

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN SERVIDOR *IOT* PARA CONVERTIR UNA CASA EN INTELIGENTE**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**DENNIS DAVID YANACALLO CASALIGLIA**

**DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ**

**DMQ, agosto 2022**

## CERTIFICACIONES

Yo, Dennis David Yanacallo Casaliglia declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



**Dennis David Yanacallo Casaliglia**

**dennis.yanacallo@epn.edu.ec**

**swf\_sinned@hotmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por Dennis David Yanacallo Casaliglia, bajo mi supervisión.



**Leandro Antonio Pazmiño Ortiz**

**DIRECTOR**

**leandro.pazmino@epn.edu.ec**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



Dennis David Yanacallo Casaliglia

CI: 172508144-0

## **DEDICATORIA**

El presente proyecto de titulación se la dedico a Dios por brindarme salud y vida para culminar mis estudios, a mi mamá Ana Casaliglia y mis hermanos Katty, Daniel y Mauricio por creer en mí y brindarme su cariño y apoyo total en el transcurso de la carrera y a mi pareja Lizbeth por su compañía, paciencia y palabras de aliento para no rendirme. Esto es gracias a ustedes.

## **AGRADECIMIENTO**

Un agradecimiento especial a los profesores que me formaron como estudiante y persona, inculcándome valores y conocimientos. Gracias a mi tutor de tesis el Ing. Leandro Pazmiño que supo guiarme durante todo el proceso con paciencia y disciplina para realizar un buen trabajo. Por último, a mis amigos Leoni, Rene, Oscar y Bryan, gracias por su amistad, horas de estudio y charlas para culminar la carrera.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDOS.....	V
RESUMEN.....	VIII
ABSTRACT.....	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO .....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos .....	2
1.3 Alcance.....	2
1.4 Marco Teórico .....	3
Casa inteligente.....	3
<i>Raspberry Pi</i> .....	3
<i>Home Assistant</i> .....	3
Sensores .....	3
Actuadores .....	4
Detector de movimiento inteligente Infrarrojo Pasivo ( <i>PIR</i> ) CT60W .....	4
Bombilla <i>LED</i> inteligente A60.....	4
Enchufe inteligente <i>TP-Link</i> HS103.....	4
YAML.....	5
2 METODOLOGÍA.....	5
3 RESULTADOS .....	6
3.1 Identificación de los requerimientos para el diseño del prototipo.....	6
Implementación basada en <i>software</i> libre .....	6
Selección de los dispositivos inteligentes en el servidor <i>IoT</i> .....	6
Rutinas de automatización de los dispositivos .....	7

Sistema de registro de funcionamiento de los dispositivos inteligentes.....	7
Funcionamiento del servidor <i>IoT</i> .....	7
Organización del prototipo.....	7
3.2 Selección del <i>hardware</i> y <i>software</i> acorde a los requerimientos establecidos	8
Selección del <i>hardware</i> .....	8
Selección del microordenador para la implementación del servidor.....	8
Selección de memoria microSD.....	9
Selección de los dispositivos inteligentes .....	9
Selección del detector de movimiento .....	10
Selección de la bombilla inteligente.....	10
Selección del enchufe inteligente .....	10
Selección de protección del prototipo .....	10
Selección del <i>software</i> .....	11
Selección del sistema operativo de código abierto.....	11
Selección del <i>software</i> de monitoreo de los dispositivos inteligentes.....	13
3.3 Diseño del prototipo del servidor <i>IoT</i> .....	13
Esquema general del proyecto.....	13
Creación de la unidad de arranque del sistema operativo .....	15
Configuración inicial del sistema operativo .....	16
Registro de los dispositivos inteligentes.....	20
Dispositivos <i>Tuya Smart</i> (Detector de movimiento <i>PIR</i> y bombilla inteligente) .	22
Enchufe inteligente .....	24
Dispositivo móvil .....	24
Creación del servicio de envío de notificaciones.....	25
Creación de las rutinas de automatización.....	26
Encendido de luces automático .....	27
Apagado de luces automático.....	30
Control del brillo de la luz.....	31
Detección de personas en la habitación .....	34

Control de carga inteligente .....	35
Diseño del tablero de resumen del <i>Home Assistant</i> .....	38
Instalación de InfluxDB para el monitoreo de los dispositivos.....	41
Creación de los tableros de monitoreo de los dispositivos en InfluxDB .....	42
3.4 Implementación del prototipo .....	44
Instalación de estuche de la <i>Raspberry Pi</i> .....	44
Distribución del prototipo en la habitación.....	46
Instalación de los elementos del prototipo .....	47
3.5 Realización de pruebas de funcionamiento del prototipo.....	50
Pruebas de funcionamiento de dispositivos .....	50
Pruebas de rutinas de automatización .....	52
Iluminación-Habitación.....	52
Apagado-Habitación .....	53
Brillo de luz .....	54
Detección Personas.....	55
Control de carga .....	56
Verificación de los registros de funcionamiento de los dispositivos .....	57
Demostración del funcionamiento del prototipo .....	59
Mantenimiento del prototipo .....	59
Costo del prototipo .....	61
4 CONCLUSIONES.....	62
5 RECOMENDACIONES .....	63
6 REFERENCIAS BIBLIOGRÁFICAS .....	64
7 ANEXOS.....	68
ANEXO I: CERTIFICADO DE ORIGINALIDAD .....	i
ANEXO II: Enlace de video de funcionamiento.....	ii

## RESUMEN

En la primera sección se presenta la introducción del proyecto, además se plantean los objetivos a cumplir y se desarrolla conceptos de temas necesarios para el entendimiento del proyecto y la implementación del servidor *IoT* con sus componentes y características.

La segunda sección presenta la metodología realizada para el cumplimiento de cada objetivo, detallando los procesos que se llevaron a cabo durante toda la implementación del servidor *IoT*.

En la tercera sección se presenta los resultados obtenidos a partir de los procesos efectuados, ahí se presenta como se realizó la selección del *hardware* y *software* a partir de los requerimientos del prototipo, además se indica las configuraciones necesarias para el diseño e implementación del servidor, en la cual se destacan la instalación del sistema operativo, creación de rutinas de automatización e instalaciones de los dispositivos en el hogar. Por último, para comprobar la funcionalidad del prototipo se muestra los resultados de las pruebas de funcionamiento junto a los registros.

La cuarta sección indica las conclusiones en relación con los resultados obtenidos a partir del cumplimiento de los objetivos y en la quinta sección se presenta las recomendaciones para evitar cometer errores y mejorar la funcionalidad del prototipo.

En las dos últimas secciones se presenta las referencias bibliográficas consultadas para el desarrollo y los anexos que contienen el certificado de originalidad y el enlace al video que demuestra el correcto funcionamiento del prototipo.

**PALABRAS CLAVE:** *Raspberry Pi*, *IoT*, *Home Assistant*, casa inteligente, automatizaciones.

## ABSTRACT

*The first section presents the introduction of the project, as well as the objectives to be met and develops concepts of topics necessary for the understanding of the project and the implementation of the IoT server with its components and features.*

*The second section presents the methodology used to achieve each objective, detailing the processes that were carried out throughout the implementation of the IoT server.*

*The third section presents the results obtained from the processes carried out, it shows how the hardware and software were selected based on the requirements of the prototype, it also indicates the necessary configurations for the design and implementation of the server, where the installation of the operating system, creation of automation routines and installation of the devices in the home are highlighted. Finally, to check the functionality of the prototype, the results of the performance tests are shown together with the logs.*

*The fourth section indicates the conclusions based on the results obtained from the fulfillment of the objectives and the fifth section presents the recommendations to avoid making mistakes and improve the functionality of the prototype.*

*The last two sections present the bibliographical references consulted for the development and the annexes that contain the certificate of originality and the link to the video that demonstrates the correct operation of the prototype.*

**KEY WORDS:** *Raspberry Pi, IoT, Home Assistant, smart home, automations.*

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El control de las tareas del hogar de forma remota ha quedado obsoleto debido al avance tecnológico y al Internet de las cosas, dando paso a las viviendas automatizadas que controlan parámetros como la iluminación, seguridad, temperatura, energía, entre otros; que busca brindar una mejor calidad de vida de los seres humanos. El Internet de las Cosas permite que los dispositivos comunes del hogar puedan conectarse a Internet, además de recibir, enviar y procesar información a través de la red sin la intervención de una persona, proporcionando así inteligencia a la vivienda que suponen un ahorro en costos y tiempo empleado en tareas.

Para llegar a esto, el hogar debe poseer una red inalámbrica y dispositivos inteligentes que tomen la información del lugar como luz, temperatura, gas, movimiento, etc., y permitan la ejecución de actividades programadas que anteriormente se las hacía manualmente, las viviendas inteligentes se han ido incrementando en los últimos años gracias a los beneficios que estas representan en comparación con un hogar tradicional.

A través de este proyecto se realizará la implementación de un prototipo de servidor del Internet de las Cosas (*IoT*) para automatizar varias tareas en el hogar, tales como: encendido y apagado de una bombilla inteligente mediante un sensor de movimiento, control automático del brillo y temperatura del color de la bombilla según la hora del día, detección de presencia de personas en un área y la automatización de un enchufe inteligente para controlar la carga de un dispositivo móvil; convirtiéndola en una vivienda inteligente. El objetivo que se busca es no depender de servidores externos en los cuales se procesa la información de los dispositivos inteligentes instalados en una vivienda, lo que se puede convertir en un problema de seguridad y privacidad. El proyecto también busca ayudar a futuros proyectos de implementaciones domóticas.

Con ayuda de una *Raspberry Pi*, seleccionada a partir de la investigación de los requerimientos y características del prototipo, se realizó el montaje del sistema operativo *Home Assistant* como central domótico de código abierto con el fin de mantener un control centralizado y visible de todos los dispositivos inteligentes instalados garantizando que el usuario tenga conocimiento de quien controla y maneja la información tomada por los nodos finales dentro de la vivienda.

Las rutinas de automatización se hicieron en base a los dispositivos configurados en el sistema y su funcionalidad sobre el hogar, para lo cual se estableció las tareas con condiciones a cumplirse para su ejecución. Se utilizó tres dispositivos inteligentes y se

implementó cinco rutinas de automatizaciones. Para los registros de funcionamiento se utilizó la plataforma *Home Assistant* y el complemento InfluxDB que hace lectura de la activación y desactivación de los nodos finales y las tareas.

## 1.1 Objetivo general

Implementación de un servidor *IoT* para convertir una casa en inteligente.

## 1.2 Objetivos específicos

- Identificar los requerimientos para el diseño del prototipo.
- Seleccionar el *hardware* y *software* acorde a los requerimientos establecidos.
- Diseñar el prototipo del servidor *IoT*.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento del prototipo.

## 1.3 Alcance

Por medio del presente proyecto se realizará la implementación de un servidor *IoT* con el cual se automatizarán varias tareas como: encendido y apagado de una bombilla inteligente mediante un sensor de movimiento, control automático del brillo y temperatura del color de la bombilla según la hora del día, detección de presencia de personas en un área y la automatización de un enchufe inteligente para controlar la carga de un dispositivo móvil; dentro de una vivienda con el objetivo de convertirla en inteligente y sin la necesidad de recurrir a servidores externos para procesar la información. El prototipo contara con las siguientes características:

- Implementación del servidor en base a *software* libre.
- Configuración de tres dispositivos inteligentes en el servidor.
- Creación de rutinas de automatización en base a los dispositivos registrados en el servidor.
- Además, el servidor mostrará los *logs* de funcionamiento de los dispositivos inteligentes configurados dentro de la vivienda.

## 1.4 Marco Teórico

### **Casa inteligente**

Es una residencia donde se centralizan varias tecnologías, permitiendo el control de distintos aspectos del hogar, tales como: seguridad, energía y comunicación; con la finalidad de facilitar la vida del usuario, además de otorgar confort y eficiencia. A esta acción de automatizar varias actividades de la vivienda se le conoce como domótica [1]. Entre las características de una casa inteligente, se tienen: control de la casa de forma remota tanto desde el interior del hogar como desde fuera del mismo, menor participación del usuario en tareas cotidianas de la vivienda, es programable, busca reducir costos energéticos, reducir tiempos y aumentar la seguridad [2].

### **Raspberry Pi**

Es un microordenador posee tamaño reducido y menor costo, comparado con los ordenadores convencionales, está especialmente dirigido para el desarrollo informático de todo tipo de usuarios, también, es utilizado para la implementación de prototipos de tecnología e informática y se caracterizan por tener *software* libre [3]. Su versión más actual la *Raspberry Pi 4B/8* (GB), posee las siguientes características: procesador *Broadcom* con 4 núcleos, *RAM* de 8 (GB), puerto *Gigabit Ethernet*, conexión *Wi-Fi*, conexión *Bluetooth*, un lector para tarjetas microSD, salida de video vía *HDMI*, dos puertos *USB 2.0* y dos puertos *USB 3.0* [4].

### **Home Assistant**

Es un sistema operativo de código abierto dedicado a la domótica, escrito en el lenguaje de programación *Python 3* y permite centralizar el control de varios dispositivos inteligentes en el hogar, automatizando así varias funciones de la vivienda con un funcionamiento básico de desatadores, condiciones y acciones. Además, permite monitorear el funcionamiento de dichos dispositivos, mediante registros dentro de la misma plataforma. Una de las características más importantes es su compatibilidad con alrededor de 2 000 dispositivos y su fácil configuración de acuerdo con las necesidades del usuario [5].

### **Sensores**

Los sensores se encargan de captar información del lugar, dentro de los sistemas domóticos los más utilizados son aquellos que identifican factores como la temperatura, clima, iluminación, movimiento, fuego, gas y humo; a partir de la información del entorno recibida por el sensor se ejecutan acciones en conjunto con los dispositivos actuadores.

En otras palabras, el sensor se encarga de traducir la información física del ambiente en señales eléctricas para el sistema de control [6].

### **Actuadores**

Los actuadores posibilitan al sistema de control para ejecutar las acciones programadas y modificar el estado de los dispositivos, generalmente poseen dos estados como encendido/apagado y funcionan en conjunto con los sensores, entre los dispositivos actuadores más utilizados se tienen: bombillas, contactores, cortinas, cámaras, sirenas, enchufes, calefactores, aire acondicionado, motores, entre otros [7].

### **Detector de movimiento inteligente Infrarrojo Pasivo (PIR) CT60W**

Es un sensor que utiliza el infrarrojo irradiado por personas u objetos que emanan calor para detectarlos mediante el movimiento de sus ondas infrarrojas, además, se denomina pasivo al no producir energía ni calor [8].

El modelo CT60W *Tuya Smart* detecta personas en un ángulo de 110 (°) a una distancia máxima de 12 (m), también es anti mascotas, debido a que no detecta el movimiento de mascotas con un peso menor a 25 (kg), su sensor infrarrojo no se ve afectado por la iluminación ni por la oscuridad, permitiendo así su correcto funcionamiento en todo momento. El sensor trabaja mediante *Wi-Fi* de 2.4 (GHz) y envía una notificación a su aplicación, manteniendo un registro de las detecciones realizadas [9].

### **Bombilla LED inteligente A60**

Es un foco *LED* capaz de acoplarse a otros dispositivos de forma inalámbrica, es decir, vía Internet o *Bluetooth* permitiendo un control avanzado con funciones extras comparadas a las de un foco tradicional [10]. El modelo A60 trabaja con *Wi-Fi* de 2.4 (GHz), permite ajustar el color y el brillo de la bombilla de forma remota por voz o aplicación. Otro aspecto importante, es la opción de configurar horarios y escenas para automatizar la iluminación de la vivienda [11].

### **Enchufe inteligente TP-Link HS103**

El modelo HS103 opera con *Wi-Fi* de 2.4 (GHz), permite controlar remotamente, ya sea por voz o a través de la aplicación a distintos dispositivos del hogar, tales como: ventilador, aire acondicionado, cocina, microondas, entre otros electrodomésticos; también, posee configuración de escenas para un control automatizado en el encendido y apagado de los dispositivos eléctricos [12].

## YAML

Es un lenguaje para serialización de datos usado en el desarrollo de archivos que poseen configuraciones, además es de fácil entendimiento para cualquier persona, funciona en secciones de bloques y sangrías similar a *Python*, para crear funciones dentro de otras. *Home Assistant* usa este lenguaje para sus archivos de configuración, añadir integraciones, crear automatizaciones, entre otros. La extensión *.yaml* está disponible en *GitHub* lo que posibilita extraer configuraciones y automatizaciones compartidas por otros usuarios, facilitando el proceso de creación de tareas en el sistema [13].

## 2 METODOLOGÍA

Para iniciar se analizó el alcance del proyecto para establecer los respectivos requerimientos, a partir de los cuales se generaron las características principales que debe tener el diseño del servidor *IoT* para solventar el primer objetivo. A partir de los requerimientos también se analizó en base a una investigación bibliográfica los elementos de *hardware* y *software* necesarios para satisfacer las necesidades del prototipo.

En la sección de selección de *hardware* se detallaron las características para la *Raspberry Pi* y se seleccionó mediante una comparación entre distintas versiones y cual brinda mejores características al prototipo, de igual manera en el resto de los componentes fueron escogidos en función de parámetros como: compatibilidad, características y funciones. En cuanto al *software* se realizó una comparación entre sistemas operativos domóticos de código abierto compatibles con la *Raspberry Pi* y complementos que cumplan con los requerimientos determinados.

A continuación, se diseñó el prototipo según el *hardware* y *software* seleccionados y se detalló el esquema general del proyecto que detalla cómo se establecieron las conexiones, también se instaló el sistema y las respectivas configuraciones de cada dispositivo seleccionado. Además, se creó las rutinas de automatización para el encendido y apagado de luces automático, un control del brillo de la iluminación, detección de personas y control de carga inteligente, estas rutinas fueron solicitadas en el plan del proyecto.

Para la implementación del sistema se realizó el montaje del *hardware* para el prototipo y se analizó la ubicación correcta de los tres dispositivos inteligentes para la correcta funcionalidad de las tareas de automatización controladas por el servidor *IoT*. Por último,

se desarrollaron las pruebas de funcionamiento para comprobar el correcto desempeño del prototipo. Los resultados obtenidos de las pruebas y los registros son importantes para la corrección de errores en el caso de que existan, de esta manera se garantizó que el prototipo cumpla con los requerimientos establecidos.

### **3 RESULTADOS**

En la presente sección se detallan los parámetros para la identificación de requerimientos, selección de *hardware* y *software*, diseño e implementación del prototipo y las pruebas de funcionamiento, con el fin de cumplir los objetivos planteados en el plan de titulación.

#### **3.1 Identificación de los requerimientos para el diseño del prototipo**

En base al análisis realizado se identificó los siguientes requerimientos para cumplir con el diseño del servidor *IoT* que cumpla con las funciones de automatizar las tareas del hogar mediante el uso de tres dispositivos inteligentes y la creación de rutinas automatizadas con el fin de no depender de servidores externos además de evitar problemas de privacidad y seguridad.

##### **Implementación basada en *software* libre**

Para el cumplimiento de este requerimiento se requiere el uso de *software* de código abierto para la implementación del servidor, que ofrezca la libertad al usuario para su manejo y modificación, además de ser gratuito e ilimitado, una de sus ventajas es la seguridad superior a la de un *software* privado gracias a la accesibilidad al código fuente [14]. Entre las características se busca soporte, compatibilidad con varios dispositivos y configuración fácil, esto último es importante ya que a partir del mismo se diseña un servidor *IoT* adecuado para convertir una casa en inteligente que brinde una buena experiencia domótica y cumpla con las exigencias del usuario.

##### **Selección de los dispositivos inteligentes en el servidor *IoT***

El prototipo requiere el uso de tres dispositivos inteligentes que son controlados de manera inalámbrica a través de la red *Wi-Fi*, con la finalidad de realizar tareas del hogar de forma automática, además, gracias a su capacidad para interactuar entre ellos se realiza acciones repetitivas en conjunto, para brindar comodidad, ahorro, productividad, y seguridad doméstica. Estos dispositivos deben ser compatibles con el *software*

seleccionado para la implementación del servidor, también que su precio sea asequible para lo cual se investiga las marcas existentes en el mercado y su disponibilidad.

### **Rutinas de automatización de los dispositivos**

Es necesario la creación de rutinas de automatización en el servidor *IoT* a partir de los dispositivos inteligentes seleccionados, con el fin de centralizar el control de estos, independientemente de su tipo y marca para minimizar su manejo manual. Para esto se analiza las tareas a implementar, que generalmente son control de apagado, encendido, abrir, cerrar, ajustar, esto en función a los dispositivos y el horario de funcionamiento de las acciones con el fin de mejorar la eficiencia del prototipo.

### **Sistema de registro de funcionamiento de los dispositivos inteligentes**

Se requiere el registro de funcionamiento de los dispositivos inteligentes que operan en el servidor *IoT*, para monitorear el funcionamiento de estos y corregir errores en caso de que se presenten; para esto, se analiza el sistema de registros que maneja el sistema operativo y los detalles que se presentan en los mismos para verificar que se cumpla el requerimiento del plan de tesis.

### **Funcionamiento del servidor *IoT***

El servidor *IoT* realiza las tareas de control y monitoreo de los dispositivos inteligentes, para lo cual se encuentra siempre en funcionamiento y así cumplir con todas las automatizaciones asignadas al hogar en todo el día, con la potencia adecuada ofrecida por una fuente de alimentación se otorga resistencia a variaciones de voltaje. Además, se requiere la protección y sistema de enfriamiento del servidor para salvaguardar la salud del dispositivo y precautelar inconvenientes ocasionados por factores externos como pueden ser el polvo, el clima, golpes, temperatura de funcionamiento, mala manipulación del equipo, entre otros; esto garantiza el cumplimiento de los requerimientos del prototipo.

### **Organización del prototipo**

Al ser un prototipo cuya ejecución es para convertir una casa en inteligente, este se encuentra implementado en un escenario real, en donde cumple las funciones de automatización configuradas a diario y se tiene así una visión más precisa del funcionamiento del prototipo sobre situaciones comunes que se presentan en el hogar.

## 3.2 Selección del *hardware* y *software* acorde a los requerimientos establecidos

### Selección del hardware

#### Selección del microordenador para la implementación del servidor

Para la selección del microordenador se toma en cuenta la *Raspberry Pi*, que es una placa de menor tamaño que la de los computadores convencionales, capaz de ejecutar tareas básicas de otros ordenadores con las ventajas de manejar *software* libre, su tamaño y costo reducido [15]. También se tiene un uso frecuente dentro de la domótica, debido a la existencia de sistemas operativos abiertos, librerías y complementos, facilitando la programación de tareas y el acceso a funciones extras, además posee varios tipos de *hardware* compatible como sensores, cámaras de video, periféricos, estuches, pantallas, entre otros. En cuanto se refiere a la conexión de red, *Raspberry Pi* cuenta con un puerto *Ethernet* además de una tarjeta de red inalámbrica, necesario para el desarrollo del prototipo.

En la Tabla 3.1 se realiza la comparación entre las dos versiones más actuales de *Raspberry Pi* que cumplen con las características necesarias para la implementación del prototipo.

**Tabla 3.1** Comparativa entre *Raspberry Pi* 3 B+ y *Raspberry Pi* 4 B [16]

	<b><i>Raspberry Pi</i> 3B+</b>	<b><i>Raspberry Pi</i> 4B</b>
<b>Procesador</b>	<i>Broadcom</i> BCM2837, <i>quad-core</i> ARMv8	<i>Broadcom</i> BCM2711, <i>quad-core</i> Cortex-A72
<b>RAM</b>	1 (GB)	1, 2, 4, 8 (GB)
<b>USB</b>	4 puertos 2.0	2 puertos 2.0 2 puertos 3.0
<b>Conectividad</b>	Puerto <i>Ethernet</i> hasta 300 (Mbps) Tarjeta de red inalámbrica <i>Bluetooth</i> 4.2	Puerto <i>Gigabit Ethernet</i> hasta 1 000 (Mbps) Tarjeta de red inalámbrica <i>Bluetooth</i> 5.0
<b>Video</b>	1 puerto <i>HDMI</i> , resolución <i>Full HD</i>	2 puertos <i>Micro HDMI</i> , resolución <i>Ultra HD</i>

A partir de la identificación de las características de cada modelo, se seleccionó la versión de *Raspberry Pi* 4B/8 (GB), al ser compatible con *software* más exigente,

además de que el servidor *IoT* será levantado en el microordenador y sus creadores ofrecen un rendimiento tres veces mayor que el 3B+, de esta manera es capaz de ejecutar el sistema operativo domótico que maneja y recibe las notificaciones de todos los dispositivos inteligentes disponibles; al poseer 8 (GB) de memoria *RAM* se reduce los problemas de funcionamiento de los dispositivos y se obtiene una mayor eficiencia, además se tiene la opción de expandir el sistema domótico a futuro, añadiendo cámaras de seguridad que vigilen el hogar las 24 horas del día.

El modelo 4B también posee un puerto *Gigabit Ethernet* sin limitaciones, lo que comparado con la versión 3B+ muestra una mejora de conectividad *Ethernet* en 700 (Mbps), aportando al prototipo mayor velocidad en la ejecución de las tareas inteligentes y un procesamiento de las funciones remotas más rápido [17].

### **Selección de memoria microSD**

La memoria microSD es un punto importante para la implementación del servidor, al ser un sistema que posee base de datos, se realizan escrituras continuas con cada cambio de estado [18]. Por esta razón se optó por una microSD *Kingston* de 32 (GB) Clase 10 A1, que ofrece velocidades de hasta 100 (MB/s) mejorando la eficiencia en el procesamiento de datos.

### **Selección de los dispositivos inteligentes**

Para las rutinas de automatización se requiere la configuración de dispositivos inteligentes, estos se caracterizan por interconectar al usuario con el sistema domótico, recopilando, procesando y realizando acciones definidas por el usuario y a su vez manejarlos desde cualquier lugar. Su instalación puede ser de forma cableada o inalámbrica, en redes internas o externas, generalmente el control remoto sobre estos actúa mediante *Wi-Fi* o *Bluetooth*. Uno de los principales requerimientos de los dispositivos es el funcionamiento a través de conexión *Wi-Fi*, para acceder al Internet mediante la red en la que se encuentra implementado el servidor y puedan ser interconectados dentro del mismo.

Para la selección de los dispositivos, se analizaron dos tipos principales para realizar las automatizaciones, los cuales son: sensores y actuadores.

A partir del análisis de los tipos de dispositivos, se procedió a la selección de una bombilla inteligente, un detector de movimiento inteligente *PIR* y un enchufe inteligente. El detector y la bombilla cumplen la función de sensor y actuador, al igual que el enchufe inteligente como actuador, permitiendo automatizaciones complementarias y útiles para el hogar.

### **Selección del detector de movimiento**

Como primer dispositivo inteligente se optó por un detector de movimiento inteligente *PIR* que cumple con las funcionalidades de seguridad y gestión de energía junto a la bombilla inteligente, además se revisa las marcas soportadas por el sistema operativo para su integración al servidor, de tal manera que se seleccionó el detector de movimiento inteligente *PIR CT60W Tuya Smart*, que detecta la presencia de personas en una amplia distancia y trabaja con conexión *Wi-Fi* de 2.4 (GHz), este dispositivo maneja un registro de las detecciones, funcionamiento óptimo en ambientes con mucha o poca iluminación, tiene un costo económico en el mercado y además cumple con las características requeridas para realizar las automatizaciones [9].

### **Selección de la bombilla inteligente**

Para la función de actuador junto con el sensor de movimiento se seleccionó una bombilla inteligente que realice las funciones de iluminación en el hogar de forma automática y programada. La bombilla LED inteligente A60 *Tuya Smart* es compatible con el sistema operativo manejado y no es de costo elevado, además de funcionar mediante *Wi-Fi* de 2.4 (GHz) posee configuraciones de color y brillo de la luz, esta se complementa con el sensor y facilita la creación de las rutinas de automatización para cumplir con el alcance del proyecto.

### **Selección del enchufe inteligente**

Como último dispositivo se optó por un enchufe inteligente para controlar el paso de energía de aparatos electrónicos y manejar su encendido o apagado, de tal manera que cumpla con la función de ahorro energético parte de la domótica. Para esto se seleccionó el enchufe inteligente *TP-Link HS103* gracias a su facilidad de acoplamiento a las tareas de automatización, su disponibilidad y precio en el mercado. Además, es compatible con el sistema operativo y al igual que el resto de los dispositivos trabaja mediante *Wi-Fi* de 2.4 (GHz), por lo cual cumple con los requisitos para ser registrado en el servidor *IoT*.

### **Selección de protección del prototipo**

Uno de los requerimientos del proyecto hace referencia al funcionamiento constante del servidor *IoT* por lo cual se requiere protecciones contra agentes externos, sobretensión y temperaturas altas para evitar inconvenientes en el desempeño de este. Para esto se seleccionó el estuche *Argon One* para *Raspberry Pi 4* que cumple con todas estas características.

El estuche está fabricado de aluminio dándole resistencia y protección a la placa, además contiene en su interior una columna que transfiere el calor de la placa hacia el aluminio para disipar el calor de forma pasiva ayudado de ranuras de ventilación. También cuenta con un ventilador en su interior de velocidad controlable para mantener la *Raspberry Pi* a temperaturas óptimas, como muestra la Figura 3.1. Otro punto importante de la carcasa es que cuenta con un botón de encendido y apagado para hacerlo de forma segura sin causar daños en su sistema, posee un puerto de alimentación para conectar la fuente de alimentación y brindar protección al servidor [19]. La fuente de alimentación seleccionada es de *USB-C* que posee voltaje de salida de 5.1 (VDC), con la corriente indicada para operar de forma segura.



Figura 3.1 Sistema de refrigeración del estuche [20]

## Selección del software

### Selección del sistema operativo de código abierto

Para la implementación del servidor *IoT* el aspecto más preocupante es la vulnerabilidad de seguridad cuando se ingresa dispositivos nuevos al hogar, debido al desconocimiento de quien los controla, registra cada movimiento de las tareas cotidianas y quien tiene acceso a esta información. También existe una limitación entre la compatibilidad de los dispositivos de distintos fabricantes, por esta razón se busca una plataforma que ofrezca diferentes soluciones inteligentes. Por tal motivo el *software* de código abierto es un punto fundamental para el sitio en el que se manejan los dispositivos conectados, de tal manera que el usuario entienda completamente el funcionamiento de los programas que operan en el hogar, además de integrar diferentes marcas de dispositivos y tecnologías centralizando su control.

En la actualidad existen distintas opciones de *software* domótico que se ejecutan en todo tipo de ordenadores, ya sea personales o microordenadores como por ejemplo la *Raspberry Pi*, que fue el *hardware* seleccionado, entre las más comunes se tienen: *Domoticz*, *OpenHab* y *Home Assistant*. A continuación, se detalla las principales características de estos sistemas.

*Domoticz* es compatible con *Windows* y *Linux* de código abierto y bajo consumo de recursos compatible con *Raspberry Pi*, cuenta con interfaz *web* y aplicación móvil que facilita el manejo de las automatizaciones, uno de los problemas es su desactualización en su interfaz y documentación impidiendo el desarrollo de mejores funcionalidades y la disminución de tecnologías soportadas [21].

*OpenHab* es compatible con *Windows*, *OSX* y *Linux*, posee una comunidad que brinda soporte y solución de errores, tiene amplia documentación que facilita su trabajo en distintos protocolos y soporta alrededor de 200 integraciones de fácil configuración para mejorar la experiencia del instalador [21].

*Home Assistant* posee varios apartados para su interoperabilidad con varios dispositivos y servicios, soporta alrededor de 1 400 integraciones para el control del clima, salud, notificaciones, social, entre otros. Posee una interfaz *web* e interfaz de aplicación para dispositivos móviles, las rutinas de automatización se pueden crear a través de la interfaz o mediante un código con lenguaje *YAML*, por lo cual está dirigido para todo tipo de usuario, también tiene conexión con *Telegram* para el envío de notificaciones. También permite la creación de *scripts* para ser añadidas a las automatizaciones y simplificarlas [5]. Su documentación oficial brinda explicación para el manejo de la plataforma a través de ejemplos prácticos.

A partir de las características expuestas, se optó por la selección del sistema operativo *Home Assistant*, debido a su extensa documentación y soporte por parte de la comunidad para el desarrollo de nuevas funcionalidades, del que carece la primera plataforma y por su extenso número de integraciones compatibles comparado con *Domoticz* y *OpenHab*, además de poseer una interfaz sencilla de usar para usuarios nuevos en el mundo de la domótica y la aplicación para celulares que permite el control del hogar desde el dispositivo móvil, convirtiéndola en la mejor opción para la implementación del prototipo.

El sistema elegido actúa como *hub* ya que permite la comunicación y creación de automatizaciones entre varios dispositivos, *Home Assistant* admite la creación de rutinas basado en reglas, que son [22]:

- **Desencadenador:** es el evento que da paso para la lectura de la condición para ejecutar la acción
- **Condición:** impone la regla para que la automatización se ejecute.
- **Acción:** realiza la tarea configurada luego de comprobar el desencadenante y la condición.

Cada dispositivo integrado se le conoce como entidad que puede ser modificado a preferencia del usuario y brinda el estado actual y los atributos del dispositivo al sistema, dependiendo de su tipo, los sensores son la entidad más usada y se encarga de leer la medición de algún parámetro dependiendo de su función para desencadenar una acción programada por el usuario [23].

### **Selección del *software* de monitoreo de los dispositivos inteligentes**

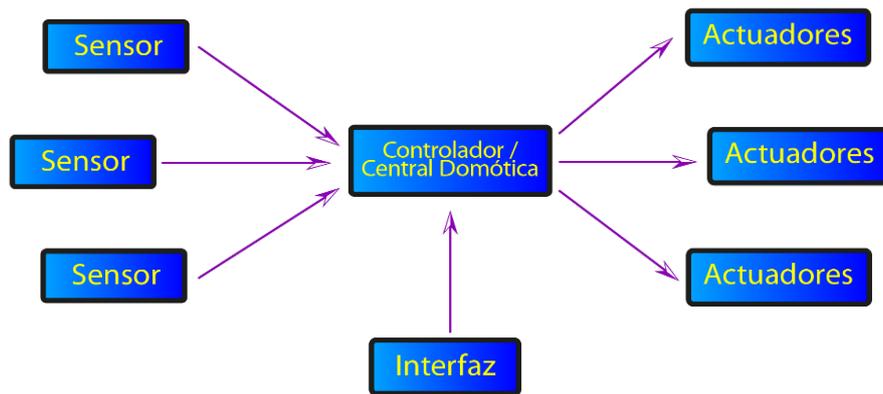
*Home Assistant*, tiene su propio sistema de registros e historial encargado de mostrar todas las acciones que se realizan dentro del servidor como una base de datos, sin embargo, se requiere una visualización gráfica del funcionamiento de los dispositivos, para lo cual se seleccionó el complemento InfluxDB.

InfluxDB es un *software* de código libre y gestiona la base de datos que guardan y monitorean información de equipos *IoT* como por ejemplo sensores, en un tiempo determinado y se procesa la fuente del dato, el valor y el tiempo en dicha base. Es mucho más rápido al procesar y escribir los datos que otras bases que miden en periodos de tiempo [24]. Este complemento se encuentra en la tienda de *Home Assistant* y cumple con el requerimiento de monitorear los registros de funcionamiento de los dispositivos seleccionados.

## **3.3 Diseño del prototipo del servidor *IoT***

### **Esquema general del proyecto**

El esquema general presenta la organización del prototipo y la manera en la que se comunican los elementos que lo conforman. El prototipo se diseñó con un *Raspberry Pi* que hace la función de sistema de control, el proyecto está desarrollado de tal manera que la placa junto con el sistema operativo *Home Assistant* centralice el control y monitoreo de todos los dispositivos integrados y actúe según las condiciones especificadas, a esta arquitectura se le denomina un control centralizado, representado en la Figura 3.2 [25].

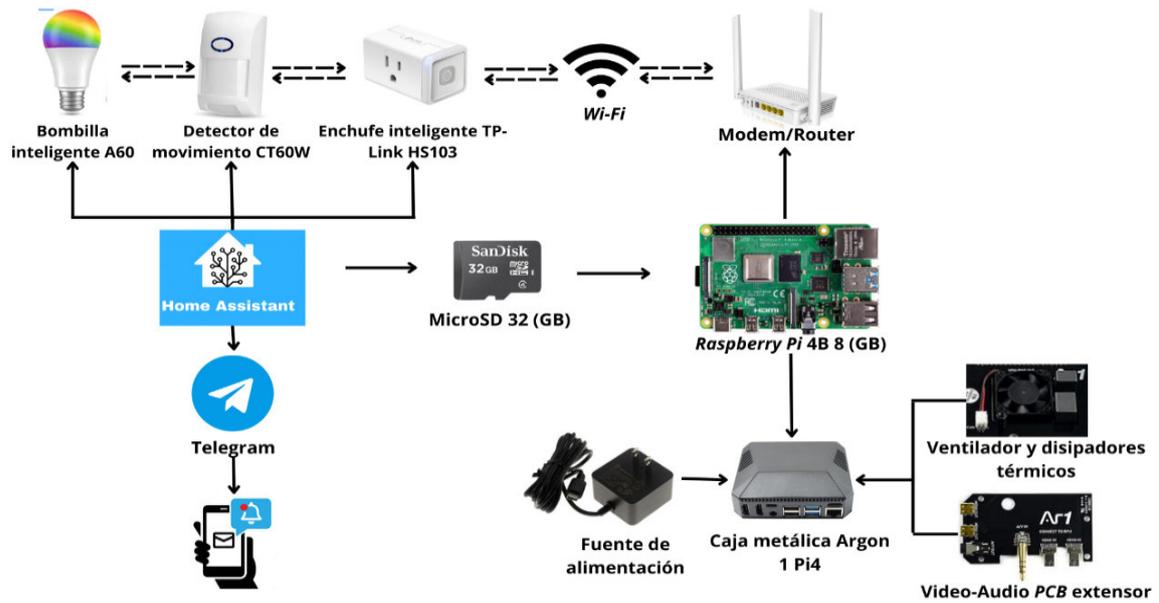


**Figura 3.2** Arquitectura de Control Centralizado [25]

A partir del modelo centralizado, en la Figura 3.3 se representa la organización del prototipo, la *Raspberry Pi* en el centro como el sistema de control domótico con conexión directa al *router* para el funcionamiento del *Home Assistant* que se encuentra en la microSD como unidad de arranque, el cual es de suma importancia ya que es el que permite la comunicación de los dispositivos entre ellos, estos se encuentran integrados en el sistema operativo y funcionan a través de la red *Wi-Fi* de la vivienda permitiendo intercomunicarse entre ellos para la ejecución de las rutinas de automatización creadas. Para el proceso de envío de notificaciones *Home Assistant* se encarga de enviarlas a través de la plataforma de Telegram a la cuenta del usuario.

Finalmente, el microordenador se encuentra al interior de la carcasa *Argon One*, la cual le brinda protección y refrigeración al servidor *IoT*, que es alimentado a través de la fuente de alimentación.

El funcionamiento de los dispositivos inteligentes es de forma automática dependiendo de las condiciones más convenientes impuestas por el usuario y también de forma manual mediante la interfaz del *Home Assistant*.



**Figura 3.3** Esquema general del funcionamiento del prototipo

### Creación de la unidad de arranque del sistema operativo

Para iniciar el *Home Assistant* es necesario crear la unidad de arranque desde la microSD, para ello se usó el software *Raspberry Pi Imager* v1.7.2, en el cual se seleccionó el sistema operativo compatible con la *Raspberry Pi 4B* el *Home Assistant OS 8.2 (RPI 4/400)* y la unidad de almacenamiento para proceder a la escritura del SO a través del cual se inicia la *Raspberry Pi*, como se muestra en la Figura 3.4.



**Figura 3.4** Creación de la unidad de arranque con *Home Assistant*

Adicionalmente se creó un archivo de texto, denominado *my-network*, para permitir al servidor trabajar en la red *Wi-Fi* del hogar; en el cual se colocó el *UUID* generado de una página web, como indica la Figura 3.5. En la Figura 3.6 se observa el documento *txt* donde se colocó el *SSID* y la contraseña de la red *Wi-Fi*.



Figura 3.5 UUID Generador

```
*my-network: Bloc de notas
Archivo Edición Formato Ver Ayuda
[connection]
id=my-network
uuid=f8b88469-efc6-4e32-b067-950e7a9d1408
type=802-11-wireless
[802-11-wireless]
mode=infrastructure
ssid=NETLIFE-TRYHARD
# Uncomment below if your SSID is not broadcasted
#hidden=true
[802-11-wireless-security]
auth-alg=open
key-mgmt=wpa-psk
psk=mrsinned250697
[ipv4]
method=auto
[ipv6]
addr-gen-mode=stable-privacy
method=auto
```

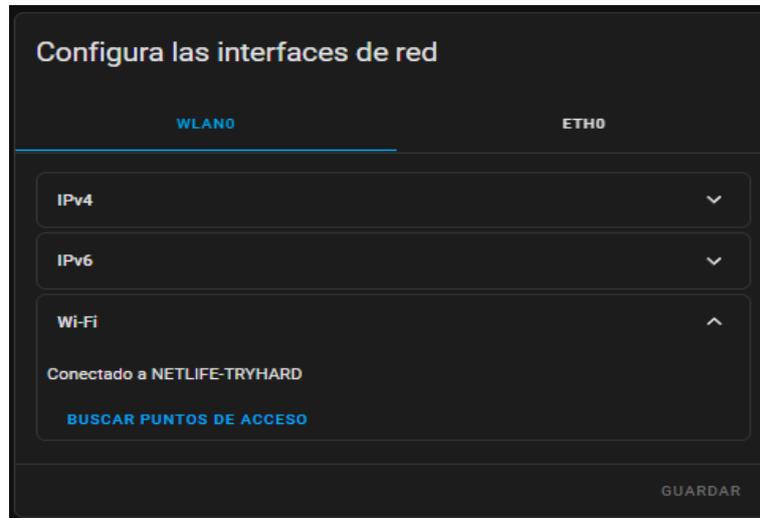
Figura 3.6 Archivo de red

### Configuración inicial del sistema operativo

Luego de finalizado la escritura del sistema y la configuración de red, se ingresó la microSD al *Raspberry Pi*, previamente encendido y conectado vía *Ethernet* al *router*, para dar inicio la instalación del *Home Assistant*. En la Figura 3.7 al conectar la *Raspberry Pi* vía *HDMI* al monitor se visualiza la dirección *IP* 192.168.100.85 asignada por el *router* vía *Ethernet* para la conexión mediante la interfaz *web* del sistema, esta dirección se usa únicamente la primera vez, para la instalación del sistema y creación del usuario.

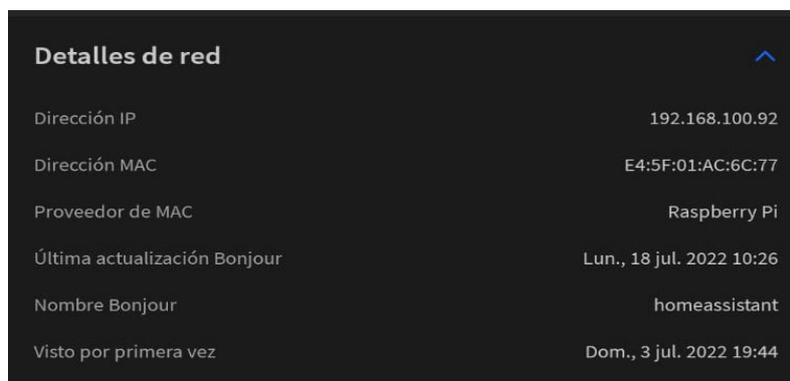


Una vez iniciado el sistema, se desconectó el cable *Ethernet* para conectarse vía *Wi-Fi* como se observa en Figura 3.9.



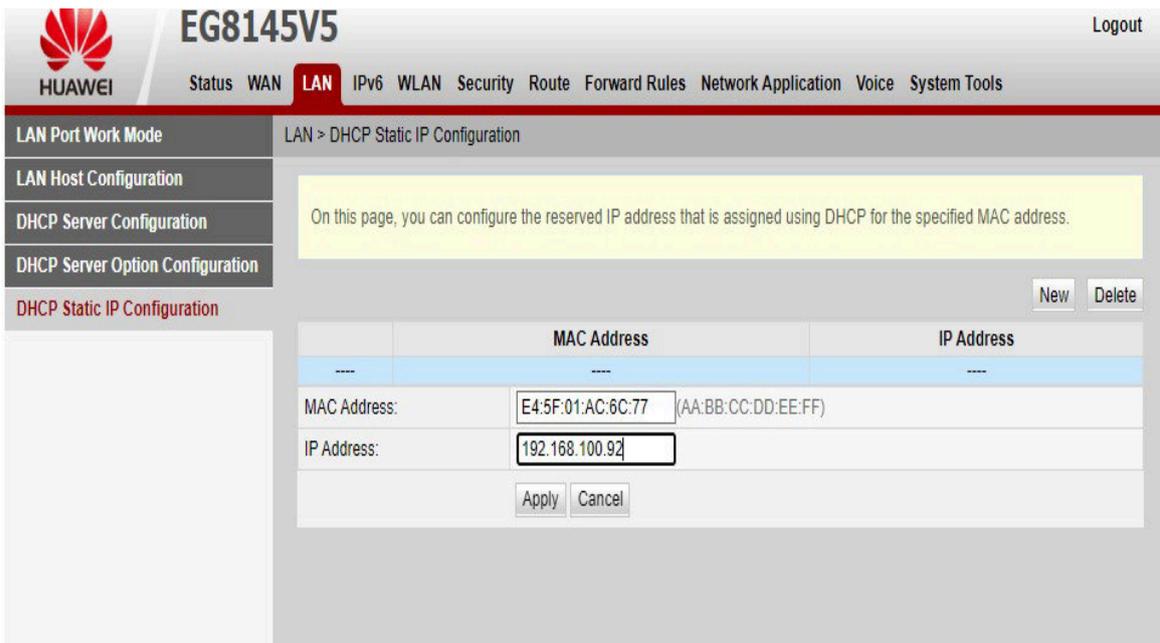
**Figura 3.9** Red *Wi-Fi*

Con la aplicación *Fing*, se visualizó la dirección *IP* asignada por el *router* a la tarjeta de red inalámbrica y la dirección *MAC* de la *Raspberry Pi*, como indica la Figura 3.10.



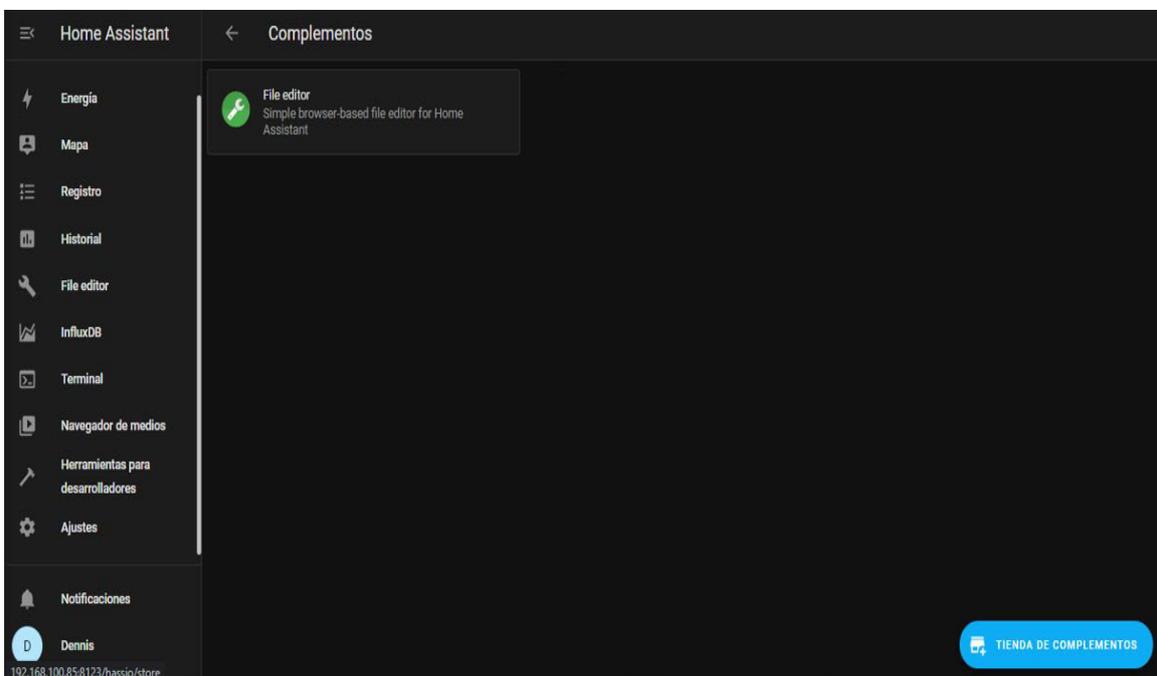
**Figura 3.10** Detalles de red *Raspberry Pi*

A continuación, en la Figura 3.11 se ingresó a la configuración del *router* para fijar la *IP* 192.168.100.92 asignada previamente a la tarjeta de red inalámbrica de la *Raspberry Pi*, como *IP* estática haciendo uso de su dirección *MAC*. De esta manera el microordenador siempre trabajará con esa *IP* fija y no será asignada a otro dispositivo. La dirección señalada será la que se use para conectarse a la interfaz del *Home Assistant*.



**Figura 3.11** Configuración *IP* estática

En la Figura 3.12 se muestra la sección de ajustes, donde se ingresa al apartado de complementos – tienda de complementos y se instala *File Editor* para editar los archivos YAML en el que *Home Assistant* maneja el código de las configuraciones, automatizaciones, *scripts*, *blueprints*, entre otros; como se muestra en la Figura 3.13.



**Figura 3.12** Complemento *File Editor*

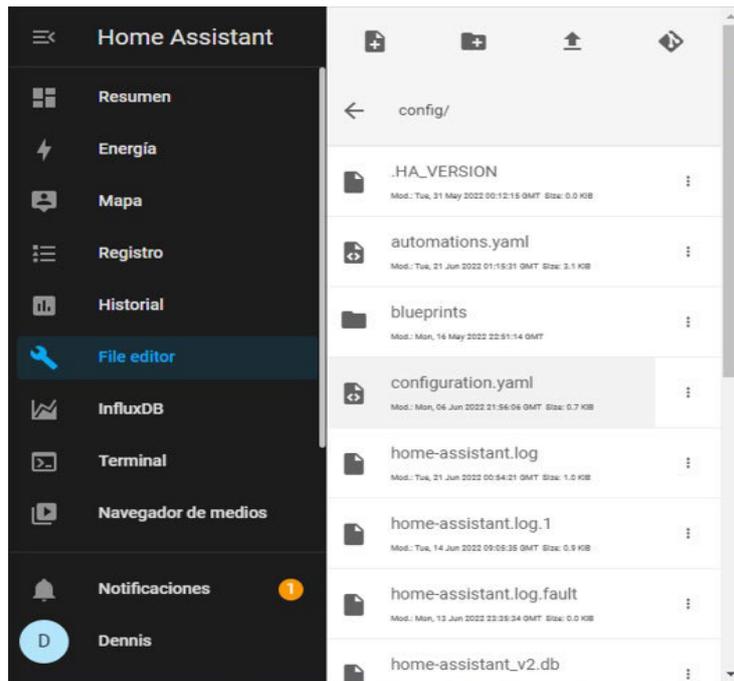


Figura 3.13 Interfaz *File Editor*

### Registro de los dispositivos inteligentes

Para el registro de los dispositivos seleccionados en el servidor *IoT* es necesario conectarlos a la red *Wi-Fi* del hogar mediante sus respectivas aplicaciones móviles dependiendo de la marca de estos. En el caso del detector de movimiento inteligente *PIR* y la bombilla inteligente se instaló la aplicación *Tuya Smart*, mientras que para el enchufe inteligente se usó la aplicación *Kasa Smart*.

Dentro de la aplicación *Tuya Smart* se agregó el dispositivo *PIR* y la fuente de iluminación inteligente, además se introdujo las credenciales de la red inalámbrica para validar la conexión, como muestran las Figura 3.14 y Figura 3.15 respectivamente.



Figura 3.14 Selección del detector de movimiento inteligente *PIR*



**Figura 3.15** Selección de red *Wi-Fi*

En la aplicación *Kasa Smart* de igual manera en la Figura 3.16 se seleccionó el enchufe inteligente HS103 y se colocó un nombre de identificación para el dispositivo, también la conexión a la red *Wi-Fi* para completar la configuración del enchufe, como se muestra en la Figura 3.17.



**Figura 3.16** Selección del enchufe inteligente



**Figura 3.17** Enchufe agregado a la red

## Dispositivos *Tuya Smart* (Detector de movimiento *PIR* y bombilla inteligente)

Para realizar el registro de ambos dispositivos *Tuya* en el servidor se creó una cuenta en <https://iot.tuya.com/> a fin de crear un proyecto en la nube y vincular los dispositivos desde la aplicación *Tuya Smart*, como se visualiza en Figura 3.18.

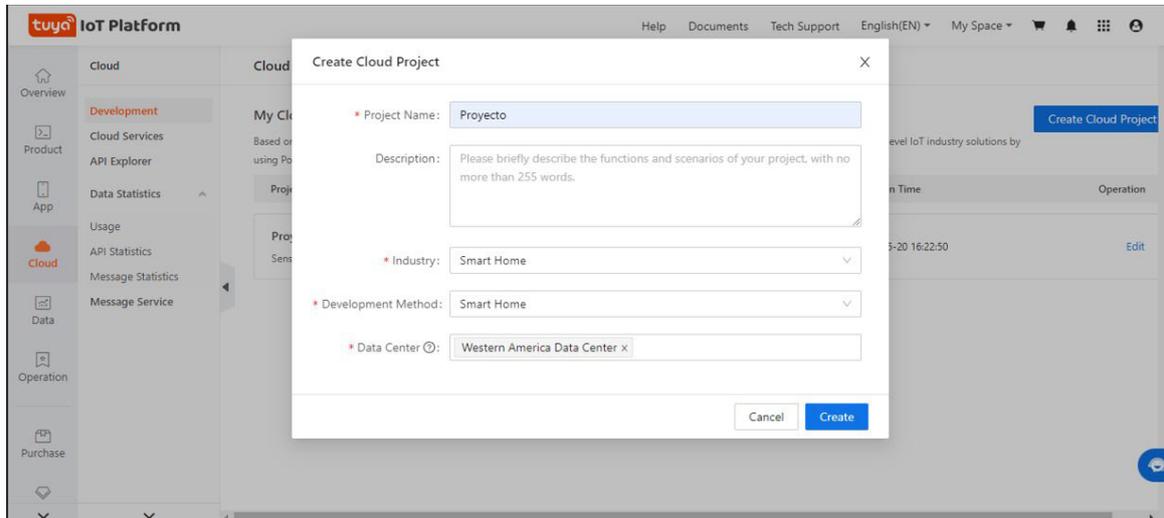


Figura 3.18 Creación del proyecto en la nube

A continuación, se vinculó los dispositivos en la pestaña *Devices*, se seleccionó *Link Tuya App Account – Add App Account* y se generó un código QR para escanear los dispositivos agregados anteriormente en la aplicación *Tuya Smart*, como se muestra en la Figura 3.19.

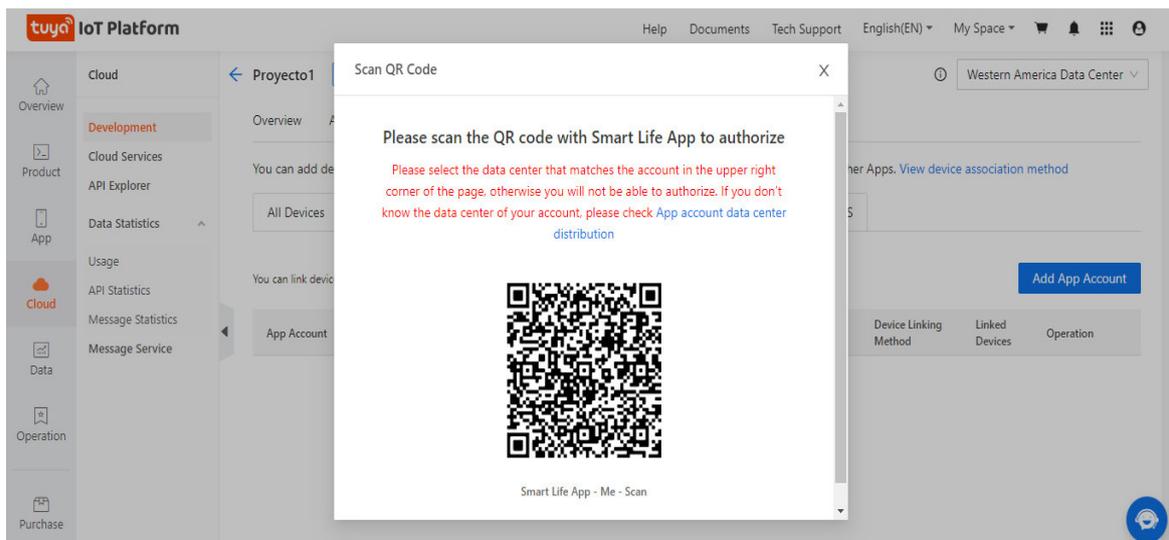
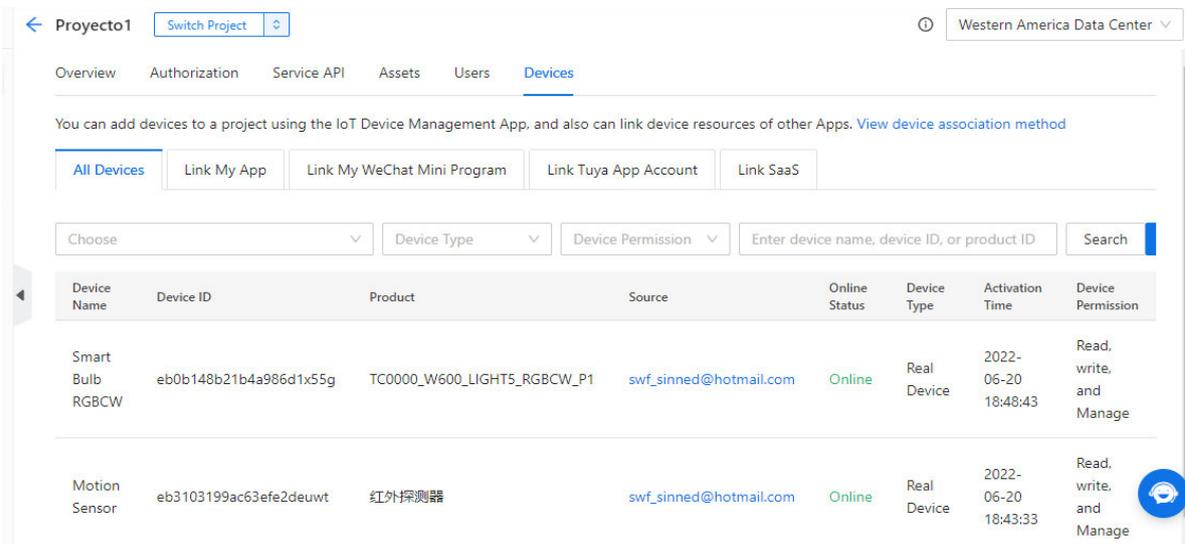


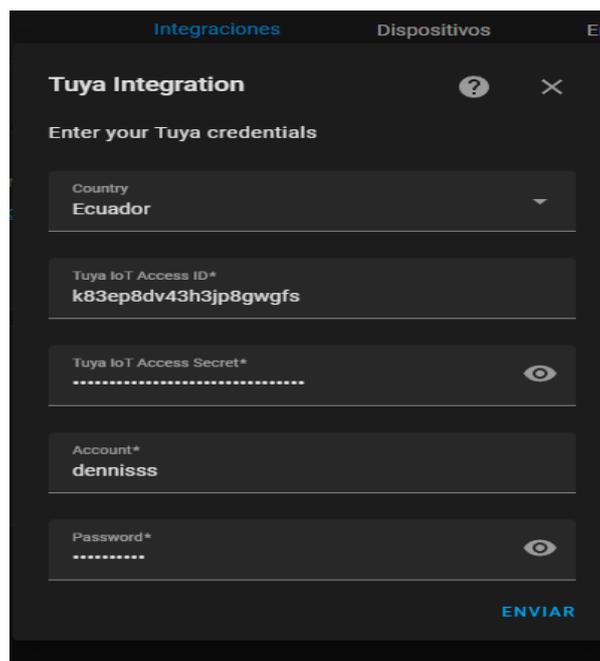
Figura 3.19 Escaneo de dispositivos *Tuya*

En la Figura 3.20, luego de realizar el escaneo de los dispositivos se los visualiza en la pestaña *All Devices*.

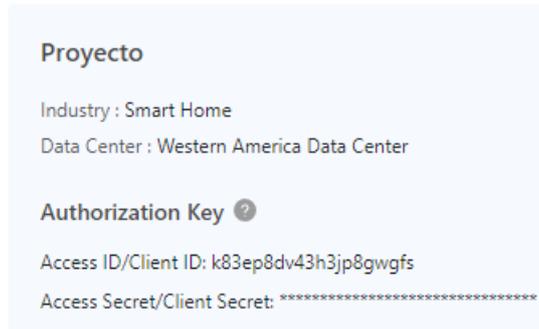


**Figura 3.20** Todos los dispositivos *Tuya*

Para el registro de los dispositivos *Tuya* dentro del *Home Assistant*, se ingresó al apartado de Ajustes – Dispositivos y servicios – Integraciones y clic en agregar integración para seleccionar la marca *Tuya*. En la Figura 3.21 se procedió a colocar los datos del proyecto creado previamente, que se encuentran en la pestaña de Overview, presentes en la Figura 3.22, completando así la agregación de los dispositivos en el servidor.



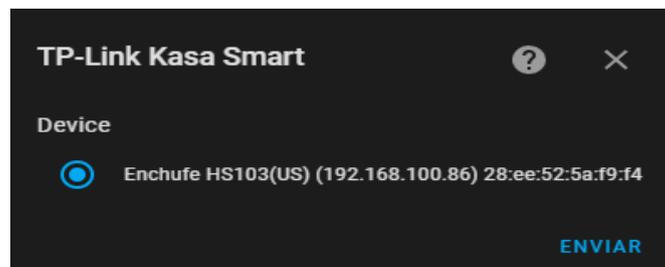
**Figura 3.21** Registro del dispositivo



**Figura 3.22** Credenciales del proyecto

### Enchufe inteligente

Al igual que los dispositivos anteriores, se agregó la integración *TP-Link Kasa Smart*, en la Figura 3.23 el enchufe HS103 se descubre automáticamente y clic en enviar para completar el registro.



**Figura 3.23** Registro de enchufe *TP-Link*

### Dispositivo móvil

Mediante un dispositivo móvil se creó automatizaciones complementarias con el enchufe inteligente. Para ello se instaló la aplicación de *Home Assistant* para *Android* y se ingresó a la cuenta del servidor para añadir el dispositivo MI 8 Lite, como se indica en la Figura 3.24. El celular añade sensores de la batería que permiten monitorear la carga del dispositivo, como indica la Figura 3.25.



**Figura 3.24** Registro del dispositivo móvil

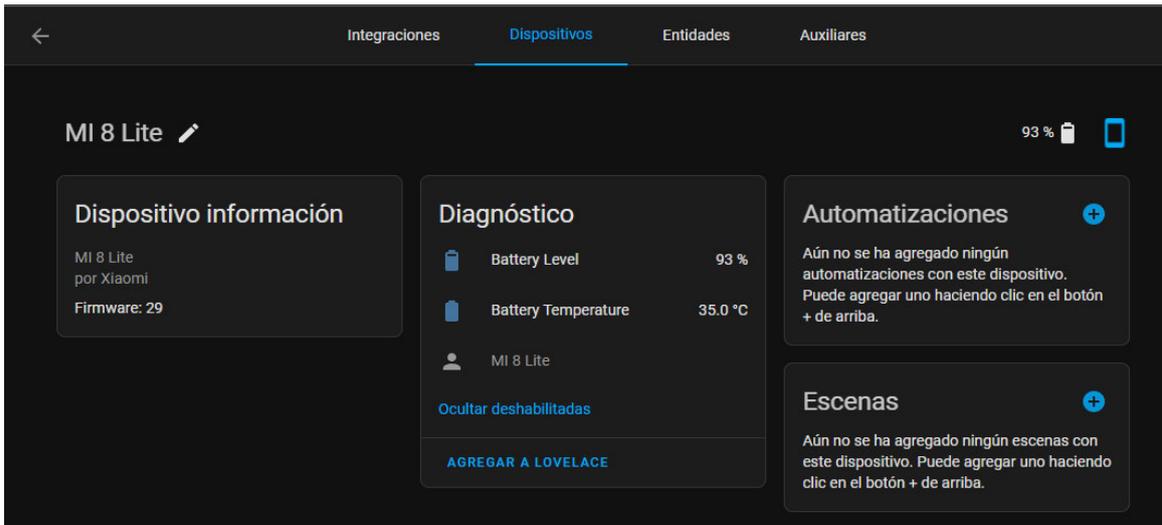


Figura 3.25 Sensores del dispositivo móvil

### Creación del servicio de envío de notificaciones

Para el servicio de notificaciones se realizó mediante la creación de un *Bot* en Telegram con el *BotFather*, dándole a iniciar y crear, indicando el nombre del *Bot* y el usuario como indica la Figura 3.26.

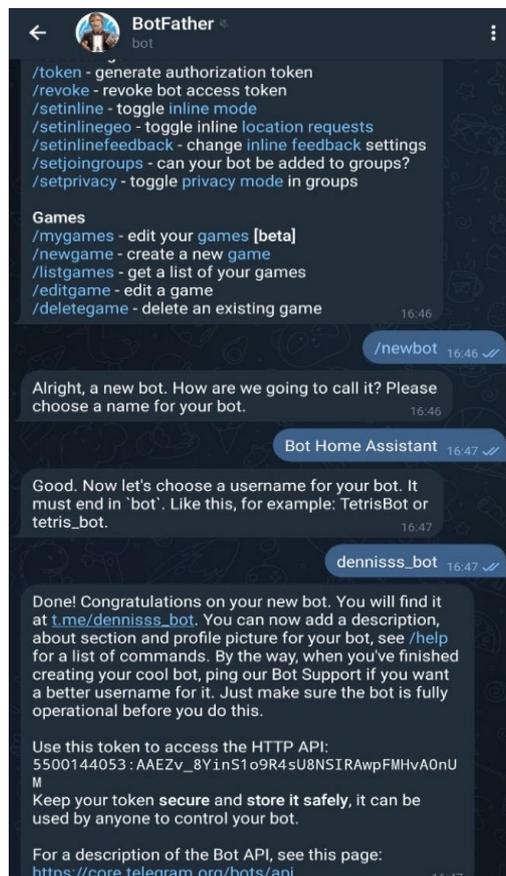


Figura 3.26 Creación del *Bot Home Assistant*

A continuación, se inicia el *Bot* creado y se envía un mensaje para sacar el *ID* del *chat* usando el *Chat ID Bot*, como muestra la Figura 3.27.



**Figura 3.27** *ID* del *chat*

Dentro del *Home Assistant* en el apartado del *File Editor* se editó el archivo *configuration.yaml* agregando el código de la página oficial de *Home Assistant* del apartado de integraciones y añadiendo el nombre, código y el *ID* del *chat* del *Bot* creado, como se indica en la Figura 3.28. Por último, se comprobó la configuración para verificar que el archivo este correctamente escrito.

```
24 # Bot de telegram-
25 telegram_bot:-
26   platform: polling-
27   api_key: 5500144053:AAEZv_8YinS1o9R4sU8NSIRAwPMHvA0nUM #clave del bot creado-
28   allowed_chat_ids:-
29     -1260649594 #id del chat del bot-
30   -
31 # Notificación del bot-
32 notify:-
33   platform: telegram #plataforma utilizada-
34   name: Bot_Home_Assistant #nombre del bot-
35   chat_id: 1260649594 #id del chat del bot-
```

**Figura 3.28** Configuración del *Bot* de Telegram

### Creación de las rutinas de automatización

Se realizó la creación de rutinas de automatización según los dispositivos registrados en el servidor, para esto se crea un total de cinco automatizaciones enfocados en el manejo de la iluminación, detección de presencia y control de la energía, dentro de la habitación del usuario. Esto se realizó en el apartado de Automatizaciones y Escenas – Agregar automatización, en donde se configuró cada una mediante los desencadenantes, condiciones y acciones.

### Encendido de luces automático

Para esta tarea se programó que la bombilla se encienda cada vez que el sensor detecte movimiento, el encendido de luces se realiza de lunes a domingo en el horario de 18:00 a 7:00 (h) debido a que son las horas en las que hay poca iluminación dentro de la habitación, además la automatización funciona siempre y cuando se encuentre activado un ayudante booleano para evitar la activación de las luces mientras el usuario duerme. Para lo cual se creó dicho ayudante en la pestaña de Auxiliares – Agregar auxiliar y se colocó el nombre Mañana, como se visualiza en la Figura 3.29.

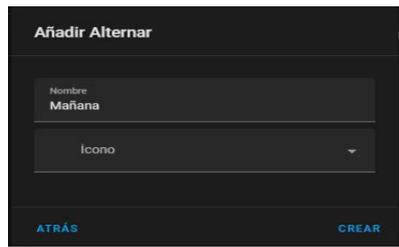


Figura 3.29 Ayudante Mañana

En el diagrama de flujo de la Figura 3.30, se detalla el funcionamiento del encendido de luces automático, en donde se indica que la luz se enciende cuando el sensor comienza a detectar movimiento siempre y cuando cumpla con las condiciones de la hora y el ayudante “Mañana” activado, caso contrario la luz no se activa y continuara con la lectura del sensor, suponiendo así un ahorro energético.

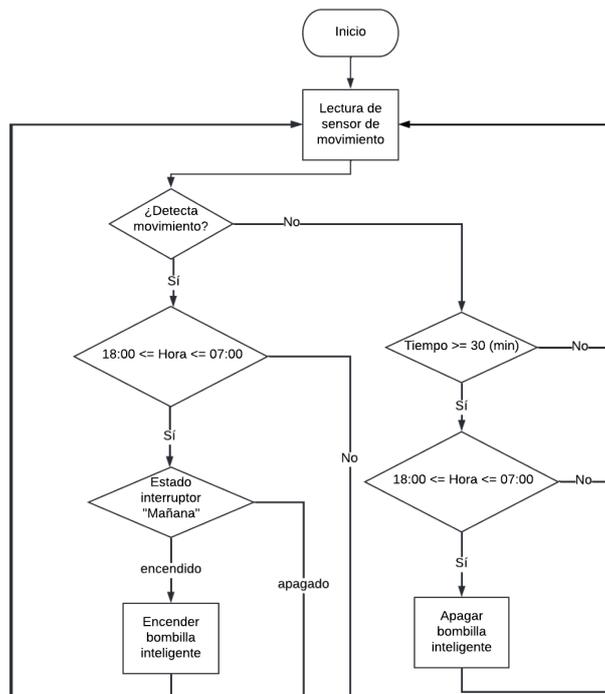
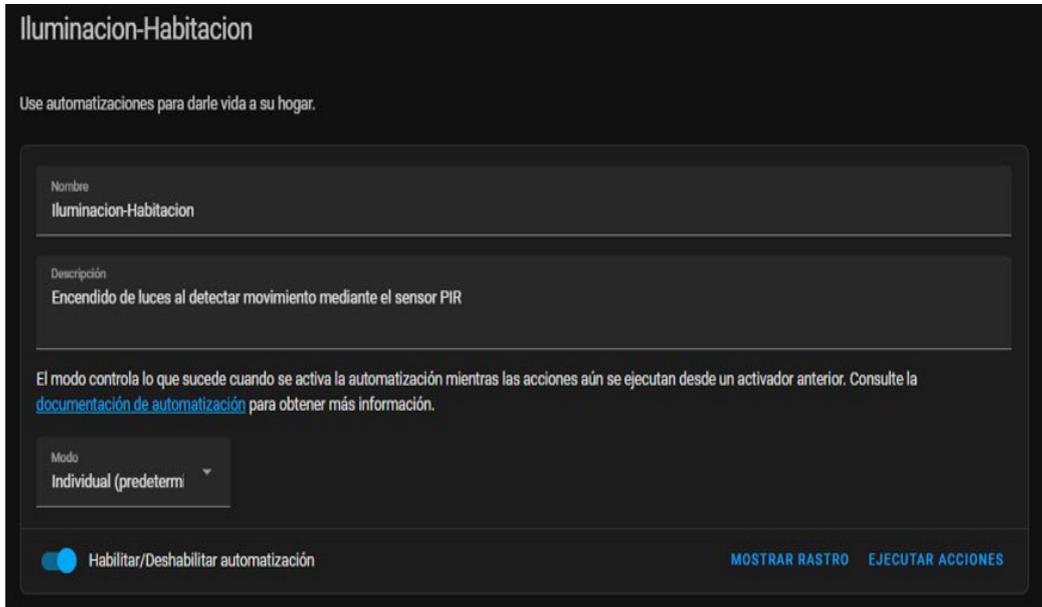


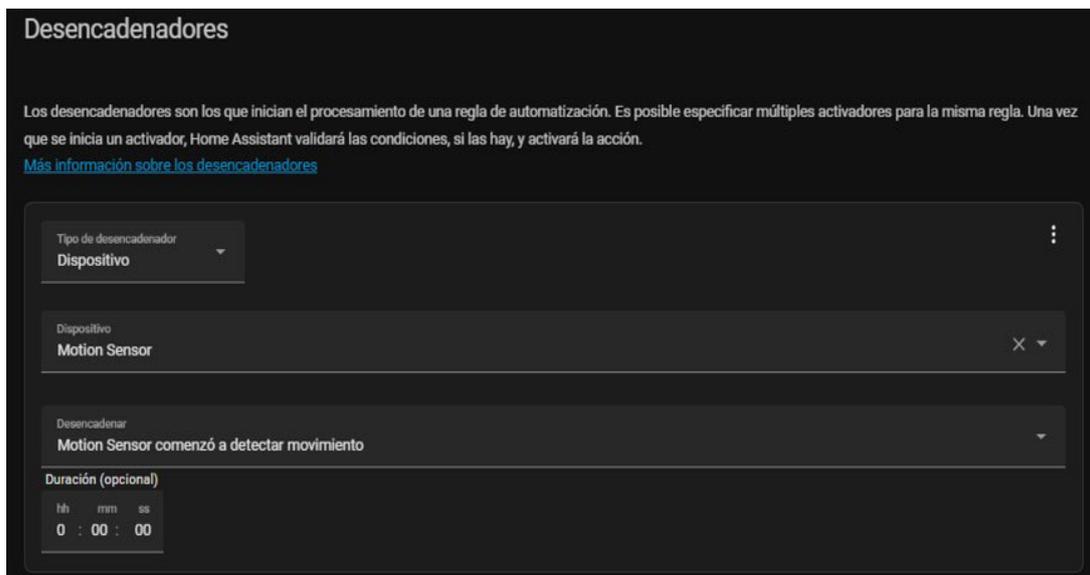
Figura 3.30 Diagrama de flujo del encendido y apagado de luces automático

A continuación, se detalla de manera general como se creó la automatización dentro del *Home Assistant*. Primero se colocó un nombre y descripción para la tarea, como muestra la Figura 3.31.



**Figura 3.31** Nombre y descripción del encendido de luces automático

En desencadenante se selecciona el tipo de dispositivo *Motion Sensor* y que inicie la lectura de la condición luego de detectar movimiento, como se muestra en la Figura 3.32.



**Figura 3.32** Desencadenador del encendido de luces automático

En la Figura 3.33 se selecciona las horas de su funcionamiento detalladas anteriormente y en la Figura 3.34 se selecciona el estado de la variable booleana Mañana se encuentre encendido.

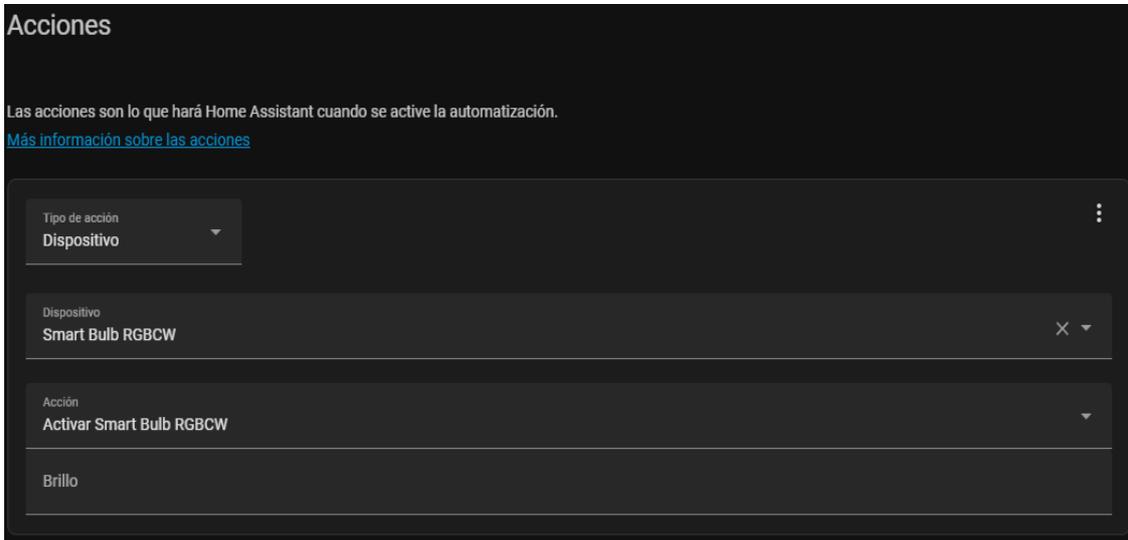
The screenshot shows a configuration window titled 'Condiciones'. At the top, it states 'Las condiciones son opcionales y evitarán una mayor ejecución a menos que se cumplan todas las condiciones.' with a link to 'Más información sobre las condiciones'. The main configuration area includes a dropdown for 'Tipo de condición' set to 'Hora'. Below this, there are two sections for 'Después de' and 'Antes de', each with radio buttons for 'Tiempo corregido' (selected) and 'Valor de un auxiliar de tipo fecha/tiempo'. The 'Después de' section has a time picker set to '18 : 00 : 00', and the 'Antes de' section has a time picker set to '7 : 00 : 00'. At the bottom, there is a dropdown for 'Días entre semana' with the selection 'mon, tue, wed, thu, fri, sat, sun'. A 'PROBAR' button and a menu icon are in the top right corner.

**Figura 3.33** Condición 1 del encendido de luces automático

The screenshot shows a configuration window titled 'Condiciones'. The 'Tipo de condición' dropdown is set to 'Estado'. The 'Entidad+' field contains 'Mañana' with a close button and a dropdown arrow. The 'Atributo' dropdown is empty. The 'Estado' field contains 'on'. At the bottom, there is a 'Por (opcional)' section with a time picker set to '0 : 00 : 00'. A 'PROBAR' button and a menu icon are in the top right corner.

**Figura 3.34** Condición 2 del encendido de luces automático

Por último, se configura la acción a realizar al cumplirse las condiciones, seleccionando la bombilla *Smart Bulb RGBCW* y la acción de activarse, como se indica en la Figura 3.35.



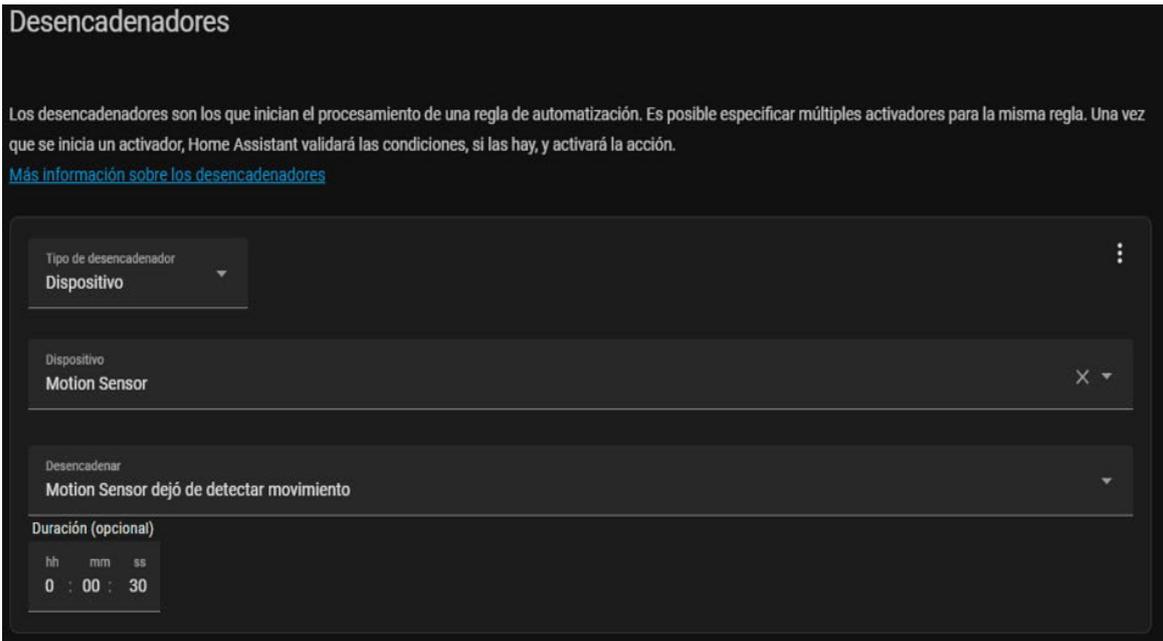
**Figura 3.35** Acción del encendido de luces automático

### **Apagado de luces automático**

Para el apagado de luces se programó que la bombilla se apague después de 30 (min) de no haber detectado movimiento, dentro del mismo horario del encendido de luces automático, al ser una tarea complementaria para el control de iluminación automático.

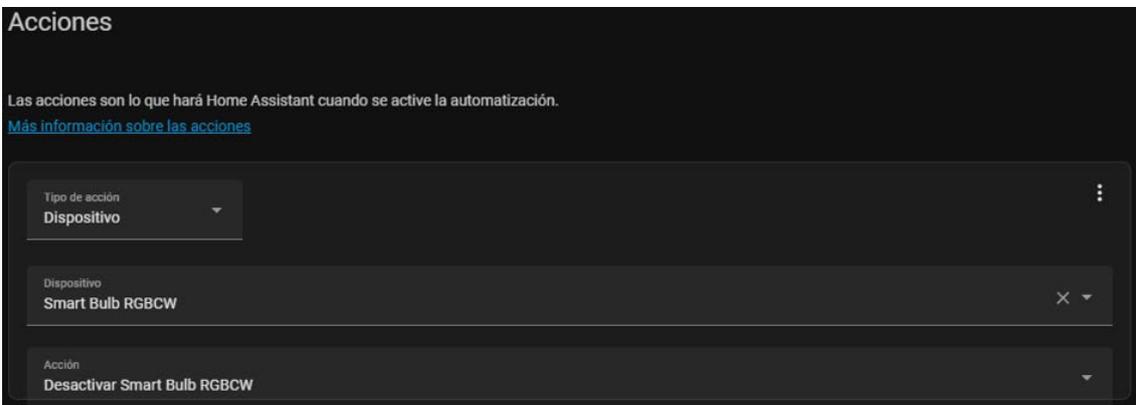
En el diagrama de flujo de la Figura 3.30, se detalla también el funcionamiento del apagado de luces automático, en donde se indica que la luz se apaga cuando el sensor deja de detectar movimiento durante 30 (min), siempre y cuando cumpla con la condición de la hora, caso contrario la luz permanecerá encendida.

La automatización dentro del *Home Assistant*, se creó colocando un nombre y descripción para la tarea, en desencadenante se configuró para que inicie la lectura de la condición luego de no detectar movimiento pasado el tiempo de 30 (min), como se muestra en la Figura 3.36; este tiempo se configuró debido a que la mayoría de sensores sensan con un tiempo mínimo de 3 (s), es decir que durante ese intervalo el dispositivo no considera el evento [26], al no detectar movimiento la bombilla se apagaría inmediatamente, por lo cual se considera un tiempo prudente dentro de las horas configuradas, ya que son las horas en las que hay poca iluminación y el usuario se encuentra realizando actividades en la habitación, con el fin de evitar que el usuario se tenga que mover todo el tiempo.



**Figura 3.36** Desencadenador del apagado de luces automático.

Además, al igual que en el encendido se coloca en condición las mismas horas de funcionamiento y, por último, se configura la acción de desactivar la bombilla al cumplirse la condición, como se indica en la Figura 3.37.



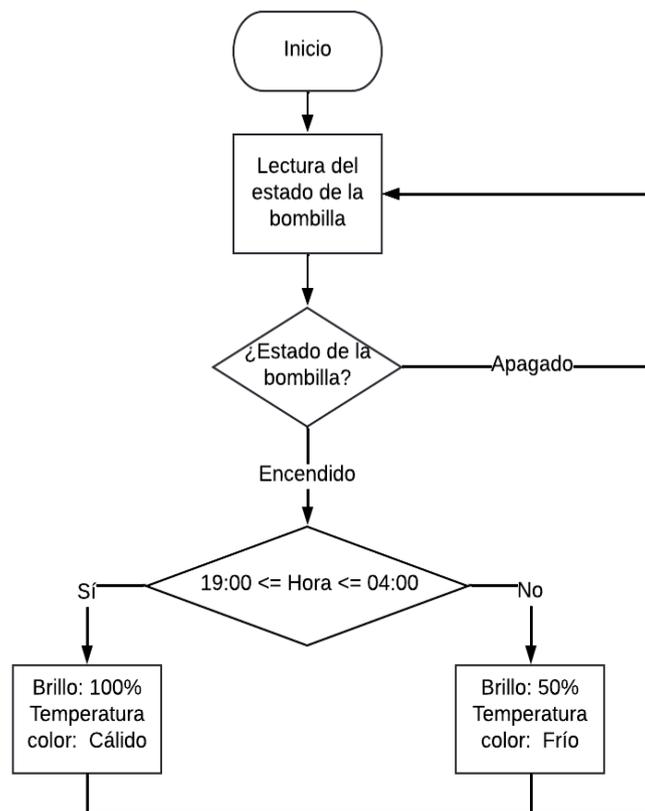
**Figura 3.37** Acción del apagado de luces automático

### Control del brillo de la luz

El control del brillo y temperatura de la luz es importante ya que según la cantidad de luz en el ambiente el brillo emitido por la bombilla complementa a la iluminación correcta [27]. Por otra parte, la temperatura de la luz tiene una influencia directa en el espacio, la luz fría es de color blanco azulado, mientras que la luz cálida se vuelve más amarilla. El uso de una luz amarillenta en la habitación crea un ambiente acogedor y relajante, mientras que la luz fría se usa en las horas del día en donde se crea un ambiente sobrio y abierto [28]. A partir del análisis anterior, se programó la rutina para que siempre y

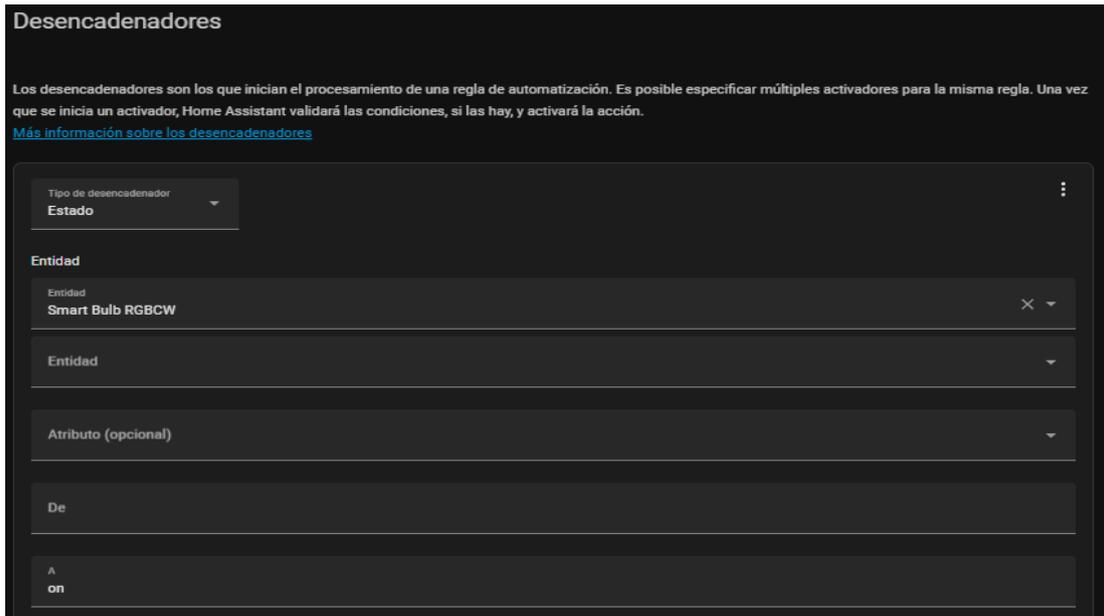
cuando se encuentre encendida la bombilla, se ajuste el brillo y temperatura de la luz de forma automática, con el fin de crear el ambiente óptimo según la hora del día. En el horario de 19:00 a 04:00 (h), que es cuando no hay la suficiente iluminación y el ambiente es frío, el brillo se configura al máximo, mientras que la temperatura de la luz se vuelve cálida, por otra parte, en el resto de las horas que la bombilla este activa el brillo se reduzca a la mitad y la temperatura sea fría al ser las horas más activas y con buena iluminación natural.

En el diagrama de flujo representado en la Figura 3.38, se especifica el funcionamiento del control del brillo y temperatura de la luz, en donde se observa que dichos parámetros se configuran según la hora específica del día y únicamente si la bombilla se encuentra encendida.



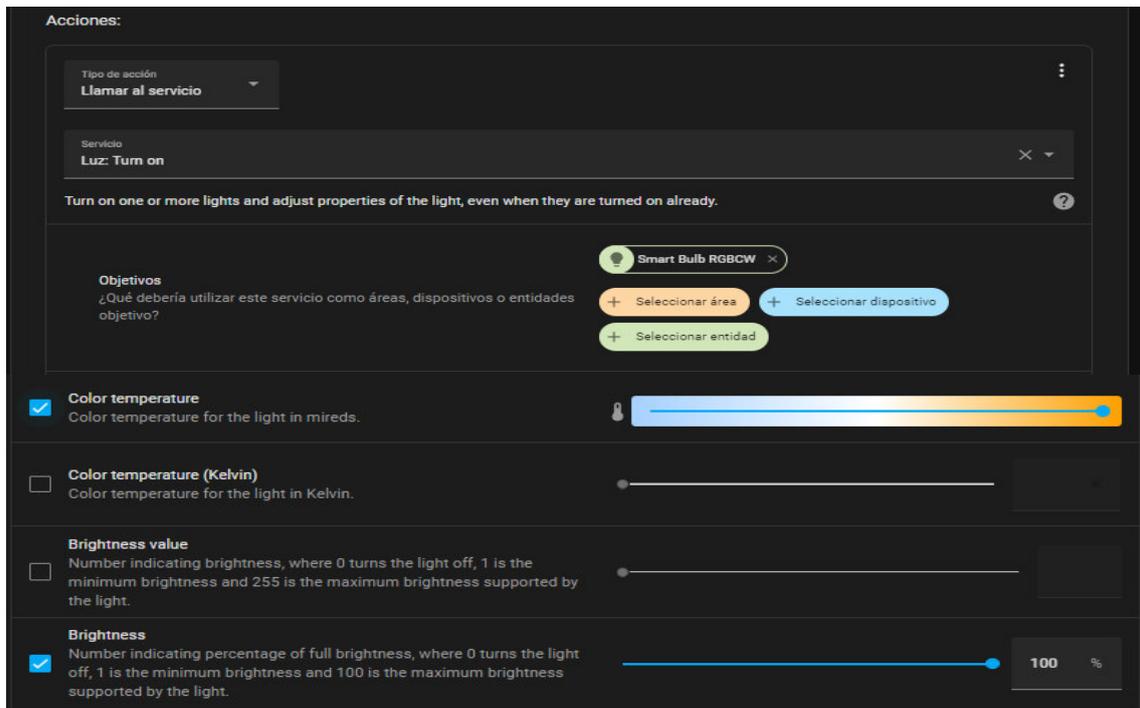
**Figura 3.38** Diagrama de flujo control del brillo de la luz

En el apartado de automatizaciones del *Home Assistant*, se creó la nueva tarea con el nombre de Brillo de luz, en donde se seleccionó el desencadenante que el estado de la bombilla se encuentre en *on*, como se muestra en la Figura 3.39.

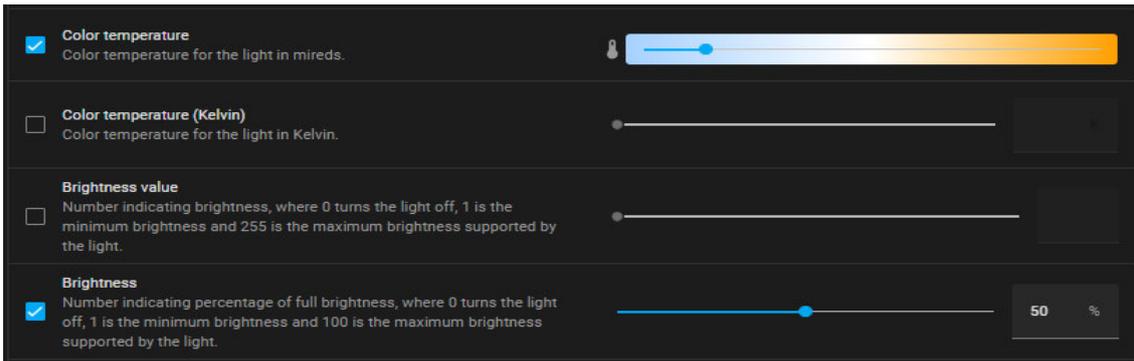


**Figura 3.39** Desencadenador de la automatización brillo de luz

Para esta automatización no se requiere ninguna condición, por lo cual se configuró dos opciones de acciones con su respectiva condición como indica la Figura 3.40, en donde la primera opción trabaja con la condición del horario entre las 19:00 a 04:00 (h), se seleccionó el dispositivo y se configuró que el brillo este al 100% y la temperatura en cálida. La segunda opción actúa en el resto de las horas y configura el brillo al 50% y la temperatura en fría, como muestra la Figura 3.41.



**Figura 3.40** Acción 1 de la automatización brillo de luz

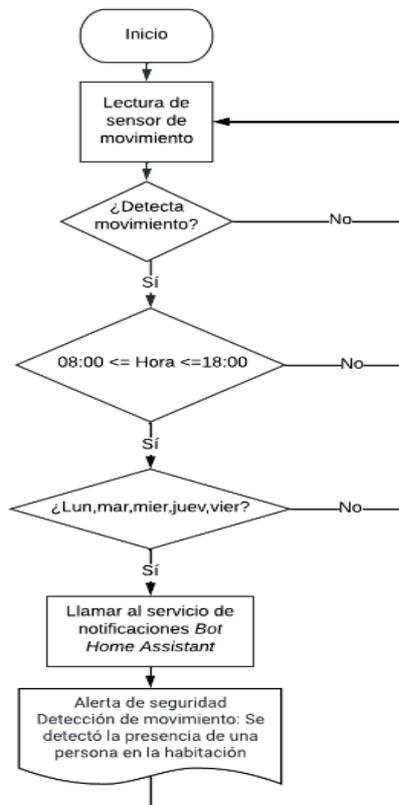


**Figura 3.41** Acción 2 de la automatización brillo de luz

### Detección de personas en la habitación

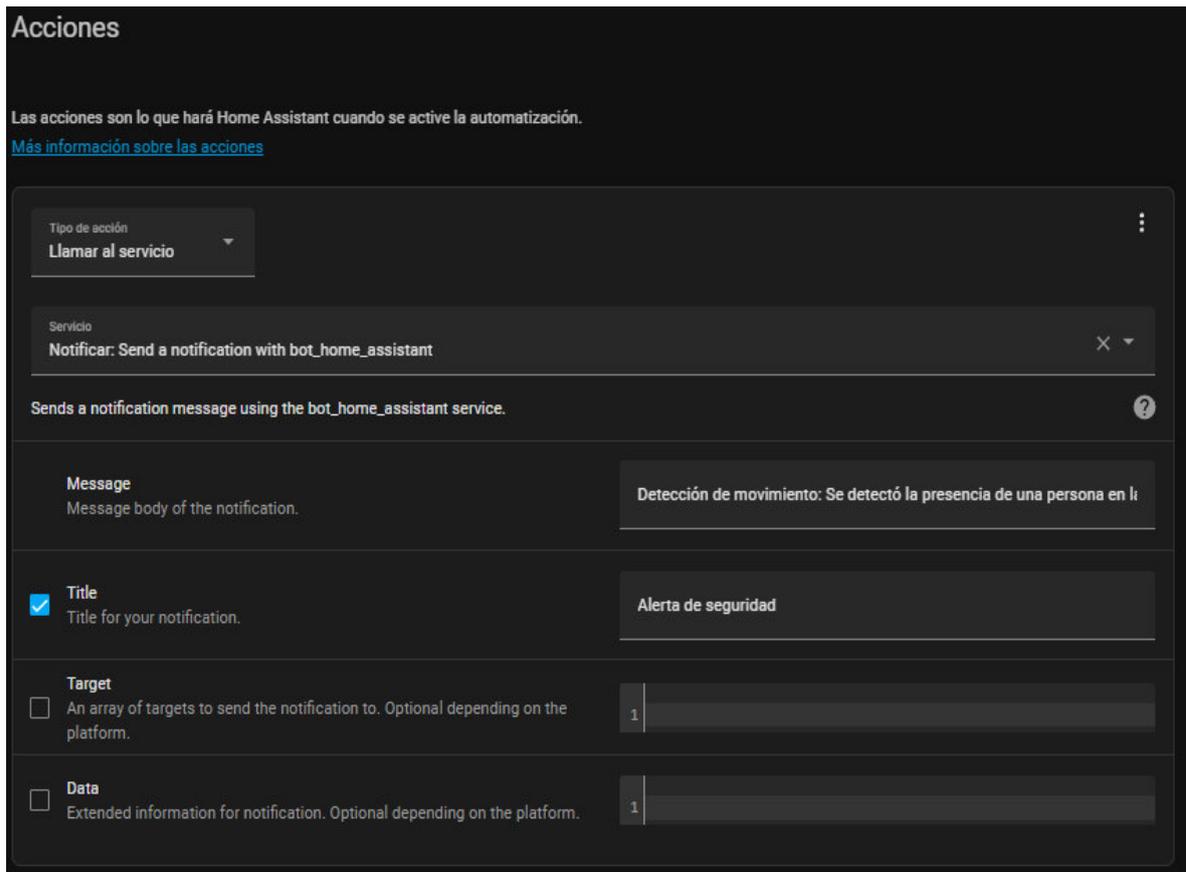
Esta automatización se programó por temas de seguridad y privacidad del usuario, en donde el sensor al detectar movimiento dentro del horario en que el usuario no se encuentra, envía una notificación al móvil vía Telegram, indicando que alguien ingresó a la habitación con un mensaje personalizado.

En la Figura 3.42 se muestra el diagrama de flujo detallando el funcionamiento de la automatización y muestra el mensaje que se envía vía Telegram al usuario cuando el sensor detecta movimiento entre las horas programadas.



**Figura 3.42** Diagrama de flujo de detección de personas

La tarea se nombró Detección Personas y como desencadenante se colocó que se inicie la lectura de la condición luego de que el sensor detecte movimiento, en el apartado de condiciones se configura las horas en las que el usuario no se encuentra en la habitación que son de 08:00 a 18:00 (h) exceptuando los fines de semana. Por último, en la Figura 3.43 se observa la acción que llama al servicio del envío de notificaciones a través del *Bot Home Assistant* previamente creado, además de configurar un título y mensaje personalizado que indica la presencia de alguien en la habitación.



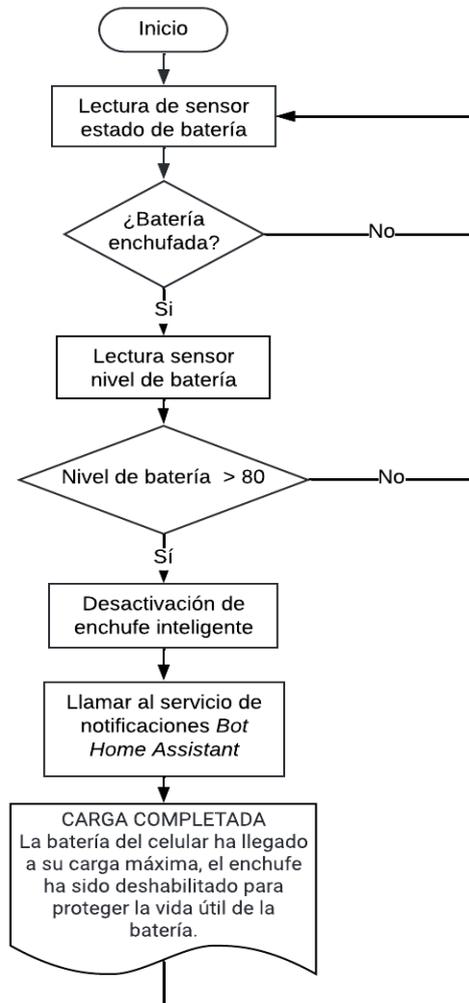
**Figura 3.43** Acción de la automatización detección de personas

### Control de carga inteligente

Uno de los puntos más importantes para el usuario a la hora de adquirir un dispositivo móvil es la duración de la batería, debido a esto la carga de la batería debe ser de forma correcta para salvaguardar la salud de esta. Una de las principales recomendaciones es evitar cargar la batería hasta su máximo porcentaje, mucho menos cuando el dispositivo se descargue completamente, el nivel óptimo de carga es mantenerlo entre el 20% y el 80% según sugerencia de los fabricantes como *Apple* y *Tesla*, con el propósito de extender la vida útil de la batería [29]. Es por esta razón que se crea un control de carga inteligente, en donde cada que el dispositivo este cargándose al llegar

a un nivel del 80% se bloquee el paso de corriente al apagar automáticamente el enchufe inteligente, y se envía una notificación al usuario de carga completada.

En el diagrama de flujo de la Figura 3.44 se detalla el funcionamiento de la automatización, en donde se realiza la lectura de los sensores de la batería para controlar el estado y nivel de carga, al llegar al porcentaje establecido el enchufe se desactiva y se envía el mensaje mostrado en el diagrama.



**Figura 3.44** Diagrama de flujo del control de carga

La tarea Control de carga hace uso de los sensores de la batería añadidos por el dispositivo móvil *MI 8 Lite*, para esto se usa el sensor del estado de la batería cuando está enchufado cargándose como desencadenador, como se observa en la Figura 3.45.

## Desencadenadores

Los desencadenadores son los que inician el procesamiento de una regla de automatización. Es posible especificar múltiples activadores para la misma regla. Una vez que se inicia un activador, Home Assistant validará las condiciones, si las hay, y activará la acción.

[Más información sobre los desencadenadores](#)

Tipo de desencadenador  
Dispositivo

Dispositivo  
MI 8 Lite

Desencadenar  
MI 8 Lite Is Charging enchufado

Duración (opcional)

hh	mm	ss
0	00	00

AGREGAR DESENCADENADOR

**Figura 3.45** Desencadenador del control de carga

En la Figura 3.46 se seleccionó el sensor del nivel de la batería en el apartado de condiciones y se detalla cuando se encuentre por encima de 80.

## Condiciones

Las condiciones son opcionales y evitarán una mayor ejecución a menos que se cumplan todas las condiciones.

[Más información sobre las condiciones](#)

Tipo de condición  
Estado numérico

PROBAR

Entidad\*  
MI 8 Lite Battery Level

Atributo

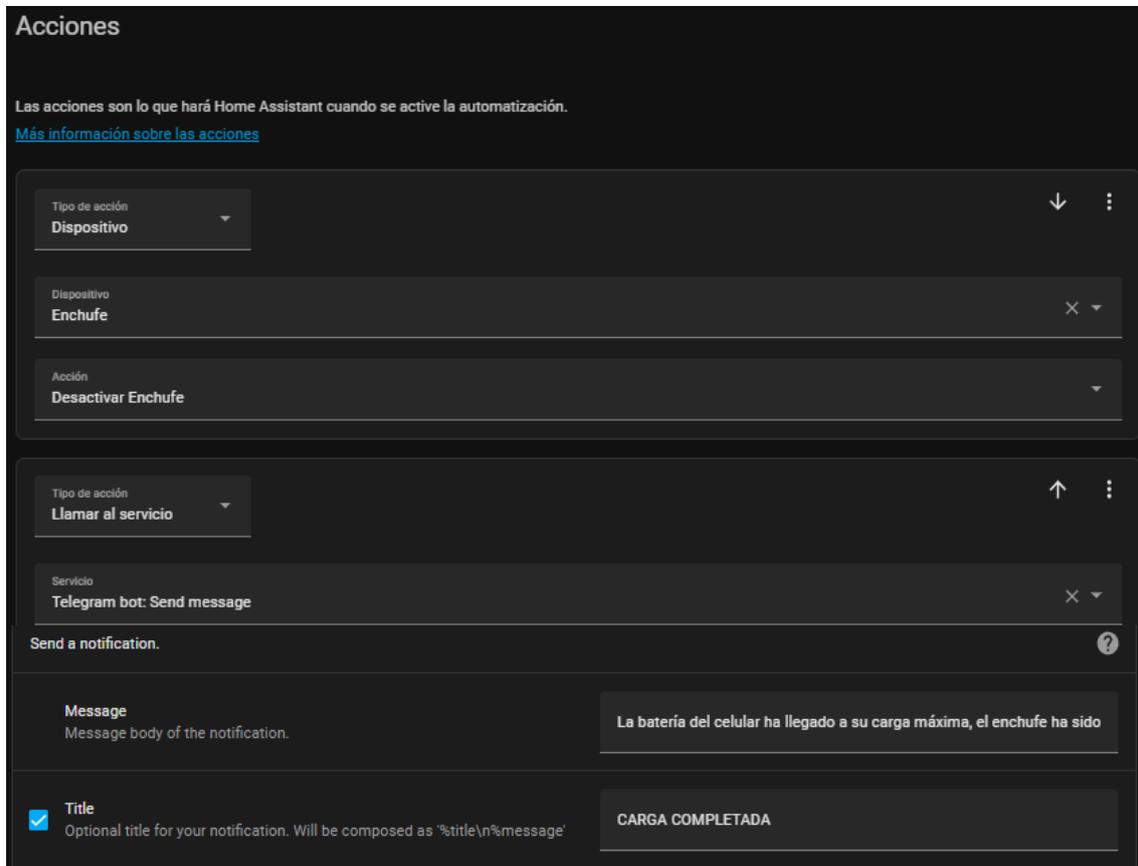
Por encima de  
80

Por debajo de

Plantilla de valor (opcional)

**Figura 3.46** Condición del control de carga

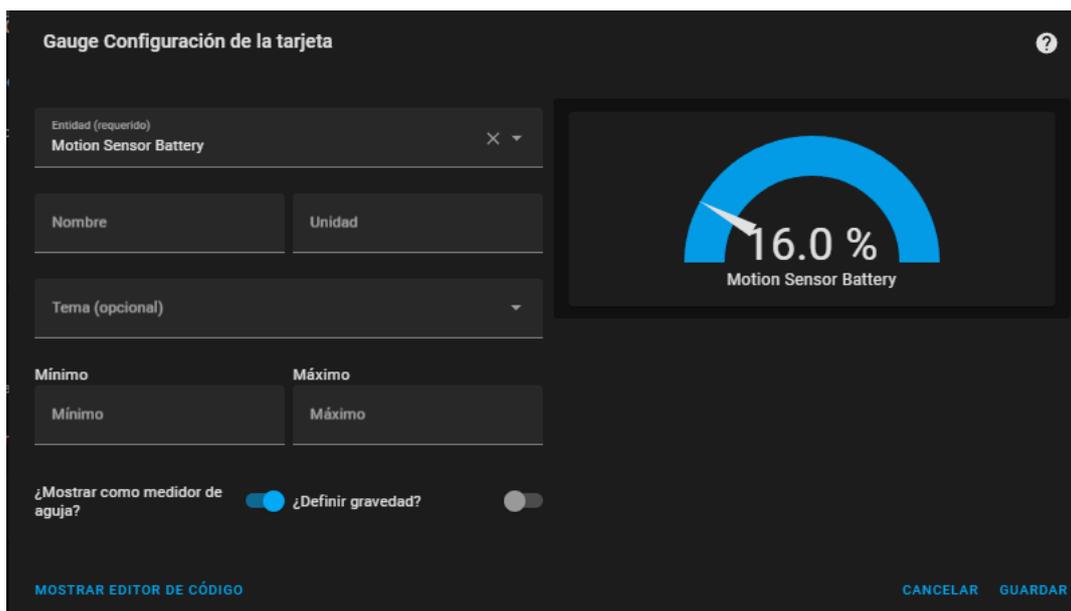
Por último, se visualiza en la Figura 3.47 el apartado de acciones se realiza la desactivación del enchufe y el envío de notificaciones a través del *Bot Home Assistant*, para indicar la finalización de carga con un mensaje personalizado.



**Figura 3.47** Acciones del control de carga

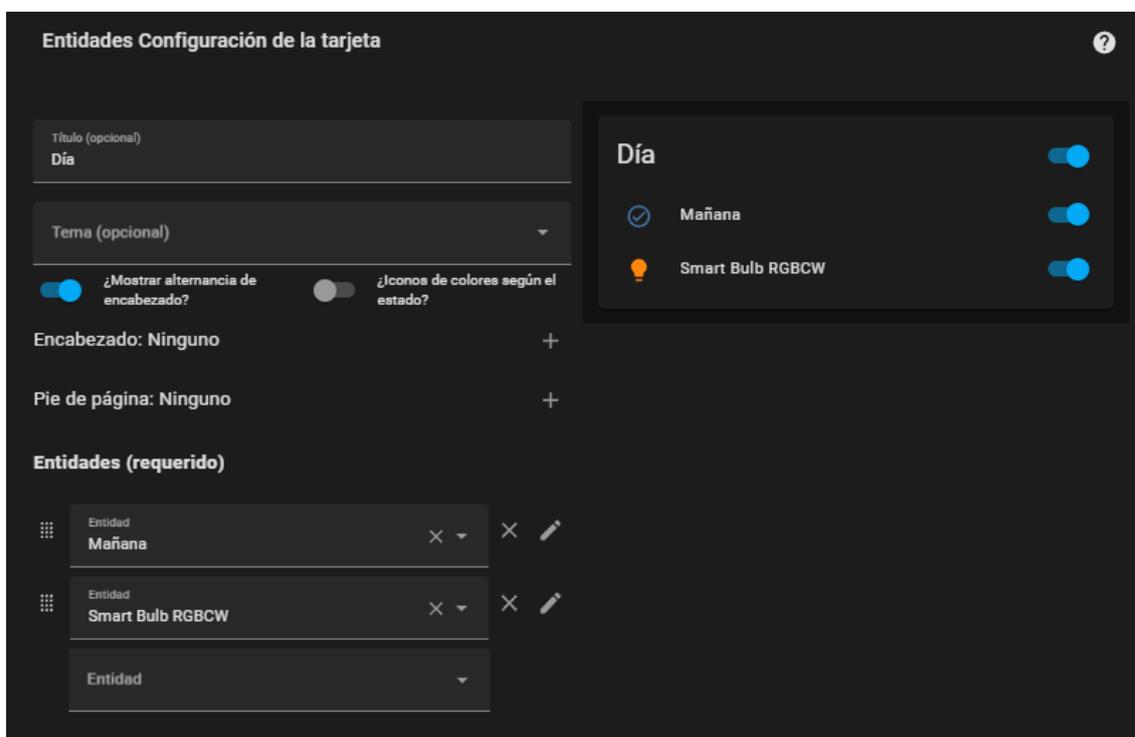
### **Diseño del tablero de resumen del *Home Assistant***

El tablero de resumen muestra los dispositivos, automatizaciones, ayudantes, clima, registros, entre otros; para lo cual se requiere personalizar su vista para tener un control más claro del sistema domótico, para lo cual se ingresa a la edición de la interfaz de usuario y se agrega tarjetas según lo que el usuario requiera visualizar. Un apartado importante es el estado de la batería del sensor de movimiento, para lo cual se selecciona la tarjeta de *Gauge* y se indica la entidad batería del sensor, como se observa en la Figura 3.48.

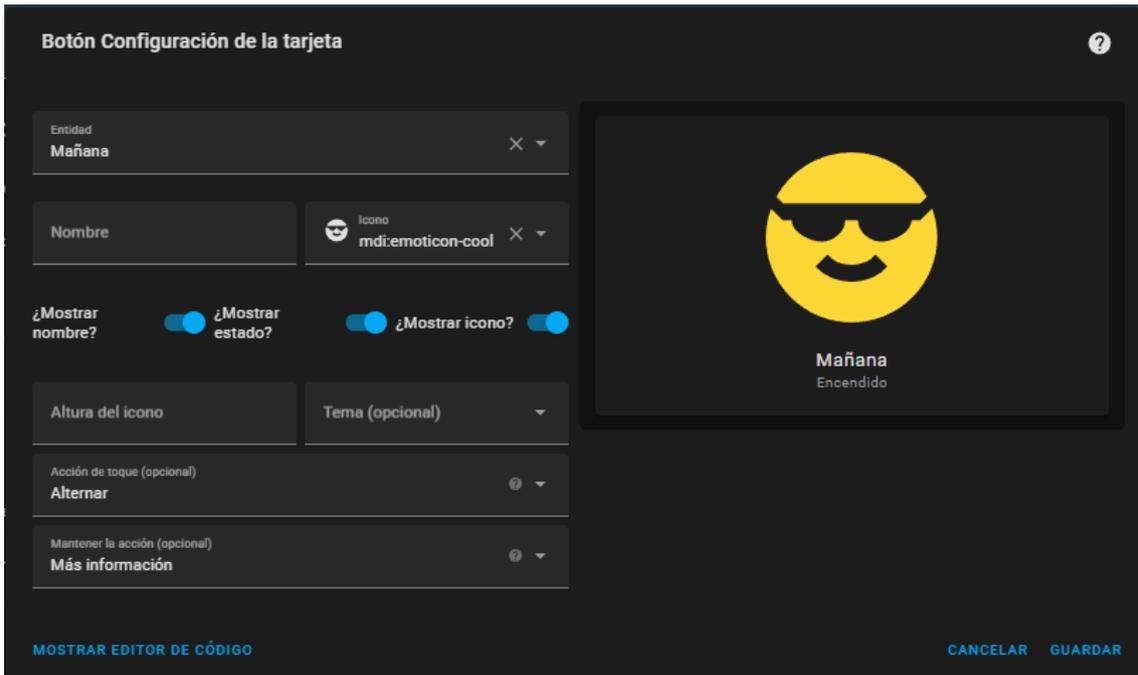


**Figura 3.48** Nivel de batería del sensor

En el tema de la iluminación se agregaron las tarjetas de entidades y luz, en la primera se colocó el nombre Día y se selecciona las entidades del ayudante Mañana y la bombilla inteligente, con el fin de visualizar si el encendido de luz automático se puede realizar, como indica la Figura 3.49, también en la Figura 3.50 se agrega la tarjeta botón en donde se selecciona el ayudante y un icono para activar siempre y cuando el usuario este despierto. La tarjeta luz permite controlar la bombilla.



**Figura 3.49** Tarjeta de entidades



**Figura 3.50** Tarjeta de ayudante Mañana

Por último, se usó la tarjeta de gráfico histórico y se seleccionaron los dispositivos inteligentes y las automatizaciones creadas, esta tarjeta muestra el registro de funcionamiento de las entidades seleccionadas en un intervalo de horas configurable, como se observa en la Figura 3.51.



**Figura 3.51** Registro de funcionamiento de las entidades

Además, se agregaron tarjetas para los sensores y el control del enchufe inteligente. En la Figura 3.52 se muestra el diseño del interfaz finalizado, en donde se controla y monitorea de manera fácil los dispositivos y automatizaciones del servidor *IoT*.



Figura 3.52 Tablero de resumen

### Instalación de InfluxDB para el monitoreo de los dispositivos

A partir de la selección del complemento InfluxDB para el sistema de monitoreo de los dispositivos, se realizó la instalación desde la tienda de complementos, una vez instalado se ingresó a la aplicación en la pestaña de administración y se creó una nueva base de datos, como se indica en la Figura 3.53.

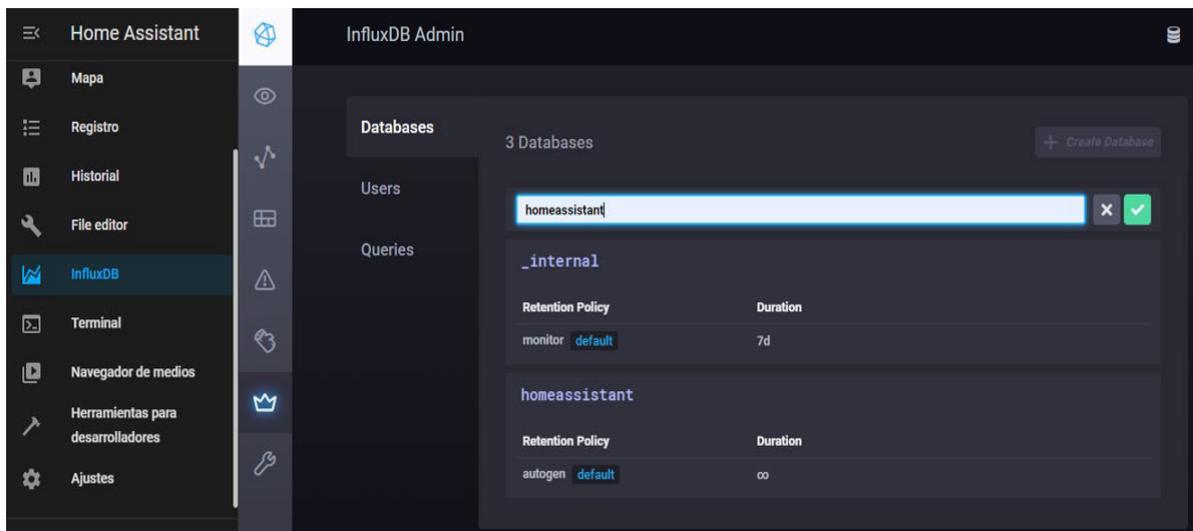
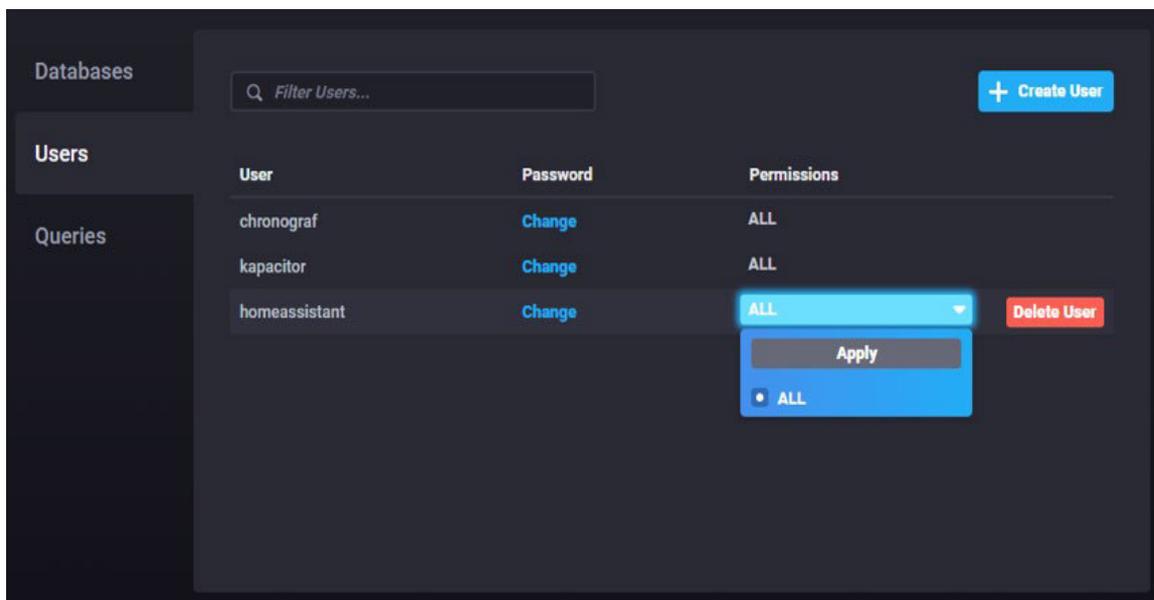


Figura 3.53 Creación de base de datos

En el apartado de usuarios se agregó un usuario, contraseña y todos los permisos para el manejo de la base de datos, como se indica en la Figura 3.54.



**Figura 3.54** Usuario de la base de datos

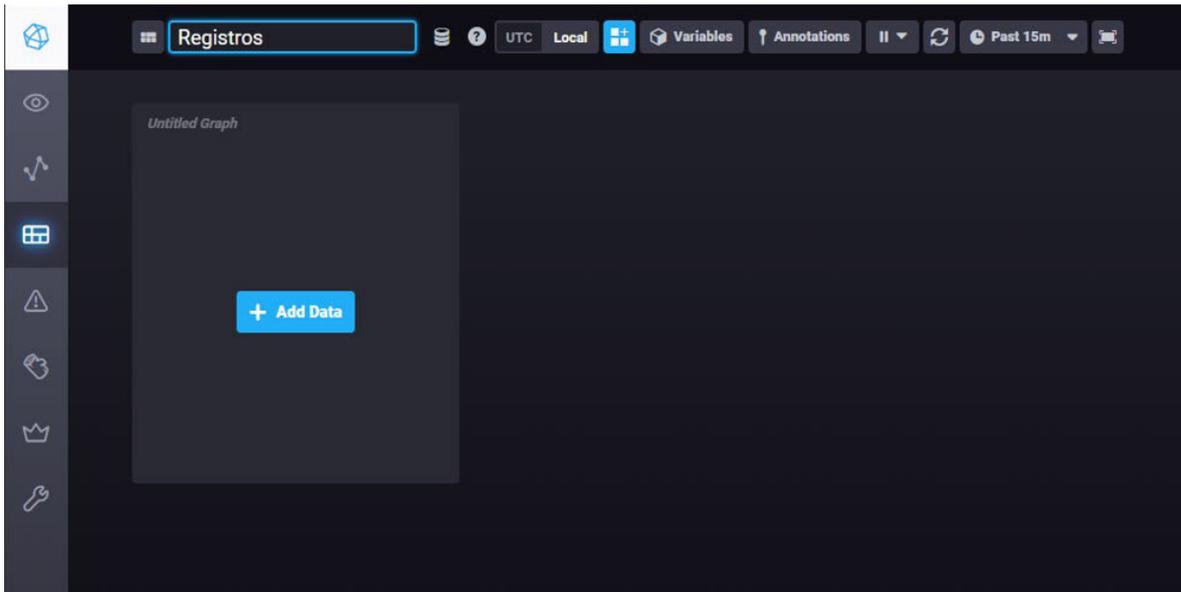
A continuación en la pestaña de Complementos – InfluxDB - Documentación se copió el código de la integración al archivo de *configuration.yaml*, en donde se edita la información pertenecientes a la base de datos creada, el usuario y contraseña, como se observa en la Figura 3.55.

```
13 influxdb:
14   - host: a0d7b954-influxdb
15   - port: 8086
16   - database: homeassistant #nombre de la base de datos creada
17   - username: homeassistant #nombre del usuario creado
18   - password: plgoryshit #contraseña del usuario creado
19   - max_retries: 3
20   - default_measurement: state
21
```

**Figura 3.55** Código de InfluxDB

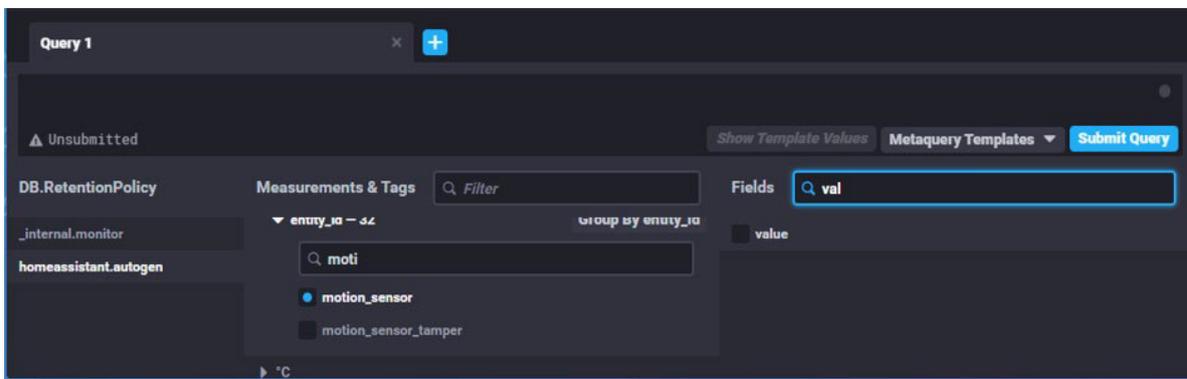
### **Creación de los tableros de monitoreo de los dispositivos en InfluxDB**

Una vez instalado y configurado el complemento, en la pestaña de Tableros se creó uno nuevo con el nombre de Registros y se dio clic en *Add Data* para generar un nuevo gráfico, como se observa en la Figura 3.56.



**Figura 3.56** Nuevo tablero

Dentro de la nueva gráfica se indicó la base de datos creada, y se buscó la entidad a monitorear y se selecciona el tipo de dato valor, además en la pestaña de visualización se elige un diagrama escalonado, como se indica en la Figura 3.57 y Figura 3.58.

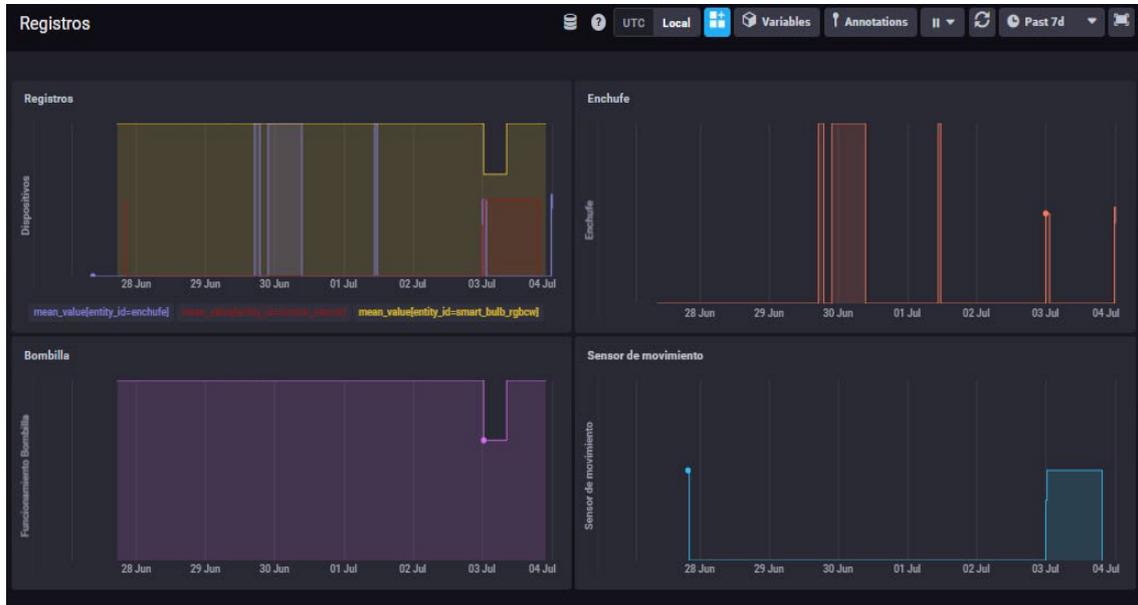


**Figura 3.57** Parámetros de la gráfica



**Figura 3.58** Visualización de la gráfica

Una vez realizado la gráfica para cada dispositivo, se puede visualizar el funcionamiento de los dispositivos mediante las gráficas del tablero configurando el tiempo de que se requiera, como se observa en la Figura 3.59.



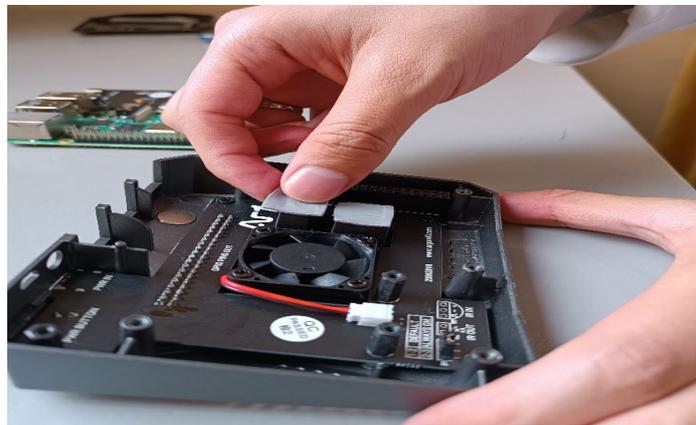
**Figura 3.59** Gráficas de funcionamiento de los dispositivos

### 3.4 Implementación del prototipo

La implementación del prototipo se realizó en una habitación como escenario real, para que el servidor se desarrolle en situaciones cotidianas y se tenga claro el funcionamiento y sus beneficios en el hogar.

#### Instalación de estuche de la *Raspberry Pi*

Se instalaron las gomas en las columnas disipadoras de calor, que transfieren la temperatura al estuche de aluminio, como muestra la Figura 3.60.



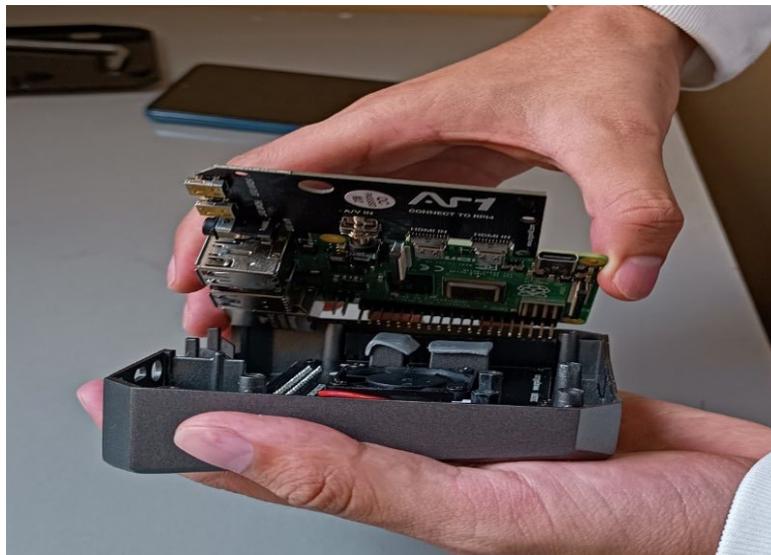
**Figura 3.60** Columnas de disipación de calor

Luego se colocó la placa PCB que adapta los puertos *USB*, *HDMI*, *Ethernet* a la carcasa y permite el encendido y apagado del servidor con un botón, como indica la Figura 3.61.



**Figura 3.61** Colocación placa *PCB*

A continuación, en la Figura 3.62 se ajustó la placa en el estuche conectando adecuadamente sus pines.



**Figura 3.62** Colocación del estuche

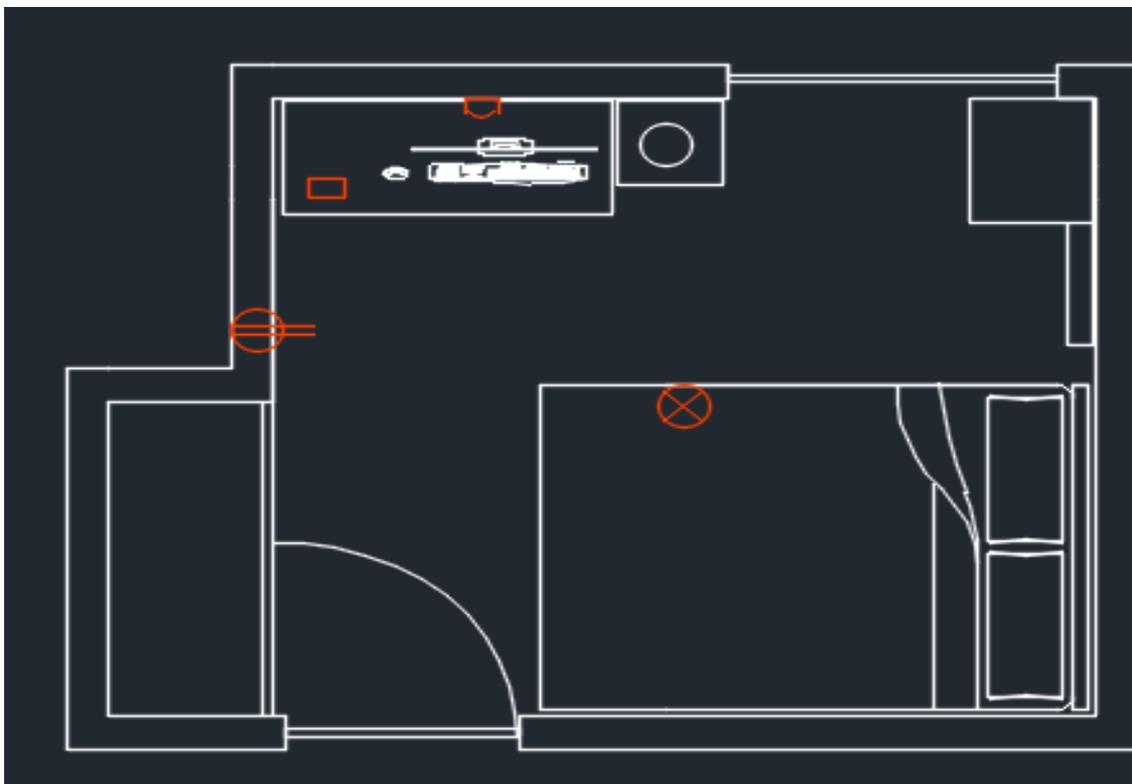
Por último, se colocó la tapa del estuche y se ajustó los tornillos para mantener la Raspberry Pi protegida y lista para su implementación, como se observa en la Figura 3.63.



**Figura 3.63** Estuche colocado correctamente

### **Distribución del prototipo en la habitación**

El prototipo consta del servidor *IoT* y los tres dispositivos inteligentes, para un mejor entendimiento de la instalación de los elementos, se diseñó un plano de la habitación real, en el que se muestra la ubicación de estos, como indica la Figura 3.64. En la Figura 3.65 se muestra la respectiva simbología para un mejor detalle.



**Figura 3.64** Plano de la habitación

	Bombilla LED
	Enchufe inteligente
	Sensor de movimiento
	Raspberry Pi

**Figura 3.65** Simbología del plano

### Instalación de los elementos del prototipo

La *Raspberry Pi* como servidor *IoT*, se instaló cerca del computador del usuario para su fácil acceso, monitoreo y manejo en caso de requerir reiniciarlo frente a fallas en el funcionamiento, como indica la Figura 3.66.



**Figura 3.66** Conexión Raspberry Pi

Para colocar el sensor de movimiento se utilizó cinta doble faz, como se observa en la Figura 3.67.



**Figura 3.67** Instalación del sensor de movimiento

En la Figura 3.68, se colocó el sensor apuntando a la entrada principal para detectar movimiento en toda la habitación y también en el momento exacto que ingrese cualquier persona.



**Figura 3.68** Colocación del detector de movimiento

Al ser una habitación, posee únicamente un punto de luz, en la Figura 3.69 se conectó la bombilla inteligente de forma tradicional.



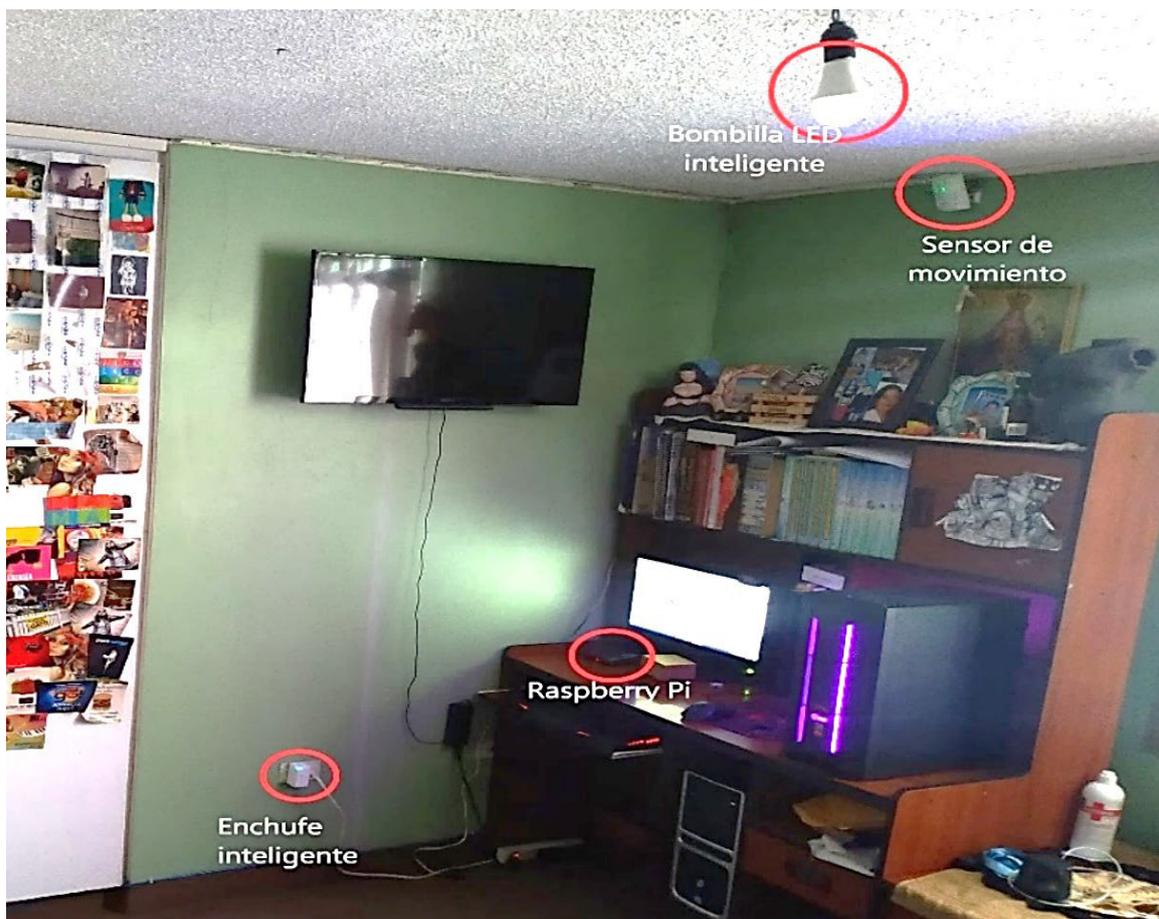
**Figura 3.69** Colocación de bombilla

El enchufe inteligente, se conectó al tomacorriente que se encuentra al lado del escritorio de trabajo, para facilitar la tarea de la carga del dispositivo móvil, como se observa en la Figura 3.70.



**Figura 3.70** Conexión de enchufe inteligente

Por último, en la Figura 3.71, se evidencia la habitación con los elementos mencionados anteriormente y su ubicación establecida por el plano de la Figura 3.64.



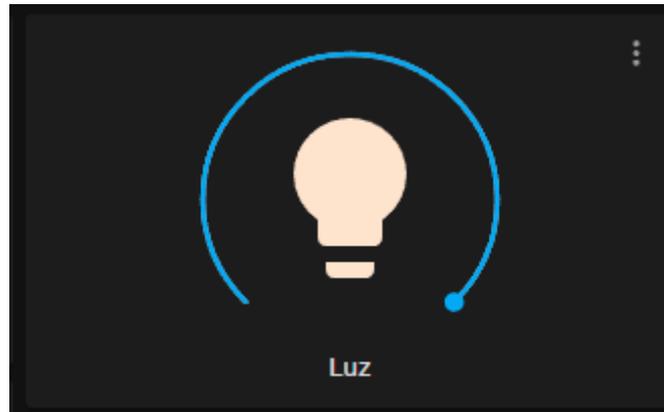
**Figura 3.71** Ubicación de los elementos

### 3.5 Realización de pruebas de funcionamiento del prototipo

Para verificar el correcto funcionamiento del prototipo se efectuaron las siguientes pruebas que demuestran la validez de este.

#### Pruebas de funcionamiento de dispositivos

Al ingresar a la interfaz web del *Home Assistant* se visualiza el tablero resumen, que permite el manejo y monitoreo de los dispositivos. En el control de la bombilla se oprime el botón de luz de la Figura 3.72 y en la Figura 3.73 se visualiza que el foco se enciende dependiendo del estado del botón.

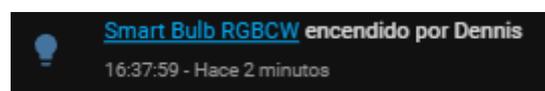


**Figura 3.72** Botón Luz encendido



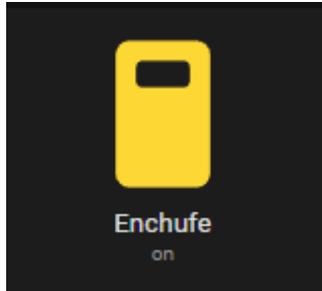
**Figura 3.73** Bombilla encendida

También, en el apartado de registros se comprueba que la bombilla fue encendida por el usuario, como muestra la Figura 3.74.



**Figura 3.74** Registro encendido de bombilla

El enchufe inteligente posee un botón que enciende o apaga el dispositivo según se requiera, en la Figura 3.75 se observa que se encuentra activado mientras que en la Figura 3.76 se presenta encendido.

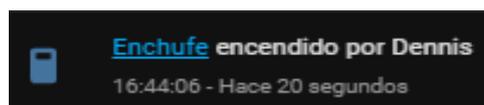


**Figura 3.75** Botón Enchufe encendido



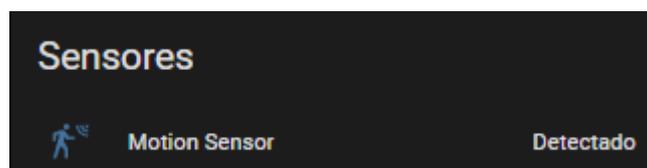
**Figura 3.76** Enchufe inteligente encendido

En la Figura 3.77 se muestra que el enchufe fue encendido por el usuario.

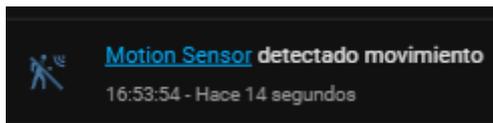


**Figura 3.77** Registro de enchufe encendido

Para verificar el correcto funcionamiento del sensor de movimiento al detectar movimiento el estado se muestra en detectado, como se observa en la Figura 3.78, y se muestra en el registro la hora que detectó movimiento en la habitación como muestra la Figura 3.79.



**Figura 3.78** Estado del sensor



**Figura 3.79** Registro del sensor

### **Pruebas de rutinas de automatización**

A continuación, se comprobó que las cinco rutinas creadas funcionan correctamente.

#### **Iluminación-Habitación**

En la Figura 3.80 se muestra que cuando el sensor detecta movimiento se realiza el encendido de luces, como se observa en la Figura 3.81.

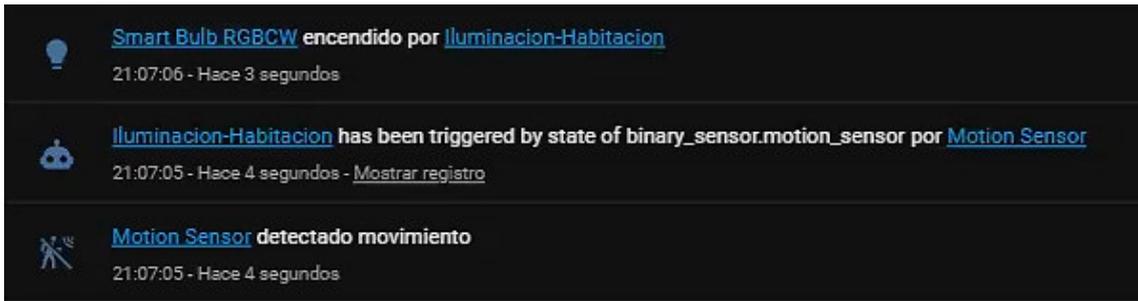


**Figura 3.80** Sensor de movimiento activado



**Figura 3.81** Bombilla encendida por Iluminación-Habitación

En el apartado de registros se visualiza que una vez que el sensor detecta movimiento se ejecuta la automatización Iluminación-Habitación por el detector y la bombilla se enciende, adicionalmente indica la hora en la que se disparó la tarea cumpliendo con las horas configuradas, como muestra la Figura 3.82.



**Figura 3.82** Iluminación-Habitación

### Apagado-Habitación

Para el apagado de luces, en la Figura 3.83 se muestra el registro en donde luego de 30 (min) de encender la bombilla mediante el sensor, el dispositivo dejó de detectar movimiento y se dispara la tarea Apagado-Habitación para apagar el foco, como muestra la Figura 3.84.



**Figura 3.83** Registro Apagado-Habitación



**Figura 3.84** Apagado de bombilla

## Brillo de luz

En la tarea brillo de luz de la Figura 3.85 se muestra que la tarea se inició por el estado de la bombilla en encendido y dependiendo la hora se modifica el brillo y temperatura de la luz. La rutina modifica el brillo y la temperatura de la luz de forma automática dependiendo la hora del día, en el horario de las 18:00 (h), la luz se configura con un brillo medio y temperatura fría, como se muestra en la Figura 3.86 la barra azul muestra el brillo de la bombilla y se encuentra en la mitad y la temperatura de la luz del foco es fría, visible en la Figura 3.87.

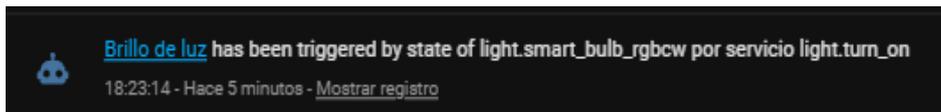


Figura 3.85 Registro brillo de luz

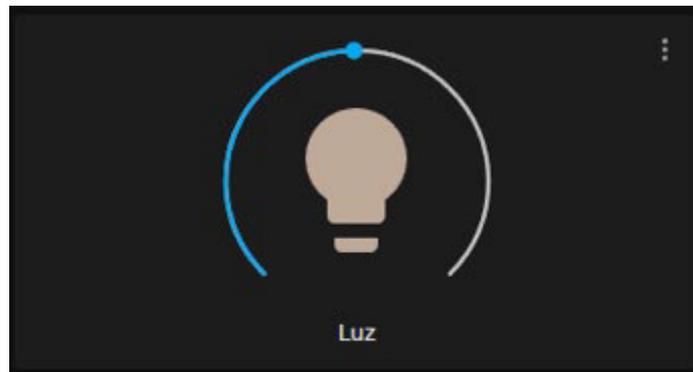
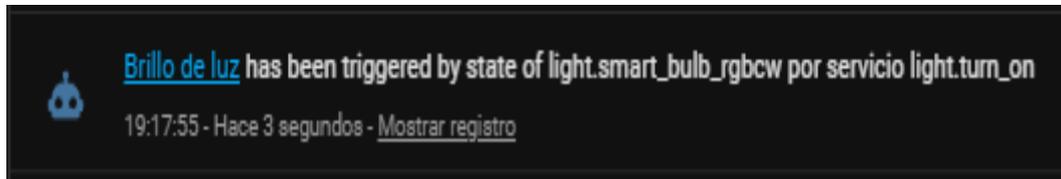


Figura 3.86 Estado de la bombilla

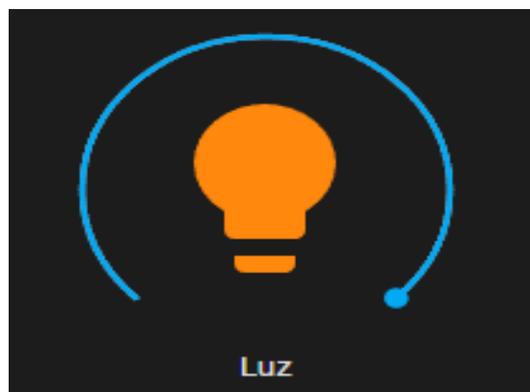


Figura 3.87 Bombilla ajustada el brillo y temperatura de la luz

En el otro horario de las 19:00 (h), se muestra que se ejecutó la tarea brillo de luz en los registros de la Figura 3.88, según la hora se ajusta el brillo al máximo y la temperatura de la luz es cálida, como se observa en el botón de la Figura 3.89 y la bombilla en la Figura 3.90.



**Figura 3.88** Registro brillo de luz



**Figura 3.89** Estado de la bombilla

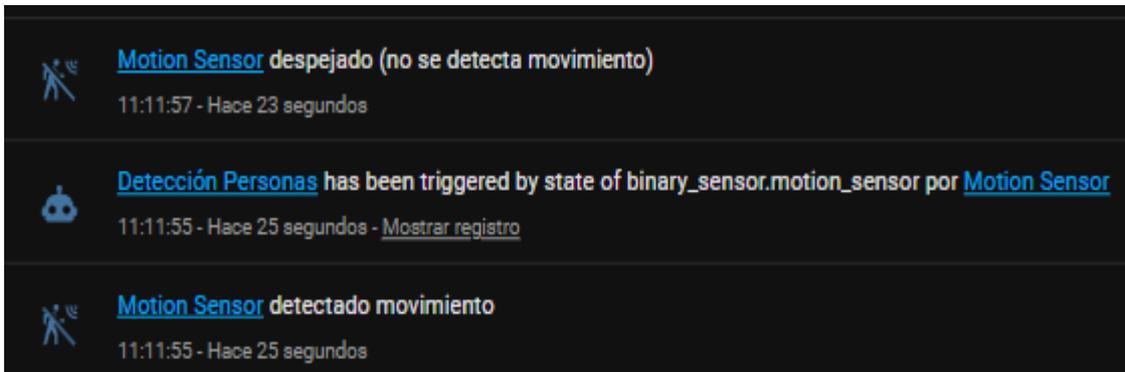


**Figura 3.90** Bombilla ajustada el brillo y temperatura de la luz

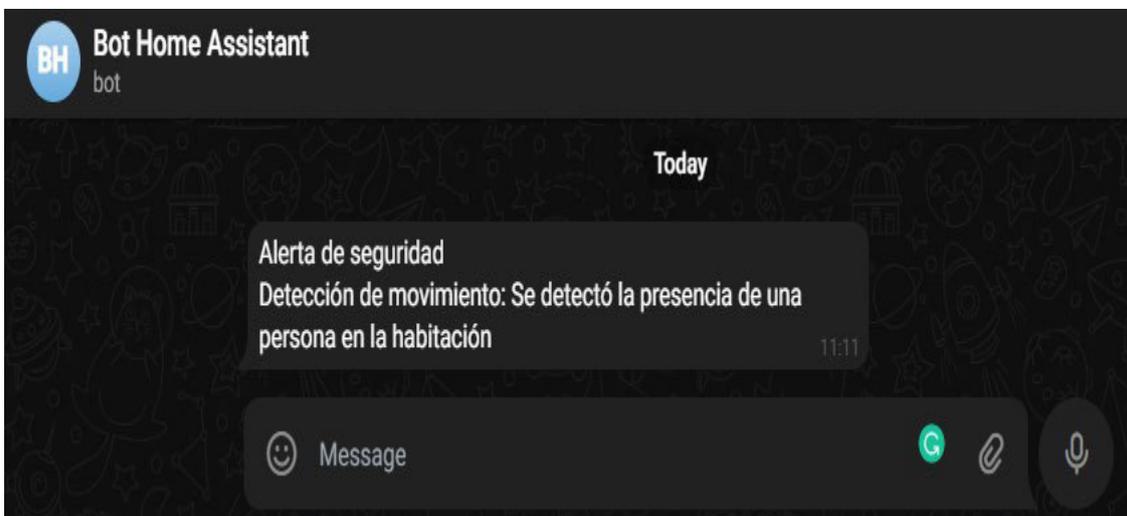
### **Detección Personas**

Para la comprobación de esta tarea, se muestra los registros en la Figura 3.91, donde se observa que luego de detectar movimiento en la habitación dentro de las horas configuradas, se disparó la rutina Detección Personas por el cambio de estado del

sensor. En la Figura 3.92 se muestra la notificación enviada a través del *Bot Home Assistant* en Telegram a la misma hora que se ejecutó la automatización.



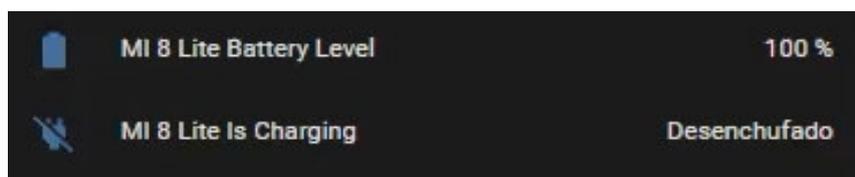
**Figura 3.91** Registro Detección Personas



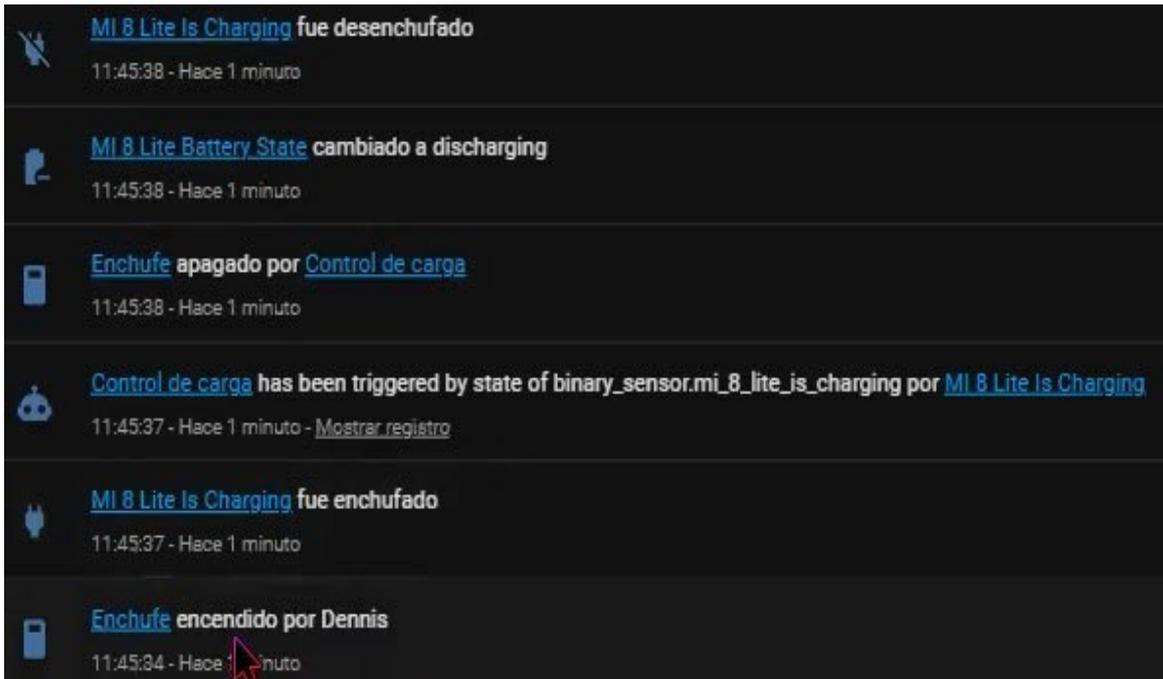
**Figura 3.92** Notificación vía Telegram

### Control de carga

La carga inteligente, en la Figura 3.93 se visualiza el estado de los sensores de la batería en desconectado y carga del 100%, por lo cual al momento de encender el enchufe inteligente se apaga instantáneamente al cumplir con el porcentaje de carga configurado. En la Figura 3.94 se muestran los registros y se indica que luego de activar el dispositivo, el celular comienza a cargar y se dispara la tarea Control de carga apagando el enchufe y cambiando el estado del móvil a desenchufado.



**Figura 3.93** Estado de los sensores de batería

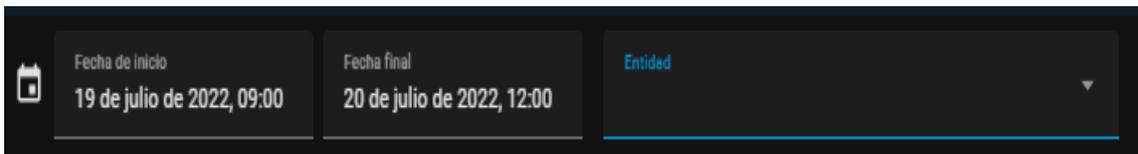


**Figura 3.94** Registro de Control de carga

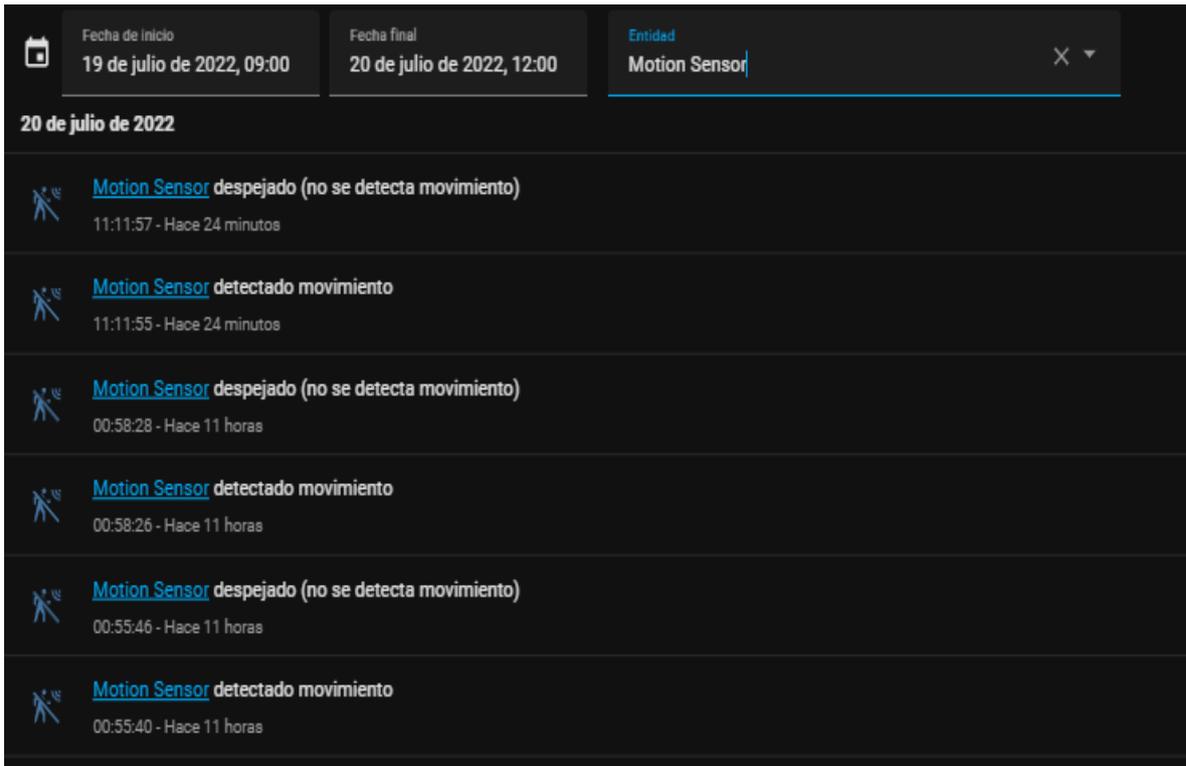
### Verificación de los registros de funcionamiento de los dispositivos

Los registros de funcionamiento se muestran en las pestañas de registros, historial e InfluxDB.

Los registros se pueden configurar el rango de fechas u horas que se necesita visualizar al igual que el dispositivo o la rutina existente, como indica la Figura 3.95. En la Figura 3.96 se seleccionó el detector de movimiento y se muestran únicamente los registros de esta entidad.

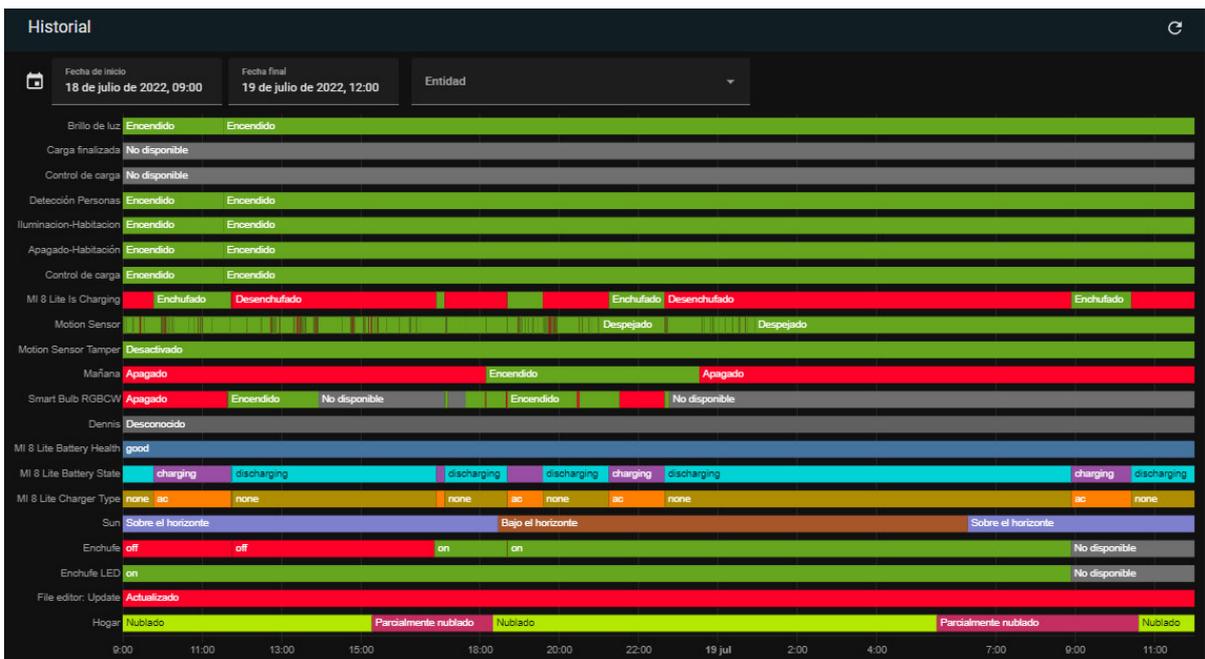


**Figura 3.95** Configuración de registros



**Figura 3.96** Registros del detector de movimiento

En el apartado de historial, se visualiza el estado de encendido o apagado de los dispositivos inteligentes y automatizaciones, también se puede configurar la visualización de la entidad que se requiera en cierta fecha, como se muestra en la Figura 3.97. En la Figura 3.98 se indica el historial de la bombilla inteligente.

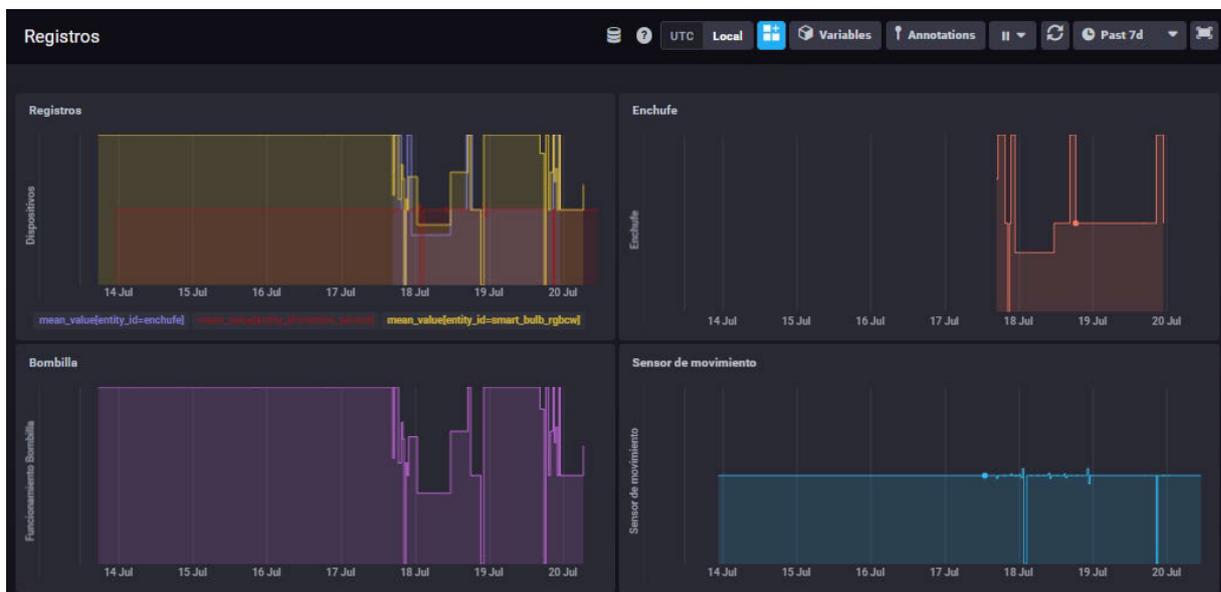


**Figura 3.97** Historial de entidades



**Figura 3.98** Historial de la bombilla

Por último, en InfluxDB se tiene un diagrama escalonado de los dispositivos instalados, en donde se representa el funcionamiento de cada uno y se puede configurar el tiempo de los registros, como indica la Figura 3.99.



**Figura 3.99** Diagramas escalonados de los dispositivos

### **Demostración del funcionamiento del prototipo**

En el ANEXO II se muestra el código QR que contiene el enlace del video demostrativo del funcionamiento del prototipo.

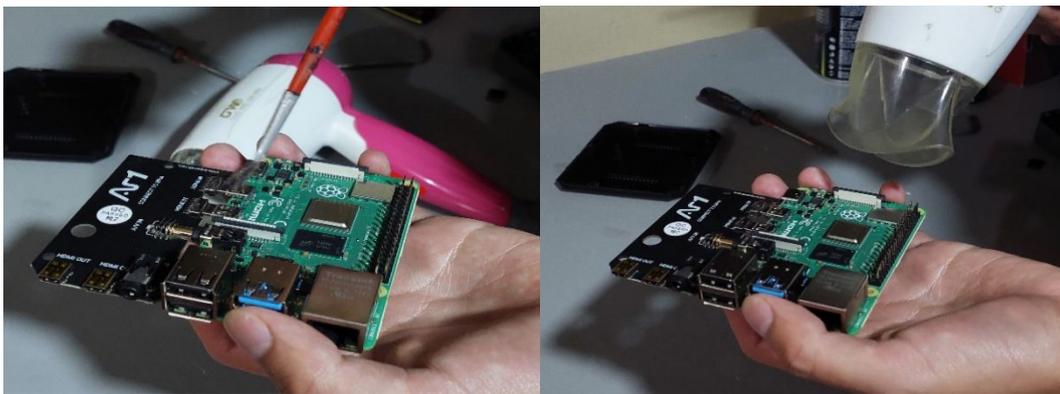
### **Mantenimiento del prototipo**

El mantenimiento del servidor *IoT* es importante para garantizar el correcto funcionamiento de este, se recomienda realizarlo cada cierto tiempo, según el lugar en el que se encuentre, en este caso la *Raspberry Pi* se encuentra en un lugar sin suciedad, por lo cual se lo hará cada 3 meses, para la limpieza se hace uso de una brocha y un secador de aire frío. Para iniciar se debe apagar el microordenador y desconectar de la fuente de alimentación, posteriormente en la Figura 3.100 se procede a abrir su estuche y sacar sus piezas para una limpieza más sencilla.



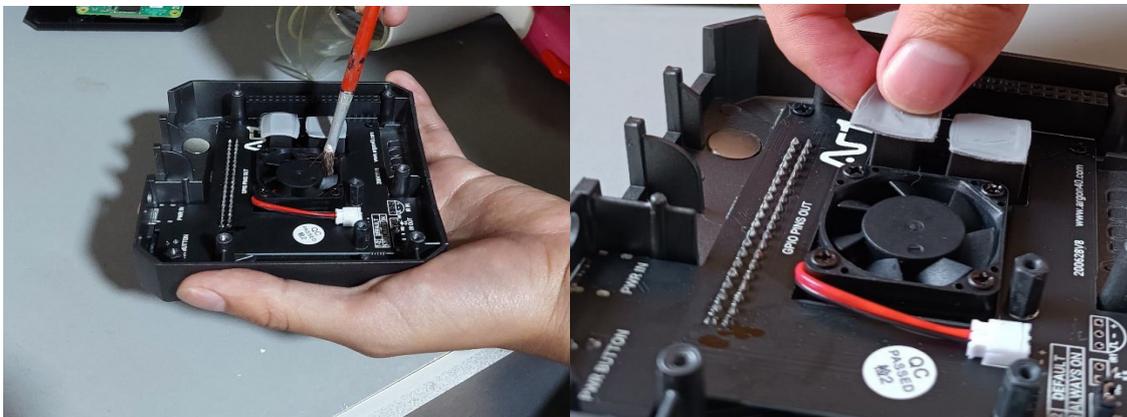
**Figura 3.100** Raspberry Pi

Con ayuda de una brocha y la secadora se descarga aire sobre los componentes y se limpia con la brocha el polvo existente sobre estos, de la misma manera se lo realiza a los puertos de la placa y las ranuras de ventilación del estuche, como se observa en la Figura 3.101.



**Figura 3.101** Limpieza componentes

A continuación, se limpia el ventilador y se reemplaza la pegatina térmica para que su sistema de enfriamiento opere correctamente, como se indica en la Figura 3.102.



**Figura 3.102** Limpieza sistema de enfriamiento

Por último, se coloca el limpia contactos en la placa y los puertos de esta, como se indica en la Figura 3.103.



**Figura 3.103** Colocación de limpia contactos

### Costo del prototipo

En la Tabla 3.2 se detalla los costos de los materiales empleados para el desarrollo e implementación del prototipo, estos son los precios de adquisición vía Mercado Libre.

**Tabla 3.2** Tabla de costos

Material	Cantidad	Precio Unitario	Precio Total
<i>Raspberry Pi</i> 4B/8 (GB)	1 (u)	\$ 150.00	\$ 150.00
Estuche <i>Argon One</i>	1 (u)	\$ 25.00	\$ 25.00
Fuente de alimentación	1 (u)	\$ 10.00	\$ 10.00
MicroSD <i>Kingston</i> 32 (GB)	1 (u)	\$ 15.00	\$ 15.00
Detector de movimiento CT60W	1 (u)	\$ 20.00	\$ 20.00
Bombilla inteligente A60	1 (u)	\$ 10.00	\$ 10.00
Enchufe inteligente HS103	1 (u)	\$ 12.00	\$ 12.00
Pilas AAA	2 (u)	\$ 2.00	\$ 4.00
Cinta doble faz	1 (u)	\$ 2.00	\$ 2.00
Mano de obra	20 (h)	\$ 10.00	\$ 200.00
<b>Total</b>			<b>\$ 448.00</b>

## 4 CONCLUSIONES

- Se concluyó que para el diseño del prototipo uno de los requerimientos más importantes para la automatización de una casa es el uso de una plataforma domótica de código abierto, con el fin de eliminar brechas de seguridad en la vivienda y permitir el control total de los dispositivos por parte del usuario. Además, el software libre brinda ventajas de compatibilidad con varios dispositivos y soporte por parte de la comunidad, facilitando la configuración del servidor sin ser un experto.
- Para la selección de hardware y software una vez analizado los requerimientos establecidos, se llegó a la conclusión de que la *Raspberry Pi* en conjunto con el sistema operativo *Home Assistant* cumplen con la principal función del servidor de convertir una casa en inteligente, la versión del microordenador 4B/8 (GB) tiene las características adecuadas y permite una expansión a futuro del sistema domótico configurando más dispositivos inteligentes.
- El diseño del prototipo centraliza el control de los dispositivos inteligentes desde un solo lugar, en donde las rutinas de automatización dependen de los sensores y actuadores seleccionados para cumplir una tarea específica en el hogar, con el sensor de movimiento, la bombilla y el enchufe inteligente se crearon tareas que otorguen iluminación automática, seguridad y ahorro energético al hogar, además del envío de notificaciones al usuario vía Telegram para el monitoreo constante del lugar.
- Para la selección de dispositivos inteligentes fue necesario revisar las integraciones compatibles con el sistema operativo para que estos puedan ser configurados en la plataforma y controlados desde ahí, además deben trabajar mediante *Wi-Fi* para que se puedan intercomunicar entre sí y programar tareas complementarias entre estos.
- Se concluyó que *Home Assistant* permite la creación de automatizaciones mediante código YAML o interfaz gráfica para usuarios menos expertos, las tareas se realizan mediante desencadenadores, condiciones y acciones, en donde al leerse el estado de una entidad se desencadena la lectura de la condición que al cumplirse dispara la acción configurada.
- En la implementación del prototipo fue necesario el uso de un estuche para la protección de la *Raspberry Pi* contra agentes externos y temperaturas altas, debido a que pueden irrumpir en el correcto funcionamiento del servidor al estar en constante uso y monitoreo del hogar.

- *Home Assistant* cuenta con su propio sistema de registros de funcionamiento de los dispositivos y automatizaciones, pero para una visualización gráfica de los datos de estos, se eligió el complemento InfluxDB que muestra la activación y desactivación de los nodos finales en cierto periodo de tiempo. Con esto se cumplió con el alcance del proyecto de monitorear la labor de cada dispositivo y corregir errores. Además, se utilizaron los *logs* para la realización de las pruebas y verificar la correcta ejecución de las rutinas creadas.
- Es importante configurar el servidor IoT con una dirección IP estática para brindar una conexión estable y evitar que el router asigne una dirección diferente ocasionando fallos en las configuraciones de los dispositivos inteligentes.

## 5 RECOMENDACIONES

- Antes de agregar un dispositivo al servidor, se recomienda verificar el apartado de integraciones de *Home Assistant* para su correcta configuración debido a que cada marca requiere procesos diferentes y así adquirir una compatible con el sistema.
- Se recomienda leer la documentación de *Home Assistant* para mejorar el sistema domótico, debido a que posee una extensa tienda de complementos entre los cuales se pueden controlar el consumo energético, realizar planos de toda la vivienda para controlar todos los dispositivos configurados con un solo botón, controlar bases de datos, realizar copias de seguridad, entre otros.
- *Home Assistant* monitorea el clima según la configuración horaria, se recomienda hacer uso de esta función para la creación de tareas en donde se programe su ejecución en conjunto con la posición del sol y facilitar así las configuraciones.
- Se recomienda revisar en InfluxDB las gráficas escalonadas de los registros de los dispositivos para observar que dispositivo tiene mayor uso y que automatizaciones son eficientes para el hogar.
- La aplicación móvil de *Home Assistant* proporciona el mismo control de la plataforma a través de la interfaz web, por lo cual se recomienda al usuario hacer uso de esta para mantener el hogar monitoreado desde cualquier lugar usando únicamente su celular.

- Para el uso de la aplicación celular, se recomienda dar permisos al dispositivo móvil para correr *Home Assistant* en segundo plano y mantener una lectura constante de los sensores y automatizaciones.
- Si se desea actualizar el sistema domótico a futuro, se recomienda migrar el sistema de la microSD a un SSD para evitar errores al manejar bases de datos pesadas y aumentar la vida útil del servidor.
- Es importante poseer una conexión a internet que sea estable y rápida, tanto para el servidor como para los dispositivos inteligentes, ya que esto influye en la rapidez con la que se ejecutan las rutinas de automatización.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] «¿Qué es una casa inteligente?,» BBVA, 20 Septiembre 2020. [En línea]. Available: <https://www.bbva.mx/educacion-financiera/blog/que-es-una-casa-inteligente.html>. [Último acceso: 15 Mayo 2022].
- [2] Euroinnova, «Que es una casa inteligente y sus características,» s.f, [En línea]. Available: <https://www.euroinnova.ec/que-es-una-casa-inteligente-y-sus-caracteristicas>. [Último acceso: 15 Mayo 2022].
- [3] R. Solé, «Raspberry Pi: Crea proyectos DIY por muy poco dinero,» Profesional Review, 18 Julio 2021. [En línea]. Available: [https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/#Que\\_es\\_Raspberry\\_Pi](https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/#Que_es_Raspberry_Pi). [Último acceso: 15 Mayo 2022].
- [4] MCI Electronics, «Raspberry Pi 4 Modelo B,» 2022. [En línea]. Available: <https://www.mcielectronics.cl/shop/product/raspberry-pi-4-modelo-b-8gb-ram-raspberry-pi-28296?category=370>. [Último acceso: 23 Mayo 2022].
- [5] D. Perez, «¿Qué es Home Assistant?,» Tecnonucleous, 19 Julio 2018. [En línea]. Available: <https://tecnonucleous.com/2018/07/19/que-es-home-assistant/>. [Último acceso: 15 Mayo 2022 ].

- [6] Gasensio., «Hogar Digital (SIHD): Elementos de control,» Micronica, 10 Noviembre 2018. [En línea]. Available: <http://www.micronica.es/?s=sihd>. [Último acceso: 25 Junio 2022].
- [7] Aragonesa, «Elementos de la domótica,» Aragonesa de PVC, 2022. [En línea]. Available: <https://bit.ly/3NoAD6Q>. [Último acceso: 25 Junio 2022].
- [8] AAI Security Systems, «¿Qué es un sensor PIR y cómo funciona?,» 24 Marzo 2015. [En línea]. Available: <https://www.aaisecurity.co.uk/news/pir-sensor/>. [Último acceso: 23 Mayo 2022].
- [9] TVCMALL, «CT60W Smart WiFi Detectores Infrarrojos,» 2022. [En línea]. Available: <https://bit.ly/3LEYtue>. [Último acceso: 23 Mayo 2022].
- [10] M. Valle, «Focos inteligentes,» CNN, 20 Agosto 2021. [En línea]. Available: <https://cnn.it/3wFoZPR>. [Último acceso: 23 Mayo 2022].
- [11] Frecuento.com, «Foco Led Smart A60,» 2022. [En línea]. Available: <https://bit.ly/3Gc6V2G>. [Último acceso: 23 Mayo 2022].
- [12] TP-Link, «Enchufe Inteligente Wi-fi Tp-link Hs103,» TP-Link Corporation Limited, 24 Enero 2021. [En línea]. Available: <https://www.tp-link.com/ec/home-networking/smart-plug/hs103p2/>. [Último acceso: 23 Mayo 2022].
- [13] Home Assistant, «YAML,» 2022. [En línea]. Available: <https://www.home-assistant.io/docs/configuration/yaml/>. [Último acceso: 18 Julio 2022].
- [14] A. Mena, «Ventajas y desventajas del 'software' libre,» Telefónica S.A., 5 Noviembre 2015. [En línea]. Available: <https://empresas.blogthinkbig.com/ventajas-y-desventajas-del-software-libre/>. [Último acceso: 18 Junio 2022].
- [15] MCI electronics, «¿Que es Raspberry Pi?,» MCIElectronics.cl, [En línea]. Available: <https://raspberrypi.cl/que-es-raspberry/>. [Último acceso: 25 Junio 2022].
- [16] 330ohms, «Raspberry Pi 4 VS Pi 3B+ ¿Realmente es mejor?,» 330ohms, 21 Noviembre 2019. [En línea]. Available: <https://blog.330ohms.com/2019/11/21/retropie-raspberry-pi-4-vs-pi-3b-realmente-es-mejor/>. [Último acceso: 25 Junio 2022].

- [17] A. Alcolea, «Raspberry Pi 4 vs Raspberry Pi 3,» Computer Hoy, 24 Junio 2019. [En línea]. Available: <https://bit.ly/3NpFaWD>. [Último acceso: 25 Junio 2022].
- [18] Cyan Automation, «La mejor tarjeta SD para Home Assistant,» Cyan Automation, 27 Julio 2020. [En línea]. Available: <https://bit.ly/3bqxG8h>. [Último acceso: 25 Junio 2022].
- [19] Didacticas Electrónicas, «Caja metálica Argon ONE,» I+D Electrónica, 2022. [En línea]. Available: <https://bit.ly/3bJsQTK>. [Último acceso: 28 Junio 2022].
- [20] Unit Electronics, «Carcasa de Aluminio Argon One,» UElectronics, 2022. [En línea]. Available: <https://uelectronics.com/producto/carcasa-de-aluminio-argon-one-pi-4-v2/>. [Último acceso: 28 Junio 2022].
- [21] Solo Electronicos, «6 herramientas de domótica de código abierto,» 14 Noviembre 2019. [En línea]. Available: <https://bit.ly/3yfQyyP>. [Último acceso: 03 Julio 2022].
- [22] Home Assistant, «Automating Home Assistant,» [En línea]. Available: <https://www.home-assistant.io/getting-started/automation/>. [Último acceso: 03 Julio 2022].
- [23] Home Assistant, «Customizing Devices,» [En línea]. Available: <https://www.home-assistant.io/docs/configuration/customizing-devices/>. [Último acceso: 03 Julio 2022].
- [24] Ionos, «InfluxDB: explicación, ventajas y primeros pasos,» IONOS Cloud, 28 Septiembre 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/que-es-influxdb/>. [Último acceso: 03 Julio 2022].
- [25] Hogar TEC, «Sistemas domóticos centralizados, descentralizados y distribuidos,» HogarTEC, 28 Julio 2015. [En línea]. Available: <https://hogartec.es/hogartec2/sistemas-domoticos-centralizados-descentralizados-y-distribuidos/>. [Último acceso: 29 Junio 2022].
- [26] Punto Flotante, «Sensor infrarrojo de movimiento PIR,» Punto Flotante S.A, 2017. [En línea]. Available: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>. [Último acceso: 30 Junio 2022].

- [27] Webedia Brand Services, «Niveles de brillo, de iluminación y luz ambiental,» Xataca, 14 Marzo 2018. [En línea]. Available: <https://bit.ly/3nDhW4G>. [Último acceso: 01 Julio 2022].
- [28] Kreon, «Luz cálida y fría: cómo influye la temperatura del color en tu interior.,» Kreon.com, 17 Julio 2018. [En línea]. Available: <https://bit.ly/3AstFuy>. [Último acceso: 01 Julio 2022].
- [29] J. Pastor, Xataca.com, 15 Septiembre 2019. [En línea]. Available: <https://bit.ly/3bJ5fmi>. [Último acceso: 01 Julio 2022].

## **7 ANEXOS**

La lista de los Anexos se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlace de video de funcionamiento



**ESCUELA POLITÉCNICA NACIONAL**  
**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**  
**CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

## **ANEXO I**

### **CERTIFICADO DE ORIGINALIDAD**

Quito, D.M. 22 de agosto de 2022

De mi consideración:

Yo, **LEANDRO ANTONIO PAZMIÑO ORTIZ**, en calidad de Director del Trabajo de Integración Curricular titulado: **IMPLEMENTACION DE UN SERVIDOR IOT PARA CONVERTIR UNA CASA EN INTELIGENTE** elaborado por el estudiante **DENNIS DAVID YANACALLO CASALIGLIA** de la carrera **TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11 %.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

[Reporte Turnitin - DYanacallo.pdf](#)

Atentamente,

---

**DIRECTOR**

Ing. Leandro Antonio Pazmiño Ortiz, MSc.

**Docente Ocasional**

**ESFOT**

## ANEXO II

Código QR del video de funcionamiento del prototipo

