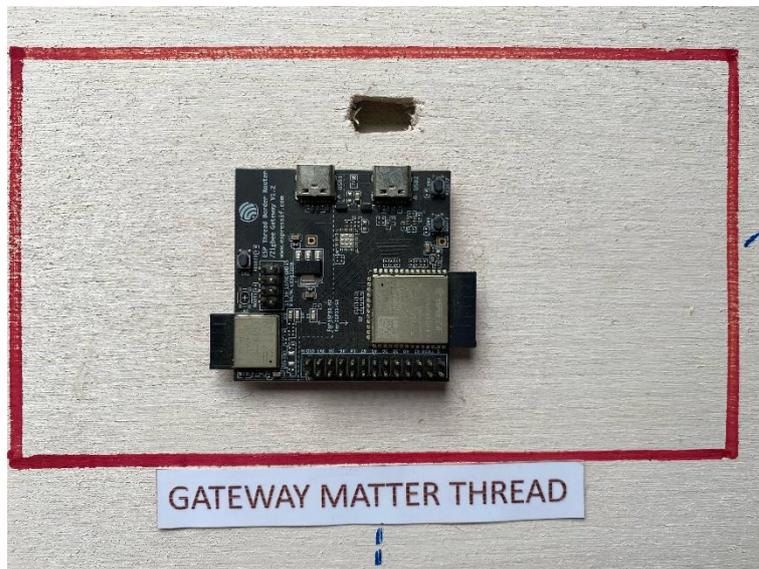




FOCO THREAD - END DEVICE



FOCO MATTER COMERCIAL - END DEVICE



GATEWAY MATTER THREAD

Anexo 2 - Observaciones de Evaluación Escenario 1

Observaciones de la evaluación					
	Tareas de evaluación por control por voz	Crterios de evaluación	Crterios de aceptación		
		Observaciones de tiempo de Respuesta	Funcionalidad	Interoperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	No se identificaron problemas, ya que la mayoría de los tiempos de respuesta fueron rápidos, oscilando entre 1 y 2 segundos, con un máximo de 3 segundos. Sin embargo, se registraron dos interacciones prolongadas de 5 segundos por parte de Hassio, lo cual podría ser motivo de preocupación.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Luz Thread	En la mayoría de las interacciones, se observó un tiempo de respuesta máximo de 3 segundos. Sin embargo, algunas interacciones presentaron tiempos de 4, 5 y 7 segundos. Se registró un máximo de 2 interacciones por segundo, provenientes de todos los asistentes, indicando que estos retrasos no son específicos de uno en particular.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Dame la Temperatura Matter	Se identificó un retraso de 9 segundos en la respuesta del asistente Siri durante una interacción, aunque no se encontraron problemas significativos aparte de este incidente puntual.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Foco Matter Comercial	No se registraron problemas significativos, ya que la mayoría de las interacciones tuvieron un tiempo de respuesta máximo de 1, 2 y 3 segundos. Sin embargo, se destacan dos interacciones en Hassio y Siri con un retraso de 4 segundos, una en Google con 8 segundos, y a través de la interfaz de Google se experimentó un retraso adicional de 15 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Switch Matter	En este contexto, los tiempos de respuesta fueron en su mayoría positivos, con solo algunas interacciones que experimentaron retrasos de 4, 5, 6 y 9 segundos. Es importante señalar que cada asistente tuvo solo una interacción con retraso, y no se observó una tendencia proporcional de retraso proveniente de algún asistente específico.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Luz Matter	No se detectaron problemas, excepto por una interacción de 9 segundos de tiempo de respuesta proveniente de Alexa en el modo de interfaz y otra de 4 segundos en Hassio.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Switch Matter	No se encontró ningún problema, teniendo solo un par de interacciones de 5 y 4 segundos de retraso uno por interacción.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Foco Matter Comercial	No se identificaron problemas, excepto por una única interacción de 5 segundos de retraso por parte de Google y ninguna otra incidencia.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Luz Thread	La mayoría de las interacciones tuvieron tiempos de respuesta perfectos, con la excepción de una única interacción prolongada de 7 segundos por parte de Google.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende la Sala (Matter Personalizada - Thread)	Los tiempos de respuesta fueron excelentes en ambos dispositivos. Sin embargo, se observaron algunas interacciones con tiempos de respuesta de 4 y 7 segundos. Estos retrasos se debieron a la secuencia en la que se encendieron los dispositivos, con una separación de 2 o 3 segundos entre ellos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Sala de Cine (Matter Personalizada - Matter Comercial)	No se encontró ningún problema los dos servicios iniciaron al mismo tiempo, pero hubo dos tiempos de respuesta de 7 y 9 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga la Sala (Matter Personalizada - Thread)	Aquí es notable destacar una prolongación de 13 segundos en el tiempo de respuesta, proveniente de Alexa. Este retraso se originó al apagar la sala donde se encontraba el dispositivo Matter personalizado. Es importante señalar que el dispositivo Matter comercial fue el que demoró en encender, lo cual puede influir en la demora mencionada anteriormente.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Sala de Cine (Matter Personalizada - Matter Comercial)	Aquí es notable destacar una prolongación, pero esta vez de 12 segundos en el tiempo de respuesta, proveniente de Alexa. Este retraso se originó al apagar la sala de cine donde se encontraba el dispositivo Matter personalizado y el comercial. Es importante señalar que el dispositivo Matter comercial fue el que demoró en encender, lo cual puede influir en la demora mencionada anteriormente.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema

Anexo 3 – Plantilla de Evaluación por voz terminada (Escenario 1)

Evaluación por control por voz												
	Tareas de evaluación por control por voz	Criterios de evaluación				Criterios de aceptación						
		Compatibilidad	Usabilidad		Optimización	Diagnóstico y Resolución	Funcionalidad	Accesibilidad	Intuitividad	Fiabilidad	Interoperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Enciende Luz Thread	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Dame la Temperatura Matter	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Eenciende Foco Matter Comercial	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Enciende Switch Matter	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Apaga Luz Matter	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Apaga Switch Matter	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Apaga Foco Matter Comercial	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Apaga Luz Thread	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	✔	⚠	Intermedio	2 Segundos	✔	✔	✔	⚠	✔	✔	✔
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	✔	⚠	Intermedio	2 Segundos	✔	✔	✔	⚠	✔	✔	✔
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✔	✔	Fácil	2 Segundos	✔	✔	✔	✔	✔	✔	✔

Anexo 4 – Plantilla de Evaluación por Interfaz terminada (Escenario 1)

		Evaluación por Interfaz									
Tareas de evaluación por control por voz	Criterios de evaluación				Criterios de aceptación						
	Compatibilidad	Usabilidad	Optimización	Diagnóstico y Resolución	Funcionalidad	Accesibilidad	Intuitividad	Fiabilidad	Interoperabilidad	Eficiencia	
Asistentes virtuales	Enciende Luz matter	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende Luz Thread	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Dame la Temperatura Matter	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende Foco Matter Comercial	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende Switch Matter	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Luz Matter	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Switch Matter	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Foco Matter Comercial	✓	✓ Fácil	1 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Luz Thread	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✓	✓ Fácil	2 Segundos	✓	✓	✓	✓	✓	✓	✓

Anexo 5 – Evaluación del rendimiento por Voz (Escenario 1)

Evaluación del rendimiento por Voz						
	Tareas de evaluación por control por voz	Velocidad de respuesta	Precisión de reconocimiento de voz	Capacidad de procesamiento	Fiabilidad	Interoperabilidad
Asistentes Virtual	Enciende Luz matter	2.0 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Luz Thread	1.9 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Dame la Temperatura Matter	2.1 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Foco Matter Comercial	2.0 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Switch Matter	1.6 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Matter	1.6 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Switch Matter	1.8 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Foco Matter Comercial	1.6 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Thread	1.8 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	1.7 Segundos	Alto	Enciende un foco luego del otro con un retraso de 1 segundo	Alto	Si
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	2.1 Segundos	Medio	Enciende un foco luego del otro con un retraso de 1 segundo	Medio	Si
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	1.8 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	1.8 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si

Anexo 6 – Evaluación del rendimiento por Interfaz (Escenario 1)

Evaluación del rendimiento por Interfaz						
	Tareas de evaluación por control por voz	Velocidad de respuesta	Precisión de reconocimiento de voz	Capacidad de procesamiento	Fiabilidad	Interoperabilidad
Asistentes Virtual	Enciende Luz matter	2.1 Segundos	Medio	No se puede solicitar más de una tarea simultánea	Medio	Si
	Enciende Luz Thread	1.9 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Dame la Temperatura Matter	2.4 Segundos	Medio	No se puede solicitar más de una tarea simultánea	Medio	Si
	Enciende Foco Matter Comercial	1.8 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Switch Matter	1.5 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Matter	2.0 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Switch Matter	1.9 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Foco Matter Comercial	1.5 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Thread	1.6 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	1.8 Segundos	Alto	Enciende un foco luego del otro con un retraso de 1 segundo	Alto	Si
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	1.6 Segundos	Perfecto	Enciende un foco luego del otro con un retraso de 1 segundo	Perfecto	Si
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	1.9 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	1.7 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si

Anexo 7 - Observaciones de Evaluación Escenario 2

Observaciones de la evaluación					
	Tareas de evaluación por control por voz	Crterios de evaluación	Crterios de aceptación		
		Observaciones de tiempo de Respuesta	Funcionalidad	Interoperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	En esta fase de interacciones, considerando la inestabilidad de la red, se observaron tiempos de retraso que oscilaron entre 5 y 12 segundos. Sin embargo, también se registraron retrasos más prolongados, que variaron desde 14 hasta 43 segundos. Los retrasos más significativos fueron de 41, 43, y 53 segundos, que fueron la mayoría por Hassio, teniendo en cuenta que solo fue una interacción por los diferentes segundos que se mencionaron.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Luz Thread	Sus tiempos de respuestas, al igual que los demás, no significan un obstáculo para poder realizar la tarea solicitada, pero también se obtuvieron, respuestas que superaron los 44 segundos y un tiempo de respuesta de 68 segundos, pero por parte de todo HASSIO.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Dame la Temperatura Matter	Al ejecutar este comando y realizar estas interacciones, se identificaron que se tenía una respuesta de entre los 5 y 12 segundos de respuestas, y solo se obtuvo una respuesta de 27 segundos de retraso por parte de Google y Siri	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Foco Matter Comercial	Los retrasos que se obtuvieron, al igual que al apagar el dispositivo, se obtuvieron máximos de 12 segundos de respuesta, pero si se tuvo interacciones que superaron el minuto, y otros que se mantenían, como el de 59 segundos por parte de Hassio, y otro de 66 por parte de Siri	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Switch Matter	las interacciones no superaban los 13 segundos, pero se tenían diferentes interacciones de manera rápida, dando paso a que se realizaba la tarea más rápido teniendo igual una conexión con la red baja, pero si se tuvo registro de un par de retrasos al ejecutarse el comando por voz que se trató de 51 segundos por parte de HASSIO y Alexa	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Luz Matter	No se tuvo problemas algunos, ya que se tenían tiempos de respuestas entre los 12 segundos como máximo, pero sí se tuvo una respuesta larga que fue de 66 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Switch Matter	Los retrasos oscilaron entre 3 y 10 segundos en las diversas interacciones. Sin embargo, se identificaron dos interacciones prolongadas que superaron el minuto de espera.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Foco Matter Comercial	Los tiempos de respuesta generalmente se mantuvieron por debajo de los 12 segundos, con algunas respuestas rápidas de 1 segundos o menos en ocasiones. Sin embargo, se registró una interacción excepcional que superó el minuto, con un tiempo de respuesta de 61 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Luz Thread	No se identificaron problemas, aunque se observaron retrasos que variaron entre 1 hasta los 12 segundos. Cabe destacar que sí hubo así los mismos retrasos en algunas interacciones que rondaban entre los 30 segundos hasta uno que sobrepasa el minuto, que fue en los asistentes de Google y SIRI.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende la Sala (Matter Personalizada - Thread)	hubo cargas de tiempos de retraso ese entre 26 segundos hasta más de 1 minuto, por parte del uso de los comandos por medio de voz, en la que rondaban entre los 32, 43, 57, 63, y 69 segundos de retraso. Siendo solo una interacción por asistentes, en los diferentes casos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Enciende Sala de Cine (Matter Personalizada - Matter Comercial)	Las interacciones se mantuvieron dentro de un límite de 12 segundos de retraso. Sin embargo, se identificaron muy poca interacción, viniendo por parte de todos los asistentes virtuales y teniendo un tiempo de retraso de 58 segundos y el otro de 61 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga la Sala (Matter Personalizada - Thread)	En este escenario, los tiempos prolongados no superaban los 12 segundos hasta que se obtuvieron algunas interacciones con un tiempo de retraso de 33 segundos y de 56 segundos viniendo de las interacciones que se realizaron con HASSIO.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema
	Apaga Sala de Cine (Matter Personalizada -Matter Comercial)	Los tiempos de respuestas fueron de entre un total de 1 a 12 segundos, siendo muy pocos los tiempos de respuestas de 1 segundo y siendo más los de 6 a 10 segundos.	No se encontró ningún problema	No se encontró ningún problema	No se encontró ningún problema

Anexo 8 - Plantilla de Evaluación por voz terminada (Escenario 2)

Evaluación por control por voz													
	Tareas de evaluación por control por voz	Criterios de evaluación				Criterios de aceptación							
		Compatibilidad	Usabilidad	Optimización	Diagnóstico y Resolución	Funcionalidad	Accesibilidad	Intuitividad	Fiabilidad	Seguridad	Inte roperabilidad	Eficiencia	
Asistentes virtuales	Enciende Luz matter	⚠	✅	Fácil	10 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Enciende Luz Thread	⚠	✅	Fácil	9 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Dame la Temperatura Matter	⚠	✅	Fácil	7 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Eenciende Foco Matter Comercial	⚠	✅	Fácil	8 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Enciende Switch Matter	⚠	✅	Fácil	8 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Apaga Luz Matter	⚠	✅	Fácil	8 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Apaga Switch Matter	⚠	✅	Fácil	8 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Apaga Foco Matter Comercial	✅	✅	Fácil	7 Segundos	✅	✅	✅	✅	✅	✅	✅	✅
	Apaga Luz Thread	⚠	✅	Fácil	7 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	⚠	⚠	Intermedio	11 Segundos	⚠	⚠	⚠	⚠	✅	✅	✅	✅
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	⚠	✅	Fácil	8 Segundos	⚠	⚠	⚠	✅	✅	✅	✅	✅
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	⚠	⚠	Intermedio	8 Segundos	⚠	⚠	⚠	⚠	✅	✅	✅	✅
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✅	✅	Fácil	5 Segundos	✅	✅	✅	✅	✅	✅	✅	✅

Anexo 9 - Plantilla de Evaluación por Interfaz terminada (Escenario 2)

Evaluación por Interfaz												
	Tareas de evaluación por control por voz	Criterios de evaluación			Criterios de aceptación							
		Compatibilidad	Usabilidad	Optimización	Diagnóstico y Resolución	Funcionalidad	Accesibilidad	Intuitividad	Fiabilidad	Seguridad	Inte roperabilidad	Eficiencia
Asistentes virtuales	Enciende Luz matter	⚠	✔ Fácil	8 Segundos	⚠	⚠	⚠	✔	✔	✔	✔	✔
	Enciende Luz Thread	⚠	✔ Fácil	7 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔
	Dame la Temperatura Matter	✔	✔ Fácil	7 Segundos	✔	✔	✔	✔	✔	✔	✔	✔
	Eenciende Foco Matter Comercial	⚠	✔ Fácil	8 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔
	Enciende Switch Matter	⚠	✔ Fácil	8 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔
	Apaga Luz Matter	⚠	✔ Fácil	10 Segundos	⚠	⚠	⚠	✔	✔	✔	✔	✔
	Apaga Switch Matter	✔	✔ Fácil	6 Segundos	✔	✔	✔	✔	✔	✔	✔	✔
	Apaga Foco Matter Comercial	✔	✔ Fácil	7 Segundos	✔	✔	✔	✔	✔	✔	✔	✔
	Apaga Luz Thread	⚠	✔ Fácil	10 Segundos	⚠	⚠	⚠	✔	✔	✔	✔	✔
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	⚠	✔ Fácil	8 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	✔	✔ Fácil	7 Segundos	✔	✔	✔	✔	✔	✔	✔	✔
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	⚠	✔ Fácil	7 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	⚠	✔ Fácil	7 Segundos	✔	⚠	⚠	✔	✔	✔	✔	✔

Anexo 10 – Evaluación del rendimiento por Voz (Escenario 2)

Evaluación del rendimiento por Voz						
Asistente Virtual	Tareas de evaluación por control por voz	Velocidad de respuesta	Precisión de reconocimiento de voz	Capacidad de procesamiento	Fiabilidad	Interoperabilidad
	Enciende Luz matter	9.5 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Luz Thread	9.4 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Dame la Temperatura Matter	7.2 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Foco Matter Comercial	7.6 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Switch Matter	7.9 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Luz Matter	8.2 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Switch Matter	8.3 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Foco Matter Comercial	6.6 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Thread	7.3 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	11.0 Segundos	Alto	Enciende un foco luego del otro con un retraso de 1 segundo	Alto	Si
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	8.3 Segundos	Alto	Enciende un foco luego del otro con un retraso de 1 segundo	Alto	Si
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	7.8 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	5.5 Segundos	Perfecto	Realiza dos Tareas al mismo Tiempo	Perfecto	Si

Anexo 11 - Evaluación del rendimiento por Interfaz (Escenario 2)

Evaluación del rendimiento por Voz						
Asistente Virtual	Tareas de evaluación por control por voz	Velocidad de respuesta	Precisión de reconocimiento de voz	Capacidad de procesamiento	Fiabilidad	Interoperabilidad
	Enciende Luz matter	8.3 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Luz Thread	7.4 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Dame la Temperatura Matter	6.8 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Enciende Foco Matter Comercial	7.8 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende Switch Matter	8.0 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Luz Matter	9.5 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Apaga Switch Matter	6.2 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Foco Matter Comercial	6.9 Segundos	Perfecto	No se puede solicitar más de una tarea simultánea	Perfecto	Si
	Apaga Luz Thread	10.1 Segundos	Alto	No se puede solicitar más de una tarea simultánea	Alto	Si
	Enciende la Sala (Luz Matter Personalizada - Foco Thread)	7.6 Segundos	Alto	Enciende un foco luego del otro con un retraso de 1 segundo	Alto	Si
	Enciende Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	6.8 Segundos	Perfecto	Enciende un foco luego del otro con un retraso de 1 segundo	Perfecto	Si
	Apaga la Sala (Luz Matter Personalizada - Foco Thread)	7.1 Segundos	Perfecto	Realiza dos Tareas al mismo Tiempo	Perfecto	Si
	Apaga Sala de Cine (Luz Matter Personalizada - Foco Matter Comercial)	7.2 Segundos	Alto	Realiza dos Tareas al mismo Tiempo	Alto	Si