

Instituto Tecnológico de Costa Rica

Carrera Ingeniería Mecatrónica



Diseño de un sistema para el control de sensores y dispositivos
inteligentes orientado a su uso en entornos educativos

Informe de Proyecto de Graduación para optar por el título de Ingeniero
en Mecatrónica con el grado académico de Licenciatura

Sebastián García Monge

Cartago, noviembre de 2021



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 24 de noviembre 2021

Sebastián García

Sebastián García Monge

Céd: 117050077

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Sebastián García Monge

Proyecto: Diseño de un sistema para el control de sensores y dispositivos inteligentes orientado a su uso en entornos educativos.

JUAN LUIS
CRESPO
MARIÑO (FIRMA)



Firmado digitalmente
por JUAN LUIS CRESPO
MARIÑO (FIRMA)
Fecha: 2021.11.25
07:47:24 -06'00'

Dr. -Ing. Juan Luis Crespo Mariño

Asesor

Cartago, 24 de noviembre 2021

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Sebastián García Monge

Proyecto: Diseño de un sistema para el control de sensores y dispositivos inteligentes orientado a su uso en entornos educativos.

Miembros del jurado evaluador

FELIPE GERARDO MEZA
OBANDO (FIRMA)

Digitally signed by FELIPE
GERARDO MEZA OBANDO (FIRMA)
Date: 2021.11.25 15:05:33 -06'00'

MSc. -Ing. Felipe Meza Obando

Jurado

IVAN ARAYA
MENESES (FIRMA)

Firmado digitalmente por IVAN
ARAYA MENESES (FIRMA)
Fecha: 2021.11.24 22:44:09
-06'00'

MSc. -Ing. Iván Araya Meneses

Jurado

MARTA EUGENIA
VILCHEZ MONGE
(FIRMA)

Firmado digitalmente
por MARTA EUGENIA
VILCHEZ MONGE (FIRMA)
Fecha: 2021.11.25
15:17:13 -06'00'

MSc. -Fís. Marta Vilchez Monge

Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 24 de noviembre 2021

Resumen

El proyecto surge de la necesidad del Grupo Integrado de Ingeniería (GII) de la Universidade da Coruña de impartir un curso sobre el Internet de las Cosas (IoT) a nivel colegial, y para ello contar con el diseño de una plataforma que permita el control de dispositivos inteligentes y a su vez la integración con el Robobo Framework. Por medio del proceso de diseño en ingeniería, se analizan las necesidades del cliente y se realiza un diseño que implica la utilización de Home Assistant, una plataforma Open Source para el control de entornos IoT. De esta manera, se establece un entorno con múltiples dispositivos inteligentes (de tipo luz, smart plugs, sensores y cámaras) conectados por medio de la plataforma, y además se diseña una aplicación en Android Studio que permite establecer una comunicación efectiva con Home Assistant y a su vez ser integrada al Robobo Framework. Se valida por medio de pruebas de confiabilidad de dispositivos integrados a la plataforma, y se realiza un análisis económico sobre los recursos implicados. Se concluye que el sistema permite cumplir con necesidades en aspectos como confiabilidad y facilidad de uso, y se provee al GII con una herramienta para la integración a su Framework.

Palabras clave: Home Assistant, dispositivos inteligentes, internet de las cosas, protocolos de comunicación, Robobo Framework, sensores

Abstract

The Project is created based on the necessity of the Coruña University's Grupo Integrado de Ingeniería (GII) to teach a high school level course on the Internet of Things (IoT), and for that they require a design for a smart device control platform, along with an integration for the Robobo Framework. Through the engineering design process, the client's needs are analyzed, and a design is proposed implementing Home Assistant, an Open Source control platform for IoT environments. This way, an environment is set up with multiple smart devices (such as lights, smart plugs, sensors, and cameras) connected through the platform, and additionally an Android Studio app is designed specifically for communicating effectively with Home Assistant and at the same time be integrated with the Robobo Framework. The project is validated reliability tests on integrated devices, and an economic analysis is conducted for the project's resources. It is concluded that the solution satisfies the established necessities in aspects such as reliability and intuitive interaction, and that the GII is provided with a tool capable of integrating with their Framework.

Keywords: Communication protocols, Home assistant, internet of things, Robobo Framework, smart devices, sensors

AGRADECIMIENTOS

Primero quiero agradecer a mi familia, por el apoyo y motivación constante. A mi madre y padres por todas sus enseñanzas y esfuerzos, que me han ayudado a alcanzar mis metas. Gracias por su apoyo incondicional, gracias a ustedes he llegado a este tan esperado momento de mi carrera.

Seguidamente al Dr. Francisco Bellas Bouza y al Dr. Alejandro Paz López, quienes me abrieron las puertas para colaborar con su proyecto, y me proporcionaron con guía, recursos y su tiempo para poder concretar el proyecto de la mejor manera.

Al Ing. Ronald Loaiza Baldares y al Dr. Juan Luis Crespo Mariño, cuya guía, paciencia y constante apoyo me ayudaron a lograr este objetivo, además de hacer de este semestre una gran experiencia.

Finalmente quiero agradecer a mis amigos y amigas, a todas aquellas personas que han estado en mi vida en estos últimos seis años, que han aportado experiencias, motivación, enseñanzas, risas y apoyo. No cambiaría estas experiencias por nada.

LISTA DE CONTENIDOS

Lista de Figuras	iv
Lista de Tablas	vii
Lista de Acrónimos	x
1 Introducción	1
1.1 Objetivos	3
1.1.1 Objetivo general	3
1.1.2 Objetivos específicos	4
1.2 Estructura del documento	4
2 Marco Teórico	6
2.1 Internet de las cosas	6
2.1.1 IoT en la educación	7
2.2 Protocolos de comunicación	8
2.2.1 Wi-fi	9
2.2.2 Zigbee	9
2.2.3 Bluetooth	10
2.2.4 Z-Wave	10
2.2.5 Conceptos relacionados para comunicaciones IoT	11
2.3 Herramientas IoT	13
2.3.1 Google Home	13
2.3.2 OpenHAB	13
2.3.3 Tuya Smart	14
2.3.4 Microcontroladores	15
2.3.5 Home Assistant	15
2.3.6 Amazon Web Services	16
2.4 Proyecto Robobo	16
2.4.1 Robobo Framework	17

2.5	Validación de Pruebas	18
2.5.1	Análisis cualitativo	18
2.5.2	Análisis cuantitativo	20
3	Metodología	22
3.1	Metodología de diseño	22
3.1.1	Determinación de necesidades y especificaciones	23
3.1.2	Generación de conceptos	24
3.1.3	Selección del concepto	25
3.1.4	Desarrollo del concepto	26
3.1.5	Pruebas de concepto	26
3.2	Entregables del proyecto	26
4	Diseño de Plataforma	28
4.1	Identificación de requerimientos	28
4.2	Selección de plataforma	30
4.2.1	Selección inicial	31
4.2.2	Iteración del proceso	36
4.2.3	Selección final	37
4.3	Selección de componentes	43
4.4	Solución por implementar	44
5	Implementación de Plataforma	47
5.1	Aplicación piloto	47
5.1.1	Instalación de Home Assistant	48
5.1.2	Conexión del dispositivo	50
5.1.3	Validación de aplicación piloto	52
5.2	Creación de entorno completo	53
5.2.1	Dispositivos de luz	55
5.2.2	Conectores (<i>smart plugs</i>)	56
5.2.3	Sensores	57
5.2.4	Cámaras	59
5.2.5	Parlantes	61
6	Integración con el Framework	63
6.1	Métodos de integración	63
6.2	Diseño de aplicación	64
6.3	Comprobación de comunicación	68
6.3.1	Implementación de un proceso automatizado	70
7	Validación del sistema	72
7.1	Definición de pruebas	72
7.1.1	Repetibilidad de conexión	72
7.1.2	Repetibilidad de lectura	73
7.1.3	Repetibilidad de actuación	73
7.1.4	Percepción de complejidad	74

7.2	Resultados y análisis	75
7.2.1	Repetibilidad de lectura, escritura y conexión en <i>Home Assistant</i>	76
7.2.2	Comunicación con la aplicación (prueba de concepto para el <i>Robobo Framework</i>)	80
7.2.3	Percepción de complejidad	82
7.2.4	Otras necesidades	88
8	Análisis económico	89
9	Conclusiones	93
9.1	Recomendaciones	94
	Referencias bibliográficas	96
A	Apéndice A: Bitácora de diseño	103
A.1	Introducción	103
A.2	Necesidades	103
A.3	Especificaciones	105
A.3.1	Métricas	105
A.3.2	Valores objetivo	105
A.4	Estudio de conceptos	107
A.4.1	Descomposiciones funcionales	107
A.5	Validación de resultados	119
B	Apéndice B: Manual de usuario	133
B.1	Instalación en Windows	133
B.2	Asignación de una IP estática en Home Assistant	142
B.3	Instalación de HACS	144
B.4	Conexión de dispositivos USB en la máquina virtual	145
B.5	Conexión de luces	147
B.5.1	Lifx Colour	148
B.5.2	Yeelight 1S	150
B.6	Conexión de conectores (Smart Plugs)	151
B.6.1	Tapo P100	151
B.6.2	Sonoff S26	154
B.7	Conexión de sensores	157
B.8	Conexión de cámaras	158
B.8.1	Tapo	159
B.8.2	Foscam	162
B.9	Parlantes	163
B.9.1	Amazon Echo Dot	163
B.10	Obtención del Long Live Token	165
B.11	Activación de Virtualización en Computador	166
C	Encuesta para pruebas de validación	168

LISTA DE FIGURAS

2.1	Robot Robobo	17
3.1	Etapas del Proceso de Diseño en Ingeniería	23
3.2	Diagrama de caja negra.	25
4.1	Diagrama de solución por implementar.	46
5.1	Dispositivo para aplicación piloto.	48
5.2	Dashboard pre-determinado de <i>Home Assistant</i>	50
5.3	Configuración en Home Assistant de Dispositivo Piloto.	51
5.4	Dispositivo piloto <i>Sonoff Smart Plug</i> integrado en <i>Home Assistant</i> . . .	51
5.5	Diagrama de flujo del funcionamiento de <i>Home Assistant</i>	54
5.6	Dispositivos de luz Yeelight y Lifx.	55
5.7	Dispositivos Smart Plugs Tapo y D-Link.	56
5.8	Sensores utilizados (varios)	57
5.9	Coordinadores Zigbee: Zigbee Sniffer y Conbee II	59
5.10	Sensores implementados en <i>Home Assistant</i>	60
5.11	Cámaras utilizadas en <i>Home Assistant</i>	61
5.12	Parlante inteligente <i>Echo Dot</i>	62
6.1	Diagrama de flujo de interacción entre aplicación y <i>Home Assistant</i> . . .	66
6.2	Interfaz gráfica de la aplicación diseñada.	68
6.3	Ejemplo de aplicación funcional.	69
6.4	Interfaz final con automatización agregada.	71
7.1	Respuestas recibidas para la primer prueba	84
7.2	Respuestas recibidas para la segunda prueba	85
7.3	Respuestas recibidas para la tercera prueba	85
7.4	Respuestas recibidas para la cuarta prueba	86
A.1	Diagrama de caja negra.	107

A.2	Descomposición funcional.	108
A.3	Árbol de clasificación para subproblema de alimentación.	108
A.4	Árbol de clasificación para subproblema de protocolos de comunicación.	109
A.5	Árbol de clasificación para subproblema de selección de herramientas.	109
A.6	Árbol de clasificación para subproblema de selección de dispositivos.	110
A.7	Tabla de combinación para concepto No.1:Android Studio	111
A.8	Tabla de combinación para concepto No.2: Google Home	112
A.9	Tabla de combinación para concepto No.3: Home Assistant	112
A.10	Tabla de combinación para concepto No.4: Ionic	113
A.11	Tabla de combinación para concepto No.5: Python	113
A.12	Tabla de combinación para concepto No.6: Tuya Smart	114
A.13	Tabla de combinación para segunda iteración de conceptos	116
B.1	Descarga imagen de <i>Virtualbox</i> de <i>Home Assistant</i>	134
B.2	Descarga de <i>Virtualbox</i>	134
B.3	Instalación de <i>Virtualbox</i>	135
B.4	Instalación de <i>Virtualbox</i> (continuación)	135
B.5	Instalación de <i>Virtualbox</i> (continuación)	135
B.6	Instalación de <i>Virtualbox</i> (continuación)	136
B.7	Instalación de <i>Virtualbox</i> (continuación)	136
B.8	Interfaz de máquina virtual	137
B.9	Configuración de máquina virtual	137
B.10	Configuración de máquina virtual (continuación)	138
B.11	Selección de imagen de <i>Home Assistant</i>	138
B.12	Selección de imagen de <i>Home Assistant</i> (continuación)	138
B.13	Selección de imagen de <i>Home Assistant</i> (continuación)	139
B.14	Selección de imagen de <i>Home Assistant</i> (continuación)	139
B.15	Selección de imagen de <i>Home Assistant</i> (continuación)	139
B.16	Configuración de máquina virtual con <i>Home Assistant</i>	140
B.17	Configuración de máquina virtual con <i>Home Assistant</i> (continuación)	140
B.18	Configuración de máquina virtual con <i>Home Assistant</i> (continuación)	140
B.19	Configuración de máquina virtual con <i>Home Assistant</i> (continuación)	141
B.20	Acceso a interfaz de <i>Home Assistant</i>	141
B.21	Acceso a interfaz de <i>Home Assistant</i> (continuación)	141
B.22	Creación de usuario en <i>Home Assistant</i>	142
B.23	Configuración de dirección IP estática	143
B.24	Configuración de dirección IP estática (continuación)	143
B.25	Descarga de Visual Studio Code en <i>Home Assistant</i>	144
B.26	Configuración de Visual Studio Code en <i>Home Assistant</i>	145
B.27	Acceso a una terminal en Visual Studio Code en <i>Home Assistant</i>	145
B.28	Comando para descarga de <i>Home Assistant Community Store</i> (HACS)	146
B.29	Configuración USB en máquina virtual	146
B.30	Dispositivo USB agregado en configuración	147
B.31	Dispositivo USB agregado en máquina virtual	147
B.32	Cambio de nombre de un dispositivo	148

B.33 Cambio de nombre de un dispositivo (continuación)	149
B.34 Cambio de nombre de un dispositivo (continuación)	149
B.35 Agregar tarjeta o visualización de un dispositivo	150
B.36 Añadir repositorio personalizado en <i>Home Assistant Community Store</i> (HACS)	152
B.37 Añadir repositorio personalizado en <i>Home Assistant Community Store</i> (HACS) (continuación)	152
B.38 Añadir repositorio personalizado en <i>Home Assistant Community Store</i> (HACS) (continuación)	153
B.39 Agregar integración de Tapo Smart Plug	153
B.40 Integración de Sonoff LAN	154
B.41 Integración de File Editor	155
B.42 Modificación de archivo de configuración	156
B.43 Dispositivo Sonoff S26 añadido	156
B.44 Integración de Zigbee Home Automation para sensores	157
B.45 Descarga en <i>Home Assistant Community Store</i> (HACS) de integración de <i>Tapo: Cameras Control</i>	159
B.46 Integración de cámara <i>Tapo</i>	160
B.47 Configuración para visualización de cámara <i>Tapo</i>	160
B.48 Configuración para visualización de cámara <i>Tapo</i> (continuación)	161
B.49 Configuración para visualización de cámara <i>Tapo</i> (continuación)	161
B.50 Integración de cámara <i>Foscam</i>	162
B.51 Configuración para visualización de cámara <i>Foscam</i>	163
B.52 Configuración para visualización de cámara <i>Foscam</i> (continuación)	163
B.53 Activación de servicio de <i>Amazon Alexa</i>	164
B.54 Activación de skill de <i>Home Assistant</i> para <i>Amazon Alexa</i>	164
B.55 Activación de skill de <i>Home Assistant</i> para <i>Amazon Alexa</i> (continuación)	165
B.56 Detección de dispositivos para <i>Alexa</i>	165
B.57 Selección de dispositivos para <i>Alexa</i>	166
B.58 Obtención del <i>Long Live Token</i>	166

LISTA DE TABLAS

2.1	Características de los principales diseños cualitativos	19
2.2	Tamaños de muestra comunes en estudios cualitativos.	20
2.3	Tamaños de muestra para estudios cuantitativos con datos binarios. . .	21
3.1	Valores para jerarquización de necesidades.	24
4.1	Tabla de necesidades.	29
4.2	Definición de Especificaciones.	30
4.3	Filtrado de conceptos inicial.	34
4.4	Evaluación de conceptos inicial.	36
4.5	Iteración de filtrado de conceptos.	41
4.6	Iteración de filtrado de conceptos (continuación)	42
4.7	Características adicionales de conceptos.	42
4.8	Evaluación final de conceptos.	43
4.9	Dispositivos seleccionados	44
5.1	Resumen de porcentaje de error en dispositivo Sonoff para prueba piloto.	53
7.1	Dispositivos utilizados para las pruebas.	76
7.2	Porcentaje de error obtenido en la lectura de dispositivos en <i>Home Assistant</i>	77
7.3	Porcentaje de error obtenido en la actuación de dispositivos en <i>Home Assistant</i>	77
7.4	Porcentaje de error obtenido en la conexión de dispositivos en <i>Home Assistant</i>	77
7.5	Intervalos de confianza (IC) por cada acción de dispositivo, para <i>Home Assistant</i>	79
7.6	Porcentaje de error obtenido en la lectura de dispositivos por medio de la aplicación diseñada	80
7.7	Porcentaje de error obtenido en la actuación de dispositivos por medio de la aplicación diseñada	81

7.8	Intervalos de confianza (IC) por cada acción de dispositivo, para llamadas de la aplicación	82
8.1	Costo de dispositivos adquiridos para el proyecto.	90
8.2	Costo total para el desarrollo del proyecto.	91
A.1	Necesidades establecidas por el cliente.	104
A.2	Valores para jerarquización de necesidades.	104
A.3	Establecimiento de métricas.	105
A.4	Establecimiento de correspondencia entre necesidades y especificaciones	106
A.5	Especificaciones objetivo.	106
A.6	Profundización de valores de métrica No. 9.	106
A.7	Filtrado de conceptos.	115
A.8	Evaluación de conceptos.	116
A.9	Filtrado de conceptos tras realimentación.	117
A.10	Evaluación de conceptos tras realimentación.	118
A.11	Resultados de pruebas de lectura en <i>Home Assistant</i> para Smart Plug Sonoff.	119
A.12	Resultados de pruebas de lectura en <i>Home Assistant</i> para Smart Plug Tapo.	119
A.13	Resultados de pruebas de tiempo promedio de lectura en <i>Home Assistant</i> para Smart Plug Tapo.	120
A.14	Resultados de pruebas de lectura en <i>Home Assistant</i> para Luz Yeelight.	120
A.15	Resultados de pruebas de lectura en <i>Home Assistant</i> para Luz Lifx.	121
A.16	Resultados de pruebas de tiempo promedio de lectura en <i>Home Assistant</i> para Luz Lifx.	121
A.17	Resultados de pruebas de lectura en <i>Home Assistant</i> para Sensor de puerta Sonoff.	122
A.18	Resultados de pruebas de lectura en <i>Home Assistant</i> para Sensor de movimiento Aqara.	122
A.19	Resultados de pruebas de actuación en <i>Home Assistant</i> para Smart Plug Sonoff.	123
A.20	Resultados de pruebas de actuación en <i>Home Assistant</i> para Smart Plug Tapo.	123
A.21	Resultados de pruebas de actuación en <i>Home Assistant</i> para Luz Yeelight.	124
A.22	Resultados de pruebas de actuación en <i>Home Assistant</i> para Luz Lifx.	124
A.23	Resultados de pruebas de conexión en <i>Home Assistant</i> para Smart Plug Sonoff.	125
A.24	Resultados de pruebas de conexión en <i>Home Assistant</i> para Smart Plug Tapo.	125
A.25	Resultados de pruebas de conexión en <i>Home Assistant</i> para Luz Yeelight.	126
A.26	Resultados de pruebas de conexión en <i>Home Assistant</i> para Luz Lifx.	126
A.27	Resultados de pruebas de conexión en <i>Home Assistant</i> para Sensor movimiento Aqara.	127

A.28 Resultados de pruebas de conexión en <i>Home Assistant</i> para Sensor de puerta Sonoff.	127
A.29 Resultados de pruebas de lectura en aplicación diseñada, para Smart plug Sonoff.	128
A.30 Resultados de pruebas de lectura en aplicación diseñada, para Smart plug Tapo.	128
A.31 Resultados de pruebas de lectura en aplicación diseñada, para Luz Yeelight.	129
A.32 Resultados de pruebas de lectura en aplicación diseñada, para Luz Lifx.	129
A.33 Resultados de pruebas de lectura en aplicación diseñada, para Sensor de movimiento Aqara.	130
A.34 Resultados de pruebas de lectura en aplicación diseñada, para Sensor de puerta Sonoff.	130
A.35 Resultados de pruebas de actuación en aplicación diseñada, para Smart Plug Sonoff.	131
A.36 Resultados de pruebas de actuación en aplicación diseñada, para Smart Plug Tapo.	131
A.37 Resultados de pruebas de actuación en aplicación diseñada, para Luz Yeelight.	132
A.38 Resultados de pruebas de actuación en aplicación diseñada, para Luz Lifx.	132

LISTA DE ACRÓNIMOS

API *Application Programming Interface*

AWS Amazon Web Services

GII Grupo Integrado de Ingeniería

HACS *Home Assistant Community Store*

HTTP Hypertext Transfer Protocol

IC Intervalo de Confianza

IETF Internet Engineering Task Force

IoT *Internet of Things* (Internet de las Cosas)

JSON JavaScript Object Notation

LIANA Laboratorio de Inteligencia Artificial para las Ciencias Naturales

MINT *Manufactura de Ingenios Tecnológicos*

REST Representational State Transfer

SDK Software Development Kit

SoC System on a Chip

TEC Tecnológico de Costa Rica

UDC *Universidade da Coruña*

CAPÍTULO 1

INTRODUCCIÓN

El proyecto presentado surge por medio de la colaboración del *Laboratorio de Inteligencia Artificial para las Ciencias Naturales* (LIANA) [1] con el *Grupo Integrado de Ingeniería* (GII) [2], de la *Universidad da Coruña* (UDC). LIANA es una unidad de investigación establecida en el Área Académica de Ingeniería en Mecatrónica del *Tecnológico de Costa Rica* (TEC) [3]. La investigación de este grupo tiene un carácter multidisciplinario, donde se busca el desarrollo de paradigmas computacionales inteligentes para la resolución de variados problemas. En el presente caso, se trabajará con el apoyo del coordinador de LIANA, el Dr. Juan Luis Crespo Mariño, y del investigador asociado Ing. Ronald Loaiza Baldares. Además, se trabajará principalmente con la UDC, siendo esta institución el cliente para quien se desarrollará el presente proyecto.

La UDC es una universidad localizada en La Coruña, España, y dentro de esta, surge el GII, un grupo interdisciplinario de investigación aplicada en ingeniería orientado a la transferencia de conocimiento y a la generación de nuevos productos en el entorno industrial [2]. En él, se desarrollan proyectos en áreas como inteligencia computacional,

robótica, sistemas de medida y sensorización, entre otros [4]. El proyecto forma parte de la línea de tecnologías avanzadas de sensorización y medida [5]. Se trabajará con el Dr. Francisco Bellas Bouza [6] quien tendrá el rol de tutor de empresa.

Adicionalmente, se debe mencionar al grupo de *Manufactura de Ingenios Tecnológicos* (MINT) , una empresa de base tecnológica que surge a través del GII, con el objetivo de crear valor por medio de las décadas de generación de conocimiento del grupo [7]. Dentro de esta empresa, se contará con la guía y colaboración del investigador Dr. Alejandro Paz López, quien, junto con un equipo de investigadores, es encargado del desarrollo del proyecto *Robobo* [8], el cual trata de un robot miniatura, portátil y móvil, para fines didácticos como un acercamiento en las áreas de robótica y computación para estudiantes.

El profesor Dr. Francisco Bellas, plantea la creación de un curso piloto de Introducción a la Inteligencia Artificial para estudiantes a nivel colegial, con el fin de brindarles un primer acercamiento a las ciencias de la computación y programación. Entre los contenidos y proyectos del curso, se encuentra una línea de trabajo con la temática del *Internet of Things (Internet de las Cosas)* (IoT) [8]. Para ello se necesita crear una plataforma computacional con funcionalidad en dispositivos móviles, y de esta manera controlar múltiples sensores o dispositivos (seleccionados específicamente para el presente proyecto) empleados para control inteligente de ambiente en un hogar o habitación (por ejemplo: cámaras, sensores de presencia-ausencia, sensores de temperatura, luces...).

Además, la propuesta mencionada anteriormente podría llegar a ser integrada al *Robobo Framework* [9], y de esta manera expandir las funcionalidades de este. El presente proyecto surge como una necesidad que se abordará con el diseño de una plataforma o herramienta que permita el desarrollo de proyectos en el área de IoT y de esta manera controlar múltiples dispositivos o sensores de un entorno de habitación, y adaptarla para los proyectos ya existentes de MINT, en este caso el Robobo.

Actualmente, el curso a impartir por el Dr. Francisco Bellas como introducción a la inteligencia artificial, está calendarizado para comenzar en febrero 2022, sin embargo, el cliente no cuenta con las herramientas necesarias para impartir el proyecto planificado en el área de IoT. Se desea que los estudiantes puedan aprender acerca de sensores y dispositivos que permitan control de una habitación, sin que sus conocimientos técnicos sean una limitante, ya que el realizar una interconexión de un dispositivo propio para comunicarse con sensores, puede representar una labor fuera de sus conocimientos actuales.

De esta manera, la realización del proyecto previamente mencionado representa un aporte a la ingeniería, mediante una solución que no solo facilite el aprendizaje del estudiante y la comprensión de la temática por impartir, sino que también promueve la innovación al permitir la interconexión de herramientas personalizadas con dispositivos externos que no están directamente conectados entre sí. Además permitiría al GII contar con un mayor rango en cuanto a temáticas y posibilidades para futuros proyectos educativos.

1.1 Objetivos

A continuación, se presentan los objetivos a desarrollar en el presente proyecto.

1.1.1 Objetivo general

Diseñar un sistema que permita una comunicación desde un lenguaje de programación estándar, con el fin de controlar sensores o dispositivos inteligentes, para proyectos educativos en el área de IoT impartidos por el Grupo Integrado de Ingeniería de la *Universidad da Coruña*.

1.1.2 Objetivos específicos

1. Identificar las herramientas existentes que podrían permitir la comunicación con dispositivos en entornos de IoT.
2. Definir un sistema que permita a un usuario la comunicación y el control de dispositivos inteligentes o sensores.
3. Desarrollar la etapa de integración del sistema para la compatibilidad con el *Robobo Framework* del proyecto *Robobo* de la empresa MINT.
4. Evaluar la solución implementada por medio de pruebas que permitan demostrar el cumplimiento de las necesidades del cliente.

1.2 Estructura del documento

En los siguientes capítulos se expone el desarrollo del proyecto. En el Capítulo 2 se cuenta con aspectos teóricos y un estado del arte relevante para comprender la temática, algunas de las tecnologías y términos utilizados en el proyecto. El Capítulo 3 indica la metodología seguida, referente a lo establecido por el proceso de diseño en ingeniería adaptada para el presente proyecto. Los siguientes capítulos se relacionan directamente con los objetivos específicos planteados para el proyecto. De esta manera, el Capítulo 4 establece los aspectos relevantes en el área del IoT, además de un estudio y selección de herramientas y dispositivos para el diseño de la plataforma. En el Capítulo 5 se documenta el proceso del establecimiento de un entorno inteligente con múltiples dispositivos, además de distintas consideraciones relevantes. El Capítulo 6 documenta la aplicación diseñada e implementada por medio de *Android Studio* que permite la comunicación externa con la plataforma de *Home Assistant*, la cual representaría la herramienta que utilizará el GII para su futura integración con el *Robobo*. Seguidamente el Capítulo 7 expone la validación de las distintas etapas y objetivos del proyecto,

junto con el correspondiente análisis de resultados. El Capítulo 8 muestra los recursos implicados en el desarrollo del proyecto mediante el análisis económico, y finalmente en el Capítulo 9 se exponen las principales conclusiones y recomendaciones del proyecto.

CAPÍTULO 2

MARCO TEÓRICO

Este capítulo presenta la información pertinente para comprender las bases teóricas que facilitan el desarrollo del presente proyecto. Se comienza con una introducción al tema del IoT, y algunas consideraciones acerca de su importancia en el área de la educación. Seguidamente, se elabora sobre los diferentes protocolos o métodos de comunicación existentes que permiten establecer la comunicación entre usuarios y dispositivos en un ecosistema IoT. Posteriormente, se exponen algunas de las principales herramientas y tecnologías en el área que permiten el establecimiento de entornos conectados y controlados. Para finalizar, se explica en qué consiste el proyecto *Robobo*, junto con el *Robobo Framework*, y consideraciones para el análisis de datos.

2.1 Internet de las cosas

El término IoT representa una red de objetos físicos, especialmente dispositivos de uso diario (incluyendo todo tipo de dispositivos como automóviles, teléfonos, cámaras,

juguetes, máquinas de hogar, luces, sensores...), que sean capaces de ser reconocidos, localizables y controlables por medio del internet [10]. Estos dispositivos, embebidos con sensores, software, y tecnologías externas son capaces tanto de permitir una comunicación con un usuario, como de intercambiar datos con otros dispositivos. De esta manera se presenta también la importancia del IoT, ya que cada vez es más sencilla la recolección, el almacenamiento y monitoreo de datos, con mínima intervención humana [11].

Surge el concepto de ambiente inteligente, que se basa en la idea de permitir un ambiente físico que está completa e invisiblemente conectado con sensores, actuadores, y elementos computacionales integrados eficientemente en los objetos y actividades del día a día [12]. El uso del IoT junto con plataformas de software especializado para la recolección, almacenamiento, pre-procesamiento y comunicación de datos permiten facilitar y automatizar cualquier tipo de proceso. Esto fomenta la introducción del IoT en la vida cotidiana en múltiples ámbitos, como por ejemplo la industria, uso personal de hogar, transporte y logística, medicina, agricultura, energía, y hasta ciudades inteligentes [13]. De esta manera, se reitera la idea fundamental del internet de las cosas en donde se busca el intercambio de datos entre dispositivos o máquinas, conectados por medio de distintos protocolos de comunicación estándar. Estos sistemas son acoplados exitosamente al lograr establecer la comunicación mediante el internet con dispositivos como sensores y actuadores, que a su vez permiten la realización de las acciones deseadas o definidas por el usuario.

2.1.1 IoT en la educación

Se estima que el área del IoT será cada vez más relevante y tendrá un impacto significativo en aplicaciones industriales. De esta forma resulta vital que estudiantes, desde tempranas etapas en su formación académica, aprendan sobre análisis, desarrollo y mantenimiento de este tipo de sistemas, por medio de exposición en esta área de

conocimiento [14, p. 248]. Para comenzar con la exposición estudiantil en esta rama tecnológica, no solo sería necesario implementar un cambio en la forma de enseñanza colegial, sino que es fundamental tomar en cuenta la formación docente, ya que estos serán el principal agente de cambio para establecer los principios y prácticas adecuadas para el uso efectivo de la tecnología en un entorno de IoT.

Distintos acercamientos a la integración del IoT con la educación han sido expuestos [15, p. 206]. A partir de ellos, se reconoce la importancia de encontrar métodos didácticos, que permitan una enseñanza clara y adaptativa para facilitar el aprendizaje de los estudiantes. Además, resulta especialmente útil la integración del IoT en proyectos o laboratorios en diferentes materias educativas, para permitir una mayor exposición a este tipo de tecnología y sus posibles aplicaciones. A lo anterior, se suma la utilización de las tecnologías de almacenamiento de datos para poder crear un mayor rango en cuanto a posibilidades de enseñanza y diversidad de proyectos. Así, es posible remarcar la importancia de adaptar las mallas curriculares de tal manera que sea posible la enseñanza de parte de la comunidad docente, y la comprensión tanto teórica como práctica por parte de la comunidad estudiantil acerca de tecnologías y aplicaciones en entornos IoT.

2.2 Protocolos de comunicación

Actualmente existen en el mercado una exorbitante cantidad y variedad de productos utilizados en el área de IoT, desde componentes electrónicos como sensores y actuadores, hasta empresas encargadas de la creación y distribución de dispositivos inteligentes como lo son Philips Hue [16], Sonoff [17] o TP-Link [18]. A pesar de que muchos de los productos que se pueden adquirir en el mercado son similares en aspecto o función, estos pueden ser capaces de operar bajo distintos protocolos de comunicación. Diferentes protocolos implican diferencias tanto en su conexión

(físicamente, o con sus respectivas aplicaciones de usuario), como en el requerimiento de hardware adicional. Existen dos categorías principales de redes para la conexión de dispositivos: largo alcance, y corto alcance. Para el presente proyecto se tomarán en cuenta los principales protocolos de comunicación para comunicaciones de corto alcance que incluye Wifi, Bluetooth, Zigbee y Z-Wave [19]. Se descartan protocolos de largo alcance debido a que su utilización es enfocada en aplicaciones industriales donde se requieren transmisiones de datos a kilómetros de distancia, o hasta en implementaciones de ciudades inteligentes.

2.2.1 Wi-fi

La implementación por Wifi utiliza el protocolo IEEE 802.11 [19], el cuál implica un funcionamiento mediante una capa de conexiones físicas, y otra de transmisión de datos. Múltiples dispositivos son capaces de conectarse a una red por diferentes puntos de acceso que la red provee. El uso del Wifi en IoT presenta una notable comodidad, ya que en la mayoría de ambientes ya se cuenta con una red Wifi (sea de hogar, o laboral). Además los dispositivos que utilizan este protocolo solo requieren conectarse a la red sin necesidad de un controlador (llamado puente o hub) adicional que se encargue de comunicar, manejar y conectar sus dispositivos. Este tipo de comunicación implica un bajo costo de adquisición y mantenimiento, pero a la vez se tiene un alto consumo de potencia y un corto alcance de comunicación [20] [21]. Es posible la operación a frecuencias de 2.4 GHz y 5GHz.

2.2.2 Zigbee

Se basa en el estándar IEEE 802.15.4, que implica comunicaciones a corta distancia por radiofrecuencia [22], implica operación a baja potencia y alta escalabilidad en cuanto a la cantidad total de nodos o dispositivos que se pueden conectar, haciéndolo ideal para entornos IoT. Un sistema Zigbee se compone de 3 tipos distintos de dispositivos[19]:

- Coordinador: encargado de procesar y almacenar información a la hora de transferir datos.
- Router: actúa como intermediario para permitir el paso de datos entre dispositivos.
- Dispositivos: tienen funcionalidad limitada, algunos ejemplos pueden ser bombillos inteligentes, *switches*, sensores.

Las comunicaciones entre dispositivos permiten distancias de 10 a 20 metros, y teóricamente permiten una cantidad total de dispositivos conectados mayor a 65 000 [23]. Opera normalmente a una frecuencia de 2.4 GHz.

2.2.3 Bluetooth

Este protocolo opera bajo el estándar IEEE 802.15.1 [22], implica un bajo consumo de potencia. El Bluetooth permite mayores distancias de comunicación (de hasta 100m) y opera a una frecuencia de 2.4 GHz. Es usado comúnmente en dispositivos móviles [24].

2.2.4 Z-Wave

Z-Wave es una comunicación manejado por Z-Wave alliance, y está categorizado junto con los estándares IEEE 802.11 y 802.15[25]. Este tipo de comunicación permite el intercambio de datos tanto con un controlador como con entre los mismos dispositivos [23], y al igual que con la comunicación Zigbee requiere de un coordinador, un router y dispositivos. Permite mayores distancias (hasta 100m) y más de 200 dispositivos conectados. La operación de este protocolo se da en las bandas de los 900 MHz [21], por lo que presenta menos posibilidad de interferencias con respecto al resto de dispositivos conectados en una red Wifi normal.

2.2.5 Conceptos relacionados para comunicaciones IoT

2.2.5.1 Websockets

El protocolo de red de *websockets* define la comunicación entre un cliente y un servidor (donde el cliente representa a una aplicación o servicio que solicita acceder a un recurso en específico, y un servidor es la aplicación o servicio que contenga dicho recurso [26]) por medio del internet, y se especifica bajo el estándar RFC 6455 de la *Internet Engineering Task Force* (IETF) [27, p. 33]. Este protocolo se presenta como una forma de comunicación *full duplex* [28] (que permite tanto envío como recepción de datos de manera simultanea por un mismo canal de tiempo/frecuencia entre emisor y receptor) entre una aplicación web y un servidor. De esta manera se presenta la conveniencia de permitir la creación de un servicio de comunicación independiente de una plataforma en específico. Los *websockets* se comunican por un formato de texto, que solamente requiere trabajar sobre un navegador web, por lo cual es posible establecer una comunicación con una aplicación cliente por medio de cualquier plataforma (sea Mac, Windows, Linux, o hasta software móvil como iOS o Android) [29]. La comunicación por *websockets* es posible de programar por medio de *JavaScript*.

2.2.5.2 Protocolo HTTP

El protocolo *Hypertext Transfer Protocol* (HTTP) es diseñado para permitir la comunicación entre clientes y servidores, de manera que se establecen pedidos y respuestas entre ambos. Existen dos métodos comunes de comunicación con HTTP, el método GET y el Método POST. GET es utilizado para solicitar información de alguna fuente en particular, el método POST funciona para enviar datos o información a un servidor [30].

2.2.5.3 REST APIs

Una interfaz programada de aplicación, o *Application Programming Interface* (API), es un conjunto de definiciones y protocolos para la creación e integración de software de aplicaciones [31]. Es un tipo de mediador entre el usuario, o cliente, y una fuente de información o servicio web del cual se quiere obtener o enviar información. Una API permite la comunicación con un sistema de manera que este comprenda las instrucciones que son requeridas de este.

Representational State Transfer (REST) es un estilo de arquitectura para implementar servicios web [32]. Esta arquitectura se estructura bajo un conjunto de reglas, entre las cuales se incluye que el servicio y la aplicación cliente deben ser independientes el uno del otro, todos los pedidos de información para un mismo recurso deben ser escritos o estructurados de la misma manera, y que cada pedido de información debe contener en sí mismo toda la información necesaria para solicitar los datos deseados (no debe depender de ningún dato almacenado en el servidor) [33].

Las REST APIs utilizan la comunicación HTTP para realizar funciones como lectura y escritura de datos en un recurso o servicio dado, por lo cual es posible conocer el estado de un recurso en particular en cualquier instante. La arquitectura REST permite la interacción con un cliente en una amplia variedad de formatos (eliminando la barrera de utilizar un lenguaje único de programación) como *JavaScript Object Notation* (JSON), *HTML*, *Python* o texto simple [26]. Por medio de sus reglas estructurales, la arquitectura REST se convierte en una herramienta para APIs flexible, rápida y escalable. Estas características hacen a las REST APIs útiles particularmente para entornos IoT.

2.3 Herramientas IoT

Actualmente, existen herramientas variadas para el establecimiento y control de un ecosistema en el área de IoT. Soluciones programables con componentes electrónicos, plataformas para el control centralizado de ambientes inteligentes, o soluciones de grandes empresas para aplicaciones industriales son algunas de las tecnologías utilizadas mundialmente en un tema cada vez más explorado como el Internet de las Cosas. A continuación, se exponen algunas de las principales herramientas disponibles en el área.

2.3.1 Google Home

Es una plataforma propietaria de *Google*, la cual permite la integración de dispositivos inteligentes de una gran variedad de proveedores. Tiene la posibilidad de utilizar el asistente de voz *Google Assistant*, y manejar la integración de dispositivos por medio de su aplicación de *Google Home* [34]. Presenta herramientas adicionales como *HomeGraph* [35], una base de datos de *Google* que permite almacenar información de los dispositivos conectados y sus habitaciones. La plataforma de IoT de *Google* no es de código abierto (*open source*), sin embargo permite una limitada manipulación externa por medio de sus *Smart Home Actions*, las cuales permiten integrar por medio de distintos lenguajes de programación algunos dispositivos que no son nativamente integrables con *Google*.

2.3.2 OpenHAB

Open Home Automation Bus (Open HAB) es una plataforma para automatización de hogar que no es dependiente de vendedores o tecnologías. Es capaz de integrar, controlar y manipular diferentes tecnologías por medio de su software, además de funcionar en múltiples sistemas operativos. La plataforma se basa en algunos

conceptos clave para la programación de dispositivos [36]:

- *Bindings*: paquetes de software que permiten a la plataforma la interacción con distintos dispositivos.
- *Things*: es algún dispositivo en específico por conectar.
- *Channels*: las acciones específicas que pueden realizar cada uno de los dispositivos o *Things*.
- *Items*: es la representación virtual de un dispositivo, de la cual se puede conocer información (o sus estados, por ejemplo encendido o apagado).
- *Rules*: son reglas que permiten la automatización de los dispositivos conectados.

Se tiene posibilidad de funcionamiento de manera local (por medio de una única red *Wifi*) o a través de la nube (por medio de internet, accesible desde cualquier ubicación) por medio de una extensión descargable llamada *MyOpenHAB* [37], la cual sigue en proceso de desarrollo.

2.3.3 Tuya Smart

Esta plataforma de IoT permite tanto el manejo como la creación de dispositivos inteligentes. Dentro de ella, se tiene la posibilidad de la creación de hardware especializado para entornos IoT por medio de *Tuya IoT Development Platform* que permite de manera gráfica diseñar el dispositivo deseado por el usuario, el cual sería posteriormente compatible con su plataforma y aplicaciones de control IoT [38]. *Tuya* cuenta con una plataforma accesible desde navegadores web, la cual permite la integración de múltiples dispositivos inteligentes y posteriormente controlarlos por una interfaz programable donde se requieren conocimientos técnicos del hardware conectado a la plataforma [39]. Adicionalmente, Tuya tiene un *Software Development Kit* (SDK) que permite a cualquier desarrollador descargar y modificar para uso

personal los servicios de la plataforma, sin embargo, el SDK solamente es ofrecido para actuar sobre dispositivos de luz, cerrojos inteligentes, o limpiadores móviles [40].

2.3.4 Microcontroladores

Los microcontroladores son dispositivos regularmente utilizados para aplicaciones en IoT, debido a su flexibilidad para ser programados y conectados con diferentes dispositivos, bibliotecas y servicios. Además, la capacidad de algunos dispositivos de ser integrados con módulos de *Wifi* o *Bluetooth* facilitan la creación de proyectos en el área de manera sencilla y económica.

Los *System on a Chip* (SoC) de las series ESP, como el ESP32 o ESP8266, producidos por la empresa *Espressif Systems* [41] son utilizados con frecuencia, ya que estos son posibles de programar desde el *Arduino IDE* [42], un entorno de programación sumamente documentado, desarrollado y que no requiere una alta curva de aprendizaje. Estos son comúnmente utilizados en conjunto con bases de datos, como por ejemplo *Google Firebase* [43]. Con estas herramientas, es posible establecer una comunicación bidireccional entre una aplicación personalizada, por ejemplo creada por medio de *Android Studio* [44], y un SoC y distintos componentes electrónicos (leds, motores, switches) [45] [46].

Adicionalmente el uso de microcomputadores como *Raspberry Pi* [47] permiten la instalación de software especializado de distintos proveedores (por ejemplo *OpenHAB*, o *Home Assistant*) para permitir un control centralizado de entornos IoT y de esta manera obtener un monitoreo continuo sobre los distintos dispositivos conectados en un entorno.

2.3.5 Home Assistant

Home Assistant es una plataforma para automatización, control y monitoreo en entornos IoT con una gran variedad de dispositivos, marcas y servicios disponibles tanto

de manera nativa como por medio de integraciones adicionales, gratuitas o por cobro [48]. Es una plataforma de código abierto que permite su instalación en una variedad de sistemas operativos, cuenta con una aplicación móvil, y además permite una función de auto-descubrimiento que permite el reconocimiento de dispositivos conectados en la misma red automáticamente. Adicionalmente, la plataforma cuenta con servicio de conexión por medio de la nube, utilizando una empresa derivada llamada *Nabu Casa* que permite el control de dispositivos a distancia [49].

2.3.6 Amazon Web Services

Es un servicio principalmente enfocado a soluciones industriales en el área de IoT. La plataforma de *Amazon Web Services* (AWS) permite manejar dispositivos a grandes escalas, enfocándose en permitir a grandes empresas la extracción, manejo, almacenamiento y hasta análisis de grandes volúmenes de datos. El servicio es actualmente utilizado en empresas de granjas de energía solar, servicios de seguridad para hogar, minería, y monitoreo en industria médica [50]. El servicio de *Amazon* permite la integración de múltiples dispositivos como por ejemplo microcontroladores, CPUs, sensores y actuadores, además de facilitar funcionalidades de control, simulación, pruebas, visualización en tiempo real, y manejo de datos [51].

2.4 Proyecto Robobo

El proyecto *Robobo* es una iniciativa educativa creada por el grupo derivado del GII, MINT. El proyecto se basa en un robot conformado por una base miniatura y móvil, y un celular corriendo el Robobo Framework que contiene la funcionalidad o software del robot [8]. El robot, observado en la figura 2.1, representa una herramienta educativa flexible y de bajo costo. En lo que respecta al *hardware*, Robobo cuenta con:

- 4 motores de corriente directa para agregar movilidad y grados de libertad al

sistema.

- 8 sensores infrarrojos.
- *Encoders* en los motores.
- Adicionalmente el *smartphone* utilizado provee al sistema con cámaras, micrófono, giroscopio, acelerómetro, magnetómetro, pantalla táctil, GPS y demás sensores dependientes del modelo de celular.



Figura 2.1: Robobo con su base y aplicación móvil. Recuperado de: [8]

El *Robobo* tiene aplicaciones tanto en *Android* como *iOS*. En lo que respecta al *software* utilizado por el *Robobo*, este es descrito por el *Robobo Framework*.

2.4.1 Robobo Framework

El *Robobo Framework* es una arquitectura modular y escalable para programar al *Robobo*, tanto en *Android* como *iOS* (en lenguajes de Java y Swift respectivamente) [8]. El *framework* está compuesto de una clase principal, y múltiples módulos con diferentes utilidades. La clase principal se encarga de manejar el ciclo de vida de cada uno de los demás módulos, su carga, actuación, y accesos entre ellos mismos. En cuanto a los módulos adicionales, estos se organizan en bibliotecas organizadas por las

diferentes capacidades del *Robobo* como lo son: sistemas de visión, procesamiento de sonido, control remoto, reconocimiento de voz, detección de tacto y de sensores.

El proyecto *Robobo* es de código abierto, ya que uno de sus principales enfoques es fomentar experimentos en el área educativa y de la robótica. El código [9] se encuentra disponible en *Github*, donde se especifica la posibilidad de añadir módulos, e incluso crear aplicaciones nuevas que permitan controlar el *Robobo* y expandir las funcionalidades ya integradas en él. Además, se muestra cómo es posible controlar el Robobo por medio de herramientas como Python, Scratch, y ROS.

2.5 Validación de Pruebas

2.5.1 Análisis cualitativo

Al analizar datos, en ocasiones es útil realizar un análisis desde el punto de vista cualitativo. Este enfoque se utiliza cuando se tiene como propósito examinar la forma en que los individuos perciben y experimentan fenómenos que los rodean. Se da un enfoque en la comprensión y exploración de procesos desde la perspectiva o interpretación del participante [52]. Es recomendado tener un conocimiento profundo del tema por investigar, y plantear de esta manera el propósito de las pruebas por realizar, su justificación y viabilidad, y el contexto en el cual se realizará el experimento. Además, es común que en este tipo de estudios, las hipótesis sean generadas a medida que se llevan a cabo los experimentos, mediante los descubrimientos del investigador y la recolección de datos, y de esta forma son afinadas constantemente.

En este tipo de estudios, el tamaño de muestra no suele ser un número fijo establecido *a priori*. En la indagación cualitativa, se establece un tipo de unidad de muestreo y de él es posible establecer un número aproximado de casos para llevar a cabo la experimentación. Este tipo de muestras suele tener un carácter flexible. En cuanto a los

datos por recolectar, son de interés conceptos, percepciones, interacciones y experiencias de los individuos bajo prueba, ya sea de manera grupal o individual. Las variables cualitativas contienen datos categóricos. Estos datos denotan cualidad y se clasifican en un número de categorías o clases. Es común obtener porcentajes o frecuencias a la hora de analizar este tipo de datos [53].

A continuación, en las tablas 2.1 y 2.2, se muestran algunos de los principales tipos de diseños cualitativos, sus características, y tamaños de muestra sugeridos. Es posible observar que para el presente proyecto, resulta especialmente útil un análisis fenomenológico, ya que en este se pretende describir y entender un fenómeno o proceso, desde el punto de vista de cada participante, y desde la perspectiva construida colectivamente, para lo cual [52] sugiere una muestra de al menos 10 individuos para las pruebas por realizar.

Tabla 2.1: Características de los principales diseños cualitativos. Adaptado de [52].

Características	Teoría Fundamentada	Etnográficos	Narrativos	Fenomenológicos	Investigación/ acción
Tipo de problema de investigación más apropiado para ser abordado por el diseño.	Cuando no se dispone de teorías o son inadecuadas para el contexto, tiempo, casos o muestra, circunstancias, etcétera.	Cuando se pretende describir, entender y explicar un sistema social. Por ejemplo, una comunidad, etnia, ciudad, grupo marginado, cultura organizacional, etcétera.	Cuando las historias detalladas ayudan a comprender el problema.	Cuando se busca entender las experiencias de personas sobre un fenómeno o múltiples perspectivas de este.	Cuando una problemática de una comunidad necesita resolverse y se pretende lograr el cambio.

Es posible destacar que al tratar de experimentos basados en herramientas de *software*, de igual manera es posible realizar análisis tanto cualitativos como cuantitativos (como se menciona previamente con respecto a los estudios fenomenológicos, en donde se busca el estudio de la experiencia del usuario en cualquier tipo de proceso). Al experimentar sobre *software*, para la recolección de datos puede resultar útil la implementación de encuestas, ya que estas permiten distinguir cuáles de las posibles variables por estudiar, pueden afectar otras variables en el sistema [54]. De esta manera, sería posible comprender cuáles de los dispositivos adquiridos representan una mayor facilidad de uso, toman menor tiempo de

Tabla 2.2: Tamaños de muestra comunes en estudios cualitativos. Adaptado de [52].

Tipo de estudio	Tamaño mínimo de muestra sugerido
Etnográfico cultural	Una comunidad o grupo cultural, 30-50 casos que lo conformen. Si es menor el grupo, incluir a todos los individuos o el mayor número posible
Etnográfico básico	Doce participantes homogéneos Si la unidad de análisis es observaciones, 100-200 unidades
Fenomenológico	Diez casos
Teoría fundamentada, entrevistas o personas bajo observación.	De 20 a 30 casos
Historia de vida familiar	Toda la familia, cada miembro es un caso
Biografía	El sujeto de estudio (si vive) y el mayor número de personas vinculadas a él, incluyendo críticos
Estudio de casos	De seis a 10. Si son en profundidad, tres a cinco
Grupos de enfoque	Siete a 10 casos por grupo, al menos un grupo por tipo de población. Si el grupo es menor, incluir a todos los individuos o el mayor número posible. Para generar teoría, tres a seis grupos

instalación, o son percibidos como intuitivos por medio de la experiencia del usuario. Las respuestas obtenidas permitirían conocer el impacto de los dispositivos sobre distintas características del sistema.

2.5.2 Análisis cuantitativo

La posibilidad de adquisición de múltiples dispositivos inteligentes, provenientes de diferentes proveedores, permite un análisis de forma cuantitativa en donde resulta óptimo analizar la confiabilidad de los dispositivos. Los dispositivos pueden llegar a ser analizados tanto en su conexión, como en su lectura y escritura, en donde se obtendrían datos binarios que indiquen una acción, conexión, o lectura exitosa o fallida. De esta manera es posible utilizar la ecuación 2.1 para determinación de tamaño de muestras [55]:

$$n = \left(\frac{Z^2}{E^2} \right) p(1 - p) \quad (2.1)$$

Donde:

- n es el tamaño de muestra.
- Z es el valor de la prueba Z para distribuciones normales reflejando el valor de

confianza que se desea utilizar.

- E es el margen de error que se permitiría.
- p es la probabilidad de éxito que se presentaría en las pruebas.

Para análisis de datos binarios, es posible asumir un valor de $p=0.5$ para obtener una maximización de la fórmula presentada, y encontrar el valor de muestra que sea posible desde un planteamiento conservativo. En [56] se plantea una tabla que contiene distintos valores de muestras posibles de obtener a partir de diferentes porcentajes de confianza y tolerancia. En la tabla 2.3 se muestran opciones para la selección de un tamaño de muestra tanto para un 90% o un 95% de confianza, presentando múltiples valores de tolerancia, o error aceptable (desde un 2 a un 10%).

Tabla 2.3: Tamaños de muestra para estudios cuantitativos con datos binarios. Adaptado de [56].

Tamaño de muestra requerido (90%)	Tamaño de muestra requerido (95%)	Nivel de tolerancia
1691	2401	2%
752	1067	3%
271	384	5%
68	96	10%

CAPÍTULO 3

METODOLOGÍA

A continuación se presenta la metodología utilizada para el desarrollo del proyecto. Se exponen los fundamentos teóricos del proceso seguido, además de las diferentes etapas que este representa y su adaptación para el proyecto.

3.1 Metodología de diseño

La metodología seguida en el proyecto corresponde a la establecida por Ulrich y Eppinger en *Diseño y Desarrollo de Productos* [57]. Este proceso de diseño en ingeniería implica el seguimiento de distintas etapas, que involucran aspectos como la comprensión del problema, determinación de necesidades para el cliente, generación de conceptos, desarrollo y evaluación del producto. En la figura 3.1 se observan las etapas que conforman el proceso seguido durante el proyecto.

Es importante recalcar que el proceso es en su totalidad iterativo, por lo cual al finalizar cada etapa se sugiere la reflexión, análisis y búsqueda de retroalimentación. De esta manera, será posible tomar en cuenta cualquier aspecto relevante descubierto

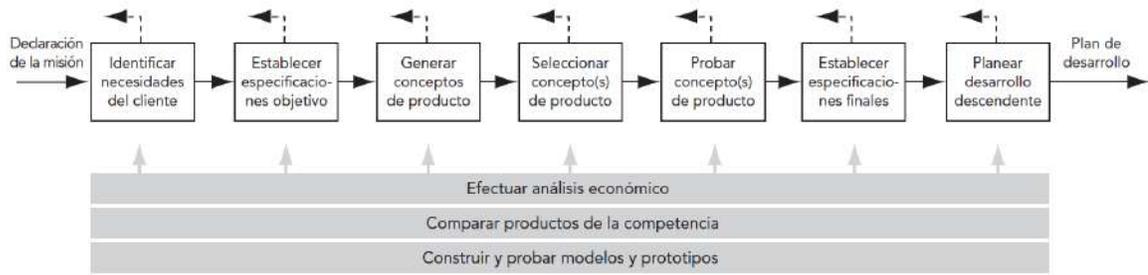


Figura 3.1: Etapas del Proceso de Diseño en Ingeniería. Recuperado de: [57]

durante cada etapa, que pueda ser usado como retroalimentación para un mejor desarrollo de etapas previas. En el presente proyecto, se toma como principal cliente al Dr. Francisco Bellas Bouza, profesor de la UDC y miembro del GII.

3.1.1 Determinación de necesidades y especificaciones

La fase de determinación necesidades se plantea para poder comprender los requerimientos del cliente, y qué características debe presentar la solución final del producto. Es importante especificar que se debe evitar el planteamiento de sesgos en el establecimiento de necesidades, de manera que el resto de la metodología no sea enfocada en una solución predeterminada que pueda afectar en el descubrimiento o propuesta de distintos o novedosos conceptos de solución. Inicialmente, se realizará al menos una entrevista con el cliente, de manera virtual, en donde se tendrá una conversación acerca de lo que considera sean requerimientos que debe cumplir el proyecto.

Se debe además tomar en cuenta que, para definir correctamente las necesidades del proyecto, puede resultar útil conocer la perspectiva de personas adicionales al Dr. Bellas que estén involucradas en el proyecto. Por ello, se contará también con la participación del Dr. Alejandro Paz López, quien aportará su perspectiva como miembro del GII y de MINT.

Tras establecer las necesidades, será posible definir un conjunto de especificaciones,

las cuales convierten cada necesidad en expresiones medibles con respecto al cumplimiento de necesidades. Las especificaciones se componen de métricas (en forma de expresiones cuantificables), y de valores objetivo (valores medibles, en ocasiones con unidades y rangos).

Las necesidades serán categorizadas, o evaluadas, de parte del cliente, por medio de la rúbrica expuesta a continuación, en la Tabla 3.1:

Tabla 3.1: Valores para jerarquización de necesidades.

Valor	Importancia
1	La función es indeseable, no quiero un producto con esta función.
2	La función no es importante, pero no me importa si el producto la tiene
3	Sería bueno tener esa función, pero no es necesaria.
4	La función es muy deseable, pero podría no tenerla.
5	La función es muy importante, no quiero un producto que no tenga esta función.

3.1.2 Generación de conceptos

En esta etapa, se cuenta con una componente de investigación importante. Se implementan búsquedas internas (que buscan los aspectos relacionados a la creatividad y pensamiento crítico del ingeniero), y búsquedas externas que requieren la investigación de todo tipo de herramientas existentes actualmente que representen posibles soluciones al problema. El proceso comienza por medio una subdivisión de problemas, con base a las distintas funciones del sistema. Para ello, se implementa un diagrama en forma de “caja negra”, en el cual se especifican los elementos tanto de entrada, como de salida que se requieren en el proyecto. Este diagrama adaptado para el presente proyecto, se muestra en la figura 3.2. En el diagrama, se tiene como energía a la alimentación que requiere cada dispositivo para su adecuado funcionamiento, el material es representado por todos los dispositivos y sensores que serán conectados al sistema, la señal de entrada corresponde al comando enviado por un usuario para solicitar a la plataforma una acción, y la señal de salida representa la señal enviada a un dispositivo para responder

ante un comando.

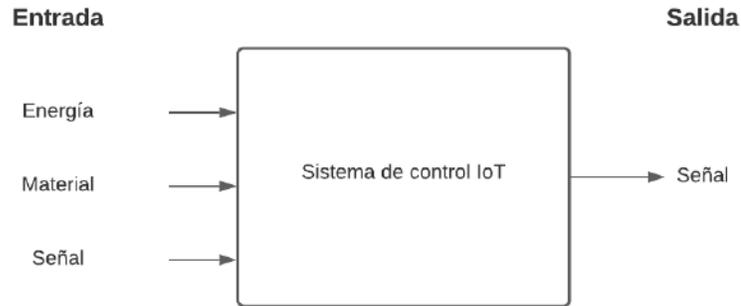


Figura 3.2: Representación de caja negra. Adaptado de: [57, p. 116]

Posteriormente, es posible analizar cada una de las diferentes funciones en el sistema, y junto con la búsqueda de información, plantear diferentes conceptos que permitan cumplir (algunos en diferente grado) con las necesidades del cliente.

3.1.3 Selección del concepto

Una vez que se cuenta con un listado de múltiples posibles conceptos, se procede a la etapa de selección de un concepto para continuar a la etapa de desarrollo. Para seleccionar dicho concepto, se trabaja por medio de dos procesos: filtrado y evaluación de conceptos. La etapa de filtrado sirve para descartar aquellos conceptos desfavorables, y continuar a la siguiente fase solamente con conceptos que cumplan en un mayor grado con las necesidades del cliente. Es posible descartar, combinar, o aprobar los conceptos para continuar a la siguiente fase. Para esto, se establece una matriz de evaluación en donde se elige un concepto como la base o referencia de evaluación. Los demás conceptos serán evaluados por medio de la siguiente simbología:

- + : concepto es mejor que la referencia en la categoría que se esté evaluando.
- - : concepto es peor que la referencia en la categoría que se esté evaluando.
- 0 : concepto es igual que la referencia en la categoría que se esté evaluando.

La etapa de evaluación de conceptos se lleva a cabo con los conceptos seleccionados de la etapa de filtrado. En esta etapa los conceptos son evaluados por medio de un proceso más preciso, donde se asigna una puntuación que permite elegir un concepto ganador para ser desarrollado. Se utiliza nuevamente un concepto de referencia, y se emplea la siguiente escala de evaluación:

- 1 -La función es notablemente peor o menor que la referencia.
- 2 -La función es ligeramente peor o menor que la referencia.
- 3 -Referencia.
- 4 -La función es ligeramente superior o mejor que la referencia.
- 5 -La función es notablemente superior o mejor que la referencia.

3.1.4 Desarrollo del concepto

Al tener un concepto evaluado como superior ante los demás, se procederá con el desarrollo de este para establecer un entorno en el área de IoT.

3.1.5 Pruebas de concepto

Esta etapa consiste en una serie de pruebas que permitirán la evaluación del grado de cumplimiento de las necesidades del cliente. Estas pruebas se apoyan tanto en las necesidades, como en las especificaciones del proyecto, y tendrán una naturaleza tanto cualitativa como cuantitativa.

3.2 Entregables del proyecto

Esta sección corresponde a todos aquellos documentos que sean producidos durante el proyecto, tanto para la adecuada documentación de este, como material diseñado

para el cliente. Inicialmente, el avance del proyecto será documentado en una bitácora del producto, la cual contendrá los resultados obtenidos a través de las distintas etapas del proceso de diseño en ingeniería. La bitácora se muestra en la sección del Apéndice A.

Seguidamente, debido a que el proyecto busca la creación de una herramienta para propósitos educativos, es necesaria la creación de un manual de usuario que permita tanto al cliente, como a futuros usuarios la comprensión del sistema, sus capacidades, y características relacionadas a instalación, *hardware* y *software* requerido, configuraciones, entre otros. El manual será detallado en la sección del Apéndice B. Además, para asegurar el cumplimiento de las necesidades del cliente, será necesario el diseño de una aplicación que sea capaz del control de dispositivos inteligentes y sensores. Para complementar, se presentará un video creado como demostración del funcionamiento de la aplicación diseñada para el proyecto.

CAPÍTULO 4

DISEÑO DE PLATAFORMA

Este capítulo abarca el proceso de diseño relacionado con la investigación de tecnologías, seguida por la selección y evaluación de conceptos, tomando en cuenta la naturaleza iterativa del proceso. Se comenzará mostrando los requerimientos solicitados por el cliente, y con base a estos, el proceso para la presentación, evaluación y selección de propuestas de solución. Adicionalmente, posterior a la selección de una plataforma de control IoT, se expone el proceso de selección de componentes o dispositivos inteligentes por utilizar en el proyecto.

4.1 Identificación de requerimientos

Inicialmente, por medio de investigación bibliográfica y entrevistas con el cliente, fue posible establecer las necesidades del proyecto. Las necesidades fueron obtenidas por medio de afirmaciones explícitas del cliente, y estas se resumen en la Tabla 4.1. En esta tabla se muestran las necesidades del proyecto en cuanto a los requerimientos de la herramienta por diseñar, su relación con el Robobo, y además necesidades con

respecto a precio y accesibilidad de dispositivos(facilidad para comprar y adquirir el producto). Además, por medio de una encuesta realizada al cliente, se define la importancia que tiene para él cada una de las necesidades. Dicha importancia se define bajo la jerarquización mostrada en la Tabla 3.1 de la sección 3.1.1.

Tabla 4.1: Tabla de necesidades.

No.	Necesidad	Importancia
1	El sistema tiene una interfaz gráfica	2
2	El sistema es capaz de realizar sus funciones objetivo correctamente fuera del Robobo Framework	4
3	El sistema permite su integración al Robobo Framework	4
4	El sistema será capaz de realizar sus funciones objetivo correctamente dentro del Robobo Framework	4
5	El sistema permite la lectura de los estados de los dispositivos inteligentes o sensores	5
6	El sistema permite la actuación sobre los estados de los dispositivos inteligentes o sensores	5
7	El sistema y sus sensores tienen un nivel de complejidad tal que sea posible su utilización por parte de personas sin un alto grado de conocimiento técnico	5
8	Se codifica la interacción con más de 2 dispositivos inteligentes o sensores	5
9	Los dispositivos inteligentes o sensores serán accesibles por plataformas internacionales que faciliten su distribución	5
10	Los dispositivos inteligentes o sensores serán de un precio accesible	5

De esta manera, es posible observar que para el cliente el funcionamiento de manera independiente de la herramienta es el principal requerimiento, es decir, que el producto opere de forma eficiente independientemente del *Robobo* y su *Framework*. Se puede observar en las necesidades número 3 y 4, que la integración es una función deseable, mas no indispensable. Es importante remarcar que la herramienta por implementar deberá permitir al usuario la capacidad de conexión, lectura y actuación de múltiples dispositivos. Además, se requiere que aquellos dispositivos por seleccionar sean posibles de adquirir con un precio accesible, y que estén disponibles fácil y regularmente en plataformas internacionales como por ejemplo *Amazon* o

Aliexpress, ya que esto facilitaría la replicabilidad del curso por impartir y sus experimentos en instituciones educativas.

Para poder establecer una referencia medible en cuanto a las necesidades, se establece un conjunto de especificaciones, que contienen distintas métricas y valores aceptables para el cumplimiento de cada necesidad. Las especificaciones se muestran en la Tabla 4.2, donde es posible notar distintos valores de medición como binario, numérico y subjetivo (se aplicará en la etapa de validación a la hora de evaluar la facilidad de uso del sistema, donde se contará con una escala adecuada para la verificación con el usuario). Adicionalmente, es posible observar la relación entre necesidades y métricas en la sección del Apéndice A, Tablas A.3 y A.4.

Tabla 4.2: Definición de Especificaciones.

No. Métrica	Métrica	Imp.	Unidades	Valor Marginal	Valor ideal
1	Error de lectura de dispositivos	5	%	5-10	<5
2	Error en actuación de dispositivos	5	%	5-10	<5
3	Error en conexión con dispositivos	5	%	5-10	<5
4	Integración con el <i>Robobo Framework</i>	4	Solicitud del cliente	Uso de Android Studio Función de lectura Función de actuación Función de listado	Uso de Android Studio Función de lectura Función de actuación Función de listado Función de automatización
5	Facilidad de utilización	5	Subjetivo	Fácil o neutro	Muy fácil
6	Cantidad de dispositivos	5	N: Número de dispositivos	2-4	>4
7	Costo total de dispositivos	5	€	500	500
			\$	590	590

4.2 Selección de plataforma

Inicialmente, el cliente establece que le gustaría explorar la posibilidad del diseño de una herramienta que permita la utilización del servicio de *Google Home* y *Google Assistant* como intermediario. Es decir, utilizar un medio que permita la comunicación, obtención de información y actuación sobre los dispositivos conectados por medio de los servicios mencionados.

4.2.1 Selección inicial

Se comienza explorando las posibilidades que ofrece *Google* con respecto a sus servicios, ya que su plataforma es propietaria (no es código abierto), sin embargo presentan la posibilidad de añadir funcionalidades personalizadas a herramientas de un usuario. Para ello, es posible utilizar *Smart Home Actions* [58], que tratan de expandir la funcionalidad de diferentes dispositivos conectados con *Google Home/Assistant* mediante *scripting*. Las *Smart Home Actions* permiten realizar funciones o acciones personalizadas de un usuario en dispositivos integrables con sus plataformas (por ejemplo, hacer parpadear una luz). Estas acciones son realizables en diferentes lenguajes, como Python, Java, o C++.

Sin embargo, estas acciones solamente permiten la expansión de la funcionalidad de una aplicación ya existente. Es decir, se requiere previamente la existencia de una herramienta o servicio que permita la conexión y manejo de dispositivos inteligentes totalmente independiente de *Google*, sea de manera local o por medio de la nube [59]. El servicio de acciones de *Google* solamente realiza una conexión con la herramienta externa, y de esta manera añade funcionalidad de voz a esta [60]. De esta manera, y por medio de una investigación exhaustiva de la documentación y las capacidades [61] [62] [63] [64] de *Google Home/Assistant/SDK*, se determina que la plataforma de control IoT de *Google* al ser propietaria, no permite ser accesada o controlada directamente con aplicaciones de terceros, pero sí es posible la extensión de una aplicación externa por medio de las *Smart Actions*. Así, se procede a investigar acerca de las posibilidades actuales para la creación de una herramienta personalizada que permita la comunicación con dispositivos inteligentes o sensores. Para ello, se investigan y comparan múltiples opciones.

Se puede observar en la Tabla 4.3 los conceptos que se consideran para una posible implementación. A continuación, se describen los conceptos planteados.

- Python: el uso de este lenguaje de programación ofrece la ventaja de ser una

herramienta ampliamente utilizada, capaz de ser integrable en numerosas aplicaciones con bibliotecas disponibles para ser integrada a servicios de terceros. Python permite la implementación con el *Google Assistant Service* [65] (previamente *Google Assistant SDK*), el cual representa la herramienta que dispone *Google* para integrar su asistente de voz en dispositivos externos a su empresa. Por medio de este servicio, es posible instalar el asistente de voz de *Google* en una *Raspberry Pi* utilizando Python, de manera que se pueden además controlar componentes compatibles con este dispositivo (leds, motores, cámaras, sensores...).

Adicionalmente, es posible la utilización de módulos de Python como *Flask* [66], un *Framework* para el desarrollo de aplicaciones web, para poder manipular componentes electrónicos por medio de una *Raspberry Pi* y de esta manera permitir el control por medio de internet de múltiples dispositivos [67]. Además, existen algunas adaptaciones de código abierto en la plataforma *Github* que permite el control de marcas específicas (aunque son pocas las implementaciones disponibles) como por ejemplo dispositivos inteligentes del vendedor *Xiaomi* [68], lo cual a pesar de permitir un mayor rango de posibilidades al utilizar un vendedor reconocido mundialmente, limita la implementación en cuanto a la variedad de dispositivos. La utilización de Python representa una alta curva de aprendizaje para el usuario que utilice una herramienta IoT, además de que este tipo de implementación no tendría una buena visualización (interfaz de usuario) comparada con otras opciones.

- **Ionic:** esta es una plataforma de código abierto programable en *Javascript*, que facilita la creación de aplicaciones móviles en diferentes sistemas operativos (e integrable a otros *frameworks* para creación de aplicaciones) [69]. El *framework* de *Ionic* puede ser utilizado junto con placas de control como por ejemplo un *ESP8266* junto con el software de *Arduino* para controlar componentes electrónicos [70] , e

incluso permitir un control por medio de plataformas como *Ubidots* que permiten el monitoreo IoT (aunque esta plataforma tiene un enfoque a nivel industrial) [71] [72]. Adicionalmente, Ionic es utilizado por desarrolladores de software que venden plantillas o *templates* de aplicaciones para control IoT [73], sin embargo es una alternativa que elimina la flexibilidad y el rango de temáticas educativas por explorar deseada por el cliente, además de involucrar un aumento en el coste del proyecto por la adquisición de la plantilla, requerimiento de contratación externa para resolución de problemas, y limitaciones a causa de licencias de distribución.

- Home Assistant, Tuya Smart y Google Home: como se ha descrito anteriormente, son plataformas para entornos IoT, que no dependen de marcas o dispositivos específicos (cada una de ellas tiene una cantidad diferente de distribuidores y dispositivos posibles de integrar). Tanto Tuya como Home Assistant son de código abierto, y permiten un mayor grado de flexibilidad y personalización al compararlas con Google Home, la cual es poco personalizable y es un software propietario. Tanto Tuya como Google Home son posibles de integrar dentro de Home Assistant.
- Android Studio: es el ambiente oficial para desarrollo de aplicaciones móviles de *Android* [44]. Es posible de programar tanto en el lenguaje de *Java* como de *Kotlin*. Esta plataforma cuenta con una extensiva documentación para su aprendizaje. Android Studio es comúnmente utilizado para entornos IoT. Su posibilidad de desarrollo de aplicaciones tanto de manera gráfica como programable permite a un desarrollador la creación de una aplicación con el grado de personalización que se desee. Android Studio es capaz de comunicarse con dispositivos inteligentes (por ejemplo, la plataforma Tuya ofrece parte de su código de manera abierta, para poder utilizarlo desde Android Studio), y también componentes electrónicos, por medio de sistemas externos para el manejo IoT. Es posible utilizar esta plataforma para el desarrollo de una

aplicación personalizada, donde se pueda utilizar la plataforma de Google Firebase (una base de datos que permite monitoreo a tiempo real de dispositivos [43]), y por medio del Software de Arduino junto con una placa ESP [45] [74] permitir una comunicación bidireccional para controlar dispositivos como sensores, motores, leds, y *switches* o relés (los cuales permitirían la conexión de cualquier electrodoméstico). Adicionalmente, esta plataforma presenta mayor facilidad de integración al *Robobo Framework*, debido a que este fue creado en Android Studio.

Tabla 4.3: Filtrado de conceptos inicial.

Criterios	Conceptos					
	Python	Home Assistant	Ionic	Android Studio	Tuya Smart	Google Home (REF)
Presentación	-	0	0	+	-	0
Funcionalidad independiente	0	0	0	0	0	0
Adaptación al Robobo Framework	+	+	-	+	+	0
Control sobre dispositivos	-	+	0	0	0	0
Facilidad para utilizar	-	-	0	-	-	0
Flexibilidad para crecimiento	-	+	-	+	-	0
Accesibilidad	0	0	0	+	-	0
—	—	—	—	—	—	—
Suma +	1	3	0	4	1	0
Suma 0	2	3	5	2	2	7
Suma -	4	1	2	1	4	0
Evaluación Neta	-5	2	-2	3	-3	0
Lugar	6	2	4	1	5	3
¿Continuar?	No	Sí	No	Sí	No	Sí

Se puede observar por medio de la tabla 4.3 los diferentes criterios establecidos, y la comparación entre cada uno de los conceptos, tomando como referencia Google Home. Es posible notar que conceptos como Python, Tuya Smart, y Ionic, a pesar de contar con herramientas útiles para el área de IoT, presentan una desventaja competitiva al lado de la referencia. Tanto Python como Tuya presentarían una interfaz de usuario poco atractiva o amigable. Además, la curva de aprendizaje en estas plataformas, sería una desventaja en cuanto a la facilidad de uso y flexibilidad para crecimiento. Por otro lado, Ionic presenta dificultades en la integración al *Robobo Framework* debido a su lenguaje de programación.

Es posible observar como Home Assistant, Android Studio y la referencia serán continuados para la etapa de evaluación de conceptos. Tanto Android Studio como Home Assistant son capaces de ser integrados al Robo Framework (Home Assistant por medio de una REST API y websockets, y Android Studio directamente integrable por medio de *Java*). Ambos permiten añadir una amplia cantidad de dispositivos, son altamente flexibles, y pueden ser utilizados en aplicaciones sencillas de aprender por parte de personas que no han tenido formación previa en la materia. En la tabla 4.4 se presenta la evaluación de los 3 conceptos seleccionados de la etapa de filtrado. En esta ocasión, no existe una única referencia. Se establece una referencia (el número 3, en negrita) que indica cuál concepto es más deseable como referencia para cada criterio. Es necesario definir un peso, en porcentaje, para cada criterio, con base a la jerarquización (o importancia) asignada por parte del cliente a las necesidades. Al establecer una calificación para cada criterio (como se menciona previamente, una escala del 1 al 5), se multiplica cada peso y calificación respectivamente para cada concepto, de manera que se genera una evaluación ponderada (mostrada como E.P en la tabla 4.4). La sumatoria de la evaluación ponderada permite definir cuáles criterios tienen una mayor importancia para cumplir con las necesidades del cliente, y así seleccionar un concepto ganador.

De esta manera es posible establecer como concepto seleccionado a la plataforma de Android Studio, con la cual se desarrollaría una aplicación personalizada para el GII, que pueda comunicarse por medio de *Firebase* con un dispositivo ESP8266, y de esta forma controlar múltiples componentes electrónicos. Además, al presentar una solución que operaría como aplicación independiente para control IoT, se podría expandir su funcionalidad por medio de las acciones de *Google* para integrar la solución al servicio del *Google Assistant*.

Tabla 4.4: Evaluación de conceptos inicial.

Criterio	Peso (%)	Concepto					
		Home Assistant		Android Studio		Google Home	
		Calificación	E.P	Calificación	E.P	Calificación	E.P
1. Presentación	5	4	0.2	3	0.15	2	0.1
2. Funcionalidad independiente	17	4	0.68	5	0.85	3	0.51
3. Adaptación al Robobo Framework	10	3	0.30	3	0.3	1	0.1
4. Control sobre dispositivos	17	3	0.51	3	0.51	1	0.17
5. Facilidad para utilizar	17	2	0.34	1	0.17	3	0.51
6. Flexibilidad para crecimiento	17	3	0.51	3	0.51	1	0.17
7. Accesibilidad	17	2	0.34	3	0.51	2	0.34
—	—	—	—	—	—	—	—
Total puntos	—	—	2.88	—	3.00	—	1.90
Lugar	—	—	2	—	1	—	3
¿Continuar?	—	—	No	—	Sí	—	No

4.2.2 Iteración del proceso

La solución de propuesta seleccionada en la sección 4.2.1 se presenta al cliente, sin embargo esta no es aceptada. El cliente establece que, a pesar de que la solución presenta una herramienta didáctica que cumpliría con los objetivos del curso por impartir, no se presenta la facilidad de utilización que se busca. En la solución propuesta, los usuarios (estudiantes colegiales) habrían tenido que realizar las conexiones necesarias físicamente; es decir, llevar a cabo el cableado necesario para conectar un ESP8266 y cualquier sensor y componente electrónico seleccionado (por medio de un manual de usuario que guiaría paso a paso al usuario para configurar el sistema). Sin embargo, el cliente indica que la situación anterior no es deseable, y debido a experiencias similares en el pasado que han tenido con instituciones educativas en España, no desean proceder con propuestas que impliquen cableado, conexiones de placas o microcontroladores. Además, al observar por medio de la

investigación realizada que la interacción con *Google Assistant* es notablemente limitada, se establece que no se continuará con la búsqueda de dicha implementación. De esta manera, el cliente indica que le resultaría llamativo una exploración a mayor profundidad de plataformas para el control IoT de código abierto (en la categoría de Home Assistant y Tuya Smart), y establecen que el criterio de selección de facilidad de utilización para el usuario deberá tener una mayor importancia.

Por este motivo, se descarta el concepto seleccionado en la tabla 4.4 de la sección 4.2.1, y se procede a realizar una iteración sobre la etapa de búsqueda externa e investigación. En esta ocasión, se enfocaría la exploración en plataformas para control y monitoreo en IoT, de manera que sea posible el uso de dispositivos inteligentes como bombillos, enchufes, sensores, y cámaras.

4.2.3 Selección final

Al comenzar la búsqueda, enfocando la investigación en plataformas para manejo de entornos IoT, es necesario tomar en cuenta todas las necesidades, especificaciones y criterios de selección que se analizaron en los conceptos de la iteración previa. Es necesario tomar en cuenta la facilidad que implica la plataforma, donde entran en juego factores como instalación, manejo y configuración de dispositivos. Además, es importante tomar en cuenta consideraciones en cuanto a la existencia de métodos para la integración con el *Robo Framework* y las distintas plataformas existentes. Es necesario investigar acerca de factores que permitan al sistema ser flexible para experimentar, y permitir su expansión con múltiples dispositivos. Adicionalmente, se busca que la plataforma permita dispositivos variados, y no sea limitado en cuanto a marcas y proveedores (los cuales sean posibles de adquirir tanto para España, como Costa Rica). También, se pueden considerar factores adicionales que impliquen un valor agregado como la disponibilidad de una aplicación móvil existente que extienda la funcionalidad de la plataforma.

En cuanto a las plataformas por evaluar, se detallan en las tablas 4.5 y 4.6, 10 plataformas distintas utilizadas en la actualidad para establecimiento, monitoreo y control de sistemas IoT. Se puede observar que nuevamente se consideran *Home Assistant* y *Tuya Smart*, ya que son parte de las tecnologías emergentes con mayor popularidad, utilización, integraciones y documentación actualmente. Se profundiza a continuación acerca de las plataformas:

- Home Assistant: tal y como se ha mencionado previamente, *Home Assistant* cuenta con herramientas que la hacen amigable con el usuario, tiene reconocimiento automático de más de 1500 marcas o proveedores, y además cuenta con métodos para integración al *Robo Framework* como websockets o una REST API. Home Assistant tiene visualización desde cualquier navegador con una interfaz de usuario, cuenta con aplicación móvil gratuita, y es controlable por medio de la línea de comando de cualquier computador en los diferentes sistemas operativos existentes (iOS, Windows, Linux, o utilización en *Raspberry Pi*). Adicionalmente, *Home Assistant*, al ser *Open Source* cuenta con la posibilidad de contribuciones de su comunidad de usuarios. De esta forma, se han creado una gran cantidad de bibliotecas o integraciones de dispositivos que no son incluidas de manera nativa por la plataforma, expandiendo aún más sus posibilidades en cuanto a tipos y marcas de dispositivos.
- Tuya Smart: cuenta con una aplicación móvil que facilita la conexión de dispositivos. Además, cuenta con una interfaz por medio del navegador que no requiere instalación de ningún *software* adicional. El manejo de esta plataforma se basa en un control programado, por lo cual, si el usuario desea darle un uso más personalizado al servicio de *Tuya*, deberá programar los comandos que desee realizar. Adicionalmente, el mercado principal de esta plataforma se encuentra en la plataforma de servicios de *AliExpress*, y presenta una cantidad limitada de proveedores ya que es un servicio emergente. Esta plataforma está

actualmente en el desarrollo de una integración que permita a sus dispositivos ser controlados por medio de *Home Assistant*.

- **Open Hab:** Permite la visualización en un navegador, además de aplicación móvil gratis, y la posibilidad de operación tanto de manera gráfica en su interfaz como programada por medio de la línea de comandos del computador en cualquier sistema operativo. Cuenta con diferentes interfaces de usuario descargables para diferentes usos, por lo que dependiendo del grado de complejidad que se le quiera dar a la plataforma, es posible la implementación de distintas interfaces de usuario. Como se menciona previamente, *OpenHab* depende de varios conceptos clave (*bindings, things, channels, items, rules*) los cuales deben ser configurados manualmente en cada dispositivo que se desee configurar, de manera que el proceso de conexión puede llegar a ser extenso.
- **Meet Hue:** es la aplicación móvil propietaria de la marca *Philips* que funciona para manejar los dispositivos de esta marca. La aplicación es amigable con el usuario y sencilla de utilizar, sin embargo se limita únicamente al control y comunicación con dispositivos *Philips* (luces inteligentes) [75].
- **Io Broker:** Es una plataforma que se utiliza desde el navegador. Su instalación no es recomendada en Windows, y además no posee una aplicación móvil. No presenta una interfaz amigable con el usuario. Se requiere la descarga de distintos módulos para las diferentes funcionalidades que se le quiera dar a la plataforma. Gran parte de la comunidad e información con respecto al uso de la plataforma se presenta en el idioma alemán, y la documentación oficial, para el año 2021, sigue en proceso de redacción, por lo que gran parte de la información disponible está incompleta [76].
- **Hubitat:** es una plataforma que permite la integración localmente de dispositivos inteligentes. Su funcionamiento e integraciones depende de un puente de control

(o *hub*), encargado de controlar, coordinar y comunicar los diferentes dispositivos. Posee una interfaz amigable con el usuario tanto en navegador como en aplicación móvil. La aplicación no es de código abierto, por lo que se presenta una limitante en cuanto a la personalización de la plataforma [77].

- AWS: es una plataforma para soluciones en IoT enfocada a nivel industrial. Ofrecen tanto servicios cobrados como gratuitos, en forma de módulos con funciones pre-establecidas. El software que se utiliza es propietario en su mayoría pero también ofrecen algunas funcionalidades de código abierto. El mantenimiento de algunos de sus servicios no está disponible ni en Costa Rica ni España.
- Particle: plataforma enfocada en soluciones industriales. Esta empresa enfoca su *software* para uso personalizado con su propio *hardware*, ya que tienen en venta dispositivos como SoCs para múltiples funcionalidades en el área IoT. Tiene aplicación móvil, uso en navegadores web, y un SDK para creación de aplicaciones en iOS y Android [78].
- Domoticz: es una plataforma programada, en javascript, de manera que se deben tener conocimientos básicos en el área para poder manejar dispositivos por medio de esta herramienta. La documentación es escasa, y presenta un proceso complejo en cuanto a la instalación y configuración. Tiene aplicación móvil, sin embargo requiere de un cobro para ser utilizada [79].
- Home Genie: plataforma emergente en desarrollo. Tiene una aplicación móvil que es cobrada, y aparte una interfaz en navegador. El control de dispositivos es programado, por lo cual se requiere experiencia previa. El uso de la plataforma se enfoca en implementaciones manuales con microcontroladores o SoC. Su documentación es escasa [80].

Adicionalmente, en la tabla 4.7 se observa a manera de resumen, algunas de las características adicionales que presentan los conceptos investigados.

Tabla 4.5: Iteración de filtrado de conceptos.

Criterios	Conceptos				
	OpenHab	Meet Hue	Home Assistant	IoBroker	Hubitat (REF)
Presentación	0	0	0	-	0
Funcionalidad independiente	0	-	0	0	0
Adaptación al Robobo Framework	+	-	+	0	0
Control sobre dispositivos	0	-	0	0	0
Facilidad para utilizar	+	+	+	0	0
Flexibilidad para crecimiento	0	-	0	-	0
Accesibilidad	0	-	0	-	0
—	—	—	—	—	—
Suma +	2	2	2	0	0
Suma 0	5	0	5	4	7
Suma -	0	5	0	3	0
Evaluación Neta	2	-3	2	-3	0
Lugar	1	5	1	5	3
¿Continuar?	Sí	No	Sí	No	No

Una vez que se filtran los conceptos, se decide proceder a la evaluación de conceptos con *Open Hab*, *Tuya Smart*, y *Home Assistant*. En la tabla 4.8 se observa la calificación obtenida para cada concepto. En esta ocasión, los pesos asignados son modificados de manera que se otorga una mayor importancia al criterio de facilidad de utilización de la plataforma. Al cliente se le presenta como propuesta la implementación de *Home Assistant* para la realización del proyecto, y se decide continuar con esta herramienta para la implementación.

Tabla 4.6: Iteración de filtrado de conceptos (continuación)

Criterios	Conceptos					
	Hubitat (REF)	Amazon Web Services	Particle	Tuya Smart	Domoticz	Home Genie
Presentación	0	0	-	-	0	0
Funcionalidad independiente	0	-	0	0	0	0
Adaptación al Robobo Framework	0	-	-	+	+	0
Control sobre dispositivos	0	0	0	0	-	0
Facilidad para utilizar	0	0	-	+	0	0
Flexibilidad para crecimiento	0	-	-	0	-	-
Accesibilidad	0	0	-	0	-	-
—	—	—	—	—	—	—
Suma +	0	0	0	2	1	0
Suma 0	7	4	2	4	3	5
Suma -	0	3	5	1	3	2
Evaluación Neta	0	-3	-5	1	-2	-2
Lugar	3	5	6	2	4	4
¿Continuar?	No	No	No	Sí	No	No

Tabla 4.7: Características adicionales de conceptos.

Plataforma	Sistema Operativo	Dispositivos/marcas asociadas	App móvil
OpenHAB	Windows,Mac,Raspberry Pi,iOS,Android	~350	Sí
MeetHue	iOS, Android	Solo dispositivos Hue	Sí
Home Assistant	Windows,Mac,Raspberry Pi,iOS,Android	~1800	Sí
Hubitat	Navegador,iOS,Android	42 marcas, múltiples dispositivos	Sí
Tuya Smart	Navegador,Raspberry Pi,iOS,Android	~5000 marcas	Sí
AWS	Windows, Linux	NA	No
Particle	Windows,Mac,Raspberry Pi,iOS,Android	Solo productos propios	Sí
Domoticz	Mac,Raspberry Pi,iOS,Android	~20 marcas/integraciones	Cobrada
IO Broker	Mac,Raspberry Pi,iOS,Android	~430	No
Home Genie	Windows,Mac,Raspberry Pi,iOS,Android	~8 servicios	Cobrada

Tabla 4.8: Evaluación final de conceptos.

Criterio	Peso (%)	Concepto					
		Open Hab		Home Assistant		Tuya Smart	
		Calificación	E.P	Calificación	E.P	Calificación	E.P
1. Presentación	5	3	0.15	3	0.15	2	0.1
2. Funcionalidad independiente	15	3	0.15	3	0.45	2	0.3
3. Adaptación al Robobo Framework	10	3	0.3	3	0.3	3	0.3
4. Control sobre dispositivos	15	3	0.45	3	0.45	1	0.15
5. Facilidad para utilizar	25	2	0.5	3	0.75	1	0.25
6. Flexibilidad para crecimiento	15	3	0.45	3	0.45	2	0.3
7. Accesibilidad	15	3	0.45	3	0.45	2	0.3
—	—	—	—	—	—	—	—
Total puntos	—	—	2.75	—	3.00	—	1.70
Lugar	—	—	2	—	1	—	3
¿Continuar?	—	—	No	—	Sí	—	No

4.3 Selección de componentes

Una vez seleccionada la plataforma por utilizar en el proyecto, es posible el análisis de dispositivos (integrables en *Home Assistant*) disponibles en el mercado. Es deseable la adquisición de dispositivos que sean integrables de manera nativa con el *software* de *Home Assistant*, es decir, que puedan ser reconocidos por la herramienta con solo ser conectados o enchufados en una habitación. No obstante, existen distintos dispositivos que requieren de integraciones adicionales (llamadas *Add-ons*) en *Home Assistant* o de hardware externo para su manejo o reconocimiento (*hubs*), por lo cual se debe tomar en cuenta los requerimientos para cada dispositivo. Se decide adquirir dispositivos bajo distintas categorías:

- Luces
- Cámaras

- Sensores: de movimiento, de ventanas/puertas, temperatura y humedad.
- Conectores (enchufes, o *Smart Plugs*).
- Parlantes inteligentes.

Para ello, se realiza una búsqueda de marcas comunes, confiables, y económicas, que sean posibles de integrar con *Home Assistant* y además estén disponibles en *Amazon* tanto para Costa Rica, como para España. Los dispositivos seleccionados, junto con sus respectivas marcas, se muestran en la tabla 4.9.

Tabla 4.9: Dispositivos seleccionados

Dispositivo	Proveedor
Luces	Lifx
	Yeelight
Conector	D-Link
	Sonoff
	Tapo
Sensor de humedad/temperatura	Sonoff
Sensor de movimiento	Aqara
	Sonoff
Sensor de puerta	Sonoff
Cámara	Tapo
	Foscam
Parlante	Amazon Echo
Adicional	
USB Zigbee Sniffer	Variados
USB Conbee II	Phoscon
Tarjeta MicroSD	Sandisk

4.4 Solución por implementar

De esta manera, se plantea el desarrollo de la solución propuesta detallada a continuación:

- Instalación y configuración de *Home Assistant*, de manera que se obtenga el establecimiento de un entorno de control IoT por medio de esta plataforma.
- Establecimiento de conexión de cada uno de los dispositivos adquiridos (tabla 4.9) con *Home Assistant*, y comprobación de funcionamiento de estos.

- Diseño de una aplicación móvil utilizando Android Studio, que sea capaz de comunicarse por medio de la REST API de *Home Assistant*, y realizar funciones de lectura y actuación sobre los dispositivos conectados en la plataforma. Esta aplicación será la herramienta utilizada por el GII para integrar al *Robobo Framework*.

En la figura 4.1 se muestra la estructura de la solución descrita. Es posible observar que inicialmente, cada dispositivo deberá ser conectado por medio de su respectiva aplicación móvil, lo cual permitiría a *Home Assistant* el reconocimiento de estos y la integración dentro de la plataforma de control IoT. *Home Assistant* estará encargado de la lectura de estados, cambio de estados sobre dispositivos, y la visualización de cada dispositivo. Adicionalmente, la aplicación móvil por diseñar se comunicará con *Home Assistant*, y le enviará a esta plataforma los comandos deseados por el usuario tanto para lectura como actuación de dispositivos. *Home Assistant* recibirá las señales enviadas por la aplicación, y responderá con las acciones o envío de información correspondiente. Este proceso será descrito con mayor detalle en los capítulos 5 y 6. Es necesario aclarar que la solución propuesta se plantea de manera que se utilice la plataforma localmente, es decir, con los dispositivos, *Home Assistant*, y la aplicación móvil conectados bajo una misma red *Wifi*.

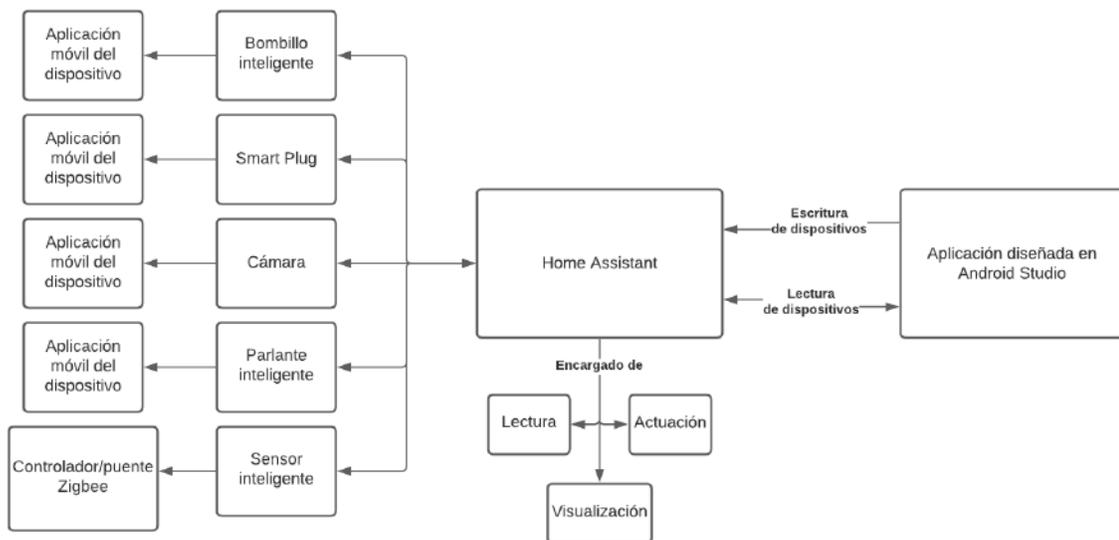


Figura 4.1: Diagrama de solución por implementar. Fuente: elaboración propia

CAPÍTULO 5

IMPLEMENTACIÓN DE PLATAFORMA

En este capítulo se detalla el proceso para la implementación de un entorno IoT con la plataforma seleccionada: *Home Assistant*. Se profundiza acerca de los requerimientos para la instalación y configuración, inicialmente para un único dispositivo a manera de comprobación funcional de la plataforma. Una vez comprobado el funcionamiento adecuado, se procede a la conexión de la totalidad de dispositivos (listados en la tabla 4.9).

5.1 Aplicación piloto

Se plantea la creación de una aplicación piloto, en donde sea posible la creación de un entorno IoT en *Home Assistant* para una máquina virtual en *Windows 10*. Para ello, se utilizará un dispositivo de tipo conector o enchufe inteligente, operado por wifi, de la marca *Sonoff* (modelo *Sonoff S26 Wifi Smart Plug*), mostrado en la figura 5.1. Este dispositivo funciona como un *switch*, de manera que es posible conectarse a cualquier enchufe de la red eléctrica de un hogar (en el caso de Costa Rica, a 120 V). Una vez

conectado el *Smart Plug* a un enchufe, se puede conectar cualquier dispositivo que requiera una conexión a la red eléctrica del hogar, y que no sobrepase una corriente máxima de 10 A. Vale la pena destacar que este *smart plug* tiene una configuración de pines distinta a los utilizados en España, debido a su compra por medio de proveedores en Costa Rica. Sin embargo, este dispositivo se encuentra disponible también por medio de la división española de *Amazon*, en donde es posible su adquisición con los pines adaptados para los estándares de conexión europea.



Figura 5.1: *Sonoff S26 Smart Plug* para aplicación piloto. Fuente: elaboración propia

5.1.1 Instalación de Home Assistant

Para la instalación de *Home Assistant*, se realiza el proceso detallado en el Manual de usuario (elaboración propia) de la sección del Apéndice B. Para este proceso, a manera de resumen, se siguen los siguientes pasos:

- Descarga de máquina virtual *Virtual Box*.
- Descarga de imagen o software de *Home Assistant* para la máquina virtual.
- Instalación y configuración de *Virtual Box*.

- Inicialización de Home Assistant en la máquina virtual, en donde se obtiene la dirección IP para el acceso a la plataforma en el navegador.
- Acceso a la plataforma web de *Home Assistant* y creación de una sesión.

Al concluir el seguimiento de los pasos anteriores, se cuenta con la plataforma de *Home Assistant* lista para su uso, con su interfaz gráfica de usuario (Dashboard) en su aspecto pre-determinado y sin dispositivos añadidos hasta este punto, como se observa en la figura 5.2. En esta figura, se pueden observar las principales pestañas (columna izquierda) que se utilizarán en *Home Assistant*. A continuación, se explican las pestañas de uso común:

- *Overview*: es el resumen de productos. Una interfaz en donde se ve información resumida en forma de tarjetas o botones, de cada uno de los dispositivos que se agreguen a la plataforma.
- *Developer Tools*: permite ver información a un nivel más técnico de los dispositivos agregados. Es posible, entre otras funciones, observar la forma en que cada dispositivo está internamente programado en *Home Assistant*, sus especificaciones, y modificar el nombre de dispositivo.
- *Supervisor*: es el módulo que contiene el catálogo de integraciones nativas de *Home Assistant*. Es decir, contiene todos los módulos oficiales, creados por los desarrolladores de la plataforma, que permiten expandir la funcionalidad de *Home Assistant*.
- *Configuration*: contiene todas las opciones para la configuración de la plataforma. Cuenta con opciones como : dispositivos, integraciones, automatizaciones, cuenta de perfil, servidores, entre otros.

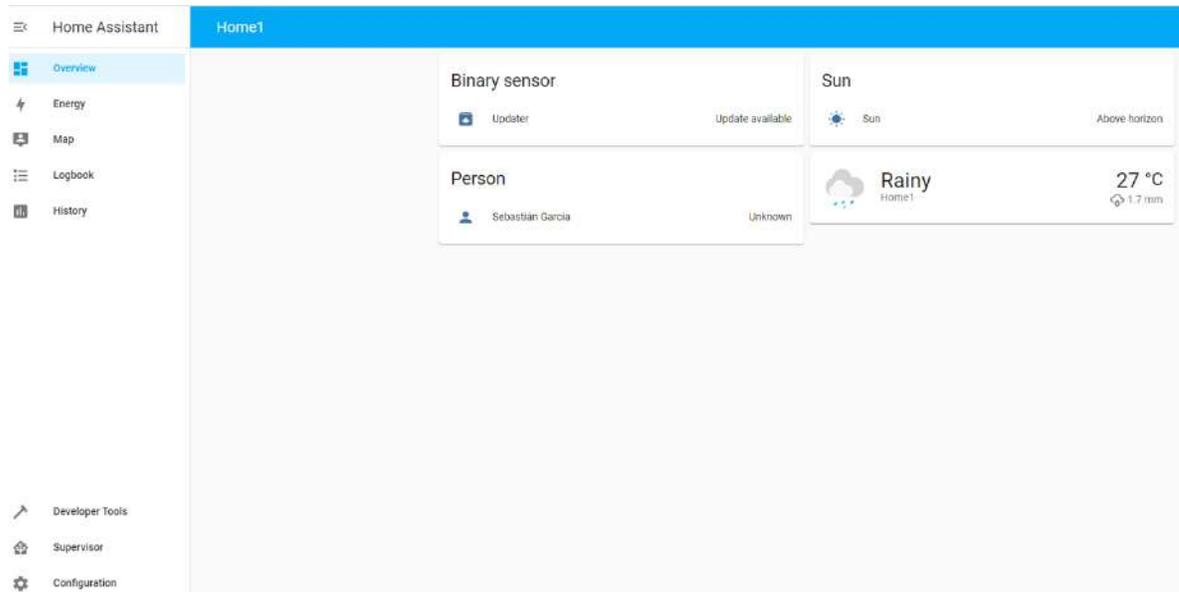


Figura 5.2: Dashboard pre-determinado de *Home Assistant*. Fuente: elaboración propia

5.1.2 Conexión del dispositivo

Para la prueba inicial con el dispositivo *Sonoff*, y al tratarse de una aplicación piloto previa a la adquisición de los demás dispositivos inteligentes, se requieren de algunos pasos adicionales para la conexión del *Smart Plug*. Para ello, es necesario:

- Adicionar el módulo integrable de *Home Assistant Community Store* (HACS) [81], el cual permite añadir a *Home Assistant* funcionalidades creadas por la comunidad de usuarios, que expanden los servicios y productos disponibles de la plataforma.
- Por medio de HACS, añadir el módulo de dispositivos *Sonoff*.
- Descargar el módulo de *File Editor* que permite el acceso a los archivos de configuración de *Home Assistant*.
- Crear una cuenta en la aplicación móvil propietaria para la operación del *Smart Plug*, llamada *eWeLink*.
- Dentro del módulo del *File Editor*, añadir 3 líneas de código que representan el

acceso a la cuenta de *eWeLink* para la conexión con *Sonoff*, como se observa en la figura 5.3.



Figura 5.3: Configuración en Home Assistant de Dispositivo Piloto. Fuente: elaboración propia

Una vez realizados los pasos anteriores, sería posible encontrar en el módulo de resumen de productos (*Overview*) una instancia de switch para la activación del *Smart Plug* como se puede observar en la figura 5.4.

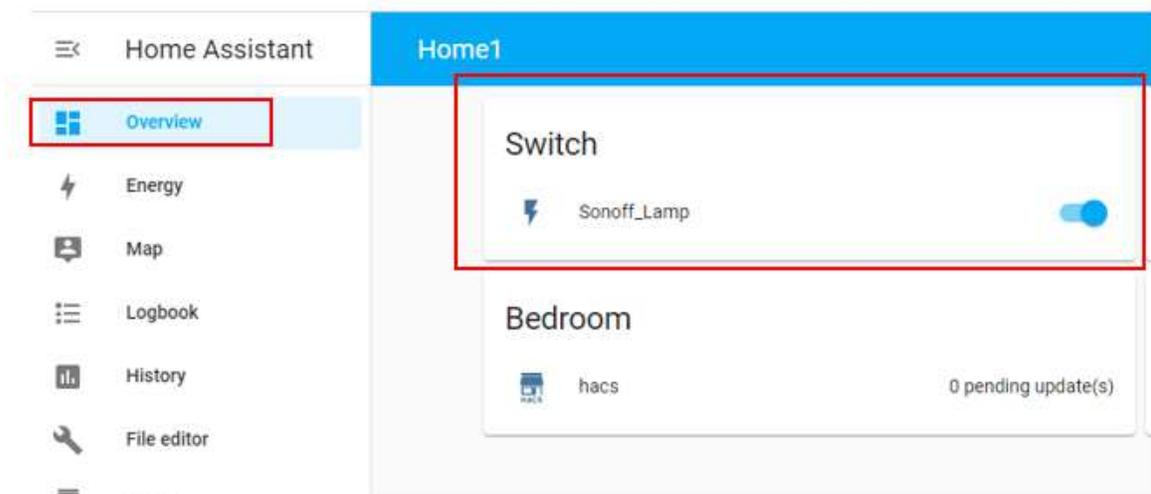


Figura 5.4: Dispositivo piloto *Sonoff Smart Plug* integrado en *Home Assistant*. Fuente: elaboración propia

5.1.3 Validación de aplicación piloto

Una vez que se establece la instalación del entorno IoT y la conexión apropiada del dispositivo *Sonoff*, se procede a realizar una etapa para validar la aplicación piloto y su adecuada funcionalidad. De esta manera se plantean 3 pruebas, para las cuales en cada ocasión se realizarán 68 iteraciones del proceso por evaluar. Este valor se define por medio de la tabla 2.3 de la sección 2.5.2, ya que se busca un tamaño de muestra con una confianza del 90% con un nivel de tolerancia de un 10%. Se describen las pruebas a continuación:

- Prueba de conexión: se propone desconectar el dispositivo de *Home Assistant*. Una vez desconectado por completo, realizar nuevamente la conexión e integración con la plataforma, y confirmar la adecuada conexión por medio de una actuación exitosa (encendido y apagado de dispositivo). Se obtiene el porcentaje de error a través de las pruebas.
- Prueba de lectura: la prueba se realiza con el dispositivo conectado adecuadamente en la plataforma y en su aplicación *eWeLink*. El objetivo de la prueba es cambiar el estado del dispositivo desde su aplicación móvil, y observar que *Home Assistant* realiza la lectura adecuada del cambio de estado del dispositivo. Para ello, se enciende o apaga el dispositivo, y se confirma el estado correspondiente. Se busca obtener el porcentaje de error en cuanto a la lectura del dispositivo.
- Prueba de actuación: la prueba se realiza con el dispositivo conectado adecuadamente en la plataforma y en su aplicación *eWeLink*. Se busca comprobar que *Home Assistant* es capaz de actuar correctamente sobre el dispositivo, tanto en su estado de encendido como de apagado. Para ello se realizan 68 iteraciones (una iteración implica un accionamiento sea encendido, o apagado), y se comprueba que en cada una el dispositivo cambie de estado

correctamente al accionar el comando correspondiente desde *Home Assistant*. Se busca obtener el porcentaje de error a través de las iteraciones.

Los resultados de la validación de la aplicación piloto se encuentran en las tablas A.11,A.19 y A.23 en la sección del Apéndice A. Es posible observar que el porcentaje de error en las 3 pruebas, resumido en la tabla 5.1, fue significativamente bajo, presentando menos de un 2% de error. Lo anterior implica que de las 68 pruebas realizadas tanto para lectura como para escritura, únicamente en una ocasión la plataforma detectó una lectura o cambio de estado incorrecto, mientras que en la prueba de conexión únicamente en una ocasión se realizó incorrectamente la integración en la plataforma. De esta manera, se confirma que *Home Assistant* permite una alta confiabilidad en la creación de la aplicación piloto en cuanto a la conexión, lectura, y actuación de un dispositivo inicial, tipo enchufe inteligente.

Tabla 5.1: Resumen de porcentaje de error en dispositivo Sonoff para prueba piloto

Dispositivo Sonoff en Home assistant	
Prueba	Error (%)
Lectura	1,471
Escritura	1,471
Conexión	1,471

5.2 Creación de entorno completo

Una vez adquiridos los dispositivos, se procede con el montaje -uno a la vez- en *Home Assistant*. Se muestra en la figura 5.5 un diagrama que representa el funcionamiento de la plataforma una vez que se establezca un entorno funcional con los dispositivos adquiridos. Se puede observar que una vez que se crea una instancia o sesión de *Home Assistant* y se conectan los dispositivos, la plataforma se encargará de manejar las solicitudes del usuario. Esta decidirá si lo que se desea es leer un estado (en donde se comunicará con el dispositivo respectivo y mostrará de vuelta la respuesta de la lectura

al usuario), o si se desea actuar sobre un dispositivo (en este caso, se enviará el cambio de estado adecuado al dispositivo).

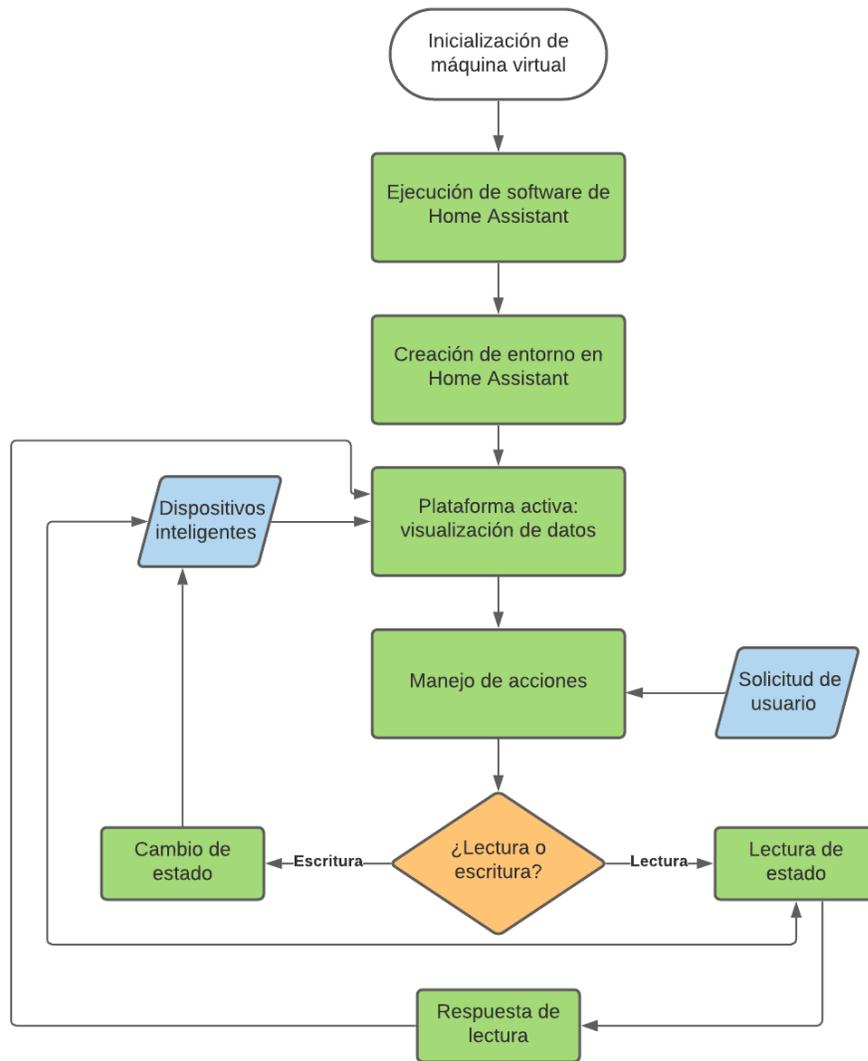


Figura 5.5: Diagrama de flujo del funcionamiento de *Home Assistant*. Fuente: elaboración propia

A continuación se describe el proceso de montaje, y aspectos relevantes a cada uno de los dispositivos.

5.2.1 Dispositivos de luz

Para la instalación de ambos dispositivos seleccionados, marca *Yeelight* y *Lifx*, se cuenta con una integración nativa (o directa) de *Home Assistant*. Es decir, la plataforma cuenta con una biblioteca que detecta automáticamente cualquier dispositivo inteligente de luz que esté conectado a la red *Wifi* del entorno. De esta manera, solo es necesario conectar cada dispositivo con su aplicación móvil respectiva, y seguidamente *Home Assistant* detecta sus presencias para ser integradas. En la figura 5.6 se observan los dispositivos. Los dispositivos al ser integrados, reciben un nombre y un código de identificación llamado *entity ID*, los cuales pueden ser personalizados. El *entity ID* es de la forma : light.nombre (donde “nombre” representa la porción del código que puede ser modificada por el usuario).



Figura 5.6: Dispositivos de luz Yeelight (izquierda) y Lifx (derecha). Fuente: elaboración propia

5.2.2 Conectores (*smart plugs*)

Para los dispositivos *smart plugs*, mostrados en la figura 5.7 no se cuenta con integraciones nativas en *Home Assistant*, por lo que se debe recurrir a las integraciones creadas por la comunidad de usuarios de la plataforma, con la herramienta HACS. Tal y como se menciona previamente con la conexión del dispositivo *Sonoff Smart Plug*, el dispositivo marca *Tapo* debe ser integrado a través de la biblioteca de *Tapo* en HACS. Nuevamente, es necesario realizar la conexión primero con su aplicación móvil, y seguidamente es posible añadirlo a *Home Assistant* por medio de la integración mencionada previamente. En este caso es importante mencionar que se debe obtener la dirección IP del dispositivo (la cual puede ser observada por medio de la aplicación móvil) ya que esta será necesaria para añadir el dispositivo a la plataforma.



Figura 5.7: Dispositivos Smart Plugs Tapo (izquierda) y D-Link (derecha). Fuente: elaboración propia

Por otro lado, el dispositivo marca *D-Link* no es capaz de ser integrado, ya que la integración nativa de *Home Assistant* que permitía la adición de esta marca, ya no es mantenida por la plataforma. De esta manera se cuenta con la posibilidad de conexión de *smart plugs* únicamente de las marcas *Sonoff* y *Tapo* para el proyecto. Para este

tipo de dispositivos, el *entity ID* es de la forma : switch.nombre (donde “nombre” representa la porción del código que puede ser modificada por el usuario).

5.2.3 Sensores

Los sensores disponibles para *Home Assistant* operan bajo los protocolos *Zigbee* o *Z-Wave*. De esta manera, el uso de estos protocolos implica que es necesario el uso de un coordinador (también llamado puente o *hub*) que se encargue de establecer y coordinar la comunicación entre cada sensor y la plataforma objetivo (en este caso *Home Assistant*). Tanto las marcas de *Aqara* como *Sonoff*, de las cuales se adquieren los sensores mostrados en la figura 5.8, cuentan con su propio *hardware* para el control *Zigbee*. No obstante, se decide el control por medio de dispositivos USB, los cuales funcionan como coordinadores “universales” *Zigbee* al no depender de una única marca y permitir la comunicación entre dispositivos tanto *Sonoff* como *Aqara*.



Figura 5.8: Sensores utilizados (varios). Fuente: elaboración propia

Es necesario tomar en cuenta, inicialmente, que se debe configurar la máquina virtual de manera que los periféricos en los puertos de la computadora, sean compartidos por la máquina virtual, y así permitir a *Home Assistant* la funcionalidad de los controladores *Zigbee* USB. Se utiliza en el entorno el dispositivo USB *Conbee II*, donde una vez que este es activado en la máquina virtual, es descubierto automáticamente por *Home*

Assistant. Posteriormente, se utiliza una integración nativa de la plataforma, llamada *Zigbee Home Automation*, que permite añadir el coordinador *Zigbee* y seguidamente integrar cualquier sensor que se desee. Para cada sensor, solamente se requiere activar el modo de sincronización (*pairing mode*) de la manera indicada en sus correspondientes instrucciones, y así serían rápidamente integrados los 4 sensores con *Home Assistant* por medio del *Conbee II* junto con la integración de *Zigbee Home Automation*. Para este tipo de dispositivos, el *entity ID* es de la forma : sensor.nombre o binary_sensor.nombre (donde “nombre” representa la porción del código que puede ser modificada por el usuario).

Adicionalmente, se adquiere el dispositivo USB *Zigbee Sniffer* como coordinador *Zigbee*. Este se adquiere a manera de experimentación ya que posee las siguientes características:

- Representa una forma, similar a la implementación del *Conbee II*, de controlar dispositivos que utilizan protocolo *Zigbee* sin depender de la marca de cada sensor.
- Es un dispositivo que no pertenece a una marca oficial. Es vendido por diferentes proveedores, por lo cual su presentación, funcionamiento y precio puede variar.
- Es posible de encontrar a precios tan bajos como 10 dólares en mercados como *Amazon*.
- No garantiza una alta confiabilidad, debido a la variabilidad presente en cada proveedor distinto. Sin embargo, el dispositivo podría representar una solución eficiente y significativamente económica.

Este dispositivo es capaz de operar bajo las mismas condiciones que el *Conbee II*, es decir, siendo conectado a la máquina virtual, y tras ser reconocido en *Home Assistant*, implementado con la integración de *Zigbee Home Automation*. Sin embargo, *Home Assistant* no fue capaz de tan siquiera reconocer el dispositivo entre su *hardware* permitido. Esto se relaciona directamente con la falta de confiabilidad

inicialmente planteada del producto, ya que tanto su hardware como su software puede presentar defectos que no permitan al componente de la plataforma identificar el producto ni su funcionalidad. De esta manera, se establece que el dispositivo *Conbee II* representa una mayor confiabilidad, no solamente al presentar un funcionamiento estandarizado, sino que al pertenecer a un proveedor establecido, este garantiza soporte, una amplia documentación, calidad en su producto, y hasta software propietario para la prueba de sus dispositivos de manera independiente.

En la figura 5.9 se muestran ambos dispositivos USB, y en la figura 5.10 se muestra con mayor detalle cada uno de los sensores.



Figura 5.9: Coordinadores Zigbee: Zigbee Sniffer (izquierda) y Conbee II (derecha). Fuente: elaboración propia

5.2.4 Cámaras

Las distintas cámaras adquiridas para el proyecto, figura 5.11, presentan dos métodos de conexión: una presenta una integración nativa, y la otra requiere la integración por medio de HACS. Inicialmente, ambos dispositivos deben ser configurados por medio de sus aplicaciones móviles respectivas (*Tapo* y *Foscam*). Seguidamente, dentro de *Home Assistant*, para la cámara *Tapo* se requiere descargar la integración de HACS, pero posterior a este procedimiento, ambas cámaras pueden



(a) Sensor de puertas o ventanas *Sonoff*



(b) Sensor de temperatura *Sonoff*



(c) Sensor de movimiento *Sonoff*



(d) Sensor de movimiento *Aqara*

Figura 5.10: Sensores implementados en *Home Assistant*. Fuente: elaboración propia

ser añadidas directamente en la plataforma. Es necesario recalcar que se requiere obtener la dirección IP de ambos dispositivos para permitir a *Home Assistant* el reconocimiento adecuado de los dispositivos. Para este tipo de dispositivos, el *entity ID* es de la forma : *camera.nombre* (donde “nombre” representa la porción del código que puede ser modificada por el usuario).

Al conectar las cámaras exitosamente en la plataforma, se observa una diferencia entre ambas. La integración de la cámara tipo *Tapo* permite ser visualizada “en tiempo real”, es decir su video se puede ver de manera fluida en la interfaz de usuario de *Home Assistant*. Por otro lado, la integración de la cámara *Foscam* no permite la transmisión

de video directamente, solamente es capaz de mostrar imágenes captadas por la cámara por cada 10 segundos de video. Esta diferenciación puede significar una desventaja al comparar las dos marcas adquiridas, y podría implicar para el cliente una preferencia por la cámara *Tapo*.



(a) Cámara *Foscam*



(b) Cámara *Tapo*

Figura 5.11: Cámaras utilizadas en *Home Assistant*. Fuente: elaboración propia

5.2.5 Parlantes

En lo que respecta a la categoría de parlantes inteligentes, se elige el *Echo Dot gen 3* de *Amazon*. Este parlante cumple la función de un asistente personal, el cual cuenta con bibliotecas que permiten expandir su funcionalidad. De esta manera es posible integrar el paquete de *Home Assistant*, el cual permite que la asistente personal de *Amazon*, llamada Alexa, reconozca y sea capaz de actuar sobre cada uno de los dispositivos ya conectados en *Home Assistant*. Esta funcionalidad permite al parlante de *Amazon* el control sobre una mayor cantidad de dispositivos, los cuales no requieren tener una integración con Alexa y solamente necesitan ser capaces de integrarse a *Home Assistant*. Se debe recalcar que este dispositivo, a pesar de establecer una comunicación con *Home Assistant*, funciona independientemente de la plataforma. Es decir, *Home Assistant* no es capaz de manipular o controlar el *Echo Dot*, solamente puede proveer

a este con la información necesaria de aquellos dispositivos conectados para que Alexa pueda controlar. El dispositivo se muestra en la figura 5.12.



Figura 5.12: Parlante inteligente *Echo Dot*. Fuente: elaboración propia

CAPÍTULO 6

INTEGRACIÓN CON EL FRAMEWORK

Se presentan a continuación los requerimientos para asegurar una integración a futuro con el *Robobo Framework* por parte del grupo MINT. Se exponen las posibilidades para establecer una comunicación entre *Home Assistant* y el *Robobo*, además de la selección y comprobación de comunicación con *Home Assistant* desde un medio externo. Finalmente, tras comprobar la comunicación mencionada, se plantean pruebas de validación del método seleccionado, con el fin de comprobar si este es adecuado para la integración para el *Robobo*.

6.1 Métodos de integración

El grupo del GII desea asegurar que sea posible de realizar a futuro una integración del presente proyecto, con el proyecto *Robobo*. Para ello, a través del proceso de diseño en ingeniería es necesaria la investigación acerca de los posibles métodos existentes para facilitar una comunicación o interacción entre ambos proyectos. En *Home Assistant*, se establece por medio de la documentación de la plataforma, que existen métodos para

interactuar con *Home Assistant* a través de aplicaciones externas. Para ello es posible utilizar tanto la REST API de la plataforma o utilizando *websockets*. Se debe destacar que, dentro del alcance del proyecto, no se contempla la integración dentro del *Robo Framework*. Sin embargo, sí se establece como objetivo asegurar que la herramienta seleccionada cuente con la capacidad y los medios para proveer al GII con un método para poder realizar dicha integración.

6.2 Diseño de aplicación

Se establece que se procederá con la utilización de la REST API de *Home Assistant*, ya que se presenta una mejor -y más amplia- documentación y material didáctico para comprender este método, al compararlo con la alternativa de la implementación de *Websockets*. Para ello, se procede con el diseño y creación de una aplicación en la plataforma de *Android Studio*, debido a que el *Robo Framework*, sus extensiones, bibliotecas, y aplicaciones, se encuentran escritas en Java y por medio de *Android Studio*.

Se establece por medio de las necesidades del cliente, que la aplicación por diseñar deberá ser capaz de:

- Permitir la visualización de una lista de todos los dispositivos conectados a *Home Assistant*, con el motivo de permitir al usuario conocer las conexiones activas en la plataforma sin tener algún conocimiento previo de estas.
- Actuar sobre dispositivos: debe ser necesario actuar sobre aquellos dispositivos que lo permitan, como sería el caso de las luces inteligentes, y de los *Smart Plugs* que permiten el cambio entre estados (*on/off*).
- Lectura de dispositivos: debe ser posible la lectura de los estados de cualquier dispositivo que se desee observar. Algunos dispositivos solamente podrán

presentar esta función (es decir, no podrán ser actuados, solamente leídos), como lo sería el caso de los sensores, los cuales debido a su naturaleza, solamente podrán ser activados por acciones externas en el ambiente, y no de manera programada.

Para lograr alcanzar la comunicación externamente con *Home Assistant*, se utilizan llamados *GET* y *POST*. El llamado *GET* se encarga de la obtención de información (las funciones referentes a la lectura de dispositivos), y el llamado *POST* se encarga del envío de información (la función referente a la actuación de dispositivos). Además, se debe tomar en cuenta que para cada comando que sea enviado a la plataforma, se deben enviar algunos elementos adicionales junto con cada llamada:

- Dirección IP: es necesario enviar en cada llamada la dirección correspondiente a la instancia de *Home Assistant* para establecer la comunicación.
- Long Live Token: es un código único creado por *Home Assistant*, que funciona como una llave exclusiva para acceder a una sesión específica de la plataforma.
- Entity ID: este es el código asignado por *Home Assistant* para cada dispositivo individual, como se especifica en la sección 5.2. Este código permite distinguir al dispositivo individualmente tanto por su nombre, como por el tipo de dispositivo.

En la figura 6.1 se muestra un diagrama de flujo en donde se representa el proceso de comunicación entre *Home Assistant*, los dispositivos inteligentes, y la aplicación por diseñar. Es posible observar cómo el proceso que se debe seguir comienza por el establecimiento de una sesión de *Home Assistant*, la cual estará conectada con cada uno de los dispositivos que se tengan disponibles. Adicionalmente, se presenta la aplicación, que será llamada *Robobo Home Assistant App*, la cual únicamente establecerá una comunicación con *Home Assistant*, y no con los dispositivos de manera directa. La aplicación enviará llamadas *GET/POST*, las cuales serán recibidas, e interpretadas por

Home Assistant, y este se encargará de enviar estos comandos a cada dispositivo. De ser una llamada *GET*, la plataforma devolverá el mensaje adecuado al usuario en la aplicación tras consultar la información correspondiente con los dispositivos. De ser un una llamada *POST*, la plataforma comunica el mensaje únicamente con los dispositivos para permitir el cambio de estado deseado.

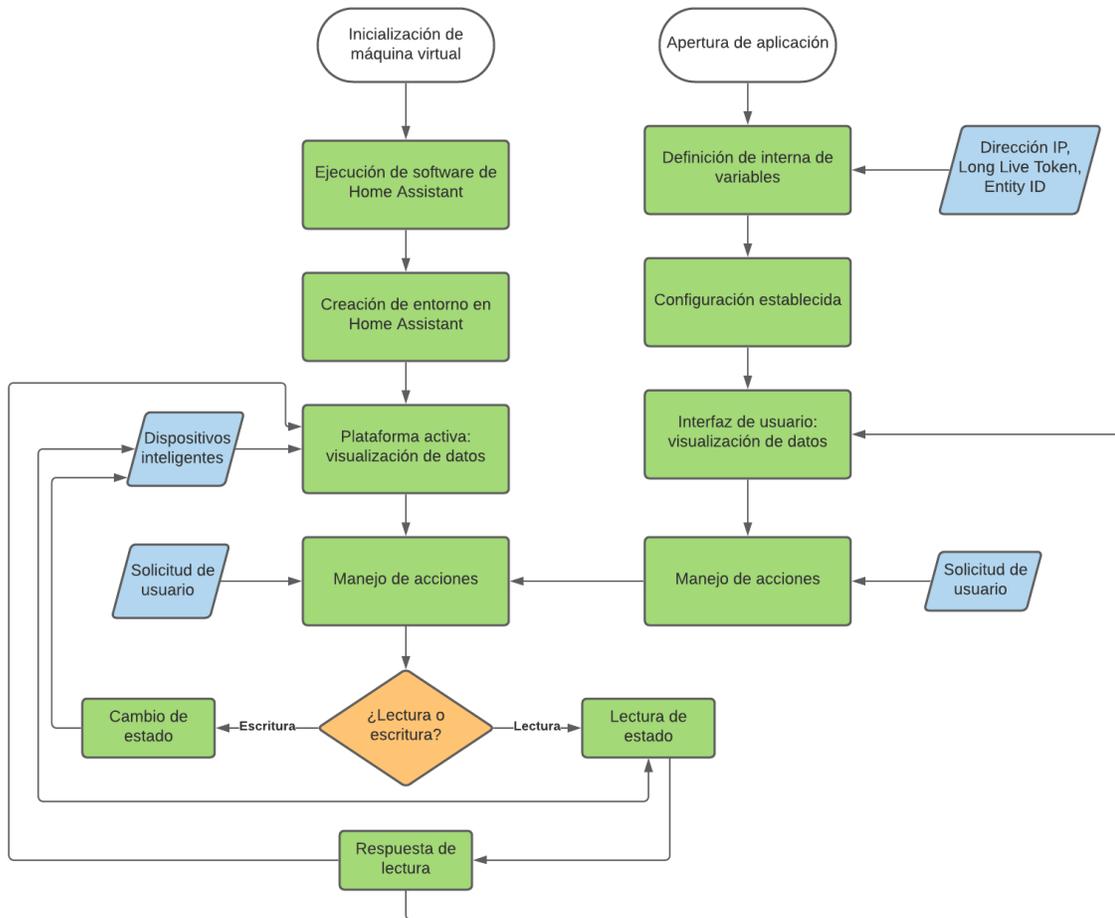


Figura 6.1: Diagrama de flujo de interacción entre aplicación y *Home Assistant*

Android Studio permite una forma de programación que comparte un diseño gráfico, y la vez un diseño programado por comandos. La funcionalidad gráfica permite agregar elementos para la visualización del usuario, es decir, para la creación de una interfaz gráfica (que permitirá a la aplicación contar con botones, o espacios de escritura o lectura de texto para la interacción con el usuario). De esta manera, se programa una

interfaz gráfica que cuenta con los siguientes elementos:

- Entradas del usuario: espacios de texto donde el usuario debe escribir textualmente la dirección IP de *Home Assistant*, el *Long Live Token*, y el *entity ID*.
- Botones: se cuenta con múltiples botones con diferentes funcionalidades. Entre ellas están: el envío de la información escrita por el usuario a variables establecidas dentro del código, actuación de estado para el dispositivo seleccionado, lectura de estado del dispositivo específico seleccionado, y extracción de todos los dispositivos conectados en *Home Assistant*.
- Visualización de texto: espacio en donde se muestra información importante para el usuario, relacionada a la respuesta de cada uno de los elementos previamente mencionados.

En la figura 6.2 se muestra la interfaz gráfica diseñada para la aplicación, donde se muestran los elementos de entrada de usuario, botones, y visualización de texto.

Posteriormente, se procede a programar la funcionalidad de cada uno de los elementos mostrados en la interfaz gráfica. Es necesario establecer las llamadas *GET/POST*, de manera que contengan toda la información necesaria mencionada previamente (el *Long Live Token*, el URL base con la dirección IP de *Home Assistant*, y en los casos necesarios, el *entity ID* del dispositivo con el que se desea interactuar). Además, para mantener una estructura ordenada, se desarrolla la funcionalidad de la aplicación por medio de funciones en el código (una para lectura, otra para escritura, y otra para el listado de dispositivos). El código se encarga de tomar las entradas o *input* del usuario, y convertirlo a variables relevantes para el proceso. Estas variables son posteriormente procesadas y utilizadas dentro de las funciones creadas, las cuales son activadas al pulsar los distintos botones definidos en la interfaz gráfica. El código

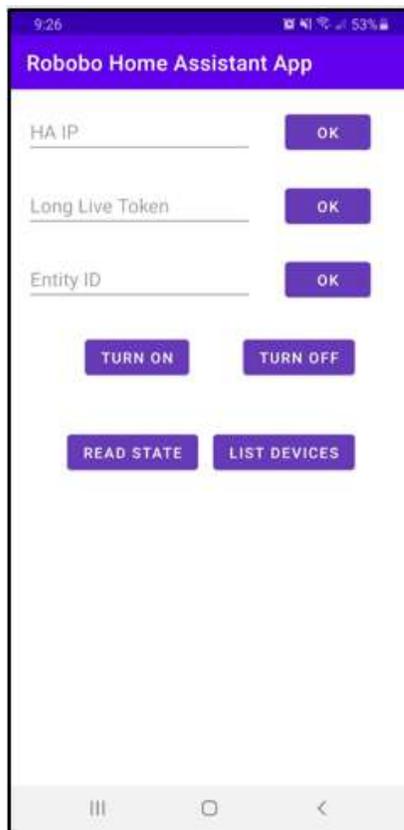


Figura 6.2: Interfaz gráfica de la aplicación diseñada

funcional y completo de la aplicación se puede observar en la plataforma de Github, mediante el siguiente enlace:

<https://github.com/sebas26gm/RoboboHomeAssistantApp.git>

Adicionalmente, se muestra en la figura 6.3 un ejemplo de la aplicación funcionando con los parámetros correspondientes a un caso de uso en una sesión de *Home Assistant* utilizada para el proyecto.

6.3 Comprobación de comunicación

Para lograr establecer la confiabilidad del método de comunicación seleccionado para simular un comportamiento que represente la interacción entre *Home Assistant* y un *Robobo*, se plantean una serie de pruebas. Las pruebas tendrán el objetivo de asegurar que el método elegido de comunicación con la REST API por medio de

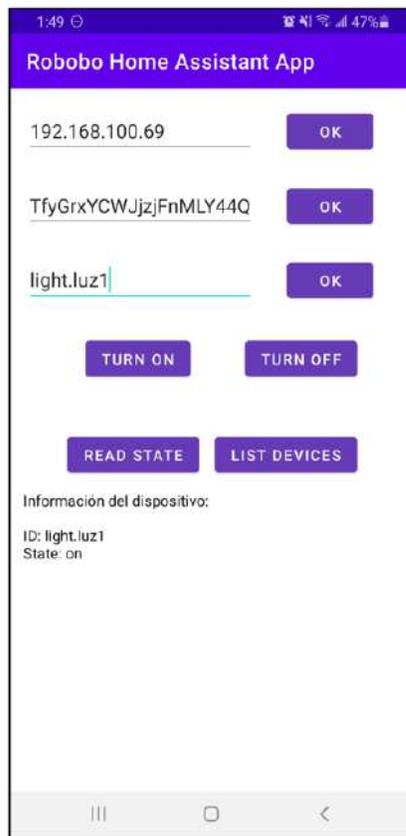


Figura 6.3: Ejemplo de aplicación funcional

llamados *GET/POST*, sea capaz de leer y actuar sobre múltiples dispositivos repetida y correctamente. Para ello, se realizará una prueba para lectura, y otra para actuación de dispositivos, aplicadas sobre todos los dispositivos disponibles, mientras exista la posibilidad (ya que en algunos casos solamente sería posible la lectura del estado del dispositivo, y no sería posible actuar sobre ellos, como lo sería el caso en los sensores). Las pruebas por realizar se detallan a continuación.

- Prueba de lectura: consiste en la solicitud de lectura para el estado de cada uno de los dispositivos (en caso de aplicar). Al solicitar la lectura, se observará tanto si el resultado obtenido es el estado correcto del dispositivo, como si el comando es comunicado correctamente. Se realizarán 68 repeticiones del proceso para cada dispositivo, en donde para cada una se modificará el estado de cada dispositivo previo a su prueba. Se evaluará al final de las pruebas el porcentaje de error (y

con ello la confiabilidad) en la comunicación realizada.

- Prueba de actuación: consiste en la solicitud de actuación sobre el estado de cada uno de los dispositivos (en caso de aplicar). Al solicitar la actuación, se observará tanto si el cambio de estado es el solicitado, como si el comando es comunicado correctamente. Se realizarán 68 repeticiones del proceso para cada dispositivo. Se evaluará al final de las pruebas el porcentaje de error (y con ello la confiabilidad) en la comunicación realizada.

Las pruebas descritas anteriormente buscan responder ante la necesidad # 4: **El sistema es capaz de realizar sus funciones objetivo correctamente dentro del *Robobo Framework***. Los resultados obtenidos de las pruebas de validación, se exponen en el Capítulo 7.

6.3.1 Implementación de un proceso automatizado

Una vez creada la aplicación, *Robobo Home Assistant App*, se busca la creación de un código dentro de esta que permita la automatización de un proceso. Esta automatización permitiría demostrar la utilidad de la aplicación y su flexibilidad para ser expandida, además de presentar una base para posibles experimentos a nivel educativo. La automatización planteada se encargaría de enlazar la lectura de un sensor, con el accionamiento de un dispositivo. Se selecciona de esta manera la funcionalidad de sensor de luz integrada en el sensor de movimiento marca *Aqara*.

Dentro de *Android Studio*, es posible leer el valor de la iluminancia (en lux, unidad lx) detectada por el sensor en un entorno por medio de un llamado con el método GET. Así, se establece un valor umbral (adecuado para el entorno de prueba) donde un valor menor que 6 lx representa la necesidad de iluminación adicional. Así, se establece una función en el código que detecta el nivel de iluminancia emitido por el sensor e indicado en *Home Assistant*, y si se reconoce un valor menor que el umbral establecido,

se realiza una llamada a la función programada previamente para el accionamiento de un dispositivo (cambio de estado), indicando que es necesario encender la luz que indique el usuario en la interfaz gráfica.

Finalmente, una vez establecida la lógica que debe seguir la automatización, se crea un botón adicional en la interfaz gráfica que permite realizar un llamado a la función. Se observa la interfaz final posterior a la creación de la automatización en la figura 6.4. Además, se encuentra un video mostrando el funcionamiento adecuado de la aplicación en conjunto con *Home Assistant*, mediante el siguiente enlace: <https://youtu.be/gd2oZK-QnUQ>

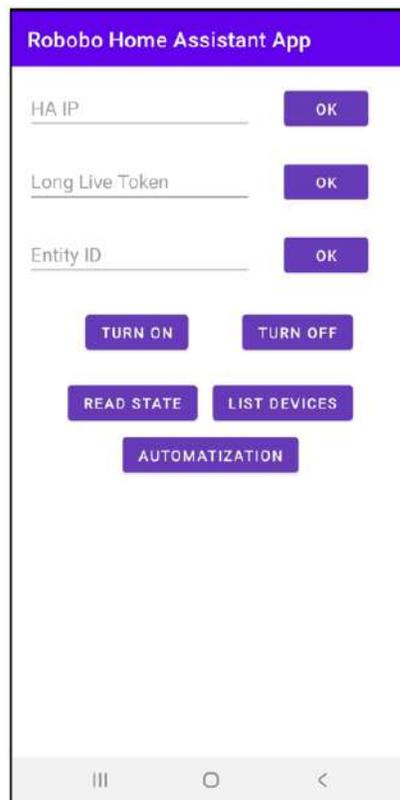


Figura 6.4: Interfaz final con automatización agregada.

CAPÍTULO 7

VALIDACIÓN DEL SISTEMA

Para poder asegurar el cumplimiento tanto de las necesidades como de sus respectivas especificaciones, es necesario realizar una adecuada validación del entorno IoT de *Home Assistant*. Para ello, se plantea la serie de pruebas descritas a continuación.

7.1 Definición de pruebas

7.1.1 Repetibilidad de conexión

En esta prueba se busca evaluar la necesidad # 2: **El sistema es capaz de realizar sus funciones objetivo correctamente fuera del *Robobo Framework***. Para ello, se complementa con la métrica # 3: **Error en conexión con dispositivos**.

La prueba planteada consiste en realizar la conexión de un dispositivo desde cero, es decir eliminar el dispositivo del *dashboard* de *Home Assistant* si este ya existe, o si no, agregar la conexión de este para poder comprobar su adecuada conexión a la plataforma. Para asegurar que el dispositivo haya sido conectado apropiadamente, se realizará un actuación del dispositivo para comprobar su funcionamiento en conjunto

con la plataforma. El procedimiento descrito será realizado para cada uno de los dispositivos comprados, y realizarán 68 iteraciones para cada dispositivo. En cada prueba, se evaluará el porcentaje de error de cada dispositivo, sea el caso de que algún dispositivo presente inconvenientes a la hora de realizar la conexión, o su posterior actuación.

7.1.2 Repetibilidad de lectura

En esta prueba se busca evaluar la necesidad # 5: **El sistema permite la lectura de los estados de los dispositivos inteligentes o sensores..** Para ello, se complementa con la métrica # 1: **Error en lectura de dispositivos.**

La prueba planteada consiste en evaluar la adecuada lectura de un cambio de estado, en un dispositivo previamente conectado a *Home Assistant*. Para ello se cambiará el estado del dispositivo desde su aplicación móvil (en aquellos casos en los cuales esto aplique), y se evaluará que el cambio de estado se vea reflejado apropiadamente en el *dashboard* de *Home Assistant*. En el caso de los sensores, en donde por su naturaleza estos no son posibles de actuar de la misma manera que una bombilla o un enchufe, se realizará una activación o cambio de estado manualmente y se procede a confirmar la lectura de cambio de estado en el *dashboard* de *Home Assistant*. El procedimiento anterior será repetido 68 veces para cada dispositivo comprado. En cada prueba, se evaluará el porcentaje de error (y con ello la confiabilidad) de cada dispositivo, sea el caso de que algún dispositivo presente inconvenientes a la hora de realizar la correspondiente lectura.

7.1.3 Repetibilidad de actuación

En esta prueba se busca evaluar la necesidad # 6: **El sistema permite la actuación sobre los estados de los dispositivos inteligentes o sensores..** Para ello, se complementa con la métrica # 2: **Error en actuación de dispositivos.**

La prueba planteada consiste en evaluar la adecuada actuación de un dispositivo desde el *dashboard* de *Home Assistant*. Para ello, se utilizará la herramienta para cambiar de estado el dispositivo, y se observará que el dispositivo responda correctamente ante la acción, y se produzca el cambio indicado. El procedimiento anterior se realizará sobre aquellos dispositivos en donde sea posible accionar un cambio de estado (es decir, se excluye en este caso los dispositivos de tipo sensor), y se repetirá el procedimiento 68 veces para cada dispositivo. En cada prueba, se evaluará el porcentaje de error (y con ello la confiabilidad) de cada dispositivo, sea el caso de que algún dispositivo presente inconvenientes a la hora de realizar la correspondiente actuación.

7.1.4 Percepción de complejidad

En esta prueba se busca evaluar la necesidad # 7: **El sistema y sus sensores tienen un nivel de complejidad tal que sea posible su utilización por parte de personas sin un alto grado de conocimiento técnico.** Esta prueba se plantea desde una perspectiva cualitativa. Por lo tanto, por medio del criterio del evaluador y conversaciones del cliente, se busca evaluar la opinión en cuanto a la usabilidad del sistema en *Home Assistant* y además de la aplicación móvil diseñada, con un grupo de usuarios. Se define una muestra de 10 usuarios (divididos en grupos de 5 personas), que seguirán la dinámica planteada a continuación:

- Los usuarios serán presentados ante una sesión de *Home Assistant* previamente instalada en un computador utilizando *Windows*. La sesión de *Home Assistant* no tendrá ningún dispositivo conectado ni configurado.
- Se proveerá a los usuarios con el manual de usuario creado en el presente proyecto, el cual contendrá las instrucciones paso a paso para la conexión de todos los dispositivos adquiridos para el proyecto.

- Se le instruirá a los usuarios, por medio de la guía de usuario, conectar 1 dispositivo de iluminación (bombilla *Lifx*), 1 dispositivo de tipo Smart Plug (marca *Tapo*), los sensores adquiridos (marca *Aqara* y *Sonoff*), y una cámara (marca *Tapo*).
- Una vez conectados los dispositivos, deberán probar la actuación y lectura de cada uno por medio de la aplicación diseñada en el presente proyecto.
- Para finalizar, se le solicitará a los usuarios responder un formulario que busca evaluar, por medio de una escala predefinida, su percepción en cuanto al proceso de instalación, y conexión de dispositivos en *Home Assistant*, junto con la funcionalidad de la aplicación utilizada.

7.2 Resultados y análisis

Los resultados obtenidos por medio de las pruebas de validación se dividen en 3 secciones:

- Pruebas de funcionamiento de dispositivos en *Home Assistant*: en donde se determina el error y la confiabilidad de lectura, escritura (actuación) y conexión de los dispositivos seleccionados.
- Pruebas de dispositivos utilizando la aplicación diseñada: se determina el error y la confiabilidad de lectura y escritura de los dispositivos seleccionados por medio de la comunicación entre la REST API de *Home Assistant* y la aplicación diseñada en *Android Studio*.
- Percepción de complejidad: en donde se evalúan las pruebas de conexión, uso y visualización de dispositivos en *Home Assistant* y en la aplicación diseñada, aplicadas a distintos usuarios.

En la tabla 7.1 se muestran los dispositivos utilizados para las distintas pruebas de conexión, lectura y escritura. Debido a la naturaleza de los sensores, estos no son posibles de actuar de manera virtual (como lo es el caso de los *Smart Plugs* o luces), por lo que estos dispositivos únicamente son sometidos a pruebas de conexión y lectura. Además, se excluye de las pruebas el sensor de movimiento de *Sonoff*, el sensor de temperatura *Sonoff*, y las cámaras. El primer dispositivo se excluye debido a que presenta un comportamiento idéntico a su contra-parte de Aqara (tanto en conexión como en lectura), y por esta razón se decide evaluar el comportamiento solamente en el sensor de movimiento de Aqara. El segundo dispositivo se excluye debido a que, bajo criterio ingenieril, los datos mostrados en la lectura no presentan la variabilidad ni potencial experimental que los demás dispositivos, ya que los valores que este indica, se muestran -para la percepción de un usuario- como un valor constante (temperatura). Finalmente, las cámaras no son evaluadas debido a que no presentan una actuación o lectura dinámica como el resto de los dispositivos bajo prueba, ya que estas solamente permiten observar imágenes o video, pero no emiten un estado ni es posible actuarlas como el resto de dispositivos.

Tabla 7.1: Dispositivos utilizados para las pruebas.

Dispositivo	Lectura	Conexión	Escritura
Smart Plug Sonoff	X	X	X
Smart Plug Tapo	X	X	X
Luz Yeelight	X	X	X
Luz Lifx	X	X	X
Sensor de puerta Sonoff	X	X	
Sensor Movimiento Aqara	X	X	

7.2.1 Repetibilidad de lectura, escritura y conexión en *Home Assistant*

Inicialmente se presentan las tablas 7.2-7.4, las cuales muestran un resumen de los porcentajes de error obtenidos para cada dispositivo en las diferentes acciones evaluadas:

lectura, escritura y conexión. Estos resultados son tomados directamente de las pruebas realizadas, para las cuales se pueden observar con mayor detalle en el Apéndice A, específicamente en las tablas A.11-A.28. Los dispositivos son evaluados en cada prueba, donde durante las 68 iteraciones de cada proceso, se anota un valor correspondiente a un 1 en caso de ser una iteración exitosa (sea conexión, lectura, o actuación), y un 0 en caso de presentar inconvenientes.

Tabla 7.2: Porcentaje de error obtenido en la lectura de dispositivos en *Home Assistant*.

Dispositivo	Error (%)
Smart Plug Sonoff	1,471
Smart Plug Tapo	2,941
Luz Yeelight	0,000
Luz Lifx	2,941
Sensor de puerta Sonoff	0,000
Sensor Movimiento Aqara	0,000

Tabla 7.3: Porcentaje de error obtenido en la actuación de dispositivos en *Home Assistant*.

Dispositivo	Error (%)
Smart Plug Sonoff	1,471
Smart Plug Tapo	8,824
Luz Yeelight	0,000
Luz Lifx	0,000

Tabla 7.4: Porcentaje de error obtenido en la conexión de dispositivos en *Home Assistant*.

Dispositivo	Error (%)
Smart Plug Sonoff	1,471
Smart Plug Tapo	2,941
Luz Yeelight	1,471
Luz Lifx	0,000
Sensor de puerta Sonoff	0,000
Sensor Movimiento Aqara	1,471

Se observa cómo para la evaluación de los dispositivos en *Home Assistant*, el porcentaje de error obtenido, en general, es inferior a un 3%, a excepción del

dispositivo *Smart Plug* de *Tapo*. El dispositivo de la marca *Tapo*, contiene el mayor error estudiado de entre las pruebas, ya que para el proceso de actuación presentó más de un 8%, lo cual representa que el dispositivo recibió múltiples veces un comando de cambio de estado, pero no fue capaz de realizar la acción exitosamente. Por otro lado, sin tomar en cuenta el error obtenido del *Smart Plug Tapo* en su actuación, las demás pruebas de validación presentan un alto porcentaje de interacciones exitosas (donde los errores en lectura representan una lectura incorrecta, y en conexión una integración fallida que no permita el manejo posterior del dispositivo en *Home Assistant*), ya que un porcentaje menor al 3% de error implica que no se presentan más de 2 fallos en cada prueba, en cada dispositivo.

Además, durante las pruebas se observa que para dos dispositivos se presenta un retraso, o *delay* en cuanto a la lectura posterior a un cambio de estado. Mientras que, en la prueba de lectura, todos los cambios de estado de los dispositivos son leídos y mostrados en *Home Assistant* inmediatamente (a percepción de un usuario), los dispositivos *Smart Plug Tapo* y la luz marca *Lifx* presentan un retraso en su visualización de cambio de estado. Este retraso se observa únicamente a nivel de lectura, y no de actuación, por lo que para cada uno de estos dos dispositivos, se cronometra manualmente el tiempo que tarda *Home Assistant* en leer correctamente cada cambio de estado de los dispositivos. Los tiempos de respuesta para ambos dispositivos en cada iteración se muestran en la sección del Apéndice A, en las tablas A.13 y A.16, donde se muestra que se presentan tiempos promedio de reacción de 21.77 s y 5.26 s para el dispositivo *Tapo* y *Lifx* respectivamente. De esta manera, es posible afirmar que, mientras que el dispositivo *Lifx* presenta un tiempo de lectura promedio aceptable, el dispositivo *Tapo* presenta un tiempo de lectura significativamente alto en comparación con los demás dispositivos, y esto podría eventualmente inducir a errores en su implementación, en donde el usuario no tenga certeza si el estado verdadero del dispositivo es el mismo que el estado mostrado en la

lectura de este.

Adicionalmente, por medio del *software Minitab*, es posible obtener el *Intervalo de Confianza* (IC) para cada una de las pruebas de validación realizadas. Se presenta en la tabla 7.5 una comparación entre cada una de las acciones evaluadas (conexión, lectura y actuación) en cuanto a los límites superiores, inferiores y probabilidad de éxito, para cada dispositivo, para un nivel de confianza del 90%. Se observa en la tabla nuevamente como el dispositivo *Smart Plug de Tapo* presenta una mayor probabilidad de fallos, como se denota mostrando la menor probabilidad de éxito en cuanto a conexión, y con un IC inferior que podría bajar hasta un 83.3%. Por otro lado, todos los demás ICs superan el 90%, al igual que una probabilidad de éxito mayor a un 97%. De esta manera, las pruebas de validación en cuanto a lectura, escritura y conexión, determinan una alta confiabilidad en 5 de los 6 dispositivos evaluados mediante la utilización de la plataforma objetivo, *Home Assistant*.

Tabla 7.5: Intervalos de confianza (IC) por cada acción de dispositivo, para *Home Assistant*.

Dispositivo	Acción	IC inferior	IC Superior	Probabilidad de éxito
Smart Plug Sonoff	Conexión	0,932	0,999	0,985
	Lectura	0,932	0,999	0,985
	Actuación	0,932	0,999	0,985
Smart Plug Tapo	Conexión	0,910	0,995	0,971
	Lectura	0,910	0,995	0,971
	Actuación	0,833	0,961	0,912
Luz Yeelight	Conexión	0,932	0,999	0,985
	Lectura	0,967	1,000	1,000
	Actuación	0,967	1,000	1,000
Luz Lixf	Conexión	0,967	1,000	1,000
	Lectura	0,910	0,995	0,971
	Actuación	0,967	1,000	1,000
Sensor Sonoff	Conexión	0,967	1,000	1,000
	Lectura	0,967	1,000	1,000
Sensor Aqara	Conexión	0,932	0,999	0,985
	Lectura	0,967	1,000	1,000

7.2.2 Comunicación con la aplicación (prueba de concepto para el *Robobo Framework*)

Posterior al diseño de una aplicación por medio de *Android Studio*, se requiere la validación de su adecuado funcionamiento de manera que se asegure que esta llegue a funcionar apropiadamente dentro del *Robobo Framework*, y los métodos implementados (llamadas de lectura con el método *GET*, y llamadas de escritura del método *POST*) sean confiables. Se muestran a continuación los resultados de las pruebas planteadas en la sección 6.3. En las tablas 7.6 y 7.7 se muestra un resumen de los porcentajes de error obtenidos para cada prueba y dispositivo. Para mayor detalle, en la sección del Apéndice A, en las tablas A.29-A.38, se muestran los resultados individuales de cada prueba para cada uno de los dispositivos.

Las pruebas de validación realizadas, muestran un porcentaje de error significativamente bajo, debido a que todos los dispositivos, a excepción del *Smart Plug Tapo*, no presentan ningún fallo tanto a nivel de lectura como de actuación. Se demuestra a través de las pruebas que la aplicación implementada es capaz de comunicarse con *Home Assistant*, y actuar o leer adecuadamente sobre los dispositivos de la plataforma sin problemas.

Tabla 7.6: Porcentaje de error obtenido en la lectura de dispositivos por medio de la aplicación diseñada

Dispositivo	Error (%)
Smart Plug Sonoff	0,000
Smart Plug Tapo	16,176
Luz Yeelight	0,000
Luz Lifx	0,000
Sensor de puerta Sonoff	0,000
Sensor Movimiento Aqara	0,000

Por otro lado, se demuestra nuevamente al igual que en la sección 7.2.1, que el dispositivo *Smart Plug Tapo* presenta inconvenientes en cuanto a su lectura, y actuación adecuada por medio de la aplicación. Los errores en las pruebas se presentan en la

Tabla 7.7: Porcentaje de error obtenido en la actuación de dispositivos por medio de la aplicación diseñada

Dispositivo	Error (%)
Smart Plug Sonoff	0,000
Smart Plug Tapo	8,824
Luz Yeelight	0,000
Luz Lifx	0,000

actuación en forma de un cambio de estado que no se logra concretar, y en la lectura por medio de la incapacidad de leer el estado actual (o una lectura incorrecta de este). En esta ocasión, el error en la lectura del dispositivo (aproximadamente 16%, tabla 7.6) es significativamente mayor que el de actuación (aproximadamente 8.8%, tabla 7.7). Esto implica que mediante el uso de la REST API la lectura del dispositivo es más propensa a fallar para el dispositivo *Tapo* en específico, no obstante la totalidad de resultados muestran que la comunicación por medio de la aplicación presenta un menor porcentaje de error para los demás dispositivos, al comparar tanto lectura como actuación con los resultados mostrados en la sección 7.2.1 utilizando el *dashboard* de la plataforma de *Home Assistant*. Se demuestra que un comando *GET/POST* puede llegar a presentar una mayor eficiencia debido a su capacidad de consultar o escribir directamente sobre los datos o información almacenada en *Home Assistant*, sin tener que procesar previamente los comandos recibidos de la interfaz gráfica del dashboard de la plataforma.

Además, para los resultados obtenidos se realiza un análisis con la ayuda de *Minitab* para determinar los ICs. Se muestra en la tabla 7.8 que, por medio de la muestra utilizada para los experimentos, se tiene con un 90 % de confianza que el menor porcentaje de éxito en la comunicación con los dispositivos por medio de la aplicación implementada, es de un 96.7% (a excepción del dispositivo *Smart Plug Tapo*).

Finalmente, es posible observar que el dispositivo *Tapo*, por medio de su IC, puede llegar a presentar un porcentaje de éxito hasta de un 74.6%, lo cual es

significativamente bajo en comparación con los demás dispositivos. De esta manera, es posible concluir que el dispositivo *Smart Plug Tapo*, tanto por medio de la aplicación diseñada como por medio de *Home Assistant*, es el dispositivo con menor confiabilidad al presentar la mayor cantidad de fallos y los menores ICs. Por lo tanto, se podría sugerir la utilización del dispositivo marca *Sonoff* a la hora de buscar una mejor funcionalidad de enchufes inteligentes en un entorno de IoT. En lo referente a los demás dispositivos en prueba, se demuestran resultados consistentes entre los distintos experimentos, donde los porcentajes de éxito tanto de lectura, conexión y actuación establecen una alta confiabilidad para cada uno de los dispositivos, tanto por medio de la aplicación diseñada, como por *Home Assistant*.

Tabla 7.8: Intervalos de confianza (IC) por cada acción de dispositivo, para llamadas de la aplicación

Dispositivo	Acción	IC inferior	IC Superior	Probabilidad de éxito
Smart Plug Sonoff	Lectura	0,967	1,000	1,000
	Actuación			
Smart Plug Tapo	Lectura	0,746	0,906	0,838
	Actuación	0,833	0,961	0,912
Luz Yeelight	Lectura	0,967	1,000	1,000
	Actuación			
Luz Lixf	Lectura			
	Actuación			
Sensor Sonoff	Conexión			
	Lectura			
Sensor Aqara	Conexión			
	Lectura			

7.2.3 Percepción de complejidad

Las pruebas realizadas fueron estructuradas con el siguiente orden:

- Prueba 1: conexión y utilización de un dispositivo de luz inteligente (bombilla marca *Lixf*) en *Home Assistant*, seguido por la interacción con este en la aplicación diseñada, “Robobo *Home Assistant* App”.

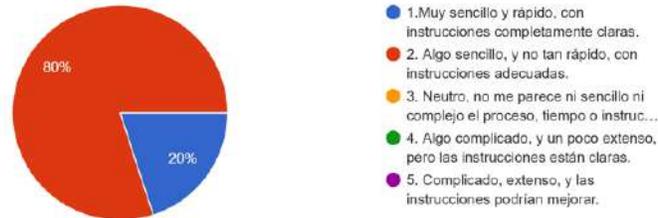
- Prueba 2: conexión y utilización de un dispositivo *Smart Plug* (marca *Sonoff*) en *Home Assistant*, seguido por la interacción con este en la aplicación diseñada, “Robobo *Home Assistant* App”.
- Prueba 3: conexión y utilización de los distintos sensores (marcas *Sonoff* y *Aqara*) en *Home Assistant*, seguido por la interacción con estos en la aplicación diseñada, “Robobo *Home Assistant* App”.
- Prueba 4: conexión y utilización de una cámara (marca *Tapo*) en *Home Assistant*, seguido por la interacción con este en la aplicación diseñada, “Robobo *Home Assistant* App”.

Posterior a la realización de las pruebas, se evalúa por medio de una encuesta, disponible en el Apéndice C, donde se muestran las preguntas realizadas a los usuarios con respecto a las 4 pruebas. Los resultados de la primera prueba, se muestran en las gráficas de la figura 7.1, en donde se muestra la distribución de las categorías seleccionadas por los participantes para expresar sus opiniones con respecto a las pruebas realizadas. La figura 7.1a muestra que a pesar de que la mayoría de los usuarios consideran que el proceso de la conexión del bombillo inteligente con *Home Assistant* no es muy rápido, todos piensan que el proceso es sencillo. Esto le otorga al bombillo inteligente un estado de preferencia en cuanto a su usabilidad y facilidad de conexión al compararlo con los demás dispositivos, ya que estos presentan una opinión más variada en cuanto a la sencillez, y rapidez de conexión.

En cuanto a las siguientes pruebas, es posible observar que los usuarios también coinciden en cuanto a la facilidad de implementación de los dispositivos *Smart Plugs*, ya que se cuenta con un alto porcentaje de usuarios (80%) que consideran este proceso sencillo y rápido (figura 7.2a). Sin embargo este patrón observado en los dos primeros dispositivos cambia para los dispositivos de las pruebas 3 y 4, en las figuras 7.3a y 7.4a respectivamente. Con el proceso de conexión de estos dispositivos, el porcentaje

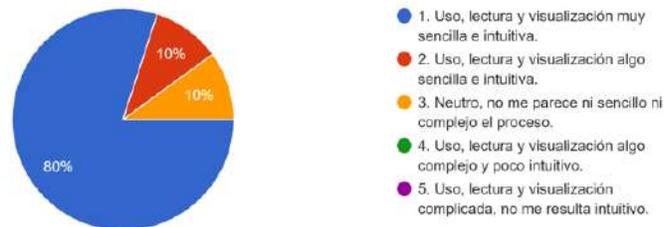
de los usuarios que consideran el proceso sencillo baja a un 60% para los sensores y 40% para las cámaras. Esto se debe a factores como un procedimiento más largo a la hora de conectar los dispositivos, su instalación y reconocimiento adecuado en *Home Assistant*. La plataforma de *Home Assistant*, como se ha mencionado previamente, permite distintas bibliotecas o integraciones para múltiples dispositivos, y cada una de ellas es diferente en su procedimiento y complejidad, debido a que cada dispositivo opera de diferente manera, e incluso hasta con distintos protocolos.

Para la Prueba #1: conexión de un dispositivo de luz inteligente. Con respecto a la instalación del dispositivo:
10 respuestas



(a) Respuestas recibidas para la primera prueba con Home Assistant

Posterior a la instalación de la Prueba #1, en cuanto al uso, cambios de estado, y lectura de los dispositivos en la aplicación Robobo Home Assistant App:
10 respuestas



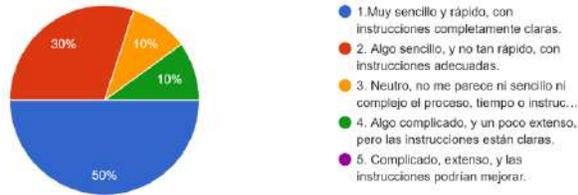
(b) Respuestas recibidas para la primera prueba con aplicación móvil

Figura 7.1: Respuestas recibidas para la primera prueba. Fuente: elaboración propia

De esta manera, resulta de vital importancia conocer la opinión de los usuarios en cuanto a sus percepciones del proceso, porque a pesar de que el proceso requerido a seguir para establecer una conexión no puede ser cambiado por *Home Assistant*, se pueden reconocer consideraciones con base en los resultados expuestos :

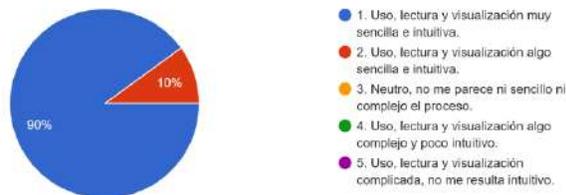
- Mejora del manual de usuario: la retroalimentación de los usuarios permite reconocer posibles puntos de mejora en el manual elaborado, donde se puede

Para la Prueba #2: conexión de un dispositivo conector (o Smart Plug). Con respecto a la instalación del dispositivo:
10 respuestas



(a) Respuestas recibidas para la segunda prueba con Home Assistant

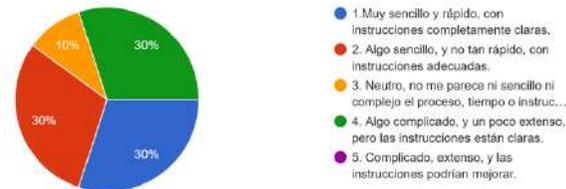
Posterior a la instalación de la Prueba #2, en cuanto al uso, cambios de estado, y lectura de los dispositivos en la aplicación Robobo Home Assistant App:
10 respuestas



(b) Respuestas recibidas para la segunda prueba con aplicación móvil

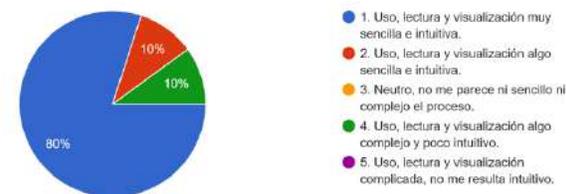
Figura 7.2: Respuestas recibidas para la segunda prueba. Fuente: elaboración propia

Para la Prueba #3: conexión de sensores inteligentes. Con respecto a la instalación de los dispositivos:
10 respuestas



(a) Respuestas recibidas para la tercera prueba con Home Assistant

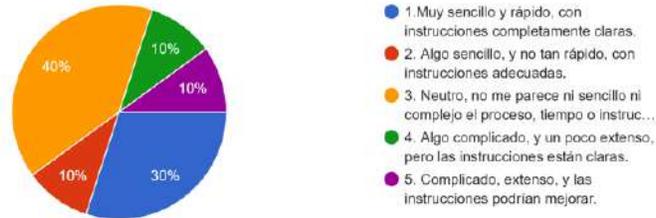
Posterior a la instalación de la Prueba #3, en cuanto al uso y lectura de los dispositivos en la aplicación Robobo Home Assistant App:
10 respuestas



(b) Respuestas recibidas para la tercera prueba con aplicación móvil

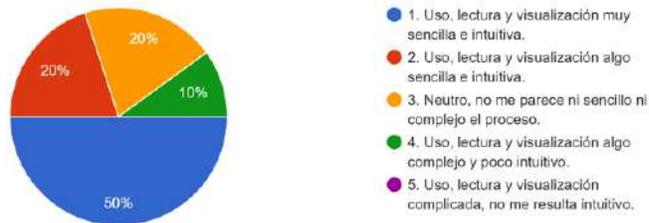
Figura 7.3: Respuestas recibidas para la tercera prueba. Fuente: elaboración propia

Para la Prueba #4: conexión de una cámara inteligente. Con respecto a la instalación del dispositivo:
10 respuestas



(a) Respuestas recibidas para la cuarta prueba con Home Assistant

Posterior a la instalación de la Prueba #4, en cuanto al uso y lectura de los dispositivos en la aplicación Robobo Home Assistant App:
10 respuestas



(b) Respuestas recibidas para la cuarta prueba con aplicación móvil

Figura 7.4: Respuestas recibidas para la cuarta prueba. Fuente: elaboración propia

presentar con mayor detalle algunas instrucciones que permitirían facilitar el proceso por seguir, a pesar de que este sea extenso por naturaleza, y de esta manera mejorar la percepción en cuanto a la sencillez del procedimiento.

- Selección de dispositivos: con base en los dispositivos que obtienen resultados más favorables (sencillez, rapidez, y claridad de instrucciones), es posible reconocer que entre los dispositivos adquiridos para el proyecto, existen categorías (luz y *smart plug*) que permiten una integración más favorable que los otros dispositivos. De esta manera es posible para el GII plantear los experimentos con conocimiento previo de cuáles dispositivos serían más versátiles, y hasta tener un mejor manejo del tiempo en cuanto a sus prácticas, al comprender que algunos dispositivos tienen un procedimiento más extenso.

- Exploración de dispositivos: se establecen algunos tipos de dispositivos que son compatibles con *Home Assistant*, y su facilidad de implementación. Sin embargo, sería posible a futuro la exploración de distintos tipos de dispositivos, como por ejemplo la utilización de microcontroladores o el uso de cortinas (o *smart blinds*). Estos dispositivos requerirían su respectiva evaluación, que podría ampliar el conocimiento existente en cuanto a la versatilidad y rango de dispositivos que podrían utilizarse para experimentación.

Por otro lado, en cuanto al uso de la aplicación móvil *Robobo Home Assistant App*, es posible observar que para las pruebas 1, 2 y 3 (figuras 7.1b, 7.2b y 7.3b respectivamente), se cuenta con la percepción de ser sencilla e intuitiva para al menos un 90% de los usuarios. Estos no indican problemas o un grado de complejidad alto para la utilización de la aplicación. Se presenta un único tipo de dispositivo, para la prueba 4 referente a una cámara (figura 7.4b), que presenta solamente un 70% de los usuarios considerando la aplicación como intuitiva. Lo anterior puede estar relacionado con el hecho de que la aplicación únicamente permite leer el estado de la cámara (sea que esté activa o inactiva), pero no permite una visualización del video registrado por *Home Assistant*. Sin embargo, por medio de conversaciones con el cliente, el GII afirma que la visualización (y el análisis) de video en cámaras, ya es un proceso utilizado y desarrollado dentro del proyecto *Robobo*, y por este motivo lo que se requiere de la aplicación diseñada *Robobo Home Assistant App* es permitir la lectura del estado de la cámara, y no la visualización de video.

Las pruebas de validación en cuanto a la percepción del usuario acerca de la facilidad de conexión, implementación y visualización en *Home Assistant* indican cuáles dispositivos permiten un proceso más sencillo y rápido de llevar a cabo, y que para la mayoría de dispositivos la aplicación diseñada es intuitiva y permite fácilmente la interacción con los dispositivos conectados a *Home Assistant*. Además, se resalta que dentro del proceso de validación desde un análisis cualitativo, se cuenta con una

retroalimentación constante tanto por medio de las observaciones del conductor de los experimentos, como de los usuarios. De esta manera se anota y trabaja sobre aquellas observaciones realizadas durante las pruebas en cuanto a posibles mejoras en el manual de usuario, las cuales son corregidas para permitir a futuros usuarios una experiencia más sencilla y rápida.

7.2.4 Otras necesidades

Adicionalmente es posible comprobar el cumplimiento con el restante de necesidades del proyecto, ya que:

- La aplicación de *Robobo Home Assistant App* cuenta con una interfaz gráfica que permite al usuario interactuar por medio de botones y texto. Adicionalmente, la plataforma de *Home Assistant* cuenta con una interfaz o *dashboard* que facilita al usuario la visualización y uso de dispositivos conectados.
- Se logra realizar la conexión exitosamente con 11 dispositivos inteligentes (2 dispositivos de luz, 2 conectores o *smart plugs*, 2 cámaras, 4 sensores y un parlante inteligente).
- Los dispositivos seleccionados tienen disponibilidad para ser adquiridos frecuentemente por medio de la plataforma de *Amazon* tanto para América como para Europa.
- Los dispositivos seleccionados cumplen con las métricas establecidas en cuanto al precio individual por dispositivo y el presupuesto total establecido por el cliente (se muestran los precios como parte del análisis económico de la sección 8).
- Al realizar la aplicación de *Robobo Home Assistant App* en *Android Studio*, se cumple tanto con los requerimientos de funcionalidad como de compatibilidad para su integración con el *Robobo Framework*.

CAPÍTULO 8

ANÁLISIS ECONÓMICO

En este capítulo se realiza un análisis que comprende el impacto en recursos involucrados para el diseño del sistema presentado, tanto a nivel monetario como de valor educativo y científico. El proyecto surge como un desarrollo con propósitos educativos para el GII, quienes buscan la enseñanza temprana a nivel colegial sobre temáticas en el área del IoT. De esta manera, se establece que el material diseñado y planteado en este proyecto, junto con posibles adiciones como lo es el *Robobo Framework*, serán completamente de software abierto u *open source*. Esto significa que las herramientas investigadas e implementadas serán presentadas como un material didáctico gratuito, que no representará una ganancia para el GII de manera inmediata.

Las instituciones educativas estarán encargadas de conseguir por sí mismas los materiales necesarios para la experimentación durante el curso impartido por el GII. En la tabla 8.1 se muestran los costos para cada uno de los dispositivos adquiridos para la implementación de un entorno inteligente, donde es posible para futuras implementaciones descartar los ítems 4 y 10, debido a que se determina en el proyecto

que estos no permiten la compatibilidad y conexión deseada por la plataforma de *Home Assistant*.

Tabla 8.1: Costo de dispositivos adquiridos para el proyecto.

Item	Dispositivo	Proveedor	Precio (₡)	Precio (€)
1	Bombilla Led Wifi	Lifx	33206,2	44,9
2	Bombilla Led Wifi	Yeelight	20633,7	27,9
3	Smart Plug	D-LINK	24331,5	32,9
4	Smart Plug	TP-Link	11759,0	15,9
5	Smart Plug	Sonoff	15974,5	21,6
6	Sensor de temperatura/humedad	Sonoff	11759,0	15,9
7	Sensor movimiento	Aqara	24331,5	32,9
8	Sensor de movimiento	Sonoff	13977,7	18,9
9	Sensor de puerta	Sonoff	13238,1	17,9
10	Zigbee Sniffer	Diferentes	8800,8	11,9
11	Conbee ii	Phoscon	25071,1	33,9
12	Camera Wifi IP	Tapo	25810,6	34,9
13	Camera Wifi IP	Foscam	55393,0	74,9
14	Tarjeta MicroSD	SanDisk	22112,8	29,9
15	Echo Dot 3rd Gen	Amazon	25810,6	34,9
Total			332210,4	449,2
Total para futuras implementaciones			311650,6	421,4

Adicionalmente, existen factores económicos por considerar, como por ejemplo el tiempo invertido de desarrollo del proyecto, según lo remunerado para asistencias estudiantiles en el Tecnológico de Costa Rica, donde se proyectan 16 semanas de trabajo en donde se invierten 20 horas semanales. Además, para el desarrollo del proyecto se utilizaron recursos computacionales: un computador personal. Para calcular la utilización de este recurso se calcula su depreciación durante las 16 semanas de duración del proyecto, donde partiendo de un precio inicial de 369 000 colones, el computador se deprecia aproximadamente en 24 600 colones (estimando una vida útil de 5 años). Por otra parte, no se presenta la necesidad de realizar inversiones en la compra de licencias de software, debido a que la plataforma utilizada es de software abierto. En la tabla 8.2 se muestra el costo total que representa la implementación del proyecto.

Tabla 8.2: Costo total para el desarrollo del proyecto.

No.	Detalle	Costo
1	Dispositivos inteligentes	332210,4
2	Depreciación de equipos	24600
3	Desarrollo del proyecto	320000
Total		676810,4

Por otra parte, el proyecto aporta al desarrollo científico del GII, ya que los avances realizados junto con los aportes diseñados para el *Robobo Framework* permiten al grupo el desarrollo de artículos científicos ya que no existen herramientas educativas sobre IoT para alumnos de secundaria, por lo que esta línea que comienza con el presente proyecto servirá para futuras publicaciones en el campo de la innovación educativa. Dicha producción científica permite generar visibilidad a nivel internacional para el GII, lo cual a futuro puede generar el interés de empresas de tecnología o inversionistas que busquen el desarrollo a mayor profundidad del presente proyecto. Según el cliente, es usual que se presente una etapa de “explotación de los resultados” en la cual empresas españolas con interés en el proyecto, contratan los servicios de la universidad para generar una versión más desarrollada de un proyecto, y de esta manera generar un producto que pueda ser vendido y así generar ingresos para el GII.

Además, es posible reconocer el impacto educativo y futuro desarrollo económico y tecnológico a nivel de España. El GII proyecta la posibilidad de impartir el curso (con la colaboración del presente proyecto) inicialmente a 6 escuelas con la participación de 7 a 10 alumnos por escuela, durante el primer año. Se estima que podría llegar a impartirse el curso en aproximadamente 10 instituciones educativas por año o incluso más, lo cual implicaría un estimado de 100 (o más) estudiantes que recibirán una educación temprana en áreas tecnológicas e ingenieriles como el IoT, anualmente. La industria del IoT ha crecido significativamente a nivel mundial, con reportes de ganancias de más de 2 mil billones de dolares en Europa Oriental para el año 2020, y con más de 8 billones de dispositivos conectados [82]. Un estudio realizado por medio de encuestas a

un total de 800 empresas europeas distintas -con usuarios desempeñando diferentes roles en sus organizaciones- muestra que un 90% de los participantes indican tener interés en el desarrollo e implementación de tecnologías en el área del IoT [83]. A pesar del interés declarado, un 60% de los participantes indican que sus empresas no han logrado implementar este tipo de tecnologías en sus labores diarias. De este porcentaje un 59% indica que la razón por la cual sus empresas no utilizan tecnologías de esta área, se debe a la falta de capacitación y experiencia con el IoT. De esta manera, la exposición temprana ante tecnologías como el IoT permitiría a jóvenes estudiantes un impulso educativo, que representaría profesionales notablemente capacitados con conocimientos en ingeniería, lo cual a su vez permitiría impulsar el desarrollo económico y tecnológico de la región.

CAPÍTULO 9

CONCLUSIONES

Al finalizar el proyecto, es posible afirmar que se provee al cliente con el material y diseño necesario para la implementación de un sistema para el control de sensores y dispositivos inteligentes en un entorno educativo del área del IoT con la plataforma de *Home Assistant*, en donde se comprueba la funcionalidad (y el grado de confiabilidad) de una serie de dispositivos inteligentes. Además de proveer una herramienta que permite al GII enlazar el presente proyecto, con proyectos propios como el *Robobo Framework*.

Las principales conclusiones del proyecto se establecen a continuación:

- Se caracterizan detalladamente diferentes tecnologías actuales que permiten el manejo de ecosistemas inteligentes, facilitando así comprender los requerimientos y beneficios de cada una, de manera que es posible la selección de una plataforma que permite cumplir con las necesidades del cliente.
- Por medio de *Home Assistant* se comprueba, valida, y documenta todos los aspectos relacionados a la comunicación y control del entorno, es decir los procesos de conexión, actuación y lectura de cada uno de los dispositivos

adquiridos para la plataforma.

- Se diseña e implementa en *Android Studio* la aplicación *Robobo Home Assistant App*, la cual se valida con porcentajes de 100% de éxito (a excepción de un único dispositivo) en el establecimiento de la comunicación para lectura y actuación con *Home Assistant*, y de esta manera cumpliendo con los requisitos de funcionamiento y de integración para el *Robobo Framework*.
- Se valida la confiabilidad de los dispositivos al ser implementados con la plataforma por medio de diferentes pruebas (de lectura, conexión y actuación) donde se comprueba un porcentaje de error menor al 5% (a excepción de un único dispositivo), y un promedio de usabilidad clasificado como “fácil e intuitivo” por la mayoría de usuarios de prueba, tanto para la plataforma de *Home Assistant* como para la aplicación diseñada.
- Se implementa adecuadamente el proceso de diseño en ingeniería al establecer un ciclo constante de conversación y retroalimentación con el cliente, de manera que se permite redireccionar aspectos en el desarrollo del proyecto, y así cumplir con las necesidades establecidas.

9.1 Recomendaciones

Es posible considerar aspectos de mejora, o recomendaciones para implementaciones futuras:

- Experimentar con otras marcas de dispositivos o diferentes tipos de dispositivos inteligentes, como podrían ser motores para cortinas inteligentes que permitan la actuación en estados de valores no binarios (por ejemplo: “abrir” o “cerrar” las cortinas a un 80%).

- Exploración de diferentes procesos para creación de automatizaciones. En la aplicación de *Robobo Home Assistant App* es posible enlazar las lecturas y actuaciones entre cualquier dispositivo actuador (luces o *smart plugs*) y los sensores adquiridos, por lo que sería posible la expansión de la funcionalidad de la aplicación para diferentes propósitos educativos.
- Adición de instalación al manual de usuario para distintos sistemas. Para el presente proyecto la disponibilidad de recursos de cómputo era la de un computador con el sistema operativo *Windows*, por lo que se podría explorar la implementación de *Home Assistant* para sistemas como *Mac*, *Linux* o hasta utilizando *Raspberry Pi*.

REFERENCIAS BIBLIOGRÁFICAS

- [1] “Laboratorio de Inteligencia Artificial para las Ciencias Naturales (LIANA) — TEC.” <https://www.tec.ac.cr/unidades/laboratorio-inteligencia-artificial-ciencias-naturales-liana>. Último Acceso: 30/05/2021.
- [2] “GII - Grupo Integrado de Ingeniería.” <https://www.gii.udc.es/>. Último Acceso: 30/05/2021.
- [3] Tecnológico de Costa Rica, “Licenciatura en Ingeniería Mecatrónica.” <https://www.tec.ac.cr/programas-academicos/licenciatura-ingenieria-mecatronica>. Último Acceso: 08/11/2021.
- [4] “Presentación - Grupo Integrado de Ingeniería.” <https://www.gii.udc.es/presentacion>. Último Acceso: 30/05/2021.
- [5] “Tecnologías avanzadas de sensorización y medida - Grupo Integrado de Ingeniería.” <https://www.gii.udc.es/investigacion/detalle/6>. Último Acceso: 30/05/2021.
- [6] “GII - Perfil de Grupo de investigación.” <https://investigacion.udc.es/gl/Research/Details/G00098>. Último Acceso: 30/05/2021.
- [7] “MINT – Manufactura de Ingenios Tecnológicos.” <https://mintforpeople.com/>. Último Acceso: 30/05/2021.
- [8] E. A. Esquivel-Barboza, L. F. Llamas, P. Vazquez, F. Bellas, and E. Arias-Mendez, “Adapting Computer Vision Algorithms to Smartphone-based Robot for Education,” *2020 4th International Conference on Multimedia Computing, Networking and Applications, MCNA 2020*, pp. 51–56, 2020.
- [9] MINT, “Robobo Programming.” <https://github.com/mintforpeople/robobo-programming/wiki/Java-Android>, 7 2021. Último Acceso: 30/05/2021.

- [10] K. K. Patel, S. M. Patel, and P. G. Scholar, “Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges,” *International Journal of Engineering Science and Computing*, 2016.
- [11] Oracle, “What Is the Internet of Things (IoT)?.” <https://www.oracle.com/internet-of-things/what-is-iot/>. Último Acceso: 16/08/2021.
- [12] G. Lampropoulos, K. Siakas, and T. Anastasiadis, “Internet of Things in the Context of Industry 4.0: An Overview,” *International Journal of Entrepreneurial Knowledge Issue*, vol. 1, no. 1, pp. 4–19, 2019.
- [13] S. Balaji, K. Nathani, and R. Santhakumar, “IoT Technology, Applications and Challenges: A Contemporary Survey,” vol. 108, pp. 363–388, 2019.
- [14] S. C. Satapathy, J. K. Mandal, S. K. Udgata, and V. Bhateja, *Information Systems Design and Intelligent Applications*, vol. 2. Springer, 2018.
- [15] A. E. Hassanien, R. Bhatnagar, N. E. M. Khalifa, and M. H. N. Taha, *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*, vol. 846. 2020.
- [16] Philips Hue, “Smart lighting .” <https://www.philips-hue.com/en-us>. Último Acceso: 29/08/2021.
- [17] SONOFF, “Smart Home automation.” <https://sonoff.tech/>. Último Acceso: 29/08/2021.
- [18] TP-Link, “WiFi Networking Equipment for Home & Business.” <https://www.tp-link.com/us/>. Último Acceso: 29/08/2021.
- [19] H. Zemrane, Y. Baddi, U. El Jadida, and A. Hasbi EMI, “Comparison between IOT protocols: ZigBee and WiFi using the OPNET simulator *,” 10 2018.
- [20] W. Chen, S. Jeong, and H. Jung, “WiFi-Based Home IoT Communication System,” *J. Inf. Commun. Converg. Eng.*, vol. 18, no. 1, pp. 8–15, 2020.
- [21] V. Singla, “Android in Home Automation.” <https://www.codingninjas.com/blog/2020/09/25/android-in-home-automation/>, 9 2020. Último Acceso: 29/08/2021.
- [22] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of Things (IoT) Communication Protocols : Review National Advanced IPv6 Centre (NAV6),” 2017.
- [23] L. Harkushko and I. Pototska, “Smart Home Automation for Android on the Example of ZigBee and Z-Wave.” <https://yalantis.com/blog/smart-home-automation-for-android/>. Último Acceso: 29/08/2021.

- [24] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, “Review of communication technologies for smart homes/building applications,” *Proceedings of the 2015 IEEE Innovative Smart Grid Technologies - Asia, ISGT ASIA 2015*, 1 2016.
- [25] Z-Wave Alliance, “About Z-Wave Technology .” https://z-wavealliance.org/about_z-wave_technology/. Último Acceso: 04/09/2021.
- [26] IBM, “What is a REST API.” <https://www.ibm.com/cloud/learn/rest-apis>, 4 2021. Último Acceso: 04/09/2021.
- [27] V. Wang, F. Salim, and P. Moskovits, “The WebSocket Protocol,” *The Definitive Guide to HTML5 WebSocket*, pp. 33–60, 2013.
- [28] IEEE Communications Society, “Full Duplex Communications Theory, Standardization and Practice .” <https://www.comsoc.org/publications/magazines/ieee-wireless-communications/cfp/full-duplex-communications-theory>, 2 2021. Último Acceso: 04/09/2021.
- [29] Y. Furukawa, “WEB-BASED CONTROL APPLICATION USING WEBSOCKET,” 2011.
- [30] W3 Schools, “HTTP Request Methods.” https://www.w3schools.com/tags/ref_httpmethods.asp. Último Acceso: 04/09/2021.
- [31] Red Hat, “What is a REST API?.” <https://www.redhat.com/en/topics/api/what-is-a-rest-api>, 5 2020. Último Acceso: 04/09/2021.
- [32] Educative, “Web Application and Software Architecture 101.” <https://www.educative.io/courses/web-application-software-architecture-101/qADAzX6yorR?aid=5082902844932096>. Último Acceso: 04/09/2021.
- [33] RESTful API, “What is REST.” <https://restfulapi.net/>, 2018. Último Acceso: 04/09/2021.
- [34] Google, “Control Your Smart Home.” <https://assistant.google.com/smart-home/>. Último Acceso: 04/09/2021.
- [35] Google Developers, “Home Graph: Actions on Google Smart Home.” <https://developers.google.com/assistant/smarthome/concepts/homegraph>, 1 2021. Último Acceso: 04/09/2021.
- [36] openHAB, “Welcome to OpenHAB: Introduction .” <https://www.openhab.org/docs/>. Último Acceso: 04/09/2021.
- [37] openHab, “myopenHAB: Cloud service.” <https://www.myopenhab.org/>. Último Acceso: 04/09/2021.
- [38] Tuya Smart, “No-Coding Development.” <https://www.tuya.com/product/iot/zero-coding-development>. Último Acceso: 04/09/2021.

- [39] Mark Watt Tech, “TUYA API (Setup, Basics and FREE FINGERBOTS).” <https://www.youtube.com/watch?v=w-BawMpxBYs>, 5 2021. Último Acceso: 04/09/2021.
- [40] Tuya Smart, “Preparation-IoT App SDK.” <https://developer.tuya.com/en/docs/app-development/preparation?id=Ka69nt983bhh5>, 8 2021. Último Acceso: 04/09/2021.
- [41] Espressif Systems, “ESP8266 Wi-Fi MCU.” <https://www.espressif.com/en/products/socs/esp8266>. Último Acceso: 04/09/2021.
- [42] Arduino, “Software.” <https://www.arduino.cc/en/software>. Último Acceso: 04/09/2021.
- [43] Firebase, “Firebase.” <https://firebase.google.com/>.
- [44] Android Developers, “Meet Android Studio.” <https://developer.android.com/studio/intro>.
- [45] H. V. Bhatnagar, P. Kumar, S. Rawat, and T. Choudhury, “Implementation model of Wi-Fi based Smart Home System,” *Proceedings on 2018 International Conference on Advances in Computing and Communication Engineering, ICACCE 2018*, pp. 23–28, 8 2018.
- [46] Lights Thing, “Proyecto IOT con ESP8266, android studio y firebase (Smart Home).” <https://www.youtube.com/watch?v=yJi90LYxVsM>, 3 2020. Último Acceso: 19/09/2021.
- [47] Raspberry Pi, “Teach, Learn, and Make with Raspberry Pi.” <https://www.raspberrypi.org/>. Último Acceso: 04/09/2021.
- [48] Home Assistant, “Home Assistant Features.” <https://www.home-assistant.io/>. Último Acceso: 04/09/2021.
- [49] Nabu Casa, “Home Assistant Cloud.” <https://www.nabucasa.com/>. Último Acceso: 04/09/2021.
- [50] Amazon Web Services, “What is AWS IoT?.” <https://www.youtube.com/watch?v=WAp6FHbhYck>, 1 2018. Último Acceso: 04/09/2021.
- [51] Amazon Web Services, “How AWS IoT works.” <https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html>. Último Acceso: 04/09/2021.
- [52] R. Hernández-Sampieri, *Metodología De La Investigación : Las Rutas Cuantitativa , cualitativa y mixta*. 2018.
- [53] M. d. l. A. Cienfuegos and A. Cienfuegos, “Lo cuantitativo y cualitativo en la investigación.” http://www.scielo.org.mx/scielo.php?pid=S2007-74672016000200015&script=sci_arttext, 12 2016. Último Acceso: 24/10/2021.

- [54] N. Juristo, *Basics of Software Engineering Experimentation*. 2013.
- [55] Boston University Medical Campus, “Sample Size for One Sample, Dichotomous Outcome.” https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_power/BS704_Power4.html, 3 2020. Último Acceso: 24/10/2021.
- [56] A. Holmes, B. Illowsky, and S. Dean, “Calculating the Sample Size n: Continuous and Binary Random Variables.” <https://opentextbc.ca/introbusinessstatopenstax/chapter/calculating-the-sample-size-n-continuous-and-binary-random-variables/>, 3 2015. Último Acceso: 24/10/2021.
- [57] K. Ulrich and S. Eppinger, *Diseño y Desarrollo de productos*. McGraw Hill, 4th ed., 2009.
- [58] Google Developers, “Device Actions Overview.” <https://developers.google.com/assistant/sdk/device-actions-overview#built-in>. Último Acceso: 04/09/2021.
- [59] Google Developers, “Connect smart home devices to the Google Assistant.” <https://codelabs.developers.google.com/codelabs/smarthome-washer#0>. Último Acceso: 04/09/2021.
- [60] D. Myers, “IoT & Google Assistant. Getting started with smart home.” <https://medium.com/google-developers/iot-google-assistant-f0908f354681>, 1 2019. Último Acceso: 19/07/2021.
- [61] Google Developers, “Google Assistant API .” <https://developers.google.com/assistant/sdk/reference/rpc>. Último Acceso: 04/09/2021.
- [62] Google Developers, “Integrate with Google Assistant.” <https://developers.google.com/assistant>. Último Acceso: 04/09/2021.
- [63] Google Developers, “Integrate the Assistant into Your Project (Other Languages).” <https://developers.google.com/assistant/sdk/guides/service/integrate#other-languages>. Último Acceso: 04/09/2021.
- [64] Google Developers, “Extend an Android app to Google Assistant with App Actions.” <https://codelabs.developers.google.com/codelabs/appactions#0>. Último Acceso: 04/09/2021.
- [65] Google Developers, “Introduction to the Google Assistant Service.” <https://developers.google.com/assistant/sdk/guides/service/python>, 11 2020. Último Acceso: 04/09/2021.
- [66] Python Basics, “What is Flask Python.” <https://pythonbasics.org/what-is-flask-python/>. Último Acceso: 16/09/2021.

- [67] A. Ibrahim, “Smart Home Automation IoT Using Raspberry Pi and Python.” <https://www.hackster.io/ahmedibrrahim/smart-home-automation-iot-using-raspberry-pi-and-python-47fb62>, 4 2018. Último Acceso: 16/09/2021.
- [68] T. Ryttilahti, “Python library & console tool for controlling Xiaomi smart appliances.” <https://github.com/rytilahti/python-miio>, 9 2021. Último Acceso: 16/09/2021.
- [69] Ionic Framework, “Cross-Platform Mobile App Development.” <https://ionicframework.com/>. Último Acceso: 16/09/2021.
- [70] C. Méndez, “Automatiza tu habitación con arduino — App movil nativa Ionic.” <https://www.youtube.com/watch?v=AUew8VBCMlc>, 6 2020. Último Acceso: 16/09/2021.
- [71] M. Hernández, “Light Control with NodeMcu and Ubidots.” <https://ubidots.com/blog/light-control-with-nodemcu-ubidots/>, 1 2017. Último Acceso: 16/09/2021.
- [72] Ubidots, “Ubidots: About.” <https://ubidots.com/>.
- [73] Ionic Marketplace, “Smart-home-app.” <https://market.ionicframework.com/themes/smart-home-app>. Último Acceso: 16/09/2021.
- [74] Lights Things, “Smart Home.” <https://lightsthing.net/proyecto/smart-home>, 3 2020.
- [75] Philips Hue, “Smart light apps – Control your lights.” <https://www.philips-hue.com/en-us/explore-hue/apps>. Último Acceso: 29/08/2021.
- [76] ioBroker, “ioBroker Smarthome.” <https://www.iobroker.net/>. Último Acceso: 29/10/2021.
- [77] Hubitat, “Local, Reliable, Fast and Private Home Automation.” <https://hubitat.com/>. Último Acceso: 29/10/2021.
- [78] Particle, “Welcome to the Particle docs.” <https://docs.particle.io/>. Último Acceso: 29/10/2021.
- [79] Domoticz, “Control at your fingertips.” <https://www.domoticz.com/>. Último Acceso: 29/10/2021.
- [80] HomeGenie, “Home Automation Server.” <https://homegenie.it/>. Último Acceso: 29/10/2021.
- [81] JuanMTech, “Install HACS in Home Assistant for themes and custom cards.” <https://www.youtube.com/watch?v=i4Jgr3tR4Ps>, 8 2021. Último Acceso: 16/09/2021.

- [82] H. Espinoza, G. Kling, F. McGroarty, M. O'Mahony, and X. Ziouvelou, "Estimating the impact of the Internet of Things on productivity in Europe," *Heliyon*, vol. 6, p. e03935, 5 2020.
- [83] D. Assante, E. Romano, M. Flamini, M. Castro, S. Martin, S. Lavirotte, G. Rey, M. Leisenberg, M. O. Migliori, I. Bagdoniene, R. T. Gallo, A. Pascoal, and M. Spatafora, "Internet of Things education: Labor market training needs and national policies," *IEEE Global Engineering Education Conference, EDUCON*, vol. 2018-April, pp. 1846–1853, 5 2018.

APPENDIX A

APÉNDICE A: BITÁCORA DE DISEÑO

A.1 Introducción

Este apéndice detalla aspectos relacionados a la documentación de las etapas del proceso de diseño en ingeniería, como se menciona en la sección 3. Se inicia exponiendo las necesidades establecidas, a través de la investigación para la comprensión del problema junto con una serie de entrevistas con el cliente. Seguidamente se detallan las especificaciones que permiten obtener valores o rangos que faciliten medir, numérica o subjetivamente, el cumplimiento de las necesidades. Finalmente, se profundiza en el proceso de estudio de conceptos, donde se presenta el proceso comenzando con la debida descomposición funcional, exploración de ideas, y análisis de conceptos.

A.2 Necesidades

Para comenzar a establecer las necesidades, se toma en cuenta para la realización de entrevistas tanto al contacto encargado de impartir el curso por parte del GII, el Dr. Francisco Bellas, como un profesional familiarizado y a cargo del proyecto *Robobo*, el Dr. Alejandro Paz. De esta manera se toma en cuenta la perspectiva en cuanto a requerimientos tanto del curso de IoT, como del *Robobo*. En la tabla A.1 se detallan las necesidades establecidas explícitamente por medio de las entrevistas con el cliente, donde se mencionan aspectos con relación al cumplimiento de diferentes funcionalidades, integración al *Robobo Framework*, dispositivos, facilidad de uso, y precios. Además, se solicita al cliente numerar la importancia que considera deberá tener el cumplimiento individual de cada necesidad. En la tabla A.2 se anotan los valores y sus respectivas definiciones para el establecimiento de importancia a cada necesidad.

Tabla A.1: Necesidades establecidas por el cliente.

No.	Necesidades del cliente	Importancia
El sistema permite al usuario distintas presentaciones para su interacción		
1	El sistema tiene una interfaz gráfica	2
2	El sistema es capaz de realizar sus funciones objetivo correctamente fuera del <i>Robobo Framework</i>	4
3	El sistema permite su integración al <i>Robobo Framework</i>	4
4	El sistema será capaz de realizar sus funciones objetivo correctamente dentro del <i>Robobo Framework</i>	4
El sistema permite la interacción con múltiples dispositivos inteligentes o sensores		
5	El sistema permite la lectura de los estados de los dispositivos inteligentes o sensores	5
6	El sistema permite la actuación sobre los estados de los dispositivos inteligentes o sensores	5
7	El sistema y sus sensores tienen un nivel de complejidad tal que sea posible su utilización por parte de personas sin un alto grado de conocimiento técnico	5
8	Se codifica la interacción con más de 2 dispositivos inteligentes o sensores	5
El sistema se ajusta a un presupuesto adecuado para proyectos educativos y con fines de replicabilidad		
9	Los dispositivos inteligentes o sensores serán accesibles por plataformas internacionales que faciliten su distribución	5
10	Los dispositivos inteligentes o sensores serán de un precio accesible	5

Tabla A.2: Valores para jerarquización de necesidades.

Valor	Importancia
1	La función es indeseable, no quiero un producto con esta función.
2	La función no es importante, pero no me importa si el producto la tiene
3	Sería bueno tener esa función, pero no es necesaria.
4	La función es muy deseable, pero podría no tenerla.
5	La función es muy importante, no quiero un producto que no tenga esta función.

A.3 Especificaciones

Se establecen especificaciones de manera que sea posible asignar valores meta que permitan evaluar el cumplimiento de las necesidades.

A.3.1 Métricas

Es posible definir un conjunto de métricas, o expresiones cuantificables que permiten definir características que la herramienta debe cumplir. En la tabla A.3 se observan las métricas junto con sus respectivas unidades para evaluación. Adicionalmente, es posible establecer la correspondencia entre métricas y necesidades en la tabla A.4. En las filas se tiene las necesidades, y en las columnas las métricas. Se observa que cada métrica se relaciona con al menos una necesidad del cliente, a excepción de las necesidades 1 y 9 que tratan de la interfaz gráfica del sistema y la accesibilidad en cuanto a los distribuidores para compra de dispositivos, respectivamente. Esto se debe a que estas necesidades, al evaluarlas, no presentarán un grado de cumplimiento (como por ejemplo los rangos de confiabilidad de las métricas 1-3), sino que solamente se evaluará el cumplimiento o no cumplimiento de las mismas.

Tabla A.3: Establecimiento de métricas.

No. Métrica	No. Necesidad	Métrica	Imp.	Unidades
1	2,4,5,6	Error de lectura de dispositivos	5	%
2	2,4,5,6	Error en actuación de dispositivos	5	%
3	2	Error en conexión con dispositivos	5	%
4	3	Integración con el framework	4	Solicitud del cliente
5	7	Facilidad de utilización	5	Subjetivo
6	8	Cantidad de dispositivos	5	N: Número de dispositivos
7	10	Costo total de dispositivos	5	€, \$

A.3.2 Valores objetivo

Una vez establecidas las métricas es necesario el establecimiento de valores objetivo, tanto marginales como ideales. En la tabla A.5 se muestran estos valores, donde se establecen valores en forma de porcentaje de error, valores binarios, y valores subjetivos que serán definidos en la etapa de validación respectiva. Además se observa que se cuenta con un presupuesto de 500 euros, por lo cual la totalidad de los dispositivos comprados, incluyendo el envío, no deben sobrepasar este valor. Para profundizar en cuanto a la métrica número 9 (precios), se crea la tabla A.6, en donde se muestra el rango de precio marginal e ideal para cada dispositivo individual, en cada una de las categorías de dispositivos por adquirir. Estos precios se basan en el rango de distintos precios posibles de adquirir por medio de *Amazon*, con el objetivo de seleccionar dispositivos confiables y económicos.

Tabla A.4: Establecimiento de correspondencia entre necesidades y especificaciones

Necesidad/Métrica	1	2	3	4	5	6	7
1							
2	X	X	X				
3				X			
4	X	X					
5	X	X					
6	X	X					
7					X		
8						X	
9							
10							X

Tabla A.5: Especificaciones objetivo.

No. Métrica	Métrica	Imp.	Unidades	Valor Marginal	Valor ideal
1	Error de lectura de dispositivos	5	%	5-10	<5
2	Error en actuación de dispositivos	5	%	5-10	<5
3	Error en conexión con dispositivos	5	%	5-10	<5
4	Integración con el <i>Robo Framework</i>	4	Solicitud del cliente	Uso de Android Studio Función de lectura Función de actuación Función de listado	Uso de Android Studio Función de lectura Función de actuación Función de listado Función de automatización
5	Facilidad de utilización	5	Subjetivo	Fácil o neutro	Muy fácil
6	Cantidad de dispositivos	5	N: Número de dispositivos	2-4	>4
7	Costo total de dispositivos	5	€ \$	500 590	500 590

Tabla A.6: Profundización de valores de métrica No. 9.

Dispositivo	Valor Marginal	Valor Ideal	Unidades
Cámara	<80	<60	\$
	<100	<70	€
Sensor	<45	<30	\$
	<45	<30	€
Conector	<30	<20	\$
	<40	<30	€
Luces	<80	<50	\$
	<70	<50	€

A.4 Estudio de conceptos

Para poder comenzar de conceptualización, es necesario comenzar por medio de un análisis que permita estudiar las distintas alternativas, y el comportamiento de la herramienta por desarrollar.

A.4.1 Descomposiciones funcionales

Inicialmente, es posible establecer un análisis por medio de una “caja negra”, y una posterior descomposición por bloques. La caja negra, figura A.1, establece la existencia de entradas y salidas. Las entradas representan la energía que se requiere para alimentar los dispositivos por utilizar, los materiales implican las diferentes posibilidades de dispositivos por adquirir, y las señales indican toda la comunicación que se debe establecer para conectar al usuario con la herramienta y los dispositivos. Finalmente se cuenta con una salida en forma de señal, que permite al usuario reconocer que sus instrucciones, o señales de entrada, fueron ejecutadas correctamente.

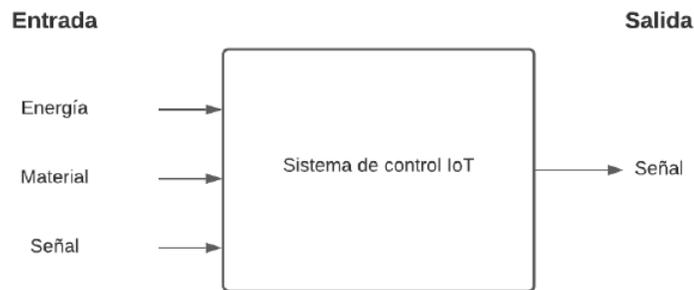


Figura A.1: Representación de caja negra. Adaptado de: [57, p. 116]

Posterior a la caja negra, es posible establecer una descomposición funcional de acuerdo a las entradas y salidas planteadas previamente. En la figura A.2 se observa la descomposición que permite relacionar los elementos necesarios en las entradas, para obtener la salida deseada. Es necesario tomar en cuenta posibilidades en cuanto a la plataforma por utilizar, tipos de dispositivos, su alimentación, y su comunicación con la plataforma.

A.4.1.1 Exploración de conceptos

Es necesaria la exploración de conceptos, para permitir el análisis de las diferentes posibilidades en cuanto a las diferentes funcionalidades del sistema. Se muestran en las figuras A.3-A.5 los arboles de clasificación de cada subproblema.

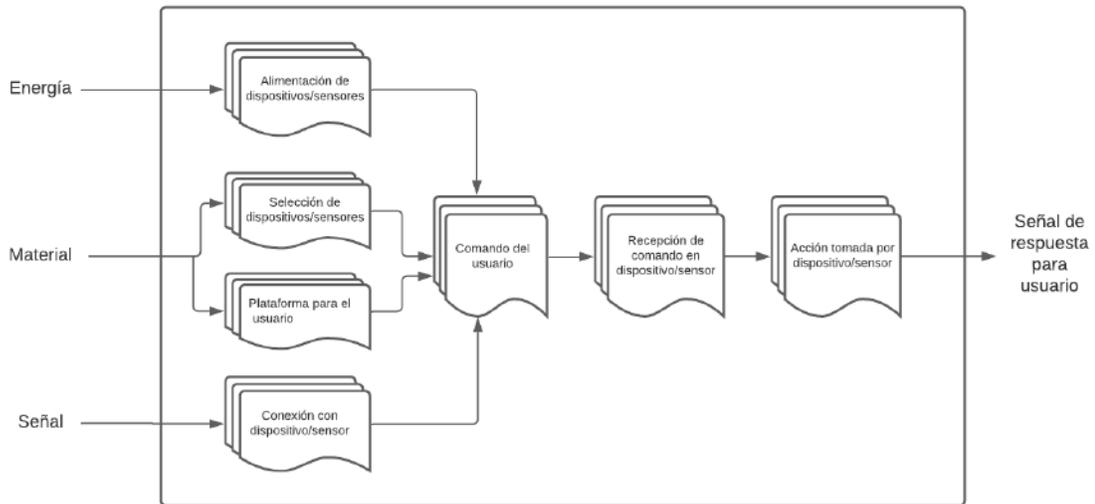


Figura A.2: Descomposición funcional del proyecto. Elaboración: fuente propia

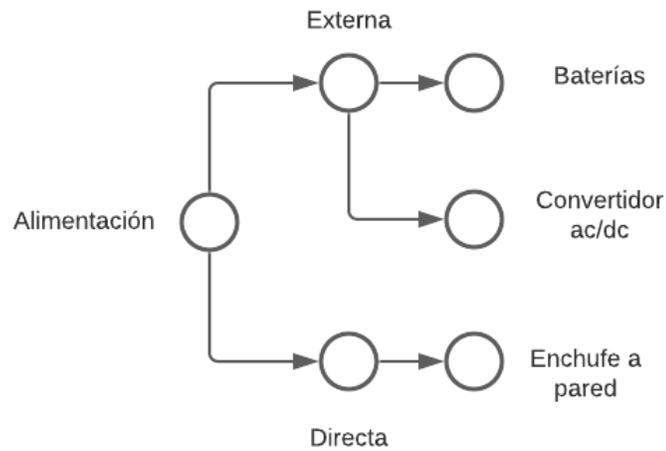


Figura A.3: Árbol de clasificación para subproblema de alimentación. Elaboración: fuente propia

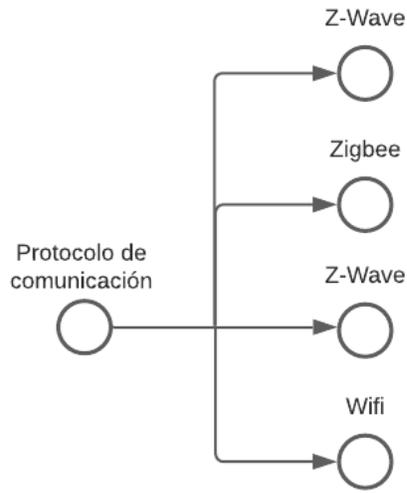


Figura A.4: Árbol de clasificación para subproblema de protocolos de comunicación. Elaboración: fuente propia

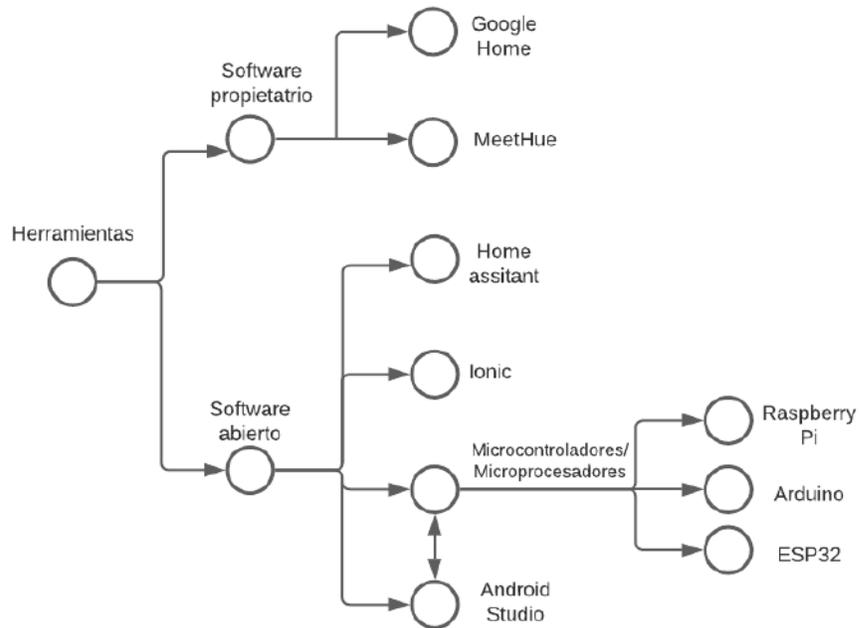
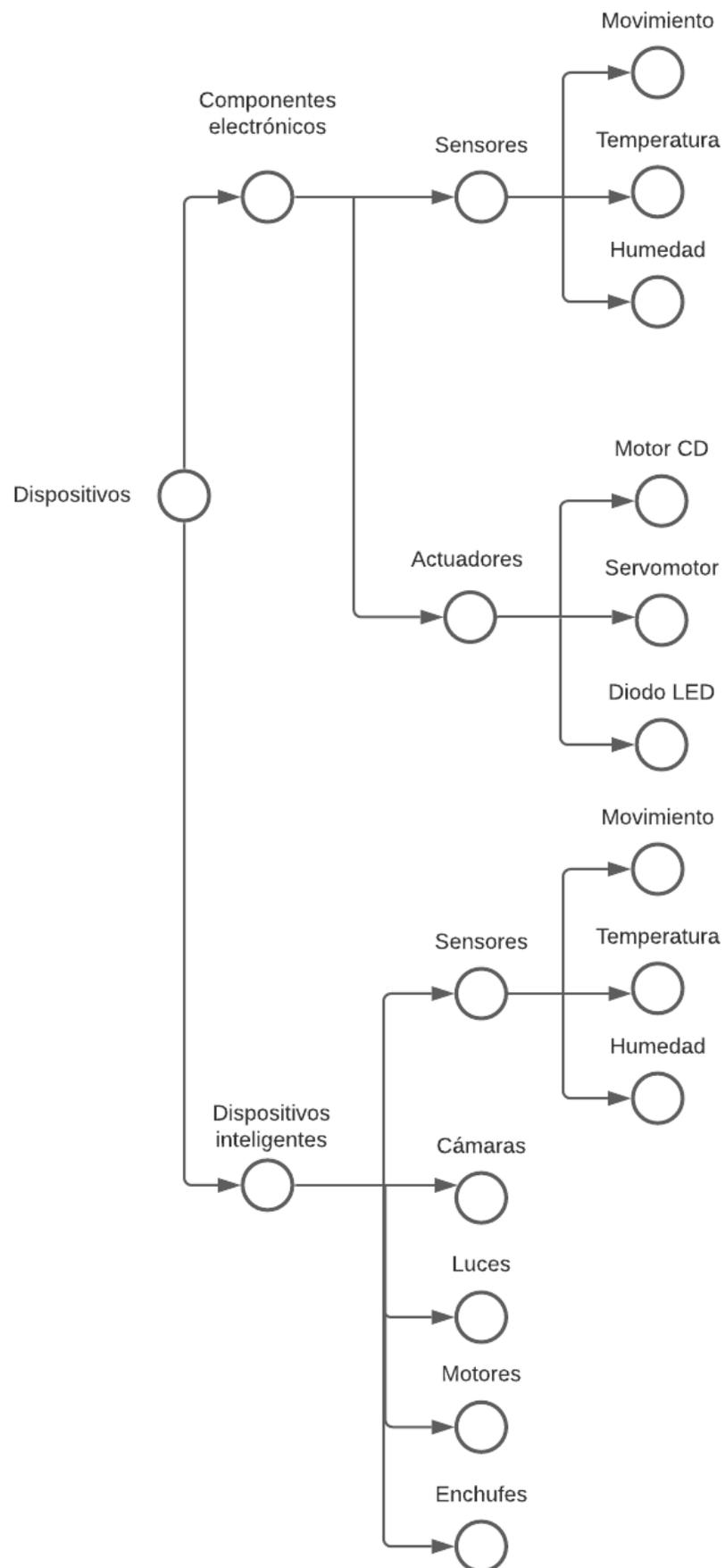


Figura A.5: Árbol de clasificación para subproblema de selección de herramientas. Elaboración: fuente propia



Posteriormente, por medio de la búsqueda tanto interna como externa, es posible establecer distintas combinaciones en cuanto a ideas, para poder generar conceptos para ser posteriormente evaluados. En las figuras A.7-A.12 se demuestran las diferentes combinaciones establecidas. Los diferentes conceptos se basan en las diferentes plataformas investigadas, y en sus diferentes posibilidades en cuanto a tipos de dispositivos y protocolos de comunicación.

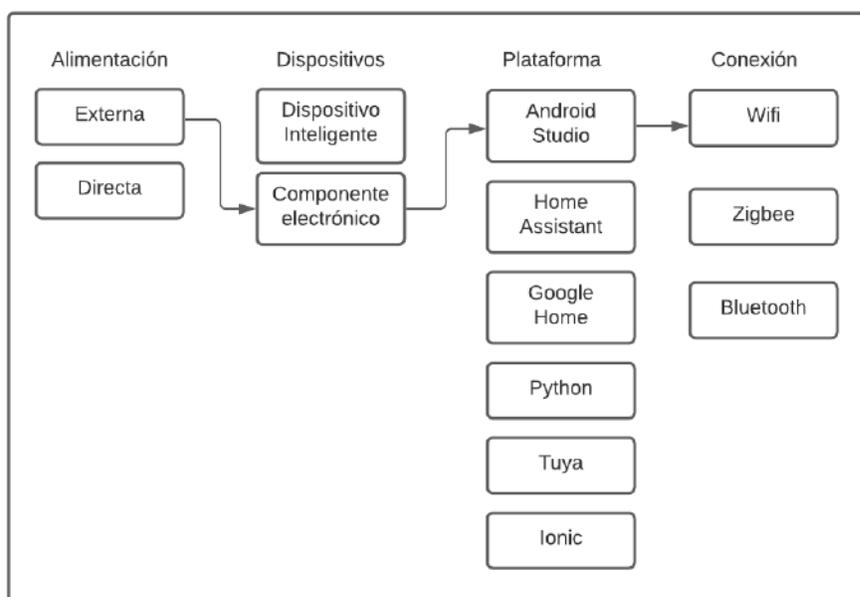


Figura A.7: Tabla de combinación para concepto No.1:Android Studio. Elaboración: fuente propia

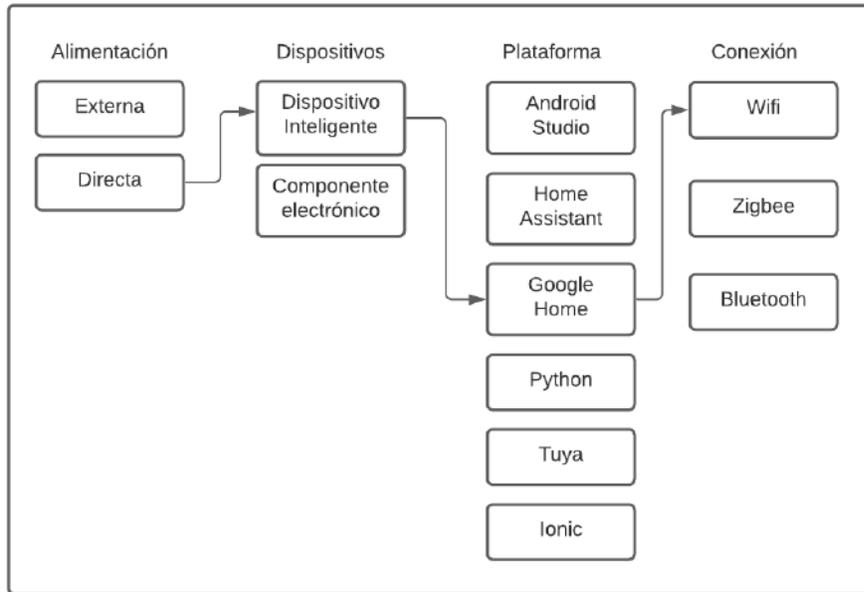


Figura A.8: Tabla de combinación para concepto No.2: Google Home. Elaboración: fuente propia

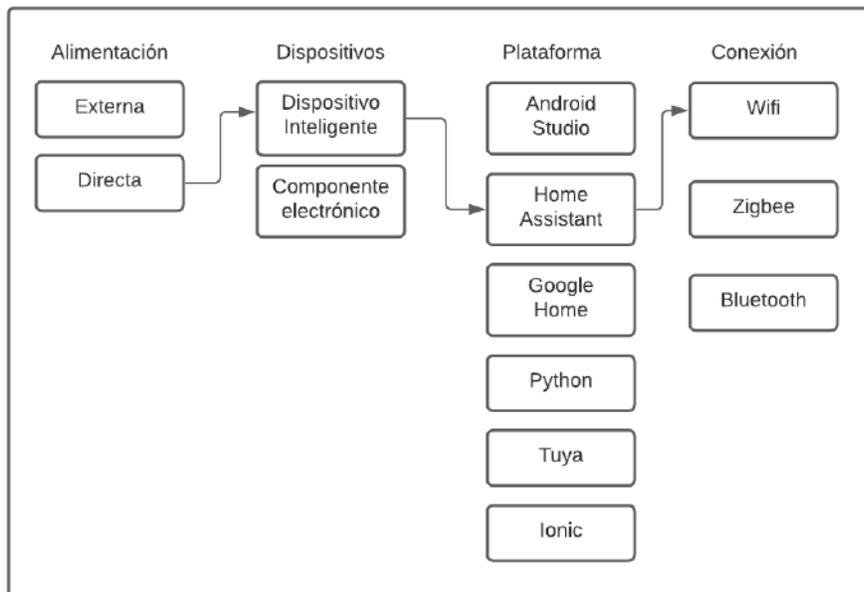


Figura A.9: Tabla de combinación para concepto No.3: Home Assistant. Elaboración: fuente propia

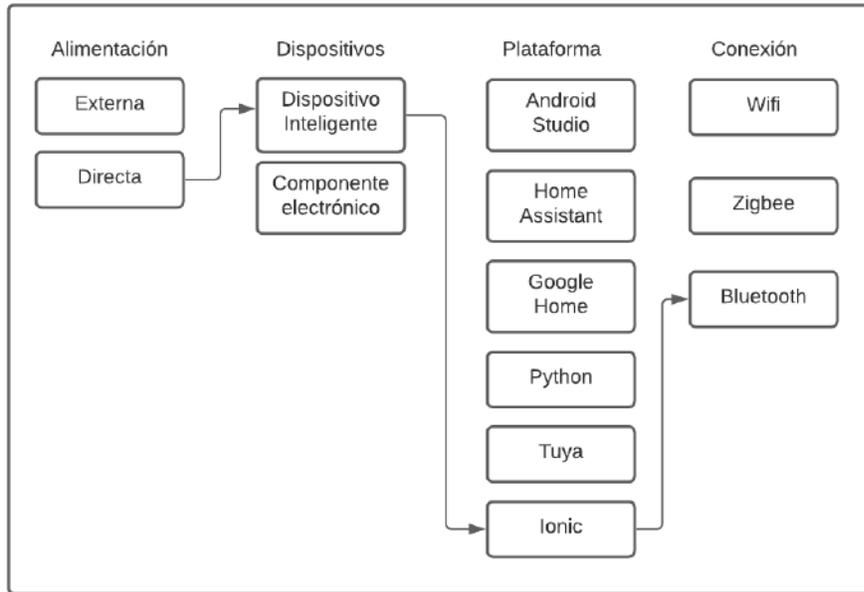


Figura A.10: Tabla de combinación para concepto No.4: Ionic. Elaboración: fuente propia

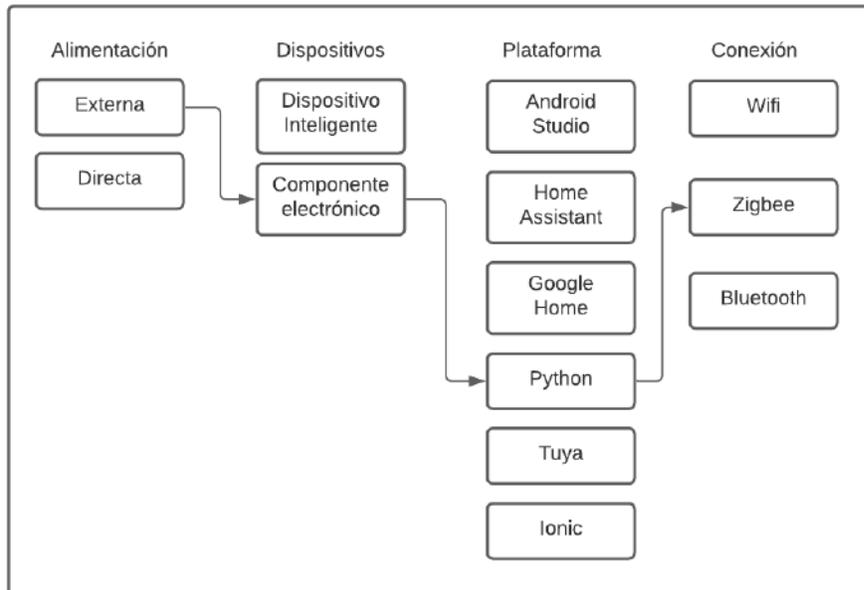


Figura A.11: Tabla de combinación para concepto No.5: Python. Elaboración: fuente propia

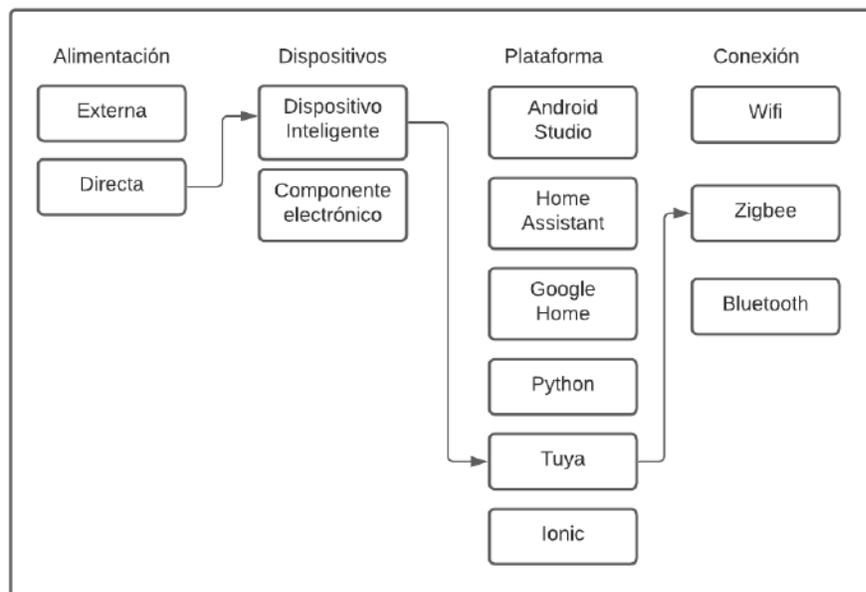


Figura A.12: Tabla de combinación para concepto No.6: Tuya Smart. Elaboración: fuente propia

A.4.1.2 Selección de conceptos

Una vez establecidas las diferentes combinaciones de alternativas, es posible continuar con el estudio de cada concepto, por medio de dos etapas: filtrado y evaluación. Se comienza filtrando los conceptos mediante criterios establecidos con base en las necesidades del cliente, y se evalúa de manera que es posible descartar, seleccionar, o combinar conceptos con base a sus características. En la tabla A.7 se observa como se seleccionan 3 conceptos distintos que permiten cumplir adecuadamente con los requerimientos del proyecto. Estos conceptos proceden a ser evaluados en la tabla A.8, se ven reflejados los pesos (basados en la jerarquización de necesidades) que permiten elegir un único concepto que permitiría continuar al desarrollo de una plataforma IoT.

Tabla A.7: Filtrado de conceptos.

Criterios	Conceptos					
	Python	Home Assistant	Ionic	Android Studio	Tuya Smart	Google Home (REF)
Presentación	-	0	0	+	-	0
Funcionalidad independiente	0	0	0	0	0	0
Adaptación al Robobo Framework	+	+	-	+	+	0
Control sobre dispositivos	-	+	0	0	0	0
Facilidad para utilizar	-	-	0	-	-	0
Flexibilidad para crecimiento	-	+	-	+	-	0
Accesibilidad	0	0	0	+	-	0
—	—	—	—	—	—	—
Suma +	1	3	0	4	1	0
Suma 0	2	3	5	2	2	7
Suma -	4	1	2	1	4	0
Evaluación Neta	-3	2	-2	3	-3	0
Lugar	6	2	4	1	5	3
¿Continuar?	No	Sí	No	Sí	No	Sí

Posterior a la selección de la plataforma de *Android studio*, y por medio de retroalimentación por parte del cliente, se decide iterar sobre la etapa de búsqueda de conceptos. El cliente descarta la opción presentada debido al nivel de complejidad que implicaría sobre los usuarios la necesidad de realizar conexiones manuales (ya que la implementación de *Android Studio* implicaría el uso de componentes electrónicos y una placa ESP8266). Por lo tanto, se redirecciona la investigación hacia plataformas de control IoT, preferiblemente de código abierto. De esta manera, se investigan 10 conceptos distintos en esta iteración del proceso de diseño. Se observa en la figura A.13 las posibilidades por estudiar. Debido a la naturaleza de las plataformas, estas operan todas por medio de dispositivos inteligentes (conectados directamente a la red eléctrica de una habitación). Además la mayor parte de las plataformas permiten el control de dispositivos en los diferentes protocolos de comunicación.

Se presenta en la tabla A.9 la iteración del filtrado de conceptos. Se utilizan los mismos criterios establecidos previamente. Se continúa hacia la evaluación después de seleccionar 3 conceptos: *Open hab*, *Home Assistant* y *Tuya Smart* debido que sobresalen con respecto a los demás en cuanto a facilidad, integración con el *Robobo Framework*, interfaz de usuario, y flexibilidad de crecimiento. En la tabla A.10 se

Tabla A.8: Evaluación de conceptos.

Criterio	Peso (%)	Concepto					
		Home Assistant		Android Studio		Google Home	
		Calificación	E.P	Calificación	E.P	Calificación	E.P
1. Presentación	5	4	0.2	3	0.15	2	0.1
2. Funcionalidad independiente	17	4	0.68	5	0.85	3	0.51
3. Adaptación al Robobo Framework	10	3	0.30	3	0.3	1	0.1
4. Control sobre dispositivos	17	3	0.51	3	0.51	1	0.17
5. Facilidad para utilizar	17	2	0.34	1	0.17	3	0.51
6. Flexibilidad para crecimiento	17	3	0.51	3	0.51	1	0.17
7. Accesibilidad	17	2	0.34	3	0.51	2	0.34
—	—	—	—	—	—	—	—
Total puntos	—	—	2.88	—	3.00	—	1.90
Lugar	—	—	2	—	1	—	3
¿Continuar?	—	—	No	—	Sí	—	No

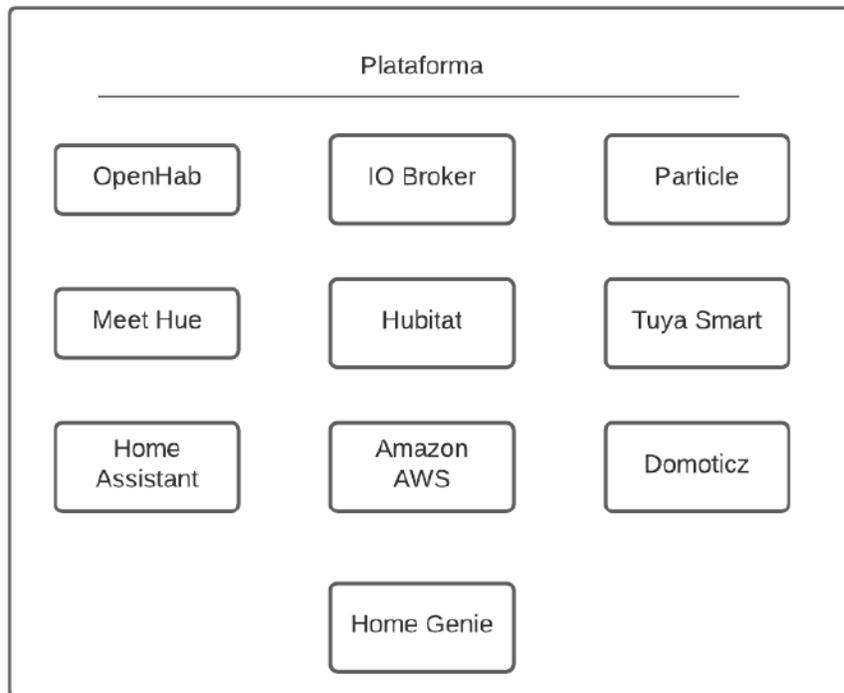


Figura A.13: Tabla de combinación para segunda iteración de conceptos. Elaboración: fuente propia

observa la evaluación de las plataformas mencionadas, donde se propone una nueva distribución de pesos de manera que se resalta la importancia en cuanto a la facilidad de uso de la plataforma. Finalmente, el concepto seleccionado consiste en la implementación de *Home Assistant* para el control IoT de múltiples dispositivos inteligentes.

Tabla A.9: Filtrado de conceptos tras realimentación.

Criterios	Conceptos									
	Open Hab	Meet Hue	Home Assistant	Io Broker	Habitat (REF)	AWS	Particle	Tuya Smart	Domoticz	Home Genie
Presentación	0	0	0	-	0	0	-	-	0	0
Funcionalidad independiente	0	-	0	0	0	-	0	0	0	0
Adaptación al Robobo Framework	+	-	+	0	0	-	-	+	+	0
Control sobre dispositivos	0	-	0	0	0	0	0	0	-	0
Facilidad para utilizar	+	+	+	0	0	0	-	+	0	0
Flexibilidad para crecimiento	0	-	0	-	0	-	-	0	-	-
Accesibilidad	0	-	0	-	0	0	-	0	-	-
—	—	—	—	—	—	—	—	—	—	—
Suma +	2	2	2	0	0	0	0	2	1	0
Suma 0	5	0	5	4	7	4	2	4	3	5
Suma -	0	5	0	3	0	3	5	1	3	2
Evaluación Neta	2	-3	2	-3	0	-3	-5	1	-2	-2
Lugar	1	5	1	5	3	5	6	2	4	4
¿Continuar?	Sí	No	Sí	No	No	No	No	Sí	No	No

Tabla A.10: Evaluación de conceptos tras realimentación.

Criterio	Peso (%)	Concepto					
		Open Hab		Home Assistant		Tuya Smart	
		Calificación	E.P	Calificación	E.P	Calificación	E.P
1. Presentación	5	3	0.15	3	0.15	2	0.1
2. Funcionalidad independiente	15	3	0.15	3	0.45	2	0.3
3. Adaptación al Robobo Framework	10	3	0.3	3	0.3	3	0.3
4. Control sobre dispositivos	15	3	0.45	3	0.45	1	0.15
5. Facilidad para utilizar	25	2	0.5	3	0.75	1	0.25
6. Flexibilidad para crecimiento	15	3	0.45	3	0.45	2	0.3
7. Accesibilidad	15	3	0.45	3	0.45	2	0.3
—	—	—	—	—	—	—	—
Total puntos	—	—	2.75	—	3.00	—	1.70
Lugar	—	—	2	—	1	—	3
¿Continuar?	—	—	No	—	Sí	—	No

A.5 Validación de resultados

Se muestra a continuación los resultados obtenidos de las pruebas de validación para cada uno de los dispositivos.

Tabla A.11: Resultados de pruebas de lectura en *Home Assistant* para Smart Plug Sonoff.

Smart Plug Sonoff							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	0	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							1,471

Tabla A.12: Resultados de pruebas de lectura en *Home Assistant* para Smart Plug Tapo.

Smart Plug Tapo							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	0	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	0	51	1	68	1
Error (%)							2,941

Tabla A.13: Resultados de pruebas de tiempo promedio de lectura en *Home Assistant* para Smart Plug Tapo.

Smart Plug Tapo							
Prueba No.	Tiempo (s)	Prueba No.	Tiempo (s)	Prueba No.	Tiempo (s)	Prueba No.	Tiempo (s)
1	11,24	18	7,59	35	24,68	52	26,73
2	23,82	19	17,03	36	24,32	53	24,54
3	20,88	20	21,04	37	14,2	54	26,00
4	7,55	21	25,65	38	24,63	55	23,53
5	22,36	22	25,17	39	12,25	56	16,87
6	11,04	23	26,08	40	16,47	57	23,48
7	10,69	24	24,47	41	19,13	58	26,27
8	21,98	25	23,93	42	22,71	59	26,52
9	25,4	26	25,45	43	23,65	60	22,11
10	25,95	27	26,03	44	16,58	61	25,79
11	23,79	28	23,95	45	23,51	62	25,58
12	18,75	29	14,95	46	25,55	63	26,01
13	20,91	30	25,41	47	26,22	64	19,71
14	23,83	31	21,43	48	20,73	65	21,09
15	20,19	32	22,51	49	26,81	66	19,84
16	24,31	33	23,49	50	23,34	67	18,21
17	24,19	34	21,51	51	22,97	68	27,69
Tiempo Promedio							21,769
Desviación Estándar							4.785

Tabla A.14: Resultados de pruebas de lectura en *Home Assistant* para Luz Yeelight.

Luz Yeelight							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.15: Resultados de pruebas de lectura en *Home Assistant* para Luz Lifx.

Luz Lifx							
Prueba No.	Lectura correcta						
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	0	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	0	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							2,941

Tabla A.16: Resultados de pruebas de tiempo promedio de lectura en *Home Assistant* para Luz Lifx.

Prueba No.	Tiempo(s)	Prueba No.	Tiempo(s)	Prueba No.	Tiempo(s)	Prueba No.	Tiempo(s)
1	5,93	18	8,62	35	9,79	52	8,93
2	8,2	19	7,23	36	5,12	53	4,98
3	4,37	20	4,33	37	4,71	54	7,28
4	2,33	21	9,84	38	4,41	55	4,61
5	6,51	22	6,16	39	6,51	56	1,92
6	5,93	23	9,78	40	8,72	57	3,68
7	5,15	24	4,43	41	6,93	58	2,95
8	1,86	25	9,24	42	5,35	59	8,14
9	7,23	26	3,96	43	7,65	60	3,44
10	5,66	27	6,42	44	4,5	61	4,34
11	3,4	28	8,02	45	5,78	62	3,76
12	3,67	29	1,6	46	6,81	63	1,14
13	6,89	30	3,91	47	4,52	64	3,96
14	6,17	31	4,69	48	6,5	65	4,87
15	2,37	32	2,38	49	0,92	66	4,48
16	4,18	33	2,26	50	5,17	67	8,14
17	3,11	34	2,12	51	9,2	68	0,81
Tiempo Promedio							5,264
Desviación Estándar							2,351

Tabla A.17: Resultados de pruebas de lectura en *Home Assistant* para Sensor de puerta Sonoff.

Sensor de puerta Sonoff							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.18: Resultados de pruebas de lectura en *Home Assistant* para Sensor de movimiento Aqara.

Sensor Movimiento Aqara							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.19: Resultados de pruebas de actuación en *Home Assistant* para Smart Plug Sonoff.

Smart Plug Sonoff							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	0	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							1,471

Tabla A.20: Resultados de pruebas de actuación en *Home Assistant* para Smart Plug Tapo.

Smart Plug Tapo							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	0	43	1	60	0
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	0	33	0	50	1	67	1
17	0	34	0	51	1	68	1
Error (%)							8,824

Tabla A.21: Resultados de pruebas de actuación en *Home Assistant* para Luz Yeelight.

Luz Yeelight							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.22: Resultados de pruebas de actuación en *Home Assistant* para Luz Lifx.

Luz Lifx							
Prueba No.	Actuación correcta						
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.23: Resultados de pruebas de conexión en *Home Assistant* para Smart Plug Sonoff.

Smart Plug Sonoff							
Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	0	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							1,471

Tabla A.24: Resultados de pruebas de conexión en *Home Assistant* para Smart Plug Tapo.

Smart Plug Tapo							
Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta
1	0	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	0	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							2,941

Tabla A.25: Resultados de pruebas de conexión en *Home Assistant* para Luz Yeelight.

Luz Yeelight							
Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	0	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							1,471

Tabla A.26: Resultados de pruebas de conexión en *Home Assistant* para Luz Lifx.

Luz Lifx							
Prueba No.	Conexión correcta						
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.27: Resultados de pruebas de conexión en *Home Assistant* para Sensor movimiento Aqara.

Sensor movimiento Aqara							
Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.28: Resultados de pruebas de conexión en *Home Assistant* para Sensor de puerta Sonoff.

Sensor de puerta Sonoff							
Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta	Prueba No.	Conexión correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	0	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							1,471

Tabla A.29: Resultados de pruebas de lectura en aplicación diseñada, para Smart plug Sonoff.

Smart plug Sonoff							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.30: Resultados de pruebas de lectura en aplicación diseñada, para Smart plug Tapo.

Smart plug Tapo							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	0
2	1	19	1	36	1	53	0
3	1	20	1	37	1	54	1
4	0	21	1	38	1	55	1
5	0	22	1	39	1	56	1
6	0	23	0	40	1	57	1
7	1	24	0	41	1	58	1
8	1	25	0	42	1	59	1
9	1	26	0	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	0	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	0	68	1
Error (%)							16,176

Tabla A.31: Resultados de pruebas de lectura en aplicación diseñada, para Luz Yeelight.

Luz Yeelight							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.32: Resultados de pruebas de lectura en aplicación diseñada, para Luz Lifx.

Luz Lifx							
Prueba No.	Lectura correcta						
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.33: Resultados de pruebas de lectura en aplicación diseñada, para Sensor de movimiento Aqara.

Sensor de movimiento Aqara							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.34: Resultados de pruebas de lectura en aplicación diseñada, para Sensor de puerta Sonoff.

Sensor de puerta Sonoff							
Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta	Prueba No.	Lectura correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.35: Resultados de pruebas de actuación en aplicación diseñada, para Smart Plug Sonoff.

Smart Plug Sonoff							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.36: Resultados de pruebas de actuación en aplicación diseñada, para Smart Plug Tapo.

Smart Plug Tapo							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	0	41	1	58	0
8	1	25	0	42	1	59	0
9	1	26	0	43	1	60	1
10	1	27	0	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							8,824

Tabla A.37: Resultados de pruebas de actuación en aplicación diseñada, para Luz Yeelight.

Smart Luz Yeelight							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

Tabla A.38: Resultados de pruebas de actuación en aplicación diseñada, para Luz Lifx.

Smart Luz Lifx							
Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta	Prueba No.	Actuación correcta
1	1	18	1	35	1	52	1
2	1	19	1	36	1	53	1
3	1	20	1	37	1	54	1
4	1	21	1	38	1	55	1
5	1	22	1	39	1	56	1
6	1	23	1	40	1	57	1
7	1	24	1	41	1	58	1
8	1	25	1	42	1	59	1
9	1	26	1	43	1	60	1
10	1	27	1	44	1	61	1
11	1	28	1	45	1	62	1
12	1	29	1	46	1	63	1
13	1	30	1	47	1	64	1
14	1	31	1	48	1	65	1
15	1	32	1	49	1	66	1
16	1	33	1	50	1	67	1
17	1	34	1	51	1	68	1
Error (%)							0,000

APPENDIX B

APÉNDICE B: MANUAL DE USUARIO

A continuación, se detalla una guía paso a paso para lograr la instalación, configuración y control de un entorno IoT en la plataforma de *Home Assistant*, junto con una serie de dispositivos de diferentes marcas, tipos, y funcionalidades

B.1 Instalación en Windows

Se describen a continuación los pasos para la instalación de *Home Assistant*.

1. Para utilizar *Home Assistant* en Windows, se requiere descargar una máquina virtual. Inicialmente, debe dirigirse a la página de *Home Assistant*, para instalaciones en Windows. En el siguiente link, dar click y descargar la imagen (o el *software*) para *VirtualBox(.vdi)*:
<https://www.home-assistant.io/installation/windows>
 - (a) Una vez descargado el archivo, se debe descomprimir (se puede usar winrar, winzip o cualquier programa para descomprimir archivos), como en la figura B.1.
2. Ahora, dirigirse a la siguiente página y dar click en *Windows Hosts* para descargar:
<https://www.virtualbox.org/wiki/Downloads> .
 - (a) Una vez descargado el archivo ejecutable, dar doble click para instalarlo.
 - (b) En la ventana de instalación, dar click en **Siguiente** hasta finalizar la instalación. La instalación puede tomar algunos minutos, estar atento a si el ordenador solicita permisos para la instalación, en caso de ser así, aceptar los permisos de instalación.
 - (c) Este proceso se observa en las figuras B.2 - B.7.
3. Al finalizar la instalación, se abrirá la ventana de la interfaz de la máquina virtual, como en la figura B.8, donde se debe seleccionar **Nueva**.

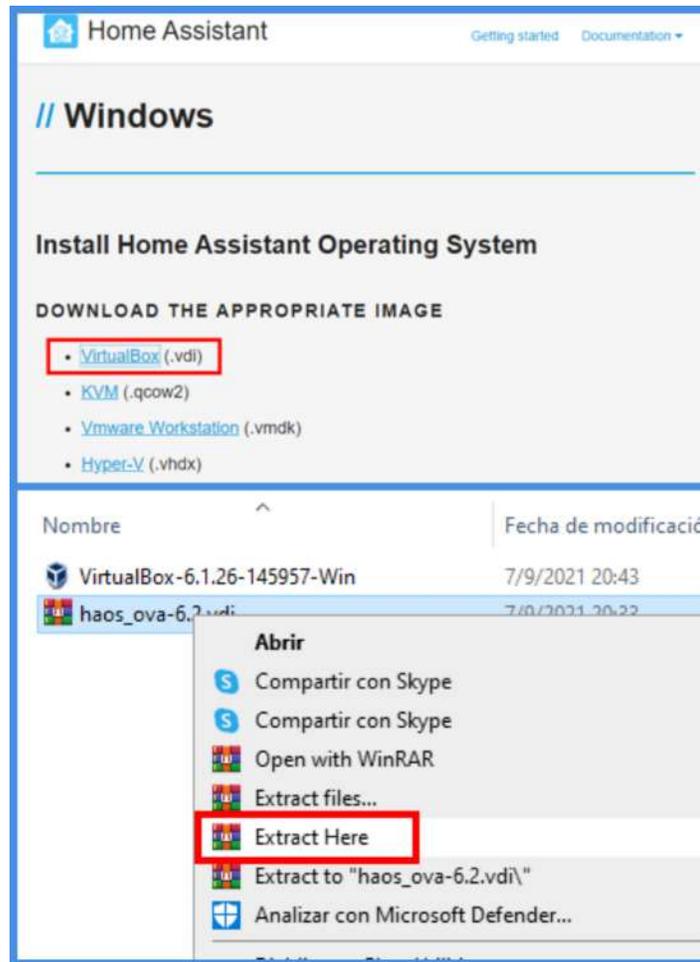


Figura B.1: Descarga imagen de *Virtualbox* de *Home Assistant*.



Figura B.2: Descarga de *Virtualbox*.



Figura B.3: Instalación de *Virtualbox*.

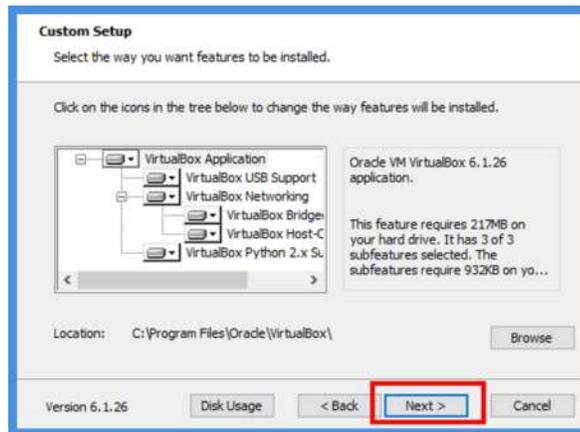


Figura B.4: Instalación de *Virtualbox* (continuación).

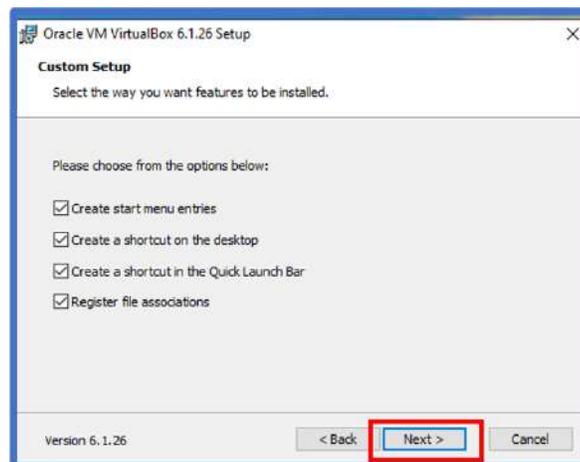


Figura B.5: Instalación de *Virtualbox* (continuación).



Figura B.6: Instalación de *Virtualbox* (continuación).

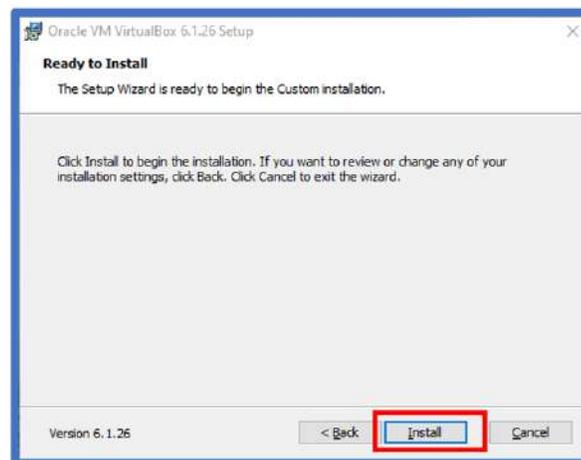


Figura B.7: Instalación de *Virtualbox* (continuación).

- (a) Seguidamente, rellenar la ventana con la información que se muestra en la figura B.9, y dar click en **Next**.
- (b) **Nota:** si la pestaña de Versión no muestra una opción para 64-bit, referirse a la sección B.11: Activación de Virtualización en Computador.
- (c) Se recomienda seleccionar al menos 2 MB de RAM (2048 MB) (figura B.10). Click en **Next**.



Figura B.8: Interfaz de máquina virtual.

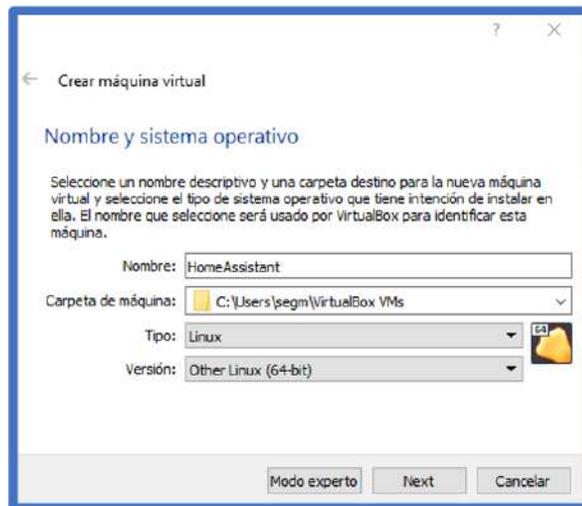


Figura B.9: Configuración de máquina virtual.

- 4. Después, seleccionar la opción de “Usar un archivo de disco duro virtual existente” como se muestra en las figuras B.11-B.15, y seleccionar la imagen descargada anteriormente de la página de Home Assistant (figura B.1).
 - (a) Al seleccionar adecuadamente la imagen de *Home Assistant*, se tendrá lista la máquina virtual con el *software* requerido para la creación de un entorno IoT.
- 5. Ahora, dar click en configuración. Hay que realizar los siguientes cambios (figuras B.16-B.19):

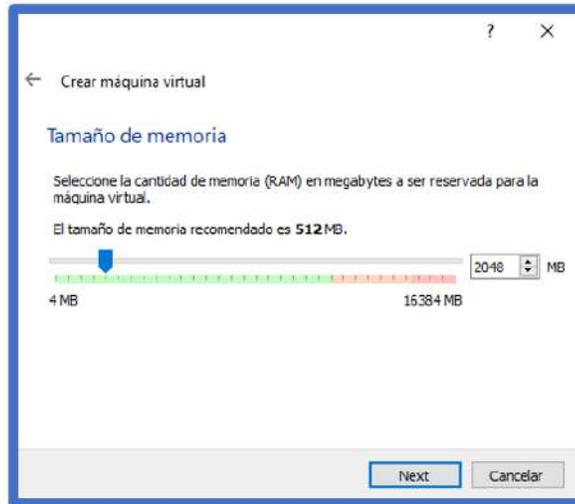


Figura B.10: Configuración de máquina virtual (continuación).

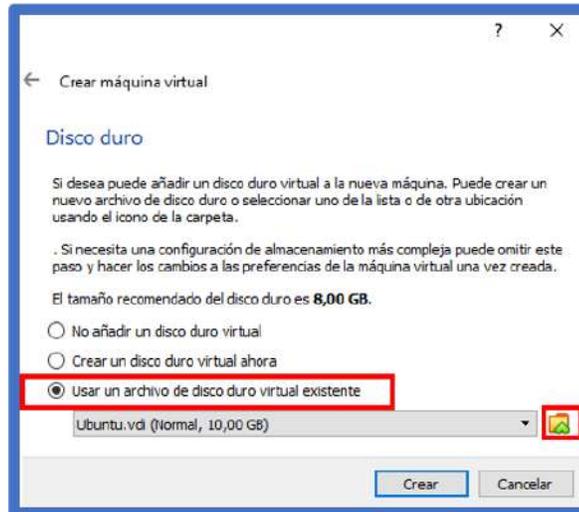


Figura B.11: Selección de imagen de *Home Assistant*.

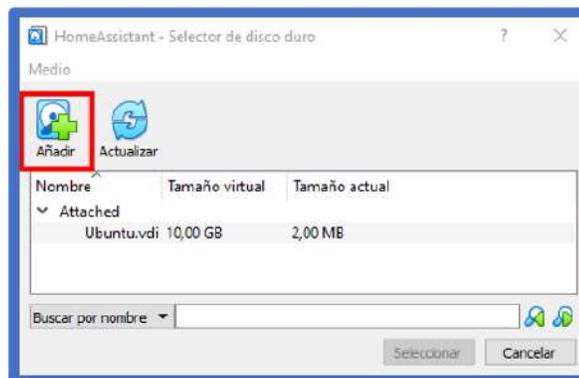


Figura B.12: Selección de imagen de *Home Assistant* (continuación).

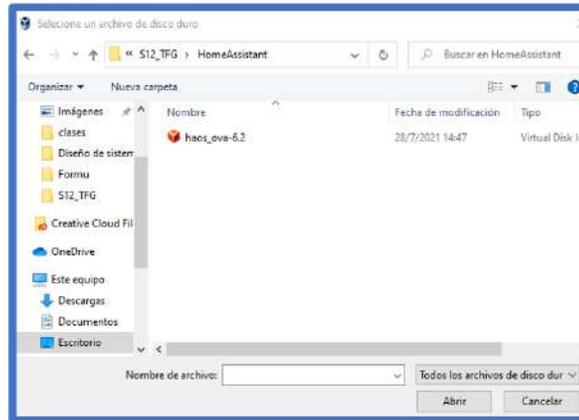


Figura B.13: Selección de imagen de *Home Assistant* (continuación).

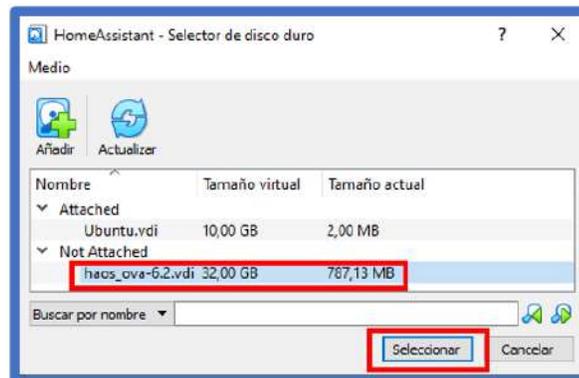


Figura B.14: Selección de imagen de *Home Assistant* (continuación).

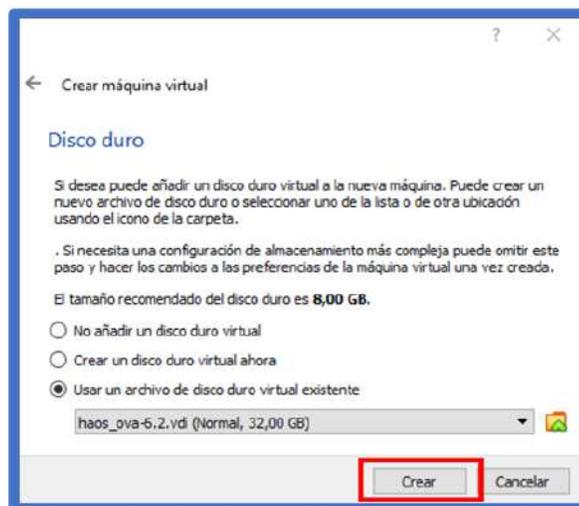


Figura B.15: Selección de imagen de *Home Assistant* (continuación).

- (a) Sistema -> Procesador -> 2 CPUs (si su sistema lo permite).
- (b) Sistema -> Placa base (Motherboard) -> Habilitar EFI.
- (c) Red-> Conectado a: Adaptador puente (bridge adapter).
- (d) Luego dar click en Aceptar.

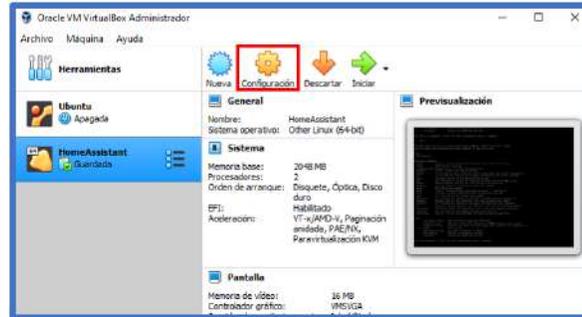


Figura B.16: Configuración de máquina virtual con *Home Assistant* .

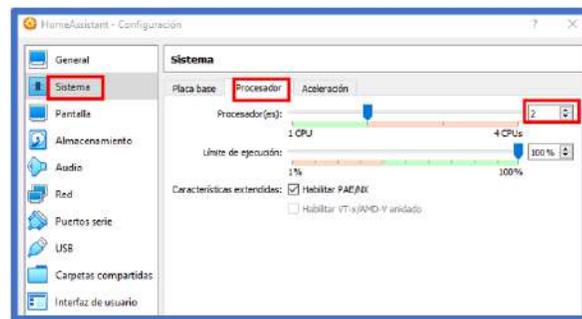


Figura B.17: Configuración de máquina virtual con *Home Assistant* (continuación).

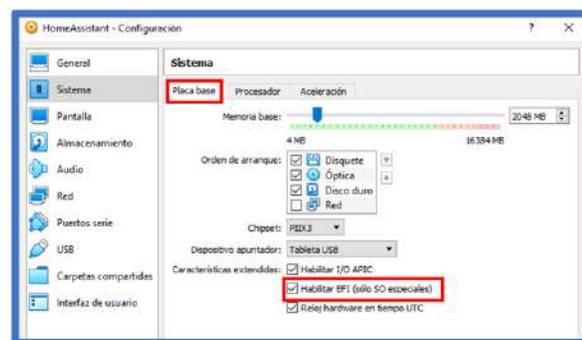


Figura B.18: Configuración de máquina virtual con *Home Assistant* (continuación).

6. Dar click en Iniciar la máquina virtual.

- (a) Comenzará el proceso de instalación, se abrirá una ventana mostrando el proceso. Esperar a que la máquina virtual termine la instalación de *Home Assistant*.

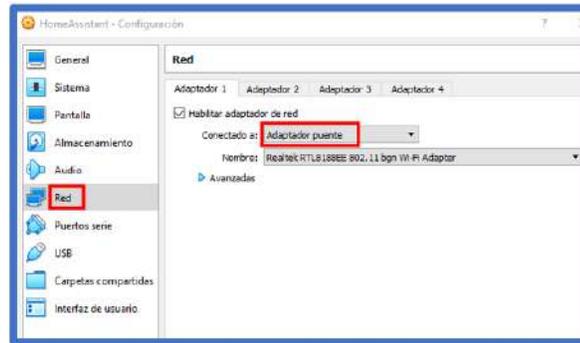


Figura B.19: Configuración de máquina virtual con *Home Assistant* (continuación).

- (b) Al terminar, se mostrará en la ventana el nombre de *Home Assistant*, y se mostrará una dirección IP y un URL de home assistant (ver ejemplo en figura B.20). Estos datos se podrán ingresar en un navegador (por ejemplo *Chrome*) para poder ingresar a la interfaz de usuario de *Home Assistant*, como se muestra en la figura B.21.
- i. En navegador web, ingresar a (donde direcciónIP representa la dirección IP personal que se obtiene en su ordenador): `http://direcciónIP:8123`
 - ii. En navegador web, ingresar a: `http://homeassistant.local:8123/`

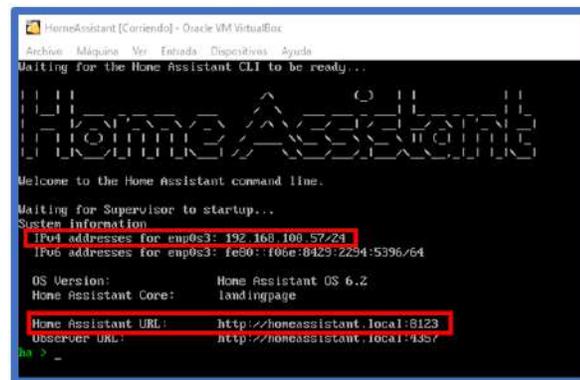


Figura B.20: Acceso a interfaz de *Home Assistant* .

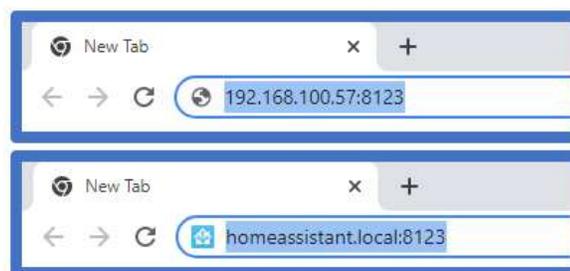


Figura B.21: Acceso a interfaz de *Home Assistant* (continuación).

7. Al acceder por primera vez, será necesario designar datos del usuario, como nombre, nombre de usuario, y contraseña.
 - (a) Después se sugiere permitir la detección de la ubicación del usuario. Además, seleccionar el tipo de moneda, y de unidades (métricas o imperiales). Dar click en Next.
 - (b) Si se desea, en la siguiente pantalla elegir la información que se desea compartir con Home Assistant, de lo contrario, solo dar click en continuar sin seleccionar ninguna casilla.
 - (c) Ver figura B.22.

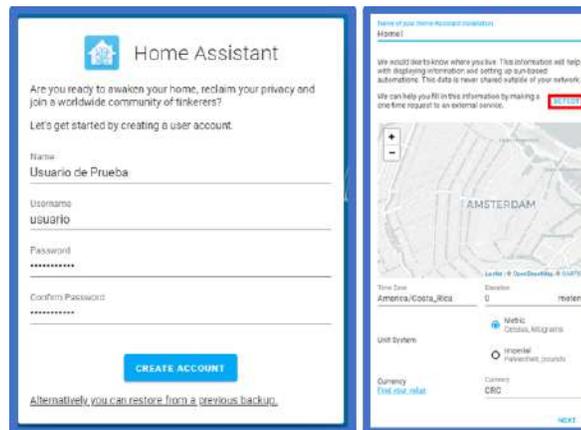


Figura B.22: Creación de usuario en *Home Assistant*.

B.2 Asignación de una IP estática en Home Assistant

Dentro de *Home Assistant*, es especialmente útil la asignación de una IP estática. Esto con el motivo de utilizar una única dirección IP en cada ocasión que se active la máquina virtual de *Home Assistant*.

1. Ingresar a *Home Assistant* en la pestaña de **Supervisor**, luego en **System (Sistema)**.
 - (a) En el cuadro de **Host**, elegir **Change (Cambiar)** al lado de *IP Address (Dirección IP)* (ver figura B.23).
2. Dentro de la configuración de dirección IP, elegir **IPv4**, y seleccionar *static* (estática), como se muestra en la figura B.24.
3. Seleccionar en guardar los cambios.

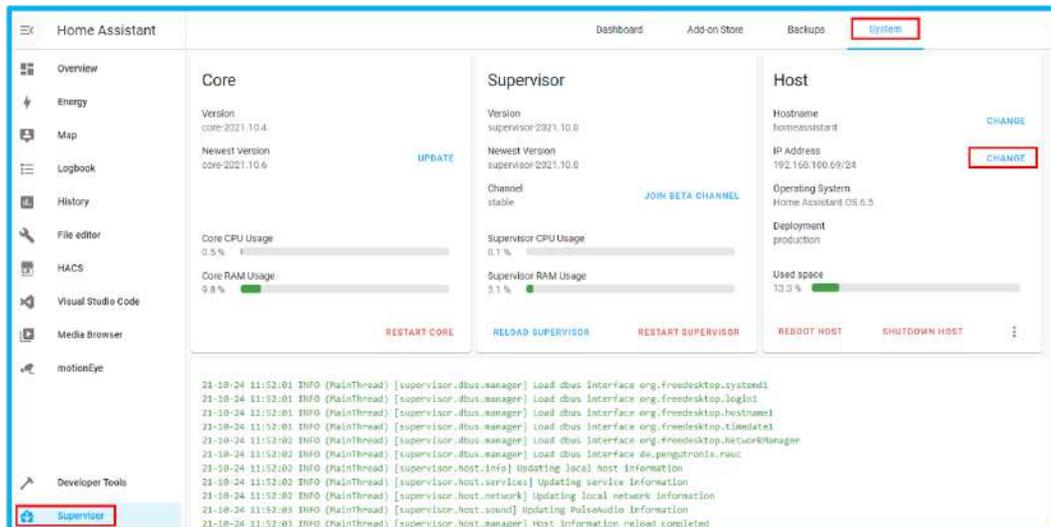


Figura B.23: Configuración de dirección IP estática

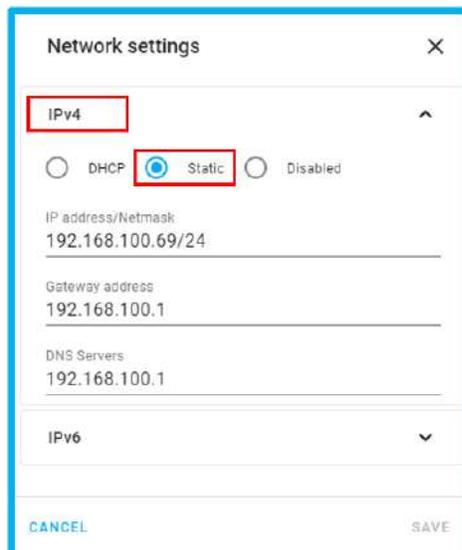


Figura B.24: Configuración de dirección IP estática (continuación)

B.3 Instalación de HACS

HACS es una integración de *Home Assistant* que permite agregar marcas, funcionalidades o dispositivos desarrollados por la comunidad de usuarios de la plataforma. De esta manera es posible agregar integraciones que no son incluidas de manera nativa en *Home Assistant*. Para agregar HACS se debe seguir los siguientes pasos:

1. Dirigirse a la pestaña de **Supervisor**, luego en **Add-On Store (Tienda de complementos)** y buscar la integración de *Visual Studio Code*, como en la figura B.25.
 - (a) Entrar en la integración y seleccionar **Instalar**. Una vez instalada la integración, habilitar la opción que se muestra en la figura B.26 y seleccionar empezar o *Start* para activar la integración en *Home Assistant*.

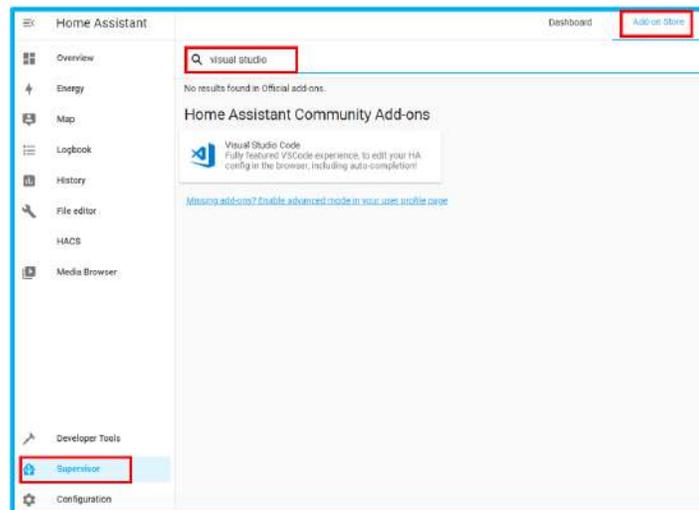


Figura B.25: Descarga de Visual Studio Code en *Home Assistant*

2. Ingresar en la nueva pestaña de *Visual Studio Code*, seleccionar opciones, terminal, nueva terminal (ver figura B.27).
 - (a) En la nueva terminal, para descargar en *Home Assistant* los archivos necesarios para HACS, escribir inicialmente el comando: `cd`
 - (b) Seguidamente el comando : `wget -O - https://get.hacs.xyz | bash -`
 - i. Ver figura B.28. O si desea, puede entrar al siguiente link, y copiar directamente la instrucción anterior bajo el título de *Home Assistant OS*: <https://hacs.xyz/docs/setup/download>
 - (c) Ahora se debe reiniciar *Home Assistant*, para lo cual escribir: `ha core restart`
3. Al finalizar el reinicio, dirigirse a **Configuration (Configuración)**, **Integrations (Integraciones)**, y se añade y busca HACS.

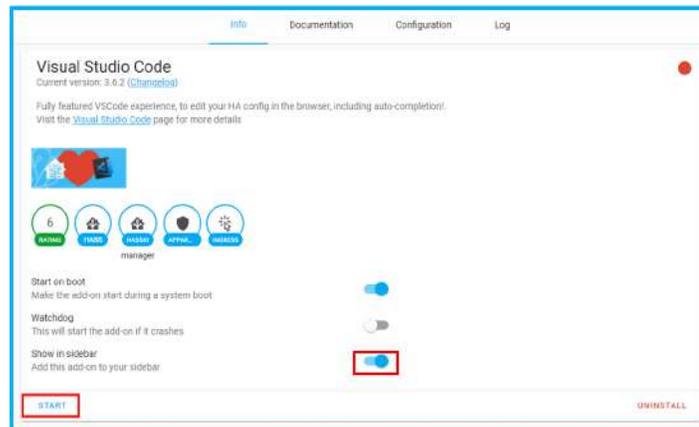


Figura B.26: Configuración de Visual Studio Code en *Home Assistant*

- (a) En caso de no aparecer la integración de HACS, refrescar el navegador.
4. Seguir las instrucciones mostradas en pantalla para completar la instalación.
 - (a) Nota: se debe contar con una cuenta de *Github* para completar la instalación.

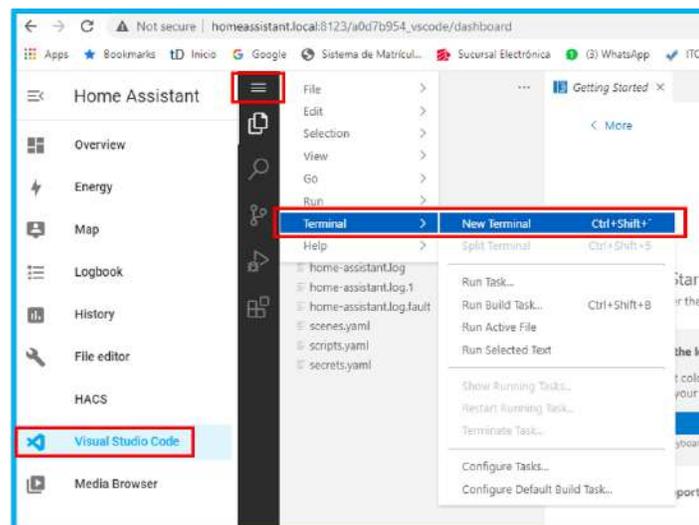


Figura B.27: Acceso a una terminal en Visual Studio Code en *Home Assistant*

B.4 Conexión de dispositivos USB en la máquina virtual

Para poder utilizar la funcionalidad de un dispositivo USB dentro de *Home Assistant*, es necesario configurar el dispositivo mediante la máquina virtual.

1. Conectar el dispositivo USB en los puertos de la computadora.



Figura B.28: Comando para descarga de acHACS

2. Entrar a la configuración de la máquina virtual (figura B.16).
3. Dirigirse a la pestaña de **USB**.
 - (a) Seleccionar **Habilitar controlador USB**, y elegir un controlador USB 2.0 o 3.0.
 - (b) Click en el ícono de **agregar un nuevo filtro USB**, y seleccionar el dispositivo que se desee agregar (figura B.29).

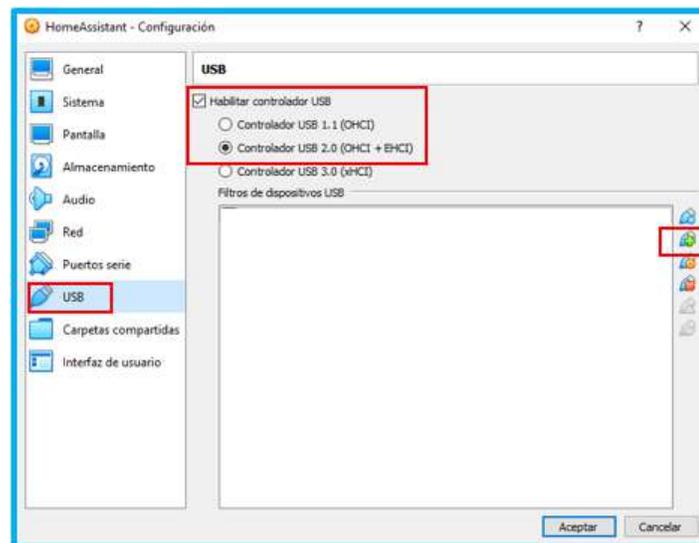


Figura B.29: Configuración USB en máquina virtual

4. El dispositivo USB seleccionado, aparecerá y deberá seleccionarlo como en la figura B.30.
5. Al finalizar este proceso, iniciar la máquina virtual.
 - (a) Dentro de la ventana de comando de *Home Assistant* de la máquina virtual, dirigirse a **Dispositivos, USB** y marcar el dispositivo USB deseado para ser reconocido por *Home Assistant*, como se muestra en la figura B.31.

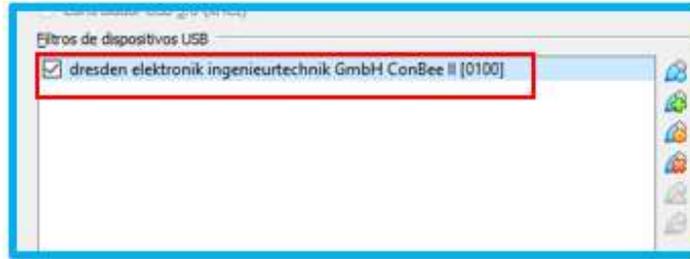


Figura B.30: Dispositivo USB agregado en configuración

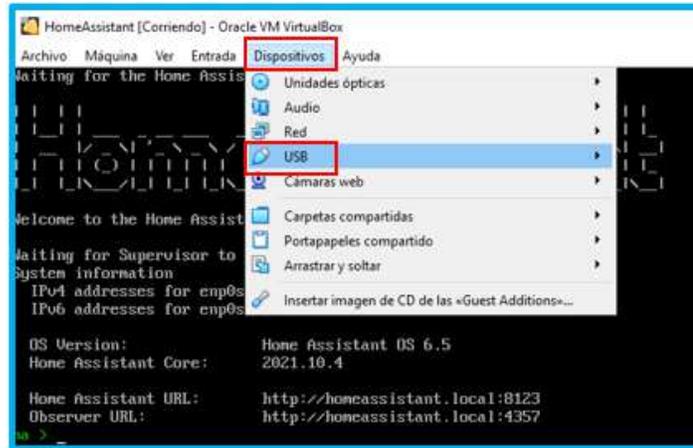


Figura B.31: Dispositivo USB agregado en máquina virtual

Nota sobre inconvenientes comunes: En ocasiones, el dispositivo USB ha mostrado un comportamiento en el cual se conecta y desconecta continuamente. Algunas de las soluciones que han logrado arreglar este inconveniente son:

- Repetir los pasos del proceso. Desconectar el USB desde la configuración de la máquina virtual, desconectar y re-conectar el USB, y volver a inicializar *Home Assistant*.
- Cambiar el dispositivo USB a otro puerto USB en su computador
- Inicializar *Home Assistant* sin el dispositivo conectado. Una vez que la plataforma se inicializa, conectar el dispositivo y reiniciar el servidor (se reinicia el servidor en **Configuración**, **Controles del servidor**, y reiniciar).

B.5 Conexión de luces

Se muestra a continuación el proceso para la integración a *Home Assistant* de dos tipos de bombillas inteligentes que se integran de manera nativa con la plataforma, de marcas: *Lifx* y *Yeelight*

B.5.1 Lifx Colour

1. Descargar en el celular la aplicación *LIFX*.
 - (a) Se debe conectar la bombilla en la habitación (asegurarse de apagar el switch de la luz en la habitación previamente a la manipulación de las bombillas).
 - (b) La aplicación le indicará las instrucciones a seguir para conectar la bombilla a la red *Wifi*. Debe tener disponible una red de 2.4 GHz.
2. Una vez conectada la bombilla con su aplicación, dentro de *Home Assistant* dirigirse a (**Configuración**), y allí, a **Integrations (Integraciones)**.
 - (a) Click en añadir integración, y escribir o buscar **LIFX**. *Home Assistant* detectará su bombilla automáticamente.
 - (b) Seleccionar la bombilla, y así se completa la integración.
3. Sugerencia 1: Cambiar el nombre y el ID de la entidad para identificar fácilmente al dispositivo.
 - (a) Dirigirse a **Herramientas de Desarrollador (Developer Tools)**, buscar el dispositivo deseado.
 - (b) Seleccionar el botón de **información** del dispositivo, y en la pantalla emergente, seleccionar configuración, como se muestra en las figuras B.32 y B.33.
 - (c) Se puede elegir el nombre de dispositivo que se desee. Para el ID de la entidad (*entity ID*), **únicamente** cambiar el nombre que va después del punto (“.”), como se indica en la figura B.34.
 - i. El *entity ID* siempre debe comenzar con el tipo de dispositivo que asigna *Home Assistant*, por ejemplo: “light.nombre”, “binary_sensor.nombre”, “switch.nombre” (en donde nombre representa el nombre que el usuario puede modificar a gusto).



Figura B.32: Cambio de nombre del dispositivo

4. Sugerencia 2: Agregar una tarjeta o visualización adecuada para el dispositivo.

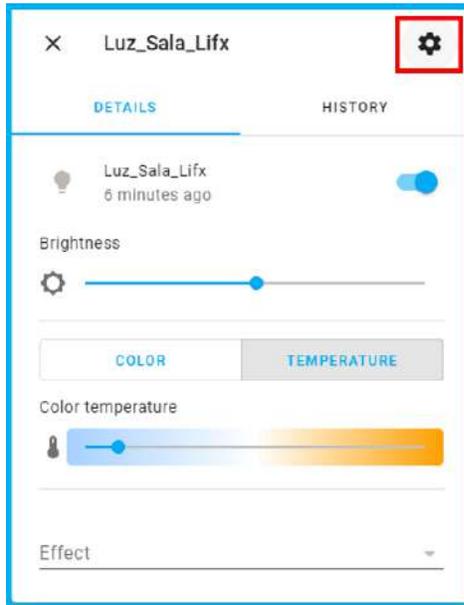


Figura B.33: Cambio de nombre de un dispositivo (continuación)

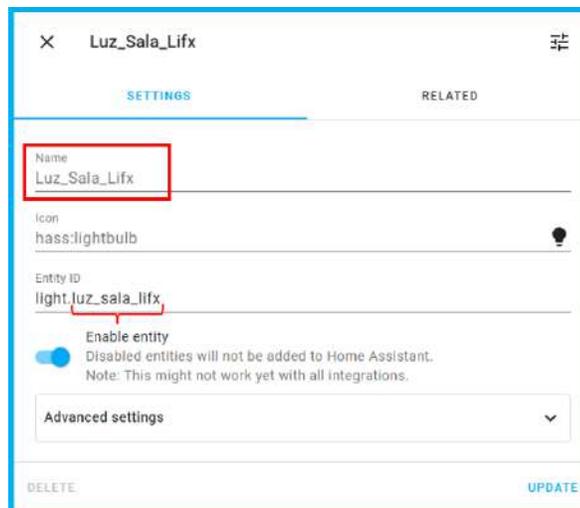


Figura B.34: Cambio de nombre de un dispositivo (continuación)

- (a) Dirigirse a la pantalla de **Resumen** (*Overview*), seleccionar en la esquina superior derecha los 3 puntos, y finalmente editar, como se muestra en la figura B.35.
- (b) Dar click sobre **Añadir tarjeta** (**Add Card**).
- (c) Seleccionar búsqueda **Por entidad** (**By entity**).
- (d) Buscar el dispositivo deseado y la presentación que guste para mostrar en el *dashboard*. Click en continuar, y añadir.

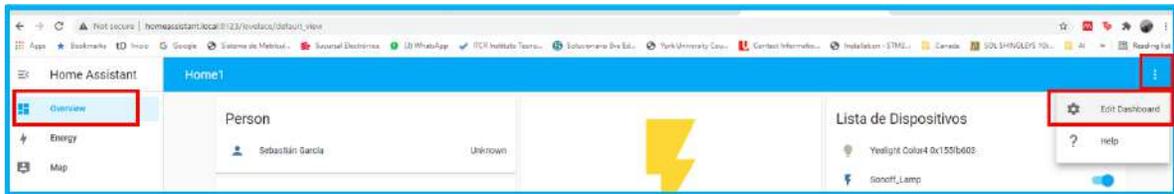


Figura B.35: Agregar tarjeta o visualización de un dispositivo

B.5.2 Yeelight 1S

1. Descargar en el celular la aplicación *Yeelight*.
 - (a) Se debe conectar la bombilla en la habitación (asegurarse de apagar el switch de la luz en la habitación previamente a la manipulación de las bombillas).
 - (b) La aplicación le indicará las instrucciones a seguir para conectar la bombilla a la red *Wifi*. Cuando la opción aparezca, aceptar la opción de **Modo LAN** en la configuración. Debe tener disponible una red de 2.4 GHz.
2. Una vez conectada la bombilla con su aplicación, dentro de *Home Assistant* dirigirse a (**Configuración**), y allí, a **Integrations** (**Integraciones**).
 - (a) Click en añadir integración, y escribir o buscar **Yeelight**. *Home Assistant* detectará su bombilla automáticamente.
 - (b) Seleccionar la bombilla, y así se completa la integración.
3. Sugerencia 1: Cambiar el nombre y el ID de la entidad para identificar fácilmente al dispositivo.
 - (a) Dirigirse a **Herramientas de Desarrollador** (**Developer Tools**), buscar el dispositivo deseado.
 - (b) Seleccionar el botón de **información** del dispositivo, y en la pantalla emergente, seleccionar configuración, como se muestra en las figuras B.32 y B.33.
 - (c) Se puede elegir el nombre de dispositivo que se desee. Para el ID de la entidad (*entity ID*), **únicamente** cambiar el nombre que va después del punto (“.”), como se indica en la figura B.34.

- i. El *entity ID* siempre debe comenzar con el tipo de dispositivo que asigna *Home Assistant*, por ejemplo: “light.nombre”, “binary_sensor.nombre”, “switch.nombre” (en donde nombre representa el nombre que el usuario puede modificar a gusto).
4. Sugerencia 2: Agregar una tarjeta o visualización adecuada para el dispositivo.
 - (a) Dirigirse a la pantalla de **Resumen** (*Overview*), seleccionar en la esquina superior derecha los 3 puntos, y finalmente editar, como se muestra en la figura B.35.
 - (b) Dar click sobre **Añadir tarjeta** (**Add Card**).
 - (c) Seleccionar búsqueda **Por entidad** (**By entity**).
 - (d) Buscar el dispositivo deseado y la presentación que guste para mostrar en el *dashboard*. Click en continuar, y añadir.

B.6 Conexión de conectores (Smart Plugs)

Se muestra a continuación el proceso para la integración a *Home Assistant* de dos tipos de conectores inteligentes, o *smart plugs*, que se integran por medio de HACS a la plataforma, de marcas: *Tapo* y *Sonoff*.

B.6.1 Tapo P100

Antes de empezar tome en cuenta que estas instrucciones muestran el procedimiento para la conexión de un **smart plug** *Tapo*. Si desea la conexión de una cámara *Tapo*, dirigirse a la sección de Conexión de Cámaras.

1. Descargar la aplicación de Tapo en el celular.
 - (a) Seguir las instrucciones en la aplicación para agregar el dispositivo.
 - (b) Una vez agregado, buscar la dirección IP del dispositivo dentro de la información del dispositivo en la aplicación (se necesitará para *Home Assistant*).
 - i. Ingresar al dispositivo, luego a su configuración (*settings*), y por último entrar a información del dispositivo (*device settings*).
2. Ahora, en *Home Assistant*, dirigirse a la pestaña de HACS, dar click a la opción de **Integraciones/Integrations**, y luego **Custom repositories/Repositorios personalizados** como se muestra en la figura B.36.
 - (a) Agregar, como se ve en la figura B.37, el siguiente link: `https://github.com/petretiandrea/home-assistant-tapo-p100.git`
 - (b) Una vez añadido el repositorio anterior, dirigirse al botón de añadir repositorio mostrado en la figura B.38, y buscar *Tapo Controller* (el repositorio que se acaba de añadir).

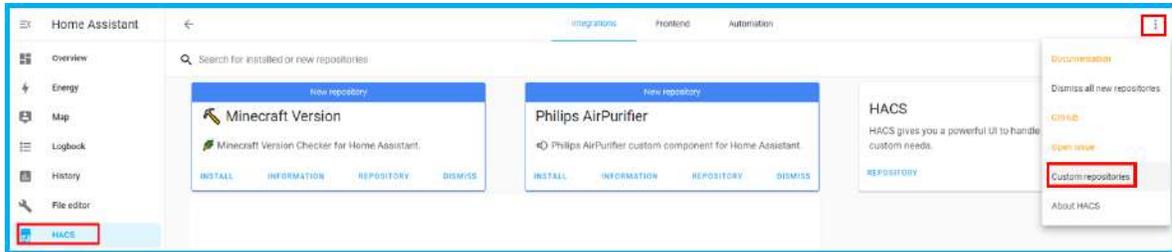


Figura B.36: Añadir repositorio personalizado en HACS

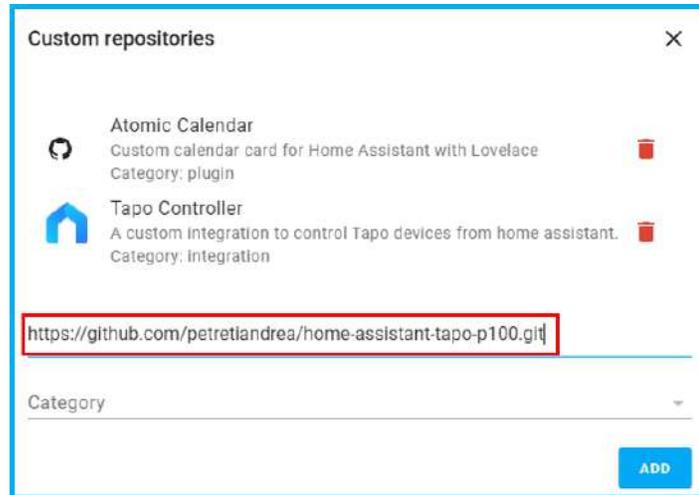


Figura B.37: Añadir repositorio personalizado en HACS (continuación)

3. Reiniciar el servidor de *Home Assistant* para poder actualizar los cambios hechos.
 - (a) Dirigirse a **Configuración, Controles del servidor/Sever Controls** y dar click en reiniciar.
4. Una vez reiniciado por completo el servidor, dirigirse a **Configuración, integraciones**, y añadir nueva integración. Buscar *Tapo* para añadir el smart plug. Aparecerá una ventana de configuración como en la figura B.39. Si la integración no se muestra en las opciones, intente refrescar el navegador.
 - (a) Host: rellenar con el número IP del enchufe.
 - (b) Username: rellenar con el correo que se utilizó en la aplicación móvil de *Tapo*.
 - (c) Password: contraseña que se usó en la aplicación móvil de *Tapo*.
5. Sugerencia 1: Cambiar el nombre y el ID de la entidad para identificar fácilmente al dispositivo.
 - (a) Dirigirse a **Herramientas de Desarrollador (Developer Tools)**, buscar el dispositivo deseado.

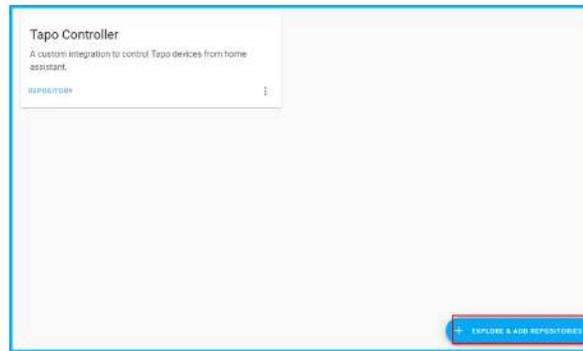


Figura B.38: Añadir repositorio personalizado en HACS (continuación)

Figura B.39: Agregar integración de Tapo Smart Plug

- (b) Seleccionar el botón de **información** del dispositivo, y en la pantalla emergente, seleccionar configuración, como se muestra en las figuras B.32 y B.33.
 - (c) Se puede elegir el nombre de dispositivo que se desee. Para el ID de la entidad (*entity ID*), **únicamente** cambiar el nombre que va después del punto (“.”), como se indica en la figura B.34.
 - i. El *entity ID* siempre debe comenzar con el tipo de dispositivo que asigna *Home Assistant*, por ejemplo: “light.nombre”, “binary_sensor.nombre”, “switch.nombre” (en donde nombre representa el nombre que el usuario puede modificar a gusto).
6. Sugerencia 2: Agregar una tarjeta o visualización adecuada para el dispositivo.
- (a) Dirigirse a la pantalla de **Resumen** (*Overview*), seleccionar en la esquina superior derecha los 3 puntos, y finalmente editar, como se muestra en la figura B.35.

- (b) Dar click sobre **Añadir tarjeta (Add Card)**.
- (c) Seleccionar búsqueda **Por entidad (By entity)**.
- (d) Buscar el dispositivo deseado y la presentación que guste para mostrar en el *dashboard*. Click en continuar, y añadir.

B.6.2 Sonoff S26

1. Descargar en el celular la aplicación *eWeLink* para utilizar dispositivos *Sonoff*.
 - (a) Seguir en la aplicación las instrucciones para conectar el dispositivo a la red *Wifi*.
2. Descargar de HACS la integración de *Sonoff*.
 - (a) Ingresar en la pestaña de HACS, dirigirse a integraciones, añadir repositorio, y buscar “Sonoff LAN”, como se observa en la figura B.40. Instalar el repositorio.
3. Reiniciar el servidor de *Home Assistant* para poder actualizar los cambios hechos.
 - (a) Dirigirse a **Configuración, Controles del servidor/Server Controls** y dar click en reiniciar.

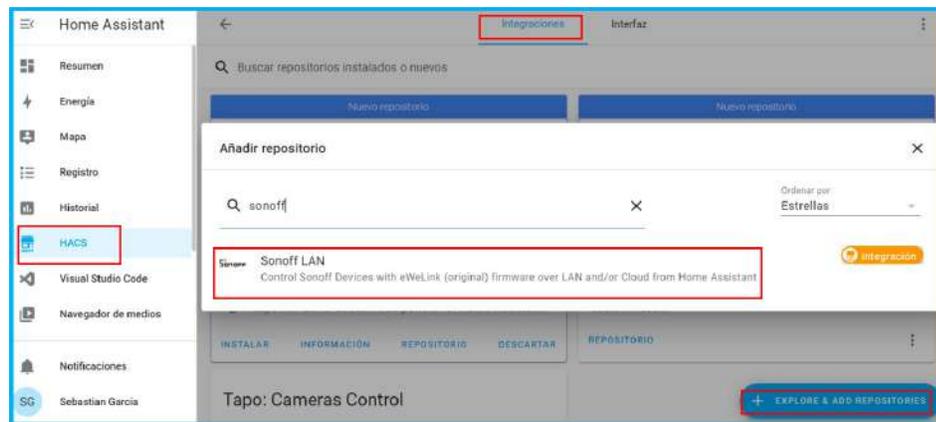


Figura B.40: Integración de Sonoff LAN

4. Será necesario añadir algunas líneas en un archivo de configuración de *Home Assistant*, para lo cual se debe descargar un *Add-on*.
 - (a) Dirigirse a la pestaña del supervisor, luego en *Add-on Store*, y buscar “File Editor”, como se muestra en la figura B.41. Instalar la integración, y asegurarse de seleccionar el botón de mostrar icono en la barra lateral (como se muestra en el ejemplo de la instalación de HACS de la figura B.26).

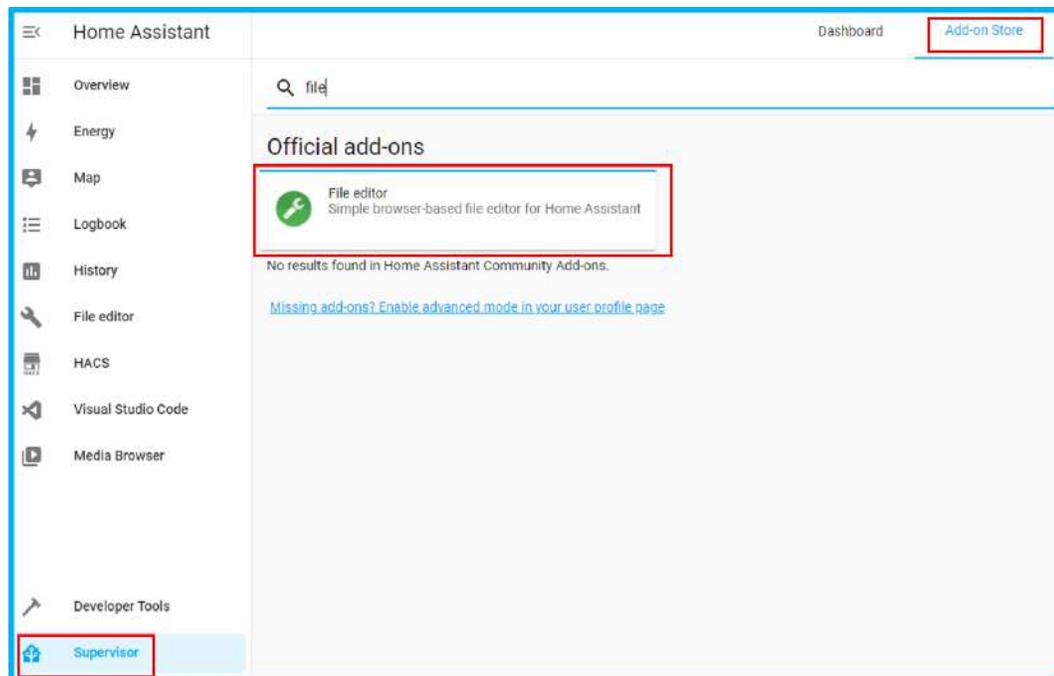


Figura B.41: Integración de File Editor

5. Entrar en la nueva pestaña del *File Editor*, seleccionar la carpeta **configuration.yaml**, y completar con sus datos como se observa en la figura B.42 o se exponen a continuación .
 - (a) Dentro de esta carpeta, añadir las líneas tal y como se muestran en la figura B.42.
 - (b) “sonoff:”, “username: ”, y “password: ”.
 - i. Donde username será el correo que se usó para la creación de su cuenta en el app *eWeLink*.
 - ii. Password será la contraseña que utilizó para *eWeLink*.
 - (c) Dar click en el icono de guardar los cambios del código que acaba de escribir. Asegúrese de utilizar la indentación (o espaciado) como se muestra en la figura B.42.
6. Reiniciar el servidor de *Home Assistant* para poder actualizar los cambios hechos.
 - (a) Dirigirse a **Configuración, Controles del servidor/Server Controls** y dar click en reiniciar.
7. Una vez reiniciado el servidor, se puede dirigir a la pestaña de **Herramientas de Desarrollador (Developer Tools)**, y encontrará el dispositivo nuevo *Sonoff*, como se muestra en la figura B.43.
8. Sugerencia 1: Cambiar el nombre y el ID de la entidad para identificar fácilmente al dispositivo.

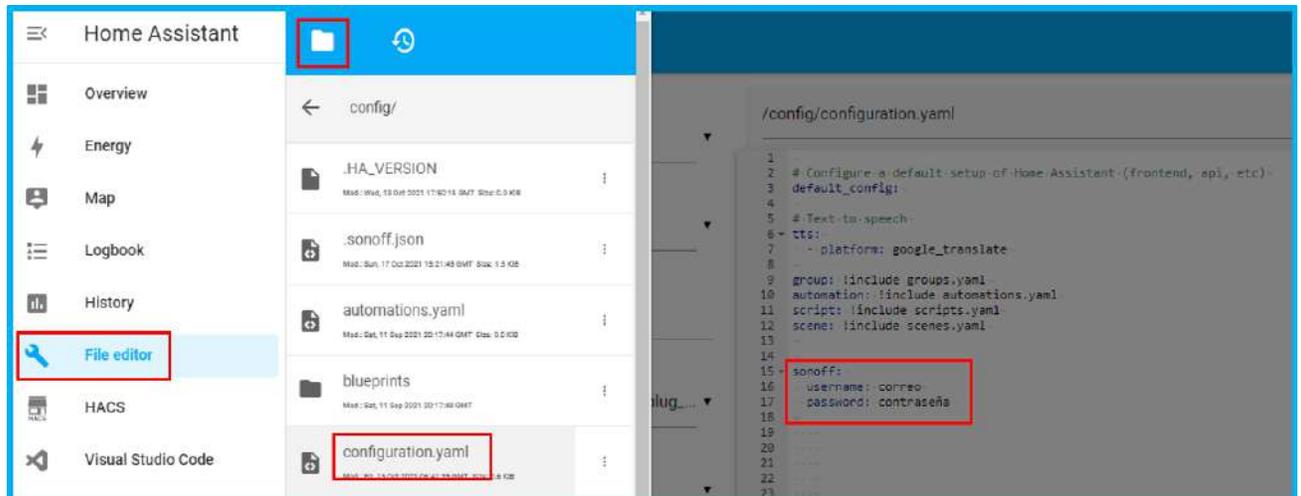


Figura B.42: Modificación de archivo de configuración



Figura B.43: Dispositivo Sonoff S26 añadido

- (a) Dirigirse a **Herramientas de Desarrollador (Developer Tools)**, buscar el dispositivo deseado.
 - (b) Seleccionar el botón de **información** del dispositivo, y en la pantalla emergente, seleccionar configuración, como se muestra en las figuras B.32 y B.33.
 - (c) Se puede elegir el nombre de dispositivo que se desee. Para el ID de la entidad (*entity ID*), **únicamente** cambiar el nombre que va después del punto (“.”), como se indica en la figura B.34.
 - i. El *entity ID* siempre debe comenzar con el tipo de dispositivo que asigna *Home Assistant*, por ejemplo: “light.nombre”, “binary_sensor.nombre”, “switch.nombre” (en donde nombre representa el nombre que el usuario puede modificar a gusto).
9. Sugerencia 2: Agregar una tarjeta o visualización adecuada para el dispositivo.
- (a) Dirigirse a la pantalla de **Resumen (Overview)**, seleccionar en la esquina superior derecha los 3 puntos, y finalmente editar, como se muestra en la figura B.35.
 - (b) Dar click sobre **Añadir tarjeta (Add Card)**.
 - (c) Seleccionar búsqueda **Por entidad (By entity)**.

- (d) Buscar el dispositivo deseado y la presentación que guste para mostrar en el *dashboard*. Click en continuar, y añadir.

B.7 Conexión de sensores

Los sensores en *Home Assistant* usualmente funcionan con los protocolos *Zigbee*, o *Z-Wave*. Por esta razón, se requiere de un puente, o hub, para poder controlarlos e integrarlos (a diferencia de otros dispositivos como luces o enchufes que son posibles de manipular directamente de sus aplicaciones o desde *Home Assistant*). Se utilizará para esto un dispositivo USB *Conbee II*, el cual funciona como coordinador de dispositivos, que permite controlar distintos sensores sin ser dependiente de una marca en específico. Así, se podrán integrar sensores tanto de la marca *Aqara* como *Sonoff*, sin requerir el uso de *hubs* propietarios o incluso de sus respectivas aplicaciones móviles.

1. Asegurarse de contar con el dispositivo USB *Conbee II* conectado y reconocido por su máquina virtual o entorno de *Home Assistant*, como se indica en la sección B.4.
2. Asegurarse de revisar en las instrucciones de cada sensor, la forma adecuada de reiniciar o poner el dispositivo en modo de emparejamiento (*pairing mode*).
 - (a) En la mayoría de los casos, es necesario presionar el botón de reinicio durante unos 3 a 5 segundos.
3. Dirigirse a la pestaña de configuración, y luego a integraciones.
4. Si *Home Assistant* no muestra que detecta una integración para *Zigbee Home Automation (ZHA)*, entonces deberá buscarla con el botón de añadir integración. (Ver figura B.44).

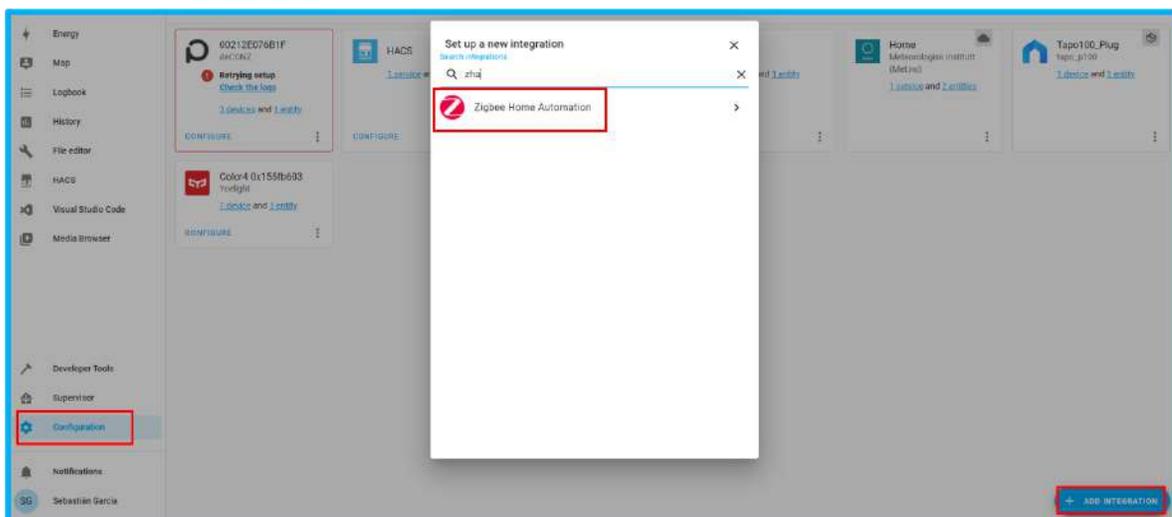


Figura B.44: Integración de Zigbee Home Automation para sensores

5. Al seleccionar la integración, esta lo guiará para seleccionar el dispositivo *Conbee II* conectado.
 - (a) Seguir las instrucciones mostradas por *Home Assistant*, y este lo pondrá en modo de detección de dispositivos.
 - (b) Si no le muestra instrucciones, dirigirse al icono de la integración de *ZHA*, dar click en configuración, y allí dentro dar click en añadir nuevo dispositivo.
 - (c) Cuando se esté en modo de detección, deberá reiniciar o poner en *pairing mode* cada uno de los sensores, **uno a la vez**. *Home Assistant* indicará cuando cada dispositivo haya sido detectado correctamente.
6. Sugerencia 1: Cambiar el nombre y el ID de la entidad para identificar fácilmente al dispositivo.
 - (a) Dirigirse a **Herramientas de Desarrollador** (*Developer Tools*), buscar el dispositivo deseado.
 - (b) Seleccionar el botón de **información** del dispositivo, y en la pantalla emergente, seleccionar configuración, como se muestra en las figuras B.32 y B.33.
 - (c) Se puede elegir el nombre de dispositivo que se desee. Para el ID de la entidad (*entity ID*), **únicamente** cambiar el nombre que va después del punto (“.”), como se indica en la figura B.34.
 - i. El *entity ID* siempre debe comenzar con el tipo de dispositivo que asigna *Home Assistant*, por ejemplo: “light.nombre”, “binary_sensor.nombre”, “switch.nombre” (en donde nombre representa el nombre que el usuario puede modificar a gusto).
7. Sugerencia 2: Agregar una tarjeta o visualización adecuada para el dispositivo.
 - (a) Dirigirse a la pantalla de **Resumen** (*Overview*), seleccionar en la esquina superior derecha los 3 puntos, y finalmente editar, como se muestra en la figura B.35.
 - (b) Dar click sobre **Añadir tarjeta** (**Add Card**).
 - (c) Seleccionar búsqueda **Por entidad** (**By entity**).
 - (d) Buscar el dispositivo deseado y la presentación que guste para mostrar en el *dashboard*. Click en continuar, y añadir.

B.8 Conexión de cámaras

Se muestra a continuación el proceso para la integración a *Home Assistant* de cámaras inteligentes. Se utilizarán las marcas *Tapo* y *Foscam*, y además se requiere el uso de integraciones de HACS.

B.8.1 Tapo

1. Descargar en el celular la aplicación *Tapo*.
 - (a) Seguir en la aplicación las instrucciones para conectar la cámara a la red *Wifi*.
 - i. La aplicación del pedirá crear un nombre de usuario. Debe nombrar ese usuario y anotarlo, ya que será necesario para su configuración en *Home Assistant*.
 - A. Si la aplicación no le pide la creación de dicho nombre de usuario, una vez que su cámara está configurada en el app, dirigirse a su cámara, luego en *settings* o configuración o ajustes, seguidamente ir a ajustes avanzados, y finalmente digitar su nombre de usuario y contraseña deseados.
 - ii. Anotar la dirección IP de la cámara ya que será necesario utilizarla en *Home Assistant*. Para ello acceder en la pantalla de visualización de la cámara, dirigirse a configuración, información de la cámara, dirección IP.
2. Descargar de HACS la integración de *Tapo: Cameras Control*. Ver figura B.45.
 - (a) Una vez descargada la integración se debe reiniciar el servidor de *Home Assistant* (ir a configuración, controles de servidor, reiniciar).

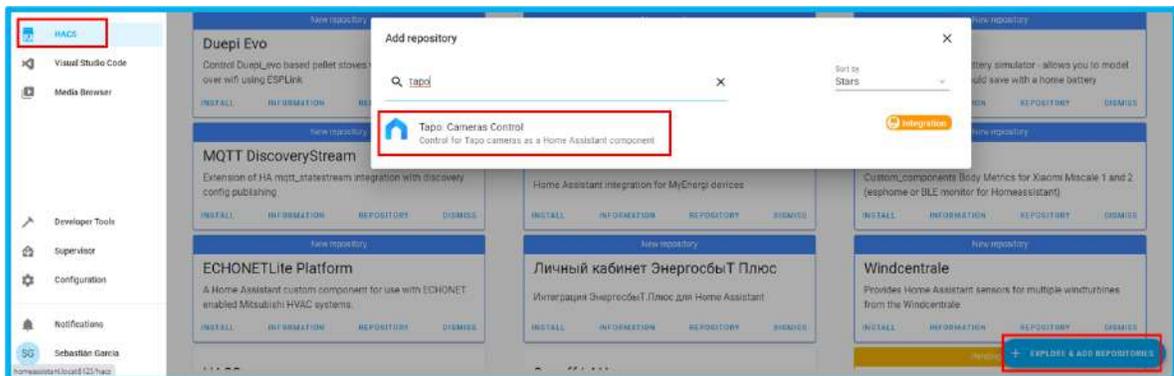


Figura B.45: Descarga en HACS de integración de *Tapo: Cameras Control*

3. Una vez completado el reinicio, dirigirse a la ventana de integraciones en configuración, y buscar *Tapo: Cameras Control*. Si la integración no aparece, puede volver a cargar la ventana en el navegador de *Home Assistant*.
 - (a) Al seleccionar la integración, se le solicitará tanto la dirección IP como el nombre de usuario y contraseña que fueron anotados previamente, como se observa en la figura B.46.
4. Una vez integrada la cámara, dirigirse al dashboard (Resumen, u *Overview*) para agregar la visualización de la cámara.

(a) Solicitud de dirección IP integración *Tapo*

(b) Solicitud de usuario y contraseña

Figura B.46: Integración de cámara *Tapo*. Fuente: elaboración propia

- (a) Entrar en modo de edición y seleccionar en añadir nueva tarjeta.
- (b) Seleccionar la búsqueda de dispositivos **By Entity** o **Por dispositivo**, y seleccionar la nueva cámara *Tapo*, como se observa en la figura B.47.

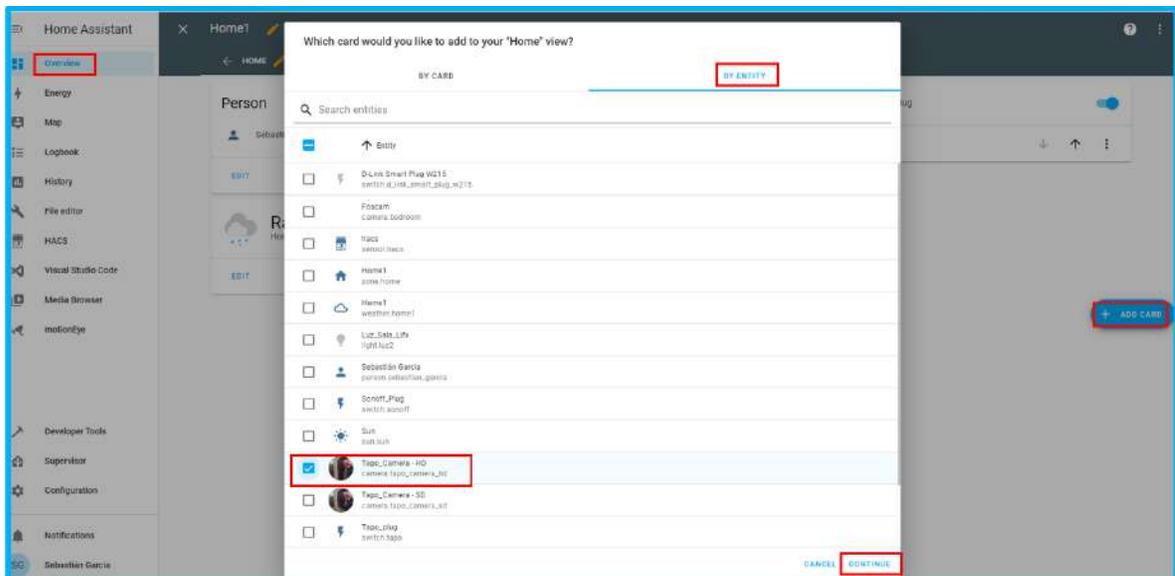


Figura B.47: Configuración para visualización de cámara *Tapo*

- (c) Después de seleccionar en **continuar**, seleccionarla opción de añadir *lovelace* (dashboard), como en la figura B.48.
5. Una vez añadida la tarjeta para la visualización, seleccionar en ella **editar**.
 - (a) Dentro de la configuración de la tarjeta, cambiar la opción de *Camera view* a *live* para poder recibir el video en “tiempo real”, como se observa en la figura B.49.

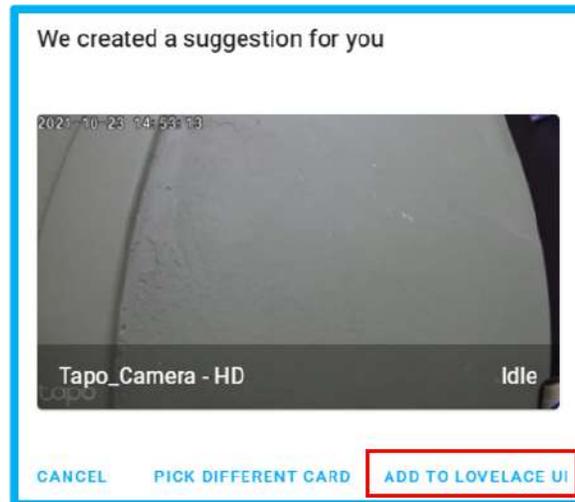
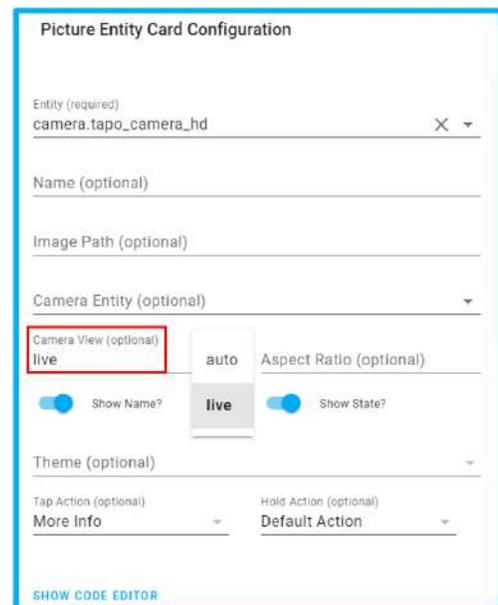


Figura B.48: Configuración para visualización de cámara *Tapo* (continuación)



(a) Edición de tarjeta de visualización de cámara



(b) Configuración de recepción de video

Figura B.49: Configuración para visualización de cámara *Tapo* (continuación). Fuente: elaboración propia

B.8.2 Foscam

1. Descargar en el celular la aplicación *Foscam*.
 - (a) Seguir en la aplicación las instrucciones para conectar la cámara a la red *Wifi*.
 - i. La aplicación del pedirá crear un nombre de usuario. Debe nombrar ese usuario y anotarlo, ya que será necesario para su configuración en *Home Assistant*.
 - ii. Anotar la dirección IP de la cámara ya que será necesario utilizarla en *Home Assistant*. Para ello acceder en la pantalla de visualización de la cámara, dirigirse a configuración, información de la cámara, dirección IP.
2. En *Home Assistant*, se cuenta con integración directa para añadir cámaras *Foscam*.
 - (a) Dirigirse a configuración, integraciones, añadir integración, y buscar *Foscam* (figura B.50a).
 - (b) Seguidamente, se solicita ingresar la dirección IP, usuario y contraseña configurados y anotados previamente (B.50b).



(a) Búsqueda de integración

(b) Configuración de cámara

Figura B.50: Integración de cámara *Foscam*. Fuente: elaboración propia

3. Una vez añadida la cámara, dirigirse al dashboard para añadir la visualización de la cámara.
 - (a) Entrar en modo de edición, y añadir nueva tarjeta.

- (b) Seleccionar elegir por entidad, y seleccionar la cámara foscam recién añadida (figura B.51).

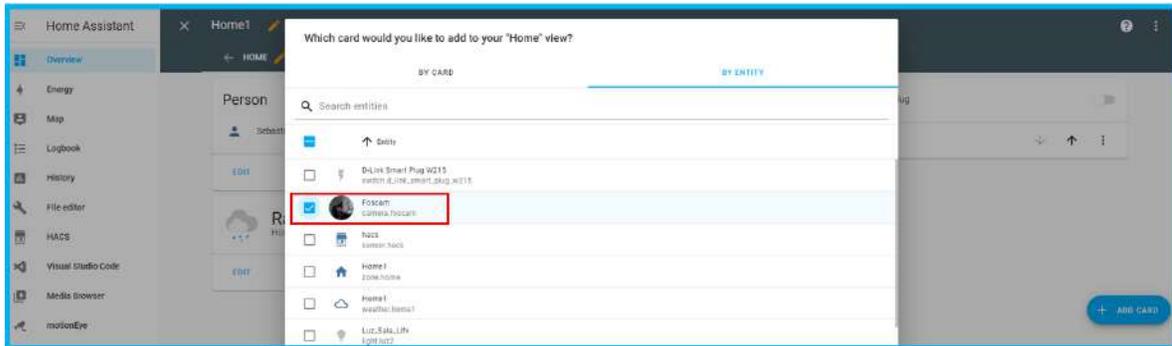


Figura B.51: Configuración para visualización de cámara *Foscam*

4. Finalmente seleccionar en continuar y añadir al dashboard (figura B.52)

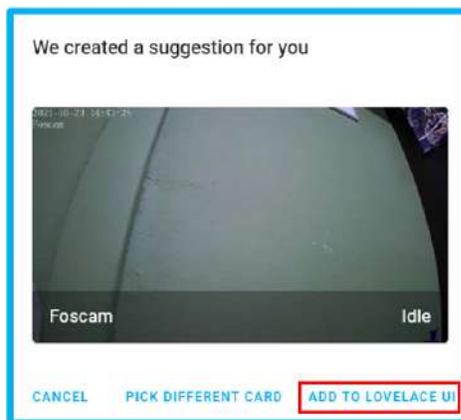


Figura B.52: Configuración para visualización de cámara *Foscam* (continuación)

B.9 Parlantes

Para añadir un parlante inteligente, es necesaria la utilización de la herramienta de la nube de *Home Assistant*, llamada *Nabu Casa*. Esta extensión de la plataforma permite conectar parlantes con asistentes inteligentes como Alexa de *Amazon* o *Google Assistant*. *Nabu Casa* es un servicio pagado (5 dolares mensuales), sin embargo, para nuevos usuarios ofrecen el primer mes de servicio gratuito. Cabe destacar que el parlante inteligente es capaz de comunicarse con *Home Assistant* para controlar sus dispositivos, pero *Home Assistant* no es capaz de controlar los servicios del parlante.

B.9.1 Amazon Echo Dot

1. Descargar la aplicación de *Amazon Alexa*.

- (a) Seguir las instrucciones para integrar a Alexa a la red *Wifi*.
2. En *Home Assistant*, dirigirse a configuración, y luego a **Home Assistant Cloud**.
 - (a) Aceptar la prueba de 1 mes, y proceder a crear una cuenta con su correo electrónico.
3. Una vez dentro del servicio de nube, dirigirse a la sección de Alexa (figura B.53) y habilitar el servicio.



Figura B.53: Activación de servicio de *Amazon* Alexa

4. Seleccionar **Activar la skill *Home Assistant para Alexa***.
 - (a) Seguir las instrucciones mostradas en pantalla para iniciar sesión con su cuenta de *Amazon*.
5. Será dirigido a una página de *Amazon* (figura B.54) en donde deberá buscar y activar el *skill* de *Home Assistant* (figura B.55).

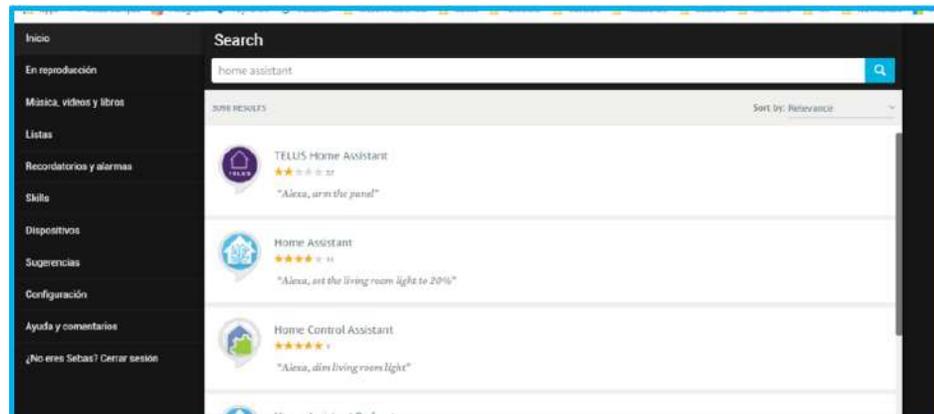


Figura B.54: Activación de skill de *Home Assistant* para *Amazon* Alexa

6. Será presentado una opción para detectar dispositivos de *Home Assistant* automáticamente, deberá aceptarla (B.56).

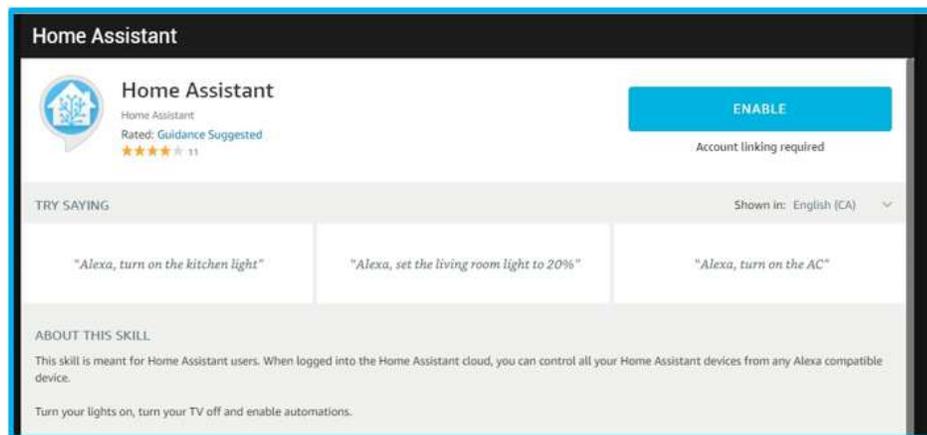


Figura B.55: Activación de skill de *Home Assistant* para *Amazon Alexa* (continuación)

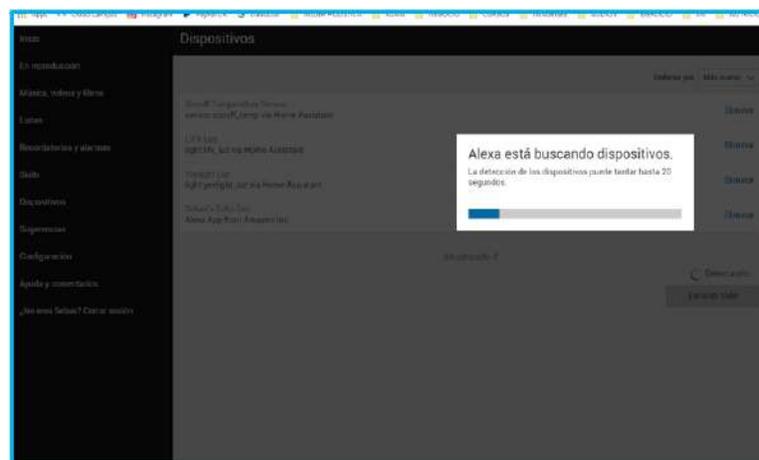


Figura B.56: Detección de dispositivos para Alexa

7. Al terminar la sincronización de dispositivos, volver a *Home Assistant*, a la sección de Alexa. Seleccionar **Administrar entidades**.
8. Dentro del administrador, habilitar o deshabilitar aquellos dispositivos que desee sean controlados y leídos por Alexa (figura B.57).

B.10 Obtención del Long Live Token

Para poder acceder remotamente a *Home Assistant*, por ejemplo utilizando una aplicación externa y comunicándose con el REST API, es necesario obtener un código llamado *Long Live Token*. Este código funciona como una llave única de su instancia (o sesión) específica de *Home Assistant*. Al obtenerlo, solo será mostrado **una única vez**, por lo que debe anotarlo y guardarlo en un lugar seguro y de fácil acceso. Para obtener este código :

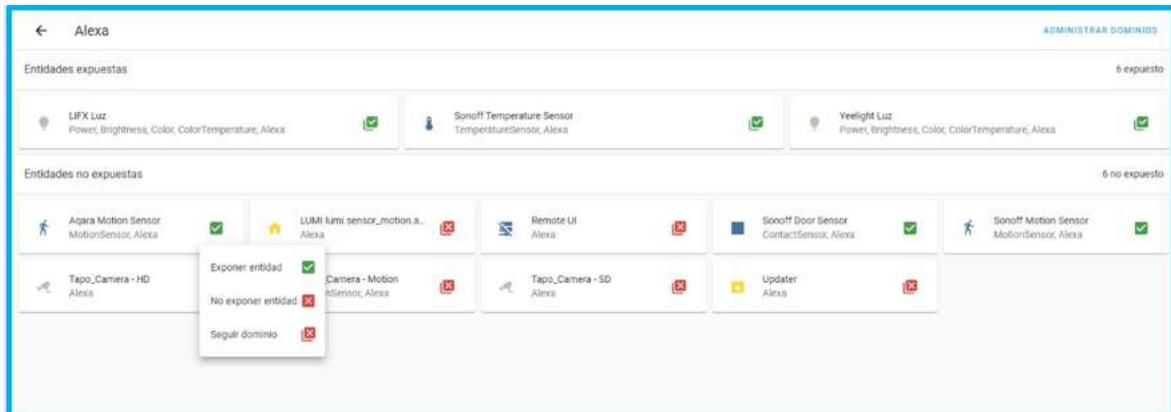


Figura B.57: Selección de dispositivos para Alexa

1. En *Home Assistant*, dirigirse a la pestaña de su perfil (la última pestaña).
2. Dirigirse al área de **Long-Lived Access Tokens** (ver figura B.58)
3. Seleccionar crear token, y copiar el código una vez que se muestre en pantalla.
 - (a) Recuerde que es **importante** anotar y guardar el token ya que solamente podrá verlo una única vez en *Home Assistant*.

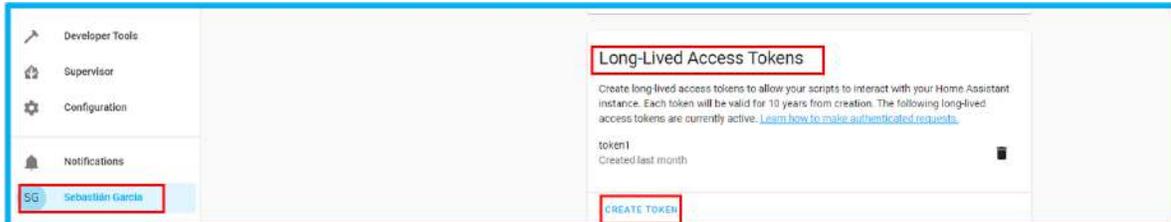


Figura B.58: Obtención del *Long Live Token*

B.11 Activación de Virtualización en Computador

En caso de que al tratar de crear la máquina virtual para *Home Assistant* no se pueda seleccionar la versión de Linux de 64 bits (como se muestra en la figura B.9), es necesario realizar una configuración a nivel del computador. Para ello, se debe:

1. Reiniciar el computador. Es necesario acceder al BIOS.
 - (a) Cuando se está inicializando el computador, se debe presionar la tecla que el computador le indique para entrar en el menú del BIOS.
 - i. Algunos computadores solicitan la tecla F10, o F12, o ESC.

2. Dentro del BIOS, buscar la opción de **System Configuration**, y activar la opción de **Virtualization Technology**.
 - (a) Guardar cambios, salir del BIOS, y terminar de encender el computador.
3. Una vez que se enciende el computador nuevamente, la opción de utilizar una versión de 64 bits ya estará habilitada y se podrá continuar con el proceso de creación de la máquina virtual.

Para tener más detalle acerca del procedimiento descrito, se recomienda seguir los mismos pasos expresados previamente en el siguiente video <https://www.youtube.com/watch?v=ojkF4iY1Yqc>.

APPENDIX C

ENCUESTA PARA PRUEBAS DE VALIDACIÓN

Se presenta a continuación la encuesta realizada para la validación de la experiencia de usuario con respecto a la usabilidad de la plataforma de *Home Assistant*, y la aplicación móvil diseñada en el presente proyecto. La encuesta consiste de 9 preguntas que deberán ser respondidas por los usuarios para evaluar sus experiencias individuales tras la ejecución de las pruebas planteadas, en donde se trabaja con la conexión de múltiples dispositivos, interacción con *Home Assistant*, y utilización de las funcionalidades de la aplicación móvil. La encuesta fue realizada digitalmente, por medio de la herramienta de *Google Forms*.

Validación del proceso de conexión y utilización de distintos dispositivos inteligentes en Home Assistant y Robobo App

Esta encuesta tiene como principal motivo evaluar la opinión del usuario en cuanto a su experiencia individual tras haber realizado la conexión de luces, conectores, sensores, y cámaras en la plataforma del Internet de las cosas (IoT), y posteriormente comprobar su funcionamiento por medio del app diseñada para el presente proyecto, llamada "Robobo Home Assistant App".

***Obligatorio**

1. Nombre de participante: *

2. Para la Prueba #1: conexión de un dispositivo de luz inteligente. Con respecto a la instalación del dispositivo: *

Marca solo un óvalo.

- 1. Muy sencillo y rápido, con instrucciones completamente claras.
- 2. Algo sencillo, y no tan rápido, con instrucciones adecuadas.
- 3. Neutro, no me parece ni sencillo ni complejo el proceso, tiempo o instrucciones.
- 4. Algo complicado, y un poco extenso, pero las instrucciones están claras.
- 5. Complicado, extenso, y las instrucciones podrían mejorar.

3. Posterior a la instalación de la Prueba #1, en cuanto al uso, cambios de estado, y lectura de los dispositivos en la aplicación Robobo Home Assistant App: *

Marca solo un óvalo.

- 1. Uso, lectura y visualización muy sencilla e intuitiva.
- 2. Uso, lectura y visualización algo sencilla e intuitiva.
- 3. Neutro, no me parece ni sencillo ni complejo el proceso.
- 4. Uso, lectura y visualización algo complejo y poco intuitivo.
- 5. Uso, lectura y visualización complicada, no me resulta intuitivo.

4. Para la Prueba #2: conexión de un dispositivo conector (o Smart Plug). Con respecto a la instalación del dispositivo: *

Marca solo un óvalo.

- 1. Muy sencillo y rápido, con instrucciones completamente claras.
- 2. Algo sencillo, y no tan rápido, con instrucciones adecuadas.
- 3. Neutro, no me parece ni sencillo ni complejo el proceso, tiempo o instrucciones.
- 4. Algo complicado, y un poco extenso, pero las instrucciones están claras.
- 5. Complicado, extenso, y las instrucciones podrían mejorar.

5. Posterior a la instalación de la Prueba #2, en cuanto al uso, cambios de estado, y lectura de los dispositivos en la aplicación Robobo Home Assistant App: *

Marca solo un óvalo.

- 1. Uso, lectura y visualización muy sencilla e intuitiva.
- 2. Uso, lectura y visualización algo sencilla e intuitiva.
- 3. Neutro, no me parece ni sencillo ni complejo el proceso.
- 4. Uso, lectura y visualización algo complejo y poco intuitivo.
- 5. Uso, lectura y visualización complicada, no me resulta intuitivo.

6. Para la Prueba #3: conexión de sensores inteligentes. Con respecto a la instalación de los dispositivos: *

Marca solo un óvalo.

1. Muy sencillo y rápido, con instrucciones completamente claras.
2. Algo sencillo, y no tan rápido, con instrucciones adecuadas.
3. Neutro, no me parece ni sencillo ni complejo el proceso, tiempo o instrucciones.
4. Algo complicado, y un poco extenso, pero las instrucciones están claras.
5. Complicado, extenso, y las instrucciones podrían mejorar.

7. Posterior a la instalación de la Prueba #3, en cuanto al uso y lectura de los dispositivos en la aplicación Robobo Home Assistant App: *

Marca solo un óvalo.

1. Uso, lectura y visualización muy sencilla e intuitiva.
2. Uso, lectura y visualización algo sencilla e intuitiva.
3. Neutro, no me parece ni sencillo ni complejo el proceso.
4. Uso, lectura y visualización algo complejo y poco intuitivo.
5. Uso, lectura y visualización complicada, no me resulta intuitivo.

8. Para la Prueba #4: conexión de una cámara inteligente. Con respecto a la instalación del dispositivo: *

Marca solo un óvalo.

1. Muy sencillo y rápido, con instrucciones completamente claras.
2. Algo sencillo, y no tan rápido, con instrucciones adecuadas.
3. Neutro, no me parece ni sencillo ni complejo el proceso, tiempo o instrucciones.
4. Algo complicado, y un poco extenso, pero las instrucciones están claras.
5. Complicado, extenso, y las instrucciones podrían mejorar.

9. Posterior a la instalación de la Prueba #4, en cuanto al uso y lectura de los dispositivos en la aplicación Robobo Home Assistant App: *

Marca solo un óvalo.

1. Uso, lectura y visualización muy sencilla e intuitiva.
2. Uso, lectura y visualización algo sencilla e intuitiva.
3. Neutro, no me parece ni sencillo ni complejo el proceso.
4. Uso, lectura y visualización algo complejo y poco intuitivo.
5. Uso, lectura y visualización complicada, no me resulta intuitivo.

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios