

ESCUELA SUPERIOR DE INGENIERIA Y TECNOLOGIA

MEMORIA DEL TRABAJO DE FIN DE GRADO

**Despliegue de red domótica utilizando dispositivos ZigBee
y Home Assistant**

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Alumno: Santiago Pérez Delgado(alu0101341347@ull.edu.es)

Tutor: D. Alberto Hamilton Castro(albham@ull.edu.es)

INDICE DE CONTENIDOS

1.Resumen	6
1.1Abstract	7
2.Introducción	8
2.1.Planteamiento del problema	8
2.2.Objetivo.....	8
2.4.Tecnologías/Conocimientos Previos	9
2.4.1.ZigBee.....	9
2.4.2.ZigBee Home Automation	10
2.4.3. Home Assistant	10
2.4.4.ZigBee dentro de Home Assistant.....	11
2.4.5.Posibilidades de Automatización en Home Assistant.....	11
2.4.6.El código de Home Assistant	12
2.4.7.Quirks en Home Assistant	13
2.4.8.Extensiones en Home Assistant	13
2.4.9.Alternativas a la integración de ZigBee Home Automation en Home Assistant	15
3.Desarrollo.....	16
3.1.Instalacion del Home Assistant	16
3.2.Integraciones.....	17
3.2.1Integracion del Coordinador	17
3.2.2.Integración Conmutadores Inteligentes	19
3.2.3.Montaje Sensor Puertas y Ventanas	23
3.2.4.Montaje Bombillas	26
3.2.5.Integración Mando Styrbar.....	27
3.2.6.Integración Mando Rodret	29
3.3.Automatizaciones	30
3.3.1.Automatizaciones con el mando Styrbar.....	31
3.3.2.Automatizaciones con el Mando Rodret.....	40
3.3.3.Automatización Con el Sensor de Puertas y Ventanas	46
3.3.4.Automatizacion con Comunicaciones	47
3.4.Estudio Rango Mandos y Sensor.....	53
3.5.Instalacion de rasgos personalizados en Home Assistant	54
4.Conclusiones y líneas abiertas	56

4.1. <i>Conclusiones</i>	56
4.2. <i>Conclusions</i>	58
4.3. <i>Lineas Abiertas</i>	59
5. <i>Presupuesto</i>	60
6. <i>Referencias</i>	62
7. <i>Anexos</i>	63
7.1 <i>Código para el cambio de color rojo</i>	63
7.2 <i>Código para el cambio de color verde</i>	63
7.3 <i>Código para el cambio de color azul</i>	63
7.4 <i>Código para aumentar el valor del color escogido</i>	63
7.5 <i>Código para disminuir el valor del color escogido</i>	63

INDICE DE FIGURAS

Figura 1. Ejemplo de red ZigBee como mesh network	9
Figura 2. Código de configuración básico de Home Assistant como ejemplo del lenguaje YAML.....	13
Figura 3. Tienda de extensiones de Home Assistant	14
Figura 4. Ejemplo de código de Node-Red usado en las automatizaciones donde se ve el diagrama de flujo los bloques y la columna de debug.....	15
Figura 5. Vista principal de la máquina virtual de Home Assistant.....	16
Figura 6. Vista principal de Home Assistant.	17
Figura 7. SonOff ZigBee 3.0 USB Dongle Plus	18
Figura 8. Vista de las integraciones del Home Assistant, donde se ve el ZigBee Home Automation.....	18
Figura 9. El conmutador inteligente SonOff ZBMINIL 2	19
Figura 10. Explicación en las instrucciones del montaje del conmutador usando dos interruptores.....	20
Figura 11. Montaje del conmutador junto con los dos interruptores y la bombilla que controla (arriba a la izquierda).	20
Figura 12. Página de características del conmutador con dos interruptores.	21
Figura 13. Explicación en las instrucciones de cómo realizar el montaje del conmutador con un único interruptor	22
Figura 14. Primer montaje realizado con el conmutador con un interruptor y una bombilla.	22
Figura 15. Página de características del conmutador con un interruptor.....	23
Figura 16. Sensor de puertas y ventanas.	24
Figura 17. Página de características del sensor de puertas y ventanas.	25
Figura 18. Sensor abierto(derecha) y sensor cerrado(izquierda).....	25
Figura 19. Página de características de la bombilla LED de 806 lúmenes.....	26
Figura 20. Página de características de la bombilla LED de 470 lúmenes.....	27
Figura 21. Mando Styrbar.	28
Figura 22. Página de características del mando Styrbar.....	29
Figura 23. Mando Rodret.	30
Figura 24. Página de características del mando Rodret.	30
Figura 25. Código con el que se encienden las dos bombillas conectadas al conmutador con un interruptor.....	31
Figura 26. Diagrama en Node-Red para regular el brillo de la bombilla dependiendo de la hora que sea.	32
Figura 27. Código con el que se encienden una única bombilla de las que están conectadas al conmutador con un interruptor.	33
Figura 28. Código con el que se enciende la bombilla LED si el conmutador con un interruptor este encendido.....	33
Figura 29. Código con el que se aumenta el brillo de la bombilla.	34
Figura 30. Código con el que se disminuye el brillo de la bombilla.	34
Figura 31. Código con el que se reestablece el brillo de la bombilla.	35
Figura 32. Detalle de la llamada a servicio con el que se reestablece el brillo de la bombilla.....	36
Figura 33. Código con el que se reestablece el color de la bombilla.	37

Figura 34. Detalle de la llamada a servicio con el que se reestablece el color de la bombilla.....	38
Figura 35. Código en para cambiar el brillo usando un único botón.....	38
Figura 36. Detalle del bloque de decisión donde se reestablece el brillo a cero si va a superar el máximo o aumenta el brillo si el máximo no se supera.....	39
Figura 37. Asignación de números mediante funciones para cambiar partes del espacio de color RGB.....	39
Figura 38. Código en Node-Red para cambiar el color de la bombilla seleccionando la parte del espacio de color RGB que se desea cambiar.	40
Figura 39. Código en Node-Red que permite reestablecer el color de la bombilla a blanco.	40
Figura 40. Código con el que se cambia el estado de la bombilla.....	41
Figura 41. Código con el que se aumenta el brillo de la bombilla.	41
Figura 42. Código con el que se disminuye el brillo de la bombilla.	42
Figura 43. Código con el que se reestablece el brillo de la bombilla.	42
Figura 44. Detalle de la llamada a servicio con el que se reestablece el brillo de la bombilla.....	43
Figura 45. Código para establecer la temperatura de la bombilla al máximo.	44
Figura 46. Detalle de la llamada al servicio de encender la luz donde se ve que la temperatura de la luz esta al máximo.....	44
Figura 47. Código para establecer la temperatura de la bombilla al mínimo.	45
Figura 48. Detalle de la llamada al servicio de encender la luz donde se ve que la temperatura de la luz esta al mínimo.	45
Figura 49. Código con el que se puede hacer que la bombilla parpadee.....	46
Figura 50. Detalle de la elección del parpadeo y sus modos en la llamada al servicio de encender la bombilla.....	46
Figura 51. Código con el que es conmutador con dos interruptores se enciende al abrirse el sensor.....	47
Figura 52. Código con el que es conmutador con dos interruptores se apaga al cerrarse el sensor.	47
Figura 53. Archivo configuration.yaml con la inclusión de la integración de Telegram.	48
Figura 54 Mensaje del bot enviado para comprobar que la integración funcionaba	49
Figura 55 Mensaje del bot enviado para comprobar que la integración funcionaba.	49
Figura 56. Detalle del disparador, donde se ve que está condicionado a que la bombilla este encendida diez minutos.....	50
Figura 57. Detalle de la llamada al servicio de envío de mensaje por Telegram donde se ve el mensaje que se enviara pasados los diez minutos.....	50
Figura 58. Vista de Home Assistant en la aplicación móvil.	51
Figura 59. Código de llamada de servicio para comprobar que la integración de la aplicación móvil funcionaba.	52
Figura 60. Notificación de Home Assistant enviada para comprobar que la integración funcionaba	52
Figura 61. Notificación de la aplicación de Home Assistant en dispositivo Android.	53
Figura 62. Información del mando Rodret, donde se ve la dirección del archivo de rasgos propio.....	54
Figura 63. Archivo de configuración con la inclusión del código para poder incluir rasgos personalizados.	55

INDICE DE TABLAS

Tabla 1. Presupuesto de realización del trabajo.....	60
Tabla 2. Presupuesto para la instalacion en un hogar común, instalando las bombillas inteligentes.	61
Tabla 3. Presupuesto para la instalacion en un hogar común, instalando los conmutadores inteligentes.....	61

1. Resumen

El objetivo del presente TFG (Trabajo de Fin de Grado) es la implementación de una red domótica usando dispositivos ZigBee y ZigBee 3.0. Para ello se realiza la instalación de Home Assistant, un software de código libre para el control de redes domóticas. Además, es necesario integrar un coordinador de red ZigBee, así como distintos dispositivos ZigBee de diferentes fabricantes, con el fin de demostrar que el protocolo es flexible.

Una vez implementados la red domótica y los dispositivos se realizan diferentes automatizaciones en maquetas, de tal manera que se pueda conocer el alcance de las posibilidades que ofrece el software y el hardware. Estas automatizaciones se realizan teniendo en cuenta primeramente su aplicación práctica, por lo que, a pesar de encontrarse el hardware en maquetas, se supuso que estas son parte de la instalación de un hogar real.

También se realizan integraciones con el objetivo de comunicarse con el usuario mediante su dispositivo móvil, bien a través de aplicaciones de mensajería o la aplicación propia del Home Assistant. Así como un estudio del rango de los mandos a distancia usados en el trabajo, para conocer su alcance real.

Palabras Clave: domótica, ZigBee, Home Assistant, ZigBee Home Automation.

1.1 Abstract

The objective of this TFG (Final Degree Project) is the implementation of a home automation network using ZigBee and ZigBee 3.0 devices. To do this, Home Assistant is installed, an open-source software for the control of home automation networks. In addition, it is necessary to integrate a ZigBee network coordinator, as well as different ZigBee devices from different manufacturers, to demonstrate that the protocol is flexible.

Once the home automation network and the devices have been implemented, different automations are carried out in models, so that the scope of the possibilities offered by the software and hardware can be known. These automations are made with their practical application in mind, so despite the hardware being found in models, it was assumed that these are part of the installation of a real home.

Integrations are also made with the aim of communicating with the user through their mobile device, either through messaging applications or the Home Assistant's own application. As well as a study of the range of the remote controls used in the work, to know their real range.

Key words: home automation, ZigBee, Home Assistant, ZigBee Home Automation.

2.Introducción

2.1.Planteamiento del problema

Recientemente la domótica ha tenido un gran aumento de demanda, viendo como grandes empresas se suman a vender dispositivos que permiten la automatización de procesos en el hogar, por lo que a su vez cada vez más gente incorpora e instala módulos de este tipo en su casa. En la mayoría de los casos, cada empresa tiene un controlador domótico propio al que vincular sus productos, por lo que les obliga a comprar únicamente dispositivos de esa empresa o llevar el control de los elementos de manera independiente unos de otros. Existen casos de centrales simples, como los de IKEA que tiene una central domótica propia y permite conectar sus dispositivos a esta. Así como casos más complejos como el KNX, donde se realiza una planificación domótica de un hogar desde el momento de la concepción de este, pues requiere de una instalación eléctrica completa. Para este tipo de casos es útil el Home Assistant, ya que permite conectar dispositivos compatibles con distintos módulos, de fabricantes distintos, y aun cuando estos están pensados para que sean controlados por centrales domóticas concretas. Permitiendo realizar el control de todos los dispositivos de una casa independientemente de su origen, puesto que todos se conectan a un control común. Además, Home Assistant es de código abierto y gratuito, por lo que únicamente requerirá de conocimientos de código para su aplicación a la hora de realizar el control de la domótica de un hogar, abaratando costes, ya que no tiene costes de utilización. Sumándose a esto también el mantenimiento de la comunidad mediante foros^[1].

2.2.Objetivo

A raíz del planteamiento del problema es que surge este trabajo, en el que se pretende conocer y estudiar las posibilidades que permite tanto el software de Home Assistant como el hardware de domótica. Para ello se realizarán montajes experimentales a modo de maquetas, y programando distintas situaciones que podrían requerir de una automatización en un hogar. En este caso se utilizarán dispositivos que funcionen mediante el estándar de comunicaciones ZigBee de distintos proveedores. Por tanto, se intentará crear una red con estos dispositivos para probar que ZigBee es un protocolo versátil.

Para estudiar las posibilidades de Home Assistant como herramienta software para la domotización se realizarán un número suficiente de automatizaciones, siempre simulando que estos fuesen casos reales. Ya que al realizarse en maquetas estas solo podrán tomarse como ejemplo, pero a la vez permiten que en un mismo montaje se realicen automatizaciones distintas. Entendiendo que los elementos conectados en la maqueta simularían los de la instalación de un hogar en distintas habitaciones incluso.

2.4.Tecnologías/Conocimientos Previos

Para la realización de este TFG, a parte de los conocimientos obtenidos durante el grado, ha sido necesario el conocimiento de tecnologías como ZigBee, que tienen implementados todos los elementos usados en el trabajo

2.4.1.ZigBee

ZigBee es un protocolo de comunicaciones de un bajo ancho de banda que se basa en señales digitales de baja potencia para permitir la conexión de elementos ZigBee de manera inalámbrica a otros elementos ZigBee locales. Fue diseñado por la ZigBee Alliance, actualmente llamada Connectivity Standards Alliance, incluyendo actualmente más de 500 empresas. ZigBee es una conexión del tipo mesh network, es decir que hay una interconectividad entre sus dispositivos. Estos se pueden clasificar en tres tipos, el coordinador, los dispositivos repetidores y los dispositivos finales, o “end-devices”. El coordinador es el que establece la comunicación de toda la red ZigBee, y a su vez realiza la comunicación con el software de gestión domótica que se use. Los dispositivos repetidores amplían el alcance de la red y el número de dispositivos que pueden haber conectados en esta, suelen ser dispositivos que están conectados permanentemente a la red eléctrica. Los end-devices son el resto de los dispositivos de ZigBee, existen algunos que se conectan directamente a la red y otros que requieren de un dispositivo repetidor para poder conectarse. Al conectarse los end-devices a través de un dispositivo funcionan como hijos de estos, por lo que es necesario que el dispositivo a través del que se conectan esté disponible en la red para facilitar la conexión de los end-devices.

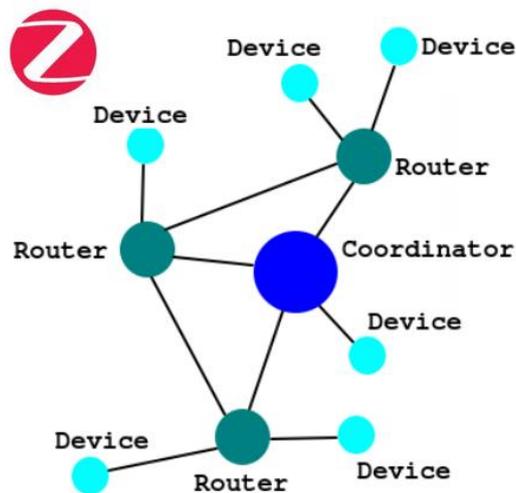


Figura 1. Ejemplo de red ZigBee como mesh network

ZigBee es un protocolo que está basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal. Este estándar especifica el nivel físico y el control de acceso al medio para redes inalámbricas de área personal con tasas bajas de transmisión de datos. Por esto usa las bandas de radio ISM, que son bandas definidas para usos

distintos a la telecomunicación. En Europa la banda que usa por tanto es la de 836 MHz y también hace uso de la banda mundial de 2,4GHz.

El estándar del que surge ZigBee nació dado a la necesidad de poder establecer comunicaciones inalámbricas que la tecnología de ese momento, Wi-Fi y Bluetooth, no podían cubrir. Comparándolo con la conexión por Bluetooth, ZigBee permite la conexión de un mayor número de nodos, usa la misma banda de frecuencia, aunque con menos canales y tiene un consumo menor. Estas características conforman las ideas que se tenían de la creación de esta nueva red, que fuese de un consumo más bajo y con una mayor conectividad que el Bluetooth. A su vez, en la comparación con la conexión mediante red Wi-Fi, ZigBee también permite la conexión de un mayor número de dispositivos, y dado a que dispone de un coordinador de red la configuración y posibles reconfiguraciones de la red son más sencillas. Así mismo, el hecho de que los dispositivos estén interconectados entre sí permite que estos envíen información entre ellos, facilitando su comunicación. Además, otra ventaja respecto a la red Wi-Fi es que la red ZigBee puede ampliar su alcance debido a su topología.

2.4.2. ZigBee Home Automation

Basado en este protocolo de comunicaciones existe el ZigBee Home Automation (ZHA), que está desarrollado mediante una librería de Python de código abierto, llamada zigpy, la cual es independiente del hardware. Revisando su documentación y código relacionado en GitHub^[2] se puede ver que ha superado el millar de “commits” durante ciertos periodos de tiempo entre el 2018 y la actualidad, habiendo llegado a los diez mil códigos desarrollados y subidos a la página a principios de este año 2024. El hecho de estar tan desarrollado permite la conexión con varios tipos de coordinadores de red distintos, siempre y cuando sean compatibles con dicha librería. Mediante esta integración se sustituye el uso de otros sistemas controladores basados en ZigBee patentados, que suelen estar pensados para unos dispositivos concretos derivados de la empresa que desarrolla el coordinador en cuestión. De esta manera se pueden conectar dispositivos de diversos fabricantes en una única red, facilitando su control e implementación general. El ZigBee Home Automation viene integrado dentro del Home Assistant, que es la herramienta software principal que se usara en este trabajo, dentro de este clasifica los dispositivos en distintos tipos, como pueden ser switches, sensores, sensores binarios o luces, que son los que principalmente se han usado en este trabajo.

2.4.3. Home Assistant

Home Assistant es un software que permite su implementación en una variedad de dispositivos^[3], entre ellos RaspberryPI, Linux, o Windows, que será el que usare en mi caso, por comodidad, ya que es el sistema operativo que acostumbro a usar. Al ser de código abierto buena parte del código nuevo se crea mediante la plataforma GitHub, donde actualmente tienen 26 autores distintos activos con más de 75 propuestas distintas. Tiene una comunidad muy activa por lo que las actualizaciones son muy frecuentes. El software de Home Assistant es también de código abierto, y permite la

implementación de diversas integraciones de domótica a parte del ZHA. Debido a esto el número de integraciones posibles y sus posibilidades son enormes, sumándole a que Home Assistant al ser la central domótica del hogar que establezcas permite cruzar distintas integraciones entre ellas. Esto y su sencillez para instalarse los dispositivos lo que lo hace una gran alternativa económica a otras opciones domóticas como pueden ser KNX, ya que estas requieren de una reinstalación completa, complicando la instalación en hogares que no sean de nueva construcción.

2.4.4. ZigBee dentro de Home Assistant

Dentro de estas integraciones permite la detección e integración de dispositivos, de los cuales detecta sus cualidades, llamadas en el software entidades, y al igual que el ZigBee Home Automation clasifica cada dispositivo, esto también se extiende a las entidades, por lo que cada entidad es de una clase concreta, permitiendo una serie de lecturas de esta. En ciertos casos solo podrá conocer su estado, para los dispositivos más sencillos, y en otros las entidades de un dispositivo dispondrán a su vez de atributos que lo definan con exactitud. Esta clasificación le diferencia del otro uso de ZigBee, el ZigBee2MQTT, que asigna una entidad individual a lo que en ZigBee Home Automation sería un atributo de una entidad. También el ZigBee Home Automation implementado dentro del Home Assistant dispone de eventos, que son las acciones que pueden llevar a cabo ciertos dispositivos, como los mandos a distancia, donde la pulsación de sus botones se registra como eventos en vez de ser acciones concretas y a parte de las entidades de los dispositivos. Todas estas acciones, aparte de permitir realizar algún tipo de programación o automatización pueden forzarse a que ocurran. Es decir, se puede forzar a cambiar el estado de una bombilla, por ejemplo, mediante la página del propio Home Assistant, llamando a un servicio, que viene a ser la manera de llamar e indicar de Home Assistant de que se ha de realizar una acción, sea esta en una automatización o no.

2.4.5. Posibilidades de Automatización en Home Assistant

En base a estos dispositivos y sus entidades permite realizar automatizaciones, scripts y escenas. En las automatizaciones se realiza un código en el que se establece un trigger o disparador de la automatización, al que se pueden añadir además varias condiciones adicionales que se pueden sumar al disparador de la automatización, para condicionar que esta progrese o no. Posteriormente se establecen las acciones que se realizan de manera secuencial, es decir, en el orden en el que se escriben en el código. Los scripts básicamente son una secuencia de código que cuando se llama, mediante una automatización o una llamada de servicio se ejecutan todas las acciones que incluye de manera secuencial, por lo que podría ser útil para realizar códigos más compactos, o en caso de que una serie de acciones se realice en más de una automatización para que exista una mayor comodidad a la hora de programarlas. Las escenas no son realmente programación en sí, realmente son una lista, donde se asigna a cada uno de los

actuadores que se tengan en Home Assistant a tener un estado concreto, de tal manera que al llamarse la escena se fuerza el estado de todos los dispositivos a uno previamente escogido.

Entrando en las automatizaciones en concreto, que conforman el grueso del trabajo realizado, todas siguen la misma estructura en cuanto a su lógica. Se dividen principalmente en tres partes, “When”, “And If” y “Then Do”. El bloque When es en el que se evalúa el disparador principal de la automatización, si esta condición no se cumple no se ejecutara la automatización. And If permite añadir condiciones adicionales tras el disparador, para permitir que la automatización se ejecute en casos más concretos. Then Do es el bloque en el que se indican las acciones a realizar por la automatización, a estos se pueden sumar bloques de evaluación para condicionar que se realicen unas acciones u otras. Es importante tener en cuenta que todos estos bloques funcionan de manera secuencial, es decir el código se ejecuta empezando en la primera línea y acabando en la última.

2.4.6.El código de Home Assistant

Home Assistant como software se basa en dos lenguajes para su funcionamiento: Python y YAML. Python lo aplica para los archivos de configuraciones concretas de dispositivos y acciones del software, como pueden ser la compilación de errores o ejecutar comandos de comunicación, así como la configuración del programa en niveles más profundo. YAML se usa en el archivo de configuración general, llamado `configuration.yaml`, del Home Assistant, así como para las automatizaciones. Esto debido a que es un lenguaje que funciona como una lista de datos del software, y resulta más legible y cercano al lenguaje humano. A parte del lenguaje YAML para las automatizaciones también existe una interfaz gráfica, que te permite realizar las automatizaciones aun desconociendo el lenguaje. Este te pone las facilidades para encontrar tanto las entidades como dispositivos para la automatización y abre menús con todas las opciones disponibles dentro de las acciones de un dispositivo o entidad, que luego traduce a YAML y es lo que carga realmente en el software.

```

1  ~
2  # Loads default set of integrations. Do not remove. ~
3  default_config: ~
4  ~
5  # Load frontend themes from the themes folder ~
6  frontend: ~
7  - themes: !include_dir_merge_named themes ~
8  ~
9  automation: !include automations.yaml ~
10 script: !include scripts.yaml ~
11 scene: !include scenes.yaml ~
12 ~
13 ~
14 |

```

Figura 2. Código de configuración básico de Home Assistant como ejemplo del lenguaje YAML

2.4.7. Quirks en Home Assistant

En el uso del Python para las configuraciones concretas de dispositivos existen un tipo de archivos llamados quirks, rasgos en inglés, que son los que dictan las características del dispositivo, su fabricante, nombre, entidades y acciones que puede realizar. Cada dispositivo tiene asignado uno concreto dentro del propio Home Assistant, aunque puede pasar, bien por actualizaciones del fabricante o porque el dispositivo sea muy reciente, que un dispositivo no disponga de un quirk propio. Para esto también existe la posibilidad de crear un archivo de Python propio para este dispositivo en el que se describe su nombre, fabricante, entradas y salidas, así como su tamaño, y las acciones que este pueda realizar. Este archivo posteriormente se puede introducir en el Home Assistant y asignarlo al dispositivo mediante el archivo configuration.yaml. Dado que es un software de código abierto existen gran cantidad de foros y creaciones de la comunidad para la implementación de dispositivos, por ejemplo, donde la gente comparte los archivos de quirks completos que estos desarrollan para ciertos dispositivos que no disponen de uno o fallan tras una actualización.

2.4.8. Extensiones en Home Assistant

También existen extensiones dentro del software, tanto nativas del propio Home Assistant, como creadas por la comunidad, que permiten distintas facilidades. Dentro del mismo Home Assistant existe lo que se llama tienda de extensiones donde hay disponibles una gran cantidad de extensiones para distintos tipos de usos. A parte de las extensiones de la tienda, en GitHub hay algunos códigos que lo que hacen es permitir la instalación de otras extensiones no visibles en la tienda. Estas extensiones se pueden descargar simplemente en un directorio de repositorios de la propia tienda, donde simplemente al introducir el enlace de GitHub ya aparecen en la tienda para descargarlas. En mi caso concreto, las extensiones que instalé fueron ,un editor de archivos, que es nativo del Home Assistant, para poder modificar la configuración, una extensión que te permite crear copias de seguridad y guardarlas en Google Drive y otra que se llama Node-Red, que permite crear diagramas de flujo para hacer

automatizaciones y dispone de ciertos bloques que no vienen de base en el Home Assistant, por lo que facilita la programación. Además, dentro de la extensión Node-Red existen bloques en los que se puede introducir código, en dos lenguajes distintos. En los bloques en los que se llama un servicio, se puede cambiar algunas características de ciertas entidades, como en el caso de las bombillas el color de estas. Estos bloques usan el lenguaje JSON, que es un formato de texto basado en Java Script para el intercambio de datos, razón por la que se utiliza al llamar servicios y modificar sus atributos. También existen en esta extensión bloques de funciones, donde se puede introducir código en Java, para ser evaluado y modificar ciertas variables dentro de un flujo, como el propio mensaje que va arrastrando un cierto flujo, para establecer variables globales dentro de un diagrama de flujo o establecer variables globales que se extienden incluso a flujos diferentes.

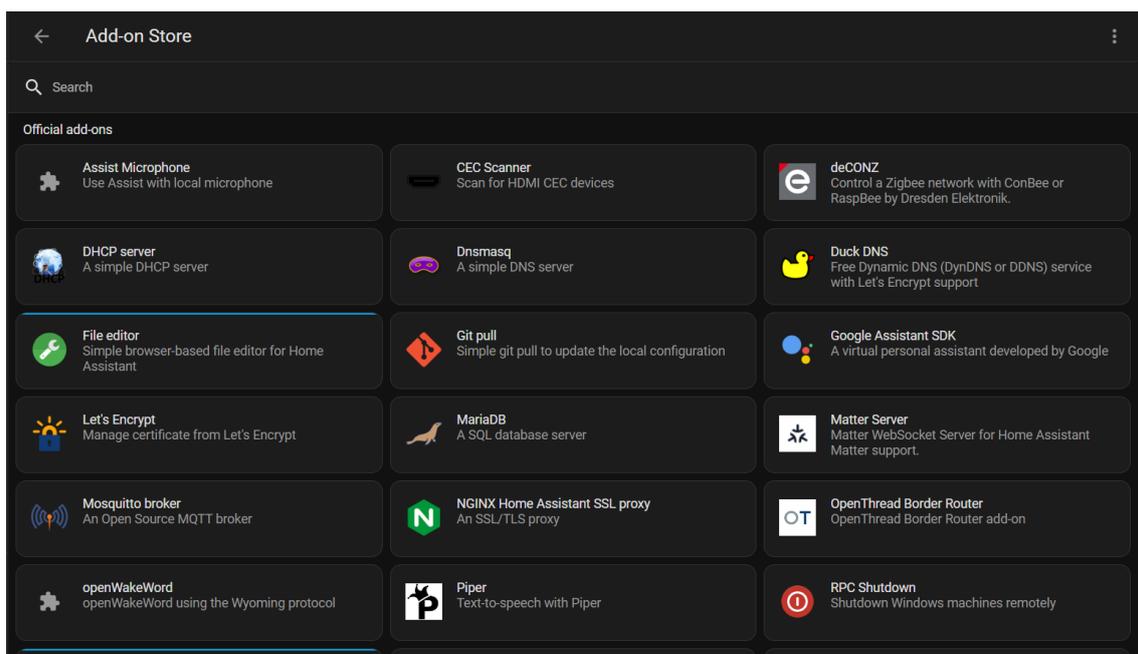


Figura 3. Tienda de extensiones de Home Assistant

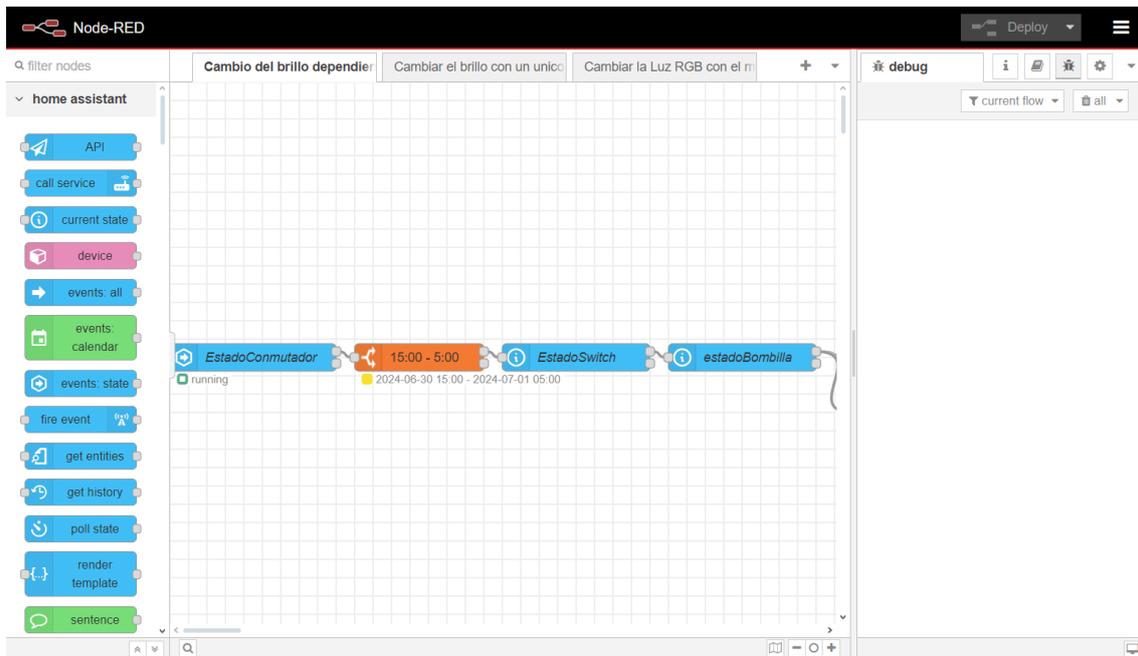


Figura 4. Ejemplo de código de Node-Red usado en las automatizaciones donde se ve el diagrama de flujo los bloques y la columna de debug.

2.4.9. Alternativas a la integración de ZigBee Home Automation en Home Assistant

Alternativamente al uso del ZigBee Home Automation hay otros protocolos que se pueden usar con ZigBee como el ZigBee2MQTT, que usa el protocolo MQTT como indica el nombre para realizar la conexión entre los dispositivos y el coordinador mediante una conexión de ancho de banda limitada y sin pérdidas. Esta conexión es del tipo publish-suscribe, donde los publicadores clasifican los mensajes recibidos en clases para así distribuirlos entre los elementos suscritos a ella. La gestión de estos mensajes se realiza mediante un bróker, que es el que clasifica y distribuye los mensajes a los elementos suscritos. Asimismo, existen alternativas al Home Assistant como las centrales domóticas que tienen distintos fabricantes para sus propios dispositivos, que suelen venir con una aplicación móvil desde la cual manejarlos, también existen otras aplicaciones para el control de un hogar inteligente como pueden ser Samsung SmartThings, Alexa, Google Home o openHAB, que está basado en Java.

Las diferencias de, por ejemplo, el Samsung SmartThings es que es un controlador domótico mediante una aplicación pensado para la conexión con dispositivos Samsung únicamente. Esto obliga al usuario únicamente disponer de dispositivos de este fabricante, por lo que perdería la posibilidad de integrar dispositivos de otros fabricantes que pudieran resultarle atractivos para su instalación.

3.Desarrollo

3.1.Instalacion del Home Assistant

Para poder realizar las automatizaciones y vincular los dispositivos primeramente fue necesario instalar HomeAssistant. En el caso de su instalacion en Windows, esta requería de la creación de una máquina virtual de Linux, con una huella que el propio Home Assistant provee para poder crear un servidor en tu red local en el que establecer el Home Assistant. Siguiendo los pasos dados en la página web^[4] sobre la instalación, donde se establecen los requerimientos de memoria RAM y procesadores que usara la máquina virtual, asi como la configuración de puertos, finalice con la instalacion de la maquina virtual. Es importante la configuración de puertos realizada, ya que en el caso de esta instalación de Home Assistant, la conexión no se puede realizar por red Wi-Fi, si no que requiere de darle acceso al puerto LAN a la máquina virtual, porque si no la creación del servidor fallara, evitando la conexión. Es importante denotar también que desde la consola de la máquina virtual se pueden introducir ciertos comandos que nos permiten realizar actuaciones directas sobre el Home Assistant, como puede ser restaurar copias de seguridad, que en caso de cambiar la configuración y que el Home Assistant como tal no pueda iniciarse nos sigue permitiendo restaurarlo, o ver la lista de perfiles que hay en el servidor concreto de Home Assistant, podemos también instalar extensiones o actualizarlas, asi como buscar actualizaciones del propio Home Assistant incluso.

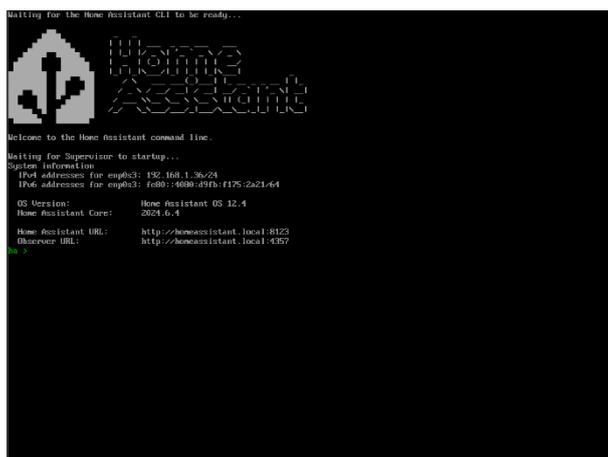


Figura 5. Vista principal de la máquina virtual de Home Assistant.

Una vez con la instalacion correcta de la máquina virtual pude acceder a Home Assistant, que como ha creado un servidor en la red es accesible mediante o bien la dirección IP más :8123 tras esta o como <http://homeassistant.local:8123>, la propia página de Home Assistant^[5] recomienda esta segunda forma por su sencillez para acceder, aunque dice que en ciertos dispositivos Windows muy antiguos puede que sea accesible únicamente mediante IP. Además, cabe recalcar que cualquier dispositivo

conectado a la red puede acceder a Home Assistant, ya esté conectado a la red mediante un cable ethernet o por Wi-Fi, aunque no sea el dispositivo en el que esta la máquina virtual ejecutándose.

Una vez se accede a esta dirección IP Home Assistant te pide que crees una cuenta, y te pide una serie de datos de localización de tu hogar, en ese momento ya estaría establecido tu hogar de Home Assistant. Por lo que en este punto se puede empezar a conectar dispositivos y realizar automatizaciones. Además, viene con unas integraciones de base, que son, el text-to-speech de Google Translate, una aplicación sobre el clima, una sobre el sol, una lista de la compra y un buscador de radio. También dispone de varias habitaciones o áreas como las llama el Home Assistant creadas de base, con el objetivo de poder clasificar los dispositivos dependiendo de donde estén estos instalados. No use ninguna de las integraciones que vienen con el Home Assistant en este trabajo concreto, pero dependiendo del objetivo para el que se instale el Home Assistant puede llegar a ser útiles.

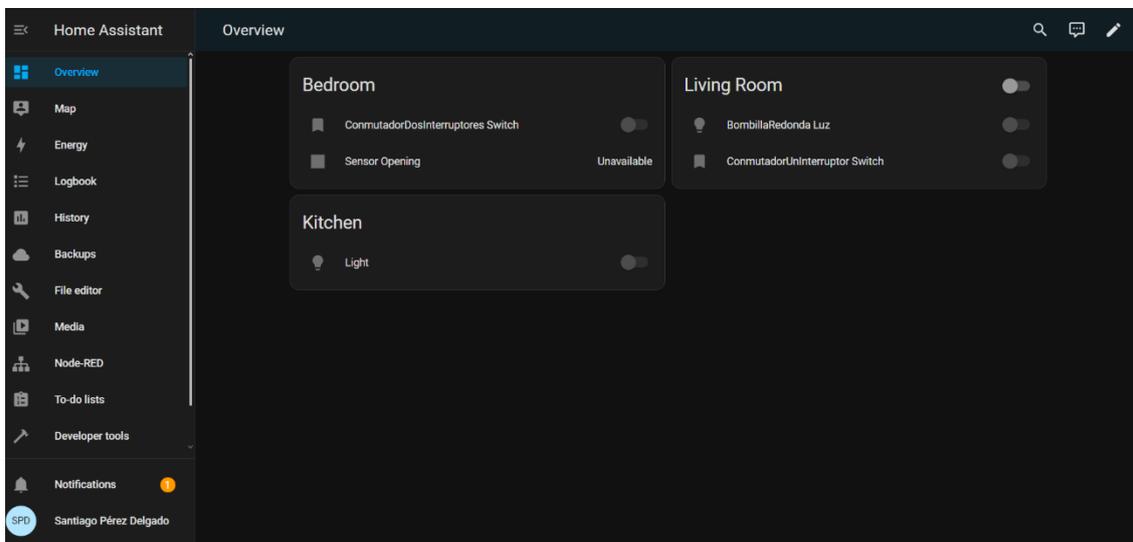


Figura 6. Vista principal de Home Assistant.

3.2.Integraciones

3.2.1Integracion del Coordinador

Una vez que ya había accedido al Home Assistant conecté el coordinador de la red ZigBee, el SonOff ZigBee 3.0 USB Dongle Plus, para poder empezar la integración del ZigBee Home Automation. Para eso siguiendo los pasos de la página de integraciones^[6] de Home Assistant intenté realizar la configuración manual del coordinador, pero no me detectaba el puerto en el que estaba conectado. Revisando en mi propio ordenador vi que no detectaba correctamente el dispositivo, dado que no tenía instalados unos controladores concretos para este, ya que al parecer o bien no vienen de base con

Windows o a la hora de conectarlo no detecto que requería de otros controladores y no salto una notificación de la necesidad de actualización.

Con los controladores del Dongle Plus ya instalados y reiniciando la máquina virtual y por tanto el Home Assistant para que fuese capaz de detectar los cambios, pude conectar el coordinador y por tanto establecer la integración de ZigBee Home Automation. Esta apareció directamente como una notificación de que se había detectado un posible elemento a conectar en la red de Home Assistant, facilitando la integración. Es por esto por lo que vi que el Home Assistant directamente notifica al usuario cuando detecta en la red dispositivos que pueden integrarse.



Figura 7. SonOff ZigBee 3.0 USB Dongle Plus

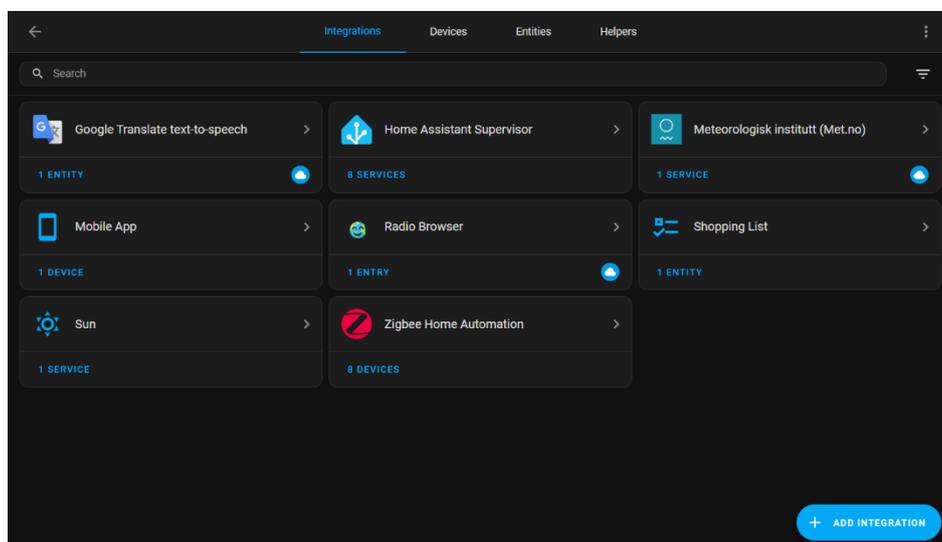


Figura 8. Vista de las integraciones del Home Assistant, donde se ve el ZigBee Home Automation.

3.2.2. Integración Conmutadores Inteligentes

Una vez con la posibilidad de poder realizar la integración de dispositivos mediante el ZigBee Home Automation, en mi casa, intente conectar uno de los conmutadores inteligentes ZBMINIL2, pensando que al ser del mismo fabricante que el coordinador daría menos problemas a la hora de conectarse a la red de Home Assistant.



Figura 9. El conmutador inteligente SonOff ZBMINIL 2

Esta conexión fue errónea, ya que al conectarse el conmutador inteligente este requiere que se le conecte aun así un interruptor común. Este interruptor lo usa como apoyo y también para cambiar el estado del relé interno incluso en uno de sus posibles montajes. Dado que no lo instale con ningún interruptor el conmutador se conectó, pero no correctamente por lo que hizo parpadear la luz ligeramente y después de eso fue imposible forzar el cambio de estado. Dado a que un montaje así era complicado y tenía un cierto riesgo, pues me requería manipular el cableado de la red de mi casa tomamos como alternativa realizar montajes de maquetas con material común, como cables de enchufes, los propios enchufes e interruptores y conectarlos a la red a ver si así funcionaban.

Para realizar los montajes de las maquetas con los conmutadores, primero realice montajes sin estas, para saber que tenía todo bien conectado y claro de como conectar posteriormente el conmutador. Por ello realice un pequeño montaje sencillo con un solo portalámparas, con una bombilla y un interruptor, que irían conectados a la red mediante un cable de enchufe, que a su vez estaría enchufado a la red.

Una vez este montaje fue exitoso y tuve claro cómo debería ser el montaje incluyendo los conmutadores, procedí a incluirlos en el montaje. Dado que existen dos maneras de incluirlos en el montaje el primero de los conmutadores lo monte con dos interruptores tal como viene dado en las instrucciones de los conmutadores. La ventaja

que encontré de este montaje respecto al otro con un único interruptor es que permite cambiar el estado del conmutador manualmente usando alguno de los dos interruptores. A parte, en el montaje con un único interruptor te requiere sacar dos cables de la línea de la red, por lo que habría que juntar dos cables en la misma salida, con los posibles riesgos que eso conlleva.

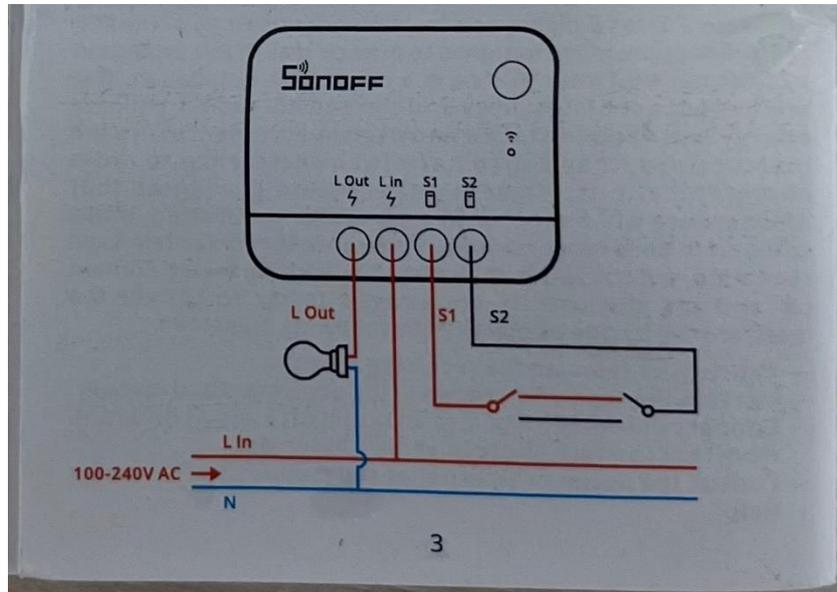


Figura 10. Explicación en las instrucciones del montaje del conmutador usando dos interruptores.



Figura 11. Montaje del conmutador junto con los dos interruptores y la bombilla que controla (arriba a la izquierda).

Una vez realizado el primer montaje con el conmutador con los dos interruptores pude conectar el conmutador a la red y ver qué características le atribuye el Home Assistant, al cual llamé dentro del programa como ConmutadorDosInterruptores. En este caso son, la batería, que es una entidad de la que disponen todos los dispositivos, aunque vayan conectados a la red, el botón identificador, que puedes pulsar desde el Home Assistant, y como característica propia la conmutación, que puedes cambiar desde el Home Assistant manualmente también o mediante automatizaciones, así como manualmente mediante los interruptores en este caso.

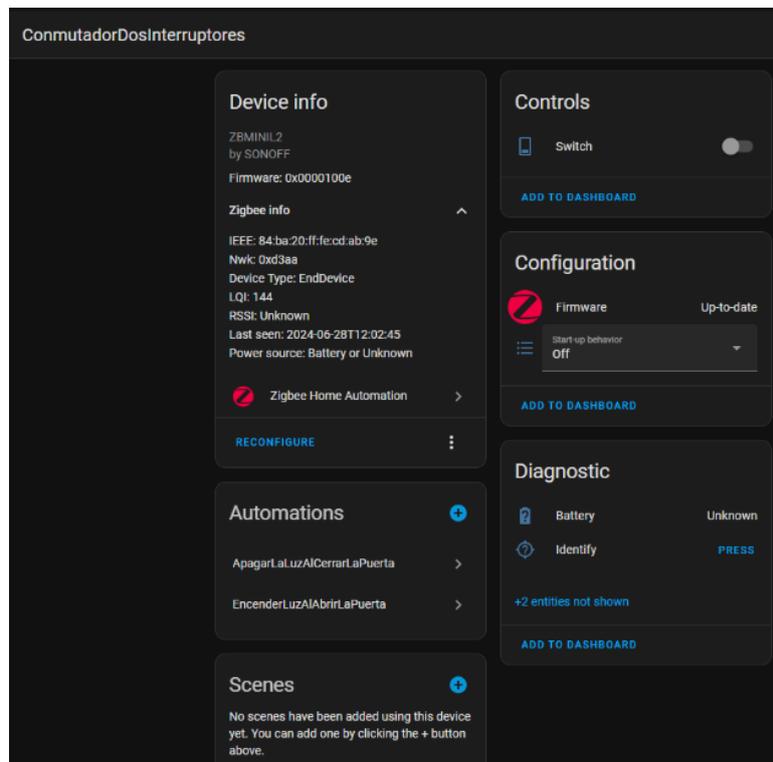


Figura 12. Página de características del conmutador con dos interruptores.

Una vez que conecté el conmutador con dos interruptores me dispuse a montar la maqueta con el conmutador y un único interruptor, para así poder saber qué diferencias podía haber entre los dos modos de implementación y compararlos, tanto en su funcionamiento como si había diferencias dentro del software dependiendo del montaje. Este conmutador fue llamado dentro del programa como ConmutadorUnInterruptor. Seguí las mismas directrices que con el conmutador con dos interruptores, conectando el conmutador tal como viene dado en las instrucciones, teniendo que conectar en un mismo punto la salida de la red tanto al conmutador como al interruptor.

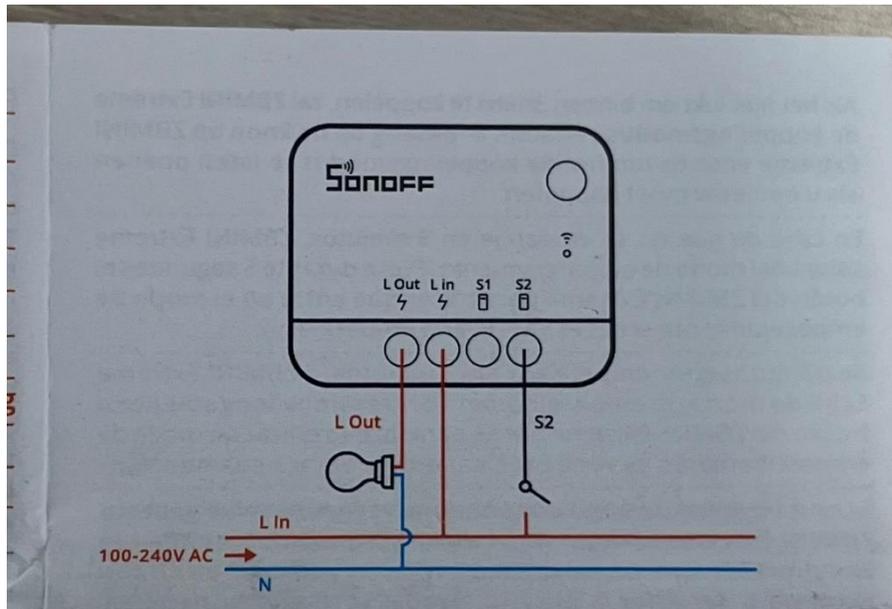


Figura 13. Explicación en las instrucciones de cómo realizar el montaje del conmutador con un único interruptor

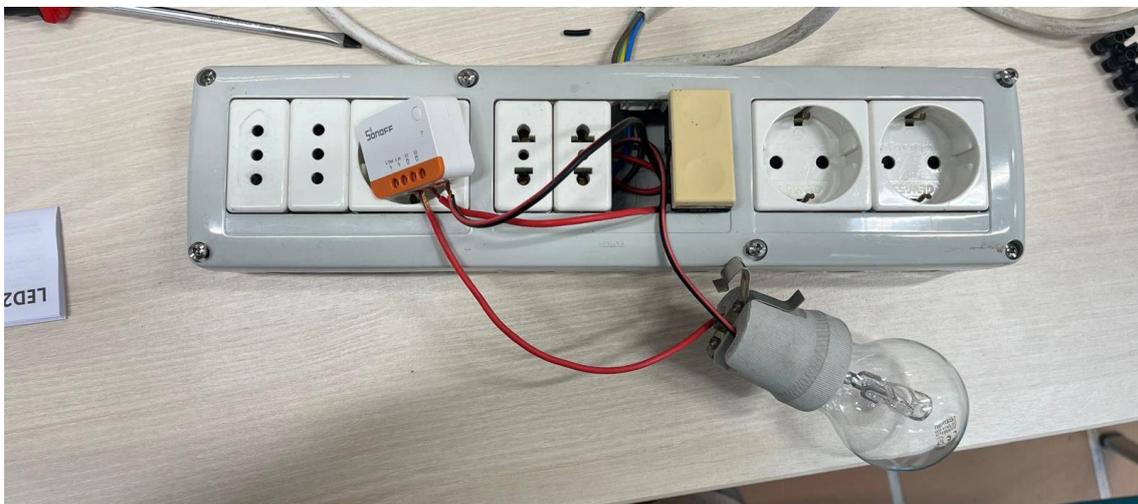


Figura 14. Primer montaje realizado con el conmutador con un interruptor y una bombilla.

Con este montaje hecho lo implementé en el Home Assistant y pude ver que a nivel de software esta implementación y la de dos interruptores no tienen ninguna diferencia, ambos tienen las mismas tres entidades, batería, identificador y conmutación, por lo que la única diferencia notable es que en el caso de la implementación con un solo interruptor no se puede cambiar manualmente el estado de la conmutación usando el interruptor con el que ha sido montado. Aunque esto me pareció raro ya que supondría que como tal el conmutador no tiene un uso real para el interruptor, por lo que pensé que a lo mejor sería debido a que algún cable no estuviese haciendo contacto suficiente en el montaje. Al principio este montaje lo realice con una sola bombilla, pero más tarde conectaría una de las bombillas LED en paralelo a la

bombilla común, cosa que a efectos de montaje no cambiaría nada, únicamente que ahora el conmutador daría corriente a dos bombillas en vez de una.

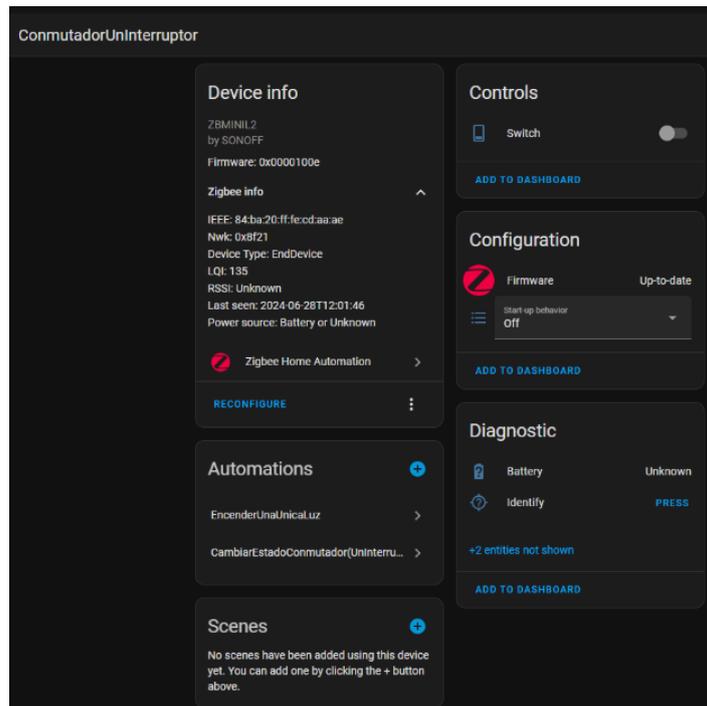


Figura 15. Página de características del conmutador con un interruptor.

3.2.3. Montaje Sensor Puertas y Ventanas

Después de montar los conmutadores pase a realizar la conexión del sensor de puertas y ventanas SilverCrest de Lidl. Este sensor se alimenta con pilas, y consta de dos piezas, las más grande que es la que dispone de la electrónica y por tanto el que se alimenta con pilas e incluso tiene un LED para indicar tanto si está en modo de emparejamiento como si ha cambiado el estado del sensor. Y una segunda pieza alargada que tendrá algún tipo de material magnético en su interior para poder detectar si la puerta está abierta o no.



Figura 16. Sensor de puertas y ventanas.

La manera de poner el sensor en modo de emparejamiento es mediante una clavija que se introduce por un orificio en uno de los lados del sensor pulsando así un botón, tras unos segundos el LED que tiene el sensor en su pieza más grande se pondrá verde y empezará a parpadear, indicando que está en modo de emparejamiento. El nombre que le asigne dentro del programa fue Sensor. No tuve problemas a la hora de que detectase el sensor de primeras con todas sus características, las entidades concretas que asigna el Home Assistant a este son, las dos generales, batería, que en este caso si presenta un número y un valor pues detecta el porcentaje de batería que le queda a las pilas, el identificador y si el sensor está abierto o cerrado.

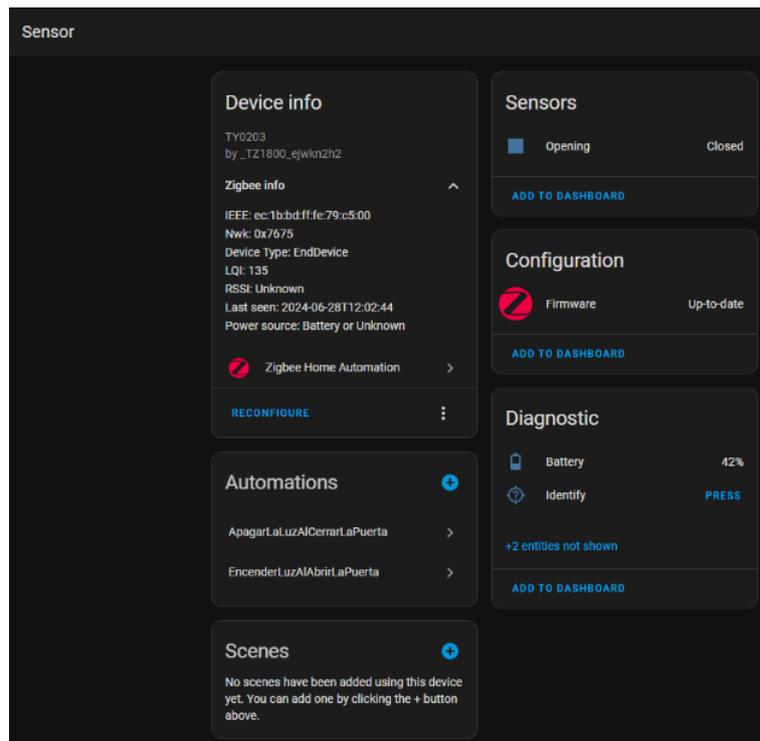


Figura 17. Página de características del sensor de puertas y ventanas.

Con el sensor ya implementado pude ver cómo es su funcionamiento concreto, cuando ambas piezas están juntas detecta está cerrado, mientras que cuando se separan una pequeña distancia, detecta que está abierta. La distancia para esto son unos dos centímetros teniendo uno pegado a otro en paralelo, aunque si los movemos manualmente tal como deberían estar montados realmente esta distancia se reduce. Esto es porque el sensor está pensado para ponerse el marco de una puerta una de las piezas, y en la propia puerta la otra pieza, haciendo que la detección de si se cierra sea cuando ambas piezas están juntas únicamente.

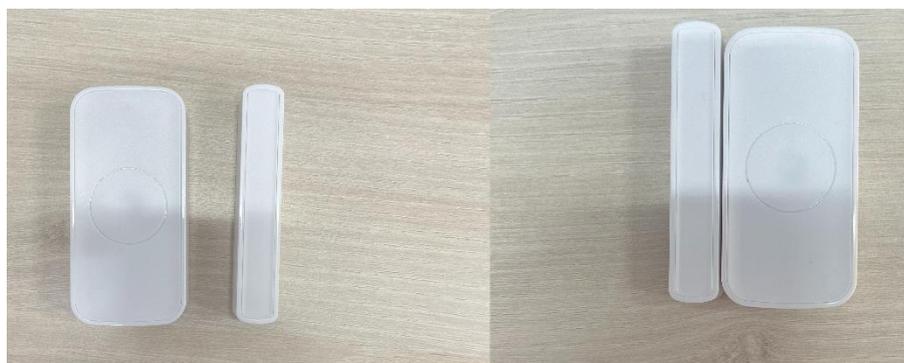


Figura 18. Sensor abierto(derecha) y sensor cerrado(izquierda).

3.2.4. Montaje Bombillas

En el caso de las bombillas LED, las TRADFRI de IKEA y la de LivarnoLux de Lidl, todas compatibles con el protocolo ZigBee. Fueron llamadas dentro del programa como BombillaRedonda, la TRADFRI de 807 lúmenes, BombillaAlargada, la TRADFRI de 470 lúmenes y BombillaAlargada2, la bombilla de LivarnoLux. Estas las monté en la misma base en la que monte los conmutadores, pero primero al menos para facilitar su conexión las conecte por separado, conectándolas directamente a la red con un interruptor de por medio únicamente. Esto porque la manera de entrar en modo de emparejamiento de las bombillas es apagarlas y encenderlas seis veces, momento en el que empezaran a parpadear ligeramente indicando que pueden conectarse correctamente. Ambas bombillas se conectaron correctamente y sin complicaciones, por lo que pude ver directamente las características de estas. En el caso de las bombillas sus entidades son, las dos generales para todos los dispositivos, batería, que no daba un valor e identificador. A parte de manera concreta en el caso de las bombillas, se pueden apagar o encender independientemente de si tienen corriente, lo que, si no disponen de corriente, aunque su estado indique que están encendidas obviamente no van a emitir luz. También se puede cambiar el nivel de encendido, la temperatura del color de la bombilla en su puesta en marcha, así como el brillo y el tiempo que tarda en cambiar su estado.

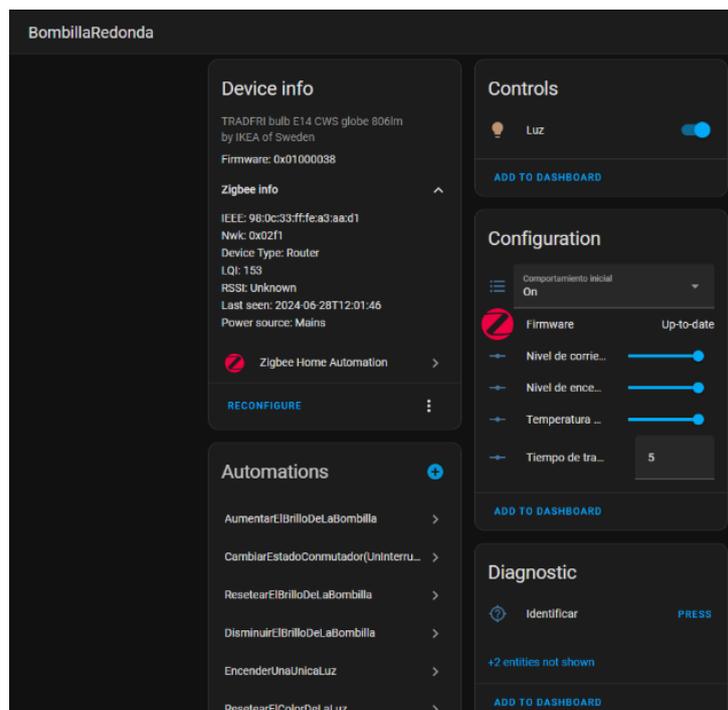


Figura 19. Página de características de la bombilla LED de 806 lúmenes.

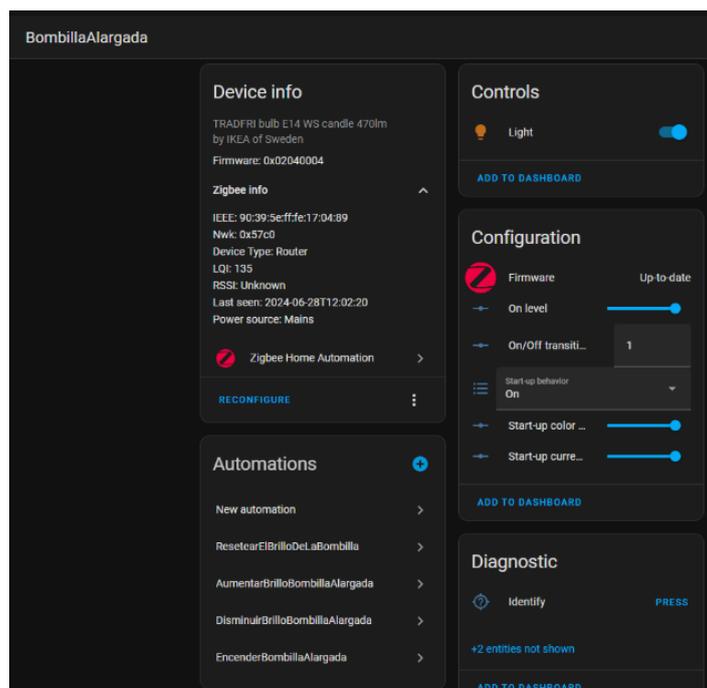


Figura 20. Página de características de la bombilla LED de 470 lúmenes.

3.2.5. Integración Mando Styrbär

Por último, conecté los mandos, primero el mando Styrbär de IKEA al cual llamé dentro del programa como MandoRectangular, que es un mando multifunción de cuatro botones. Para conectarlo al principio intente conectarlo directamente mediante el conmutador, pero la conexión no era correcta, pues no detectaba las pulsaciones de los botones, por lo que me dispuse a buscar información sobre el mando en concreto por si acaso hubiese que conectarlo de una manera concreta. Encontré en un foro del propio Home Assistant que había personas que habían tenido problemas a la hora de usarlo debido a una actualización reciente^[7], por lo que un usuario había diseñado un código de Python para asignarle un nuevo quirk y conseguir que funcionase, por lo que intente implementarlo. Seguía dándome problemas así que seguí buscando información hasta que llegué a otro foro^[8] en el que comentaban que los mandos por ser dispositivos que requieren de comunicación constante con el coordinador de manera bidireccional debían integrarse mediante un dispositivo repetidor. En el caso de Home Assistant estos dispositivos son, por ejemplo, las bombillas que ya tenía integradas, por lo que elimine el quirk de la configuración para evitar que me diese problemas en la conexión, y con la configuración básica intente conectarlo a través de la bombilla Tradfri LED 470 lm, ya que esta la instale permanentemente conectada a la red. De esta manera pude conectar correctamente el mando, que tiene únicamente las dos entidades básicas que se asignan a todos los dispositivos de batería e identificador. Esto ocurre porque los botones en ZigBee Home Automation no son entidades, si no que se registran como acciones que dispara el mando. Las acciones concretas del mando son, con sus nombres dados por el ZHA, “Turn On”, que se dispara mediante una única pulsación en el botón superior, “Turn Off”, equivalentemente al anterior es una única pulsación pero en el botón inferior, “Dim

Up”, que se dispara al mantener pulsado el botón superior, “Dim Down”, que se dispara al mantener pulsado el botón inferior, “Right”, disparado al pulsar el botón derecho como su nombre indica y “Left”, que de manera igual al anterior se dispara al pulsar el botón izquierdo. A pesar de parecer indicar ordenes concretas, los nombres de cada acción simplemente funcionan como nombres, es decir, se pueden usar de maneras diferentes a lo que el nombre original indica. En el caso de este trabajo por ejemplo no se usaran los botones para las acciones que sus nombres indican, si no que se asignarán otras acciones a estos.



Figura 21. Mando Styrbar.

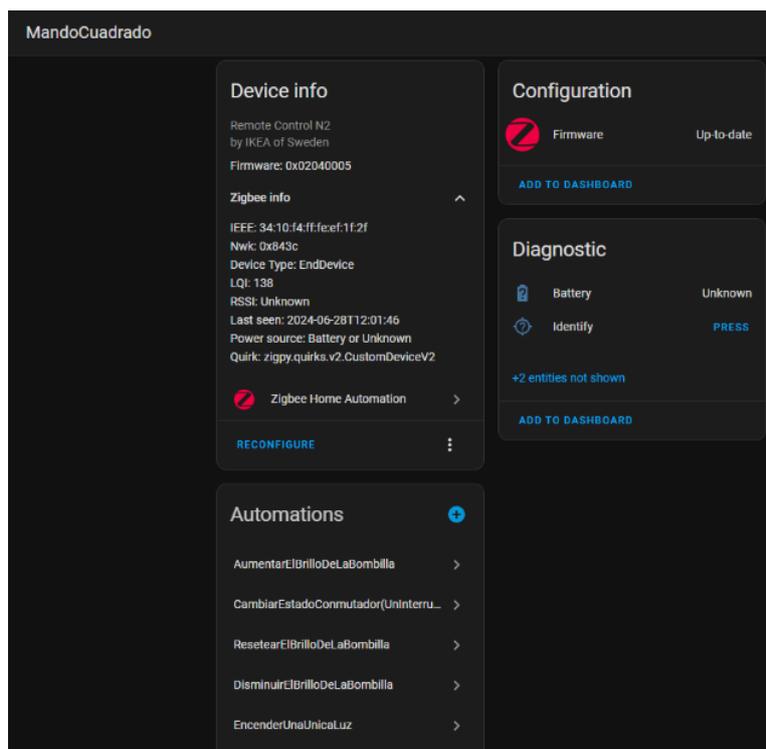


Figura 22. Página de características del mando Styrbär.

3.2.6. Integración Mando Rodret

Posteriormente conecte el mando Rodret de IKEA, que es un mando más sencillo que el Styrbär, ya que simplemente dispone de dos botones uno arriba que parece indicar el encendido y otro abajo que se entiende que indica el apagado. Para conectarlo, ya que había conectado previamente el otro mando fue más sencillo, ya que ya conocía el proceso para conectarlo, entrar en la página que describe a una de las bombillas e iniciar la integración a través de esta para que haga de repetidor. De esta manera pude conectarlo fácilmente, pudiendo acceder a la página que describe su integración, viendo que tenía como únicas entidades las dos asignadas por defecto a todos los dispositivos. Sus acciones posibles son, "Turn On", que se dispara al pulsar el botón superior del mando, que tiene además una pequeña hendidura para poder diferenciarlo, "Turn Off", disparado al pulsar el botón inferior del mando, "Dim Up", que se dispara al mantener pulsado el botón superior del mando y "Dim Down", que se dispara al mantener pulsado el botón inferior del mando. Obviamente este mando dispone de menos acciones, al disponer de menos botones, haciéndolo menos versátil que el mando anterior, pero permitiendo aun así varias automatizaciones.



Figura 23. Mando Rodret.

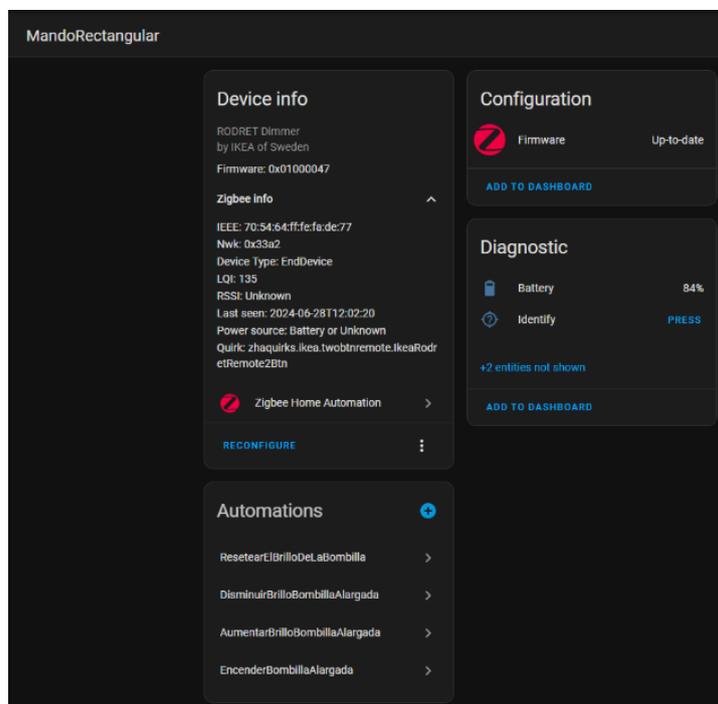


Figura 24. Página de características del mando Rodret.

3.3. Automatizaciones

Como ya comenté anteriormente las automatizaciones en Home Assistant se realizan en un código llamado YAML, aunque el propio software dispone de una interfaz gráfica, que sustituye al YAML para así facilitar la programación a personas que no conozcan el lenguaje. En mi caso opté por, en un principio, realizar las automatizaciones con la interfaz gráfica, a no ser que quisiese realizar algo más complejo por lo que optaría

por realizar automatizaciones concretas con el lenguaje YAML. Además, para algunas automatizaciones también use la extensión Node-Red.

Para las automatizaciones opté por intentar integrar todas las acciones posibles de los mandos o sensores para cambiar el estado o bien de los conmutadores o de las bombillas directamente, para así intentar extraer todo el potencial posible de los dispositivos que tenía.

3.3.1. Automatizaciones con el mando Styrbar

Con el mando Styrbar decidí controlar el conmutador conectado con un solo interruptor, además añadí una de las bombillas LED en paralelo a la salida del conmutador, por lo tanto, con el accionamiento de un solo conmutador podría controlar dos luces. Primeramente, con la acción de “Turn On” del mando decidí que fuese tanto el apagado como el encendido de las luces, es decir lo que hace el código en este caso únicamente es cambiar el estado del conmutador independientemente del estado en el que se encuentre al pulsar el botón superior del mando. Para el encendido y apagado existe la posibilidad de hacerlo por separado mediante comandos que encienden o apagan el conmutador, pero opte por opción de cambiar el estado independientemente porque me permitiría tener más botones disponible para distintas automatizaciones. En el código se puede ver que hay un bloque a parte del que cambia el estado del conmutador que indica que la bombilla se enciende, esto es para que, en caso de haberse cambiado el estado de la bombilla con otra automatización, como se supone que esta automatización lo que hace es encender ambas bombillas, para evitar cambiar el estado del conmutador y que una de las bombillas este apagada.

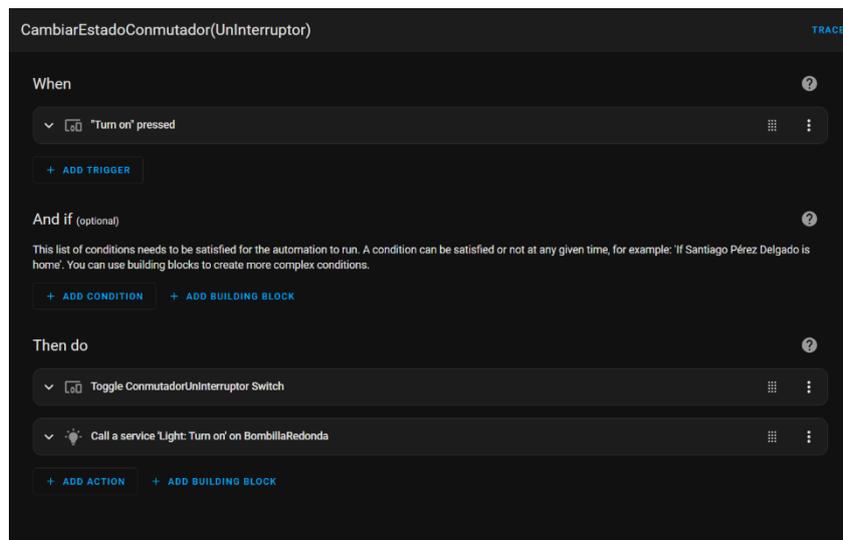


Figura 25. Código con el que se encienden las dos bombillas conectadas al conmutador con un interruptor.

Además de este código mediante la extensión de Node-Red, gracias a un bloque que permite evaluar si la hora está dentro de un rango concreto, dando verdadero si se cumple la condición y falso si no. Con este bloque, realicé una automatización que

permite evaluar el estado del conmutador, en caso de que este esté apagado y este dentro de un rango de horas, si se dispara la automatización que enciende el conmutador al encender la bombilla esta se encenderá con un brillo menor. Básicamente, esta automatización sería para disminuir el brillo de la bombilla a unas ciertas horas de la noche para, por ejemplo, mantener una mayor salud visual o incluso no deslumbrarte en exceso al encenderla dado el contraste con la luz que hay naturalmente de noche.

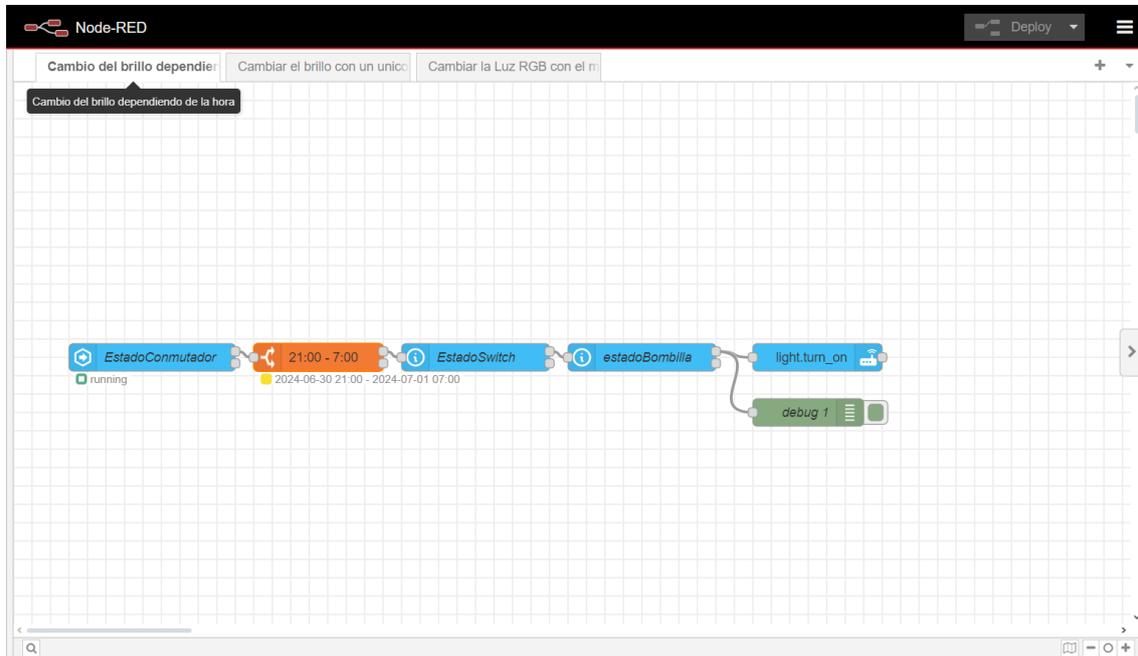


Figura 26. Diagrama en Node-Red para regular el brillo de la bombilla dependiendo de la hora que sea.

Con la acción de “Turn Off” la automatización la realizaría de manera muy similar, lo que, pensando en su uso en habitaciones con más de una luz, quise realizar una automatización donde el conmutador se encendiese, pero solo una de las bombillas se iluminase, ya que al tener integradas bombillas LED podría apagarlas aun habiendo corriente entrándole a la bombilla por el estado del conmutador. En el código esto está integrado de tal manera que primero se enciende el conmutador y tras un pequeño retardo para evitar problemas se apaga la bombilla LED

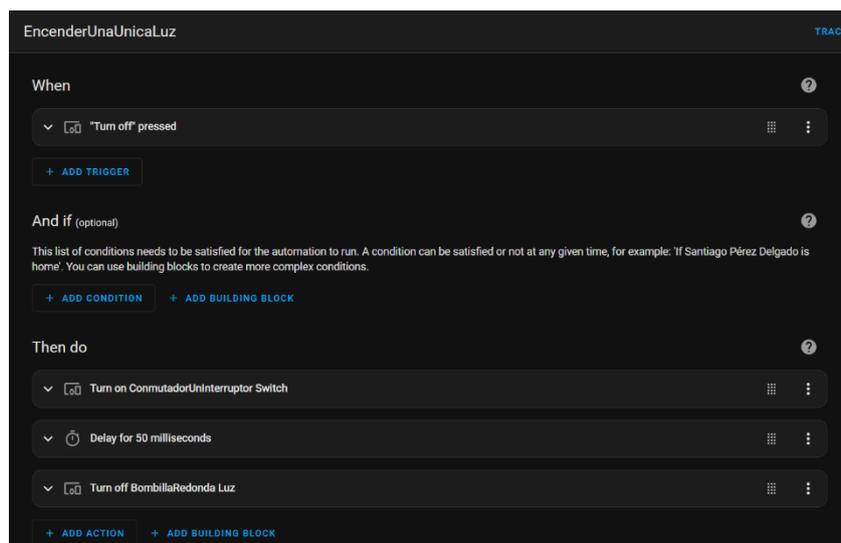


Figura 27. Código con el que se encienden una única bombilla de las que están conectadas al conmutador con un interruptor.

Además, teniendo esta automatización en cuenta pensé que sería interesante poder encender la bombilla LED sin necesidad de apagar el conmutador para luego volver a encenderlo con ambas bombillas encendidas, ya que en la situación real de una habitación con dos bombillas que iluminan partes distintas de la habitación realmente es ideal que se puedan encender las distintas bombillas por separado. Por tanto, realicé un código en el que la bombilla LED se encendería en caso de que el conmutador este encendido y se mantenga pulsado el botón superior.

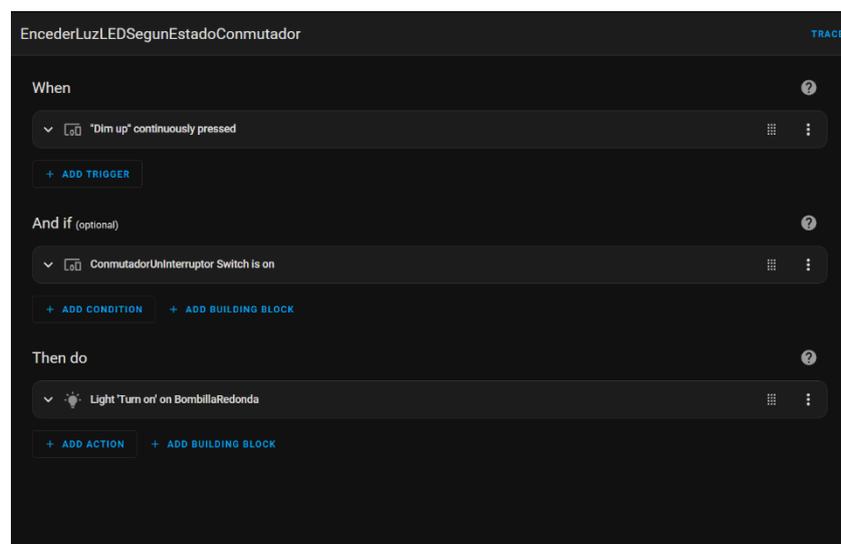


Figura 28. Código con el que se enciende la bombilla LED si el conmutador con un interruptor este encendido.

Ya pudiendo encender ambas bombillas o una nada más, me decidí a usar los botones de derecha e izquierda. Estos botones los usaría para controlar el brillo de la

bombilla LED conectada al conmutador con un solo interruptor, de esta manera al pulsar a la derecha se disminuye el brillo, y al pulsar a la izquierda se aumenta el brillo. Si se disminuye mucho el brillo la bombilla llega a apagarse, aunque sigue apareciendo como encendida para el sistema, por lo que simplemente aumentando el brillo volvería a encenderse.

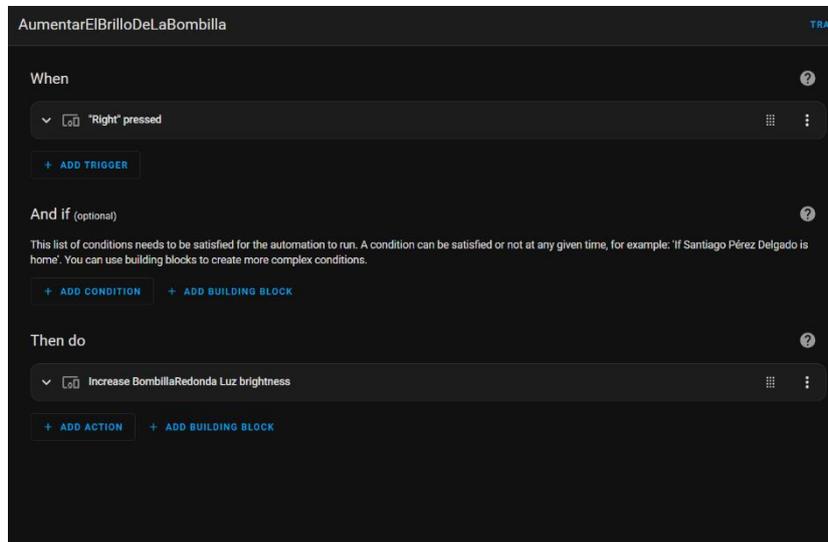


Figura 29. Código con el que se aumenta el brillo de la bombilla.

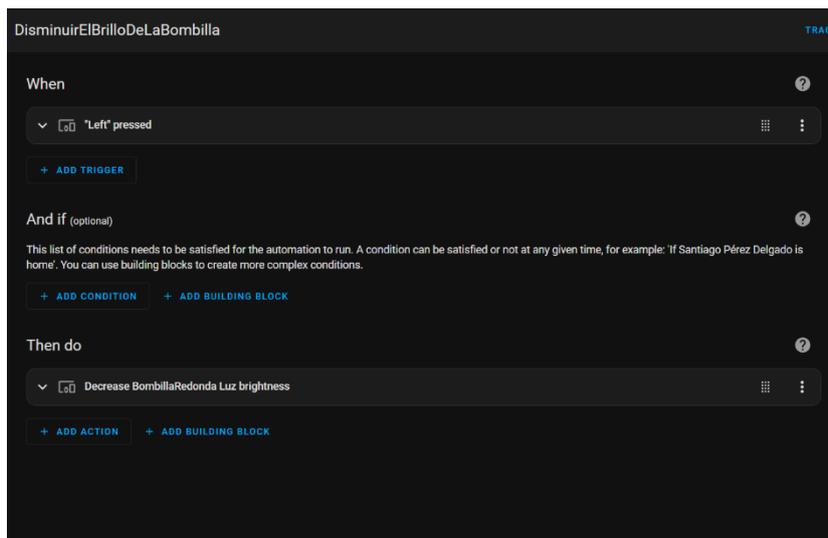


Figura 30. Código con el que se disminuye el brillo de la bombilla.

También para el caso de la bombilla LED, manteniendo el botón inferior se puede reestablecer el brillo de la bombilla a un valor medio independientemente de otras automatizaciones.

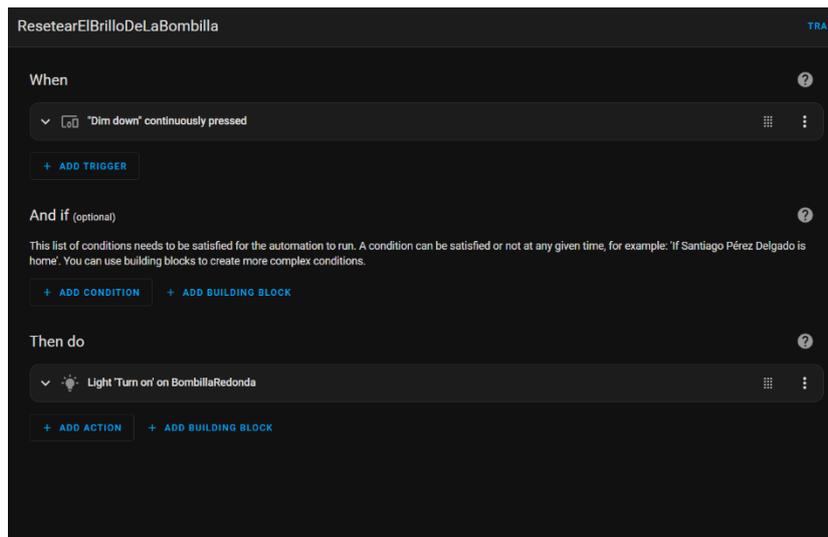


Figura 31. Código con el que se reestablece el brillo de la bombilla.

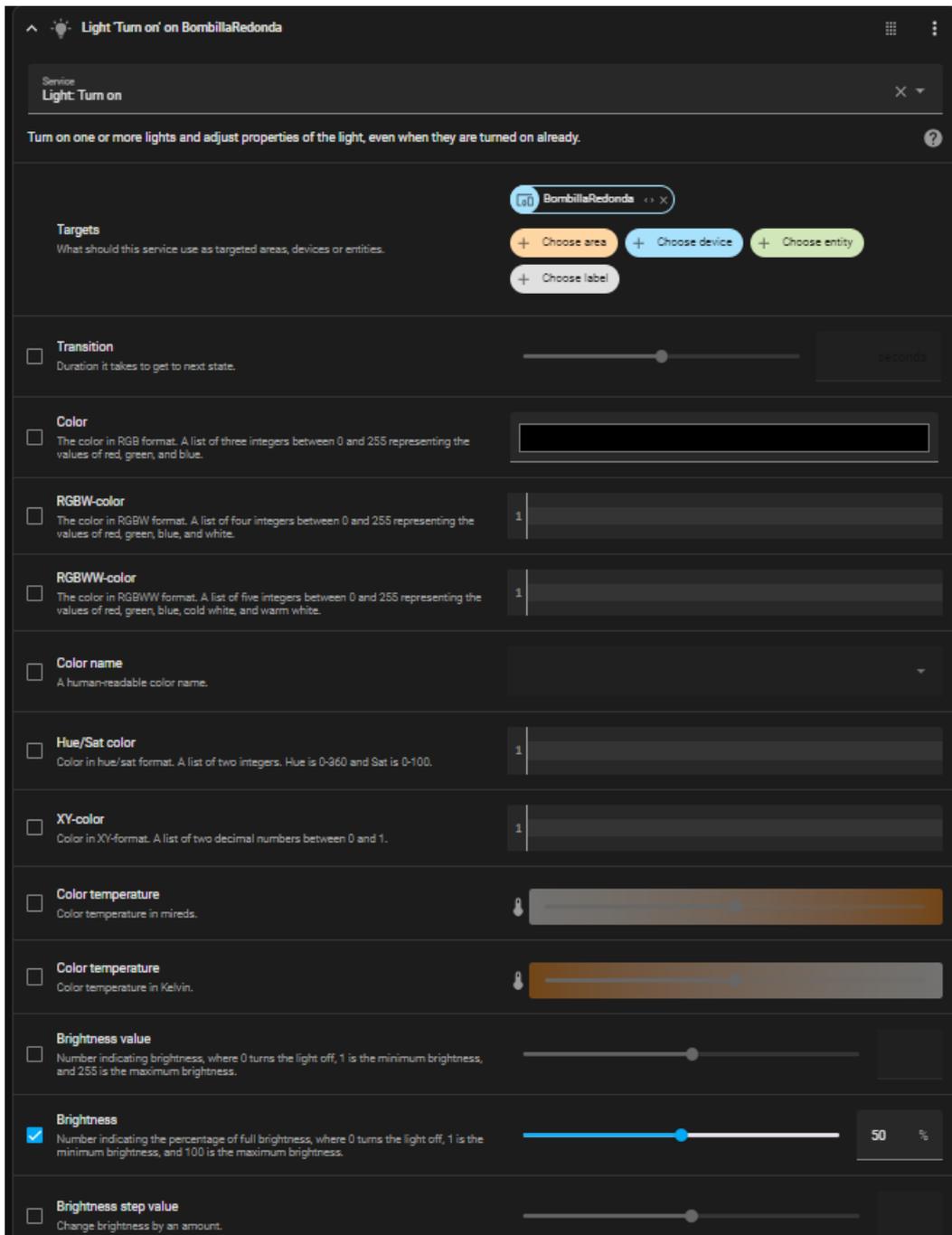


Figura 32. Detalle de la llamada a servicio con el que se reestablece el brillo de la bombilla.

También implementé, aunque no tiene tanta utilidad práctica, que se pueda devolver el color de la bombilla a un color común, blanco en este caso. Esto ya que al poder cambiarse el color de la bombilla por ser una bombilla LED, y poder cambiarse el estado de la bombilla desde el propio Home Assistant podría ser de alguna utilidad en caso de haberla cambiado. Está implementado al mantener pulsado el botón superior, aunque no de manera permanente, esta fue únicamente una prueba para ver si se podría realizar este cambio. En el código se ve que se indica que la bombilla se encienda, pero funciona aun con la bombilla encendida, ya que al intentar cambiar el estado de la bombilla a encendido no lo cambia por ya estar en ese estado, pero sí que introduce los datos del color que tiene que iluminar.

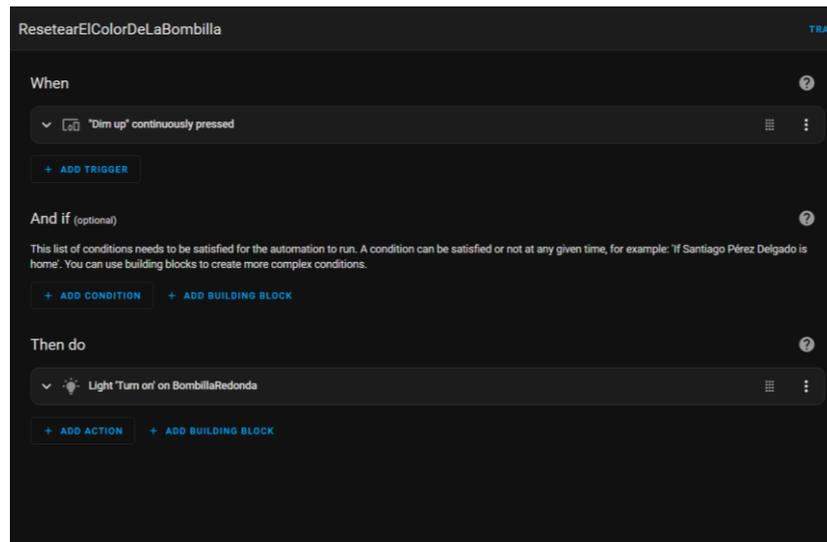


Figura 33. Código con el que se reestablece el color de la bombilla.

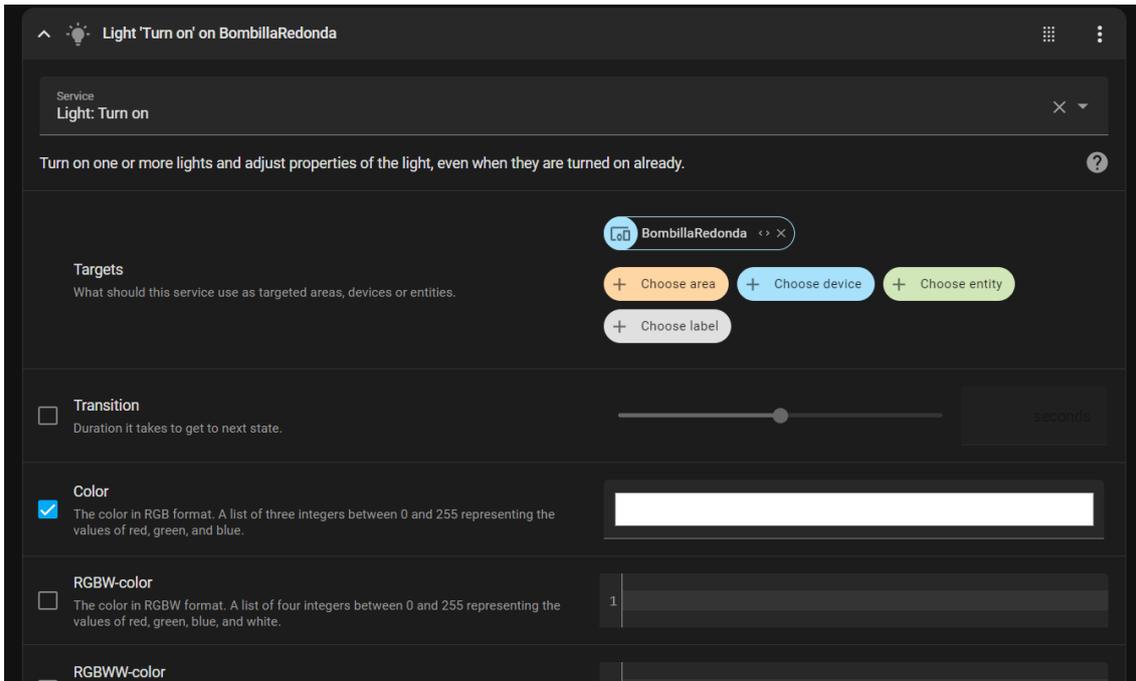


Figura 34. Detalle de la llamada a servicio con el que se reestablece el color de la bombilla.

Además, se podría programar una alternativa al aumento y disminución del brillo usando un solo botón. Tal como lo planteé yo establecí una automatización que se dispare al pulsar el botón con el que se quiere controlar el brillo, y sin ninguna acción. Dentro de una automatización de Home Assistant se puede insertar un bloque de decisión donde evalúa las condiciones para que se cumpla alguna de las acciones y las ejecuta. En este caso evalúa si el brillo de la bombilla es superior a 204, porque los cambios de brillo se harán de 51 en 51, por lo que si el brillo es superior superaría el máximo, en vez de sumarle los 51, se establecería el brillo de la bombilla a 1, que es el brillo mínimo, ya que en cero la bombilla esta apagada. El otro bloque evalúa lo contrario, es decir si el brillo no es superior a 204, caso en el cual le sumaria 51 al valor del brillo de la bombilla.

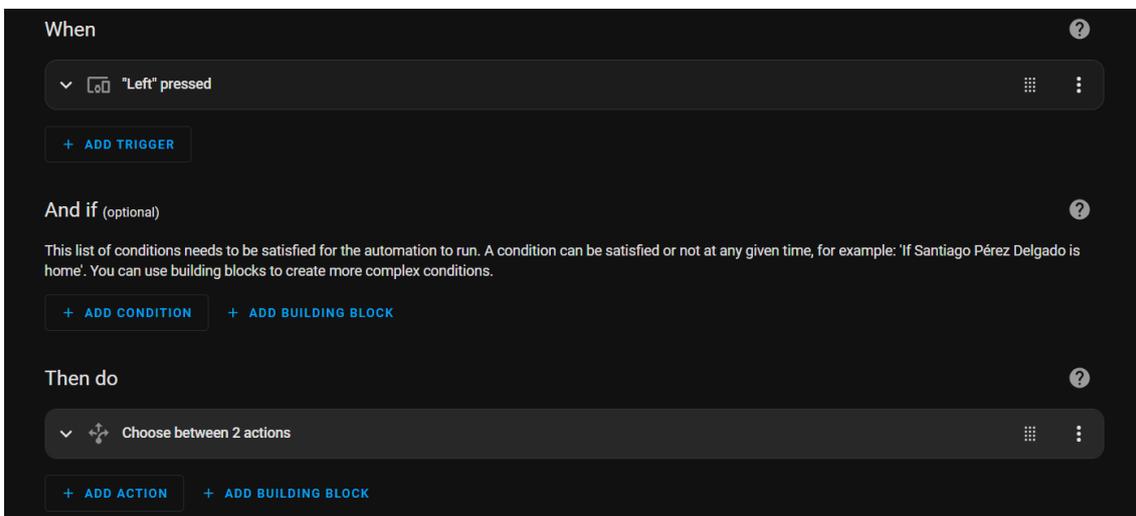


Figura 35. Código en para cambiar el brillo usando un único botón.

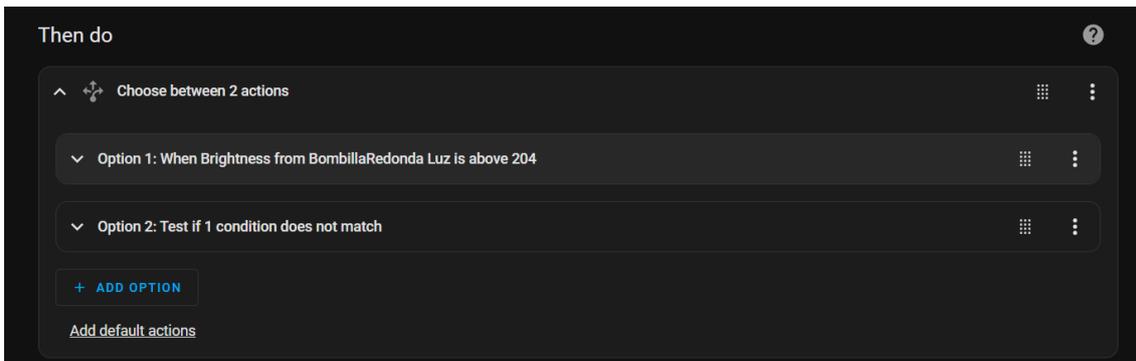


Figura 36. Detalle del bloque de decisión donde se reestablece el brillo a cero si va a superar el máximo o aumenta el brillo si el máximo no se supera.

Otra automatización que se podría realizar con el mando, aunque requeriría de casi todos los botones del mando por lo que sería solo aplicable si en una habitación hay más de un mando asignado, sería una que permitiese cambiar el color de la bombilla. Mediante Node-Red y haciendo automatizaciones donde solo incluyen el disparador y una acción sin utilidad, en este caso establecer una variable cualquiera a 1. De esta manera se podrían usar, por ejemplo, los tres botones inferiores para escoger dentro del espacio de color RGB cuál de los tres valores se quiere cambiar, por ejemplo, tal como los asigne yo, el Rojo sería el botón izquierdo, el Verde el botón inferior y el Azul el botón derecho. Cuando se dispara alguna de estas automatizaciones, es decir, se pulsa alguno de los botones se ejecuta una función que establece una variable para todo el diagrama de flujo que indica que color se ha seleccionado. En principio, los números que escogí fueron el 1,2 y el 3, aunque posteriormente pasé a asignar el 0,1 y 2 para facilitar la realización del código. Los números se asignan en orden, es decir el Rojo se indica con el 0, el Verde se indica con el 1 y el Azul se indica con el 2. El código dentro de los bloques de funciones, en naranja en la figura 38, se detallará en el anexo 7.1, 7.2 y 7.3 .



Figura 37. Asignación de números mediante funciones para cambiar partes del espacio de color RGB.

Luego en otra parte que no está unida como tal a este diagrama se evalúa primero si se mantiene el botón superior y luego si la luz esta encendida, dependiendo de que valor se haya establecido en la variable del flujo aumentara el valor de uno de los colores del espacio de color RGB. Como valor para aumentar o disminuir el valor RGB de la bombilla es de 51, por ser 255 divisible entre este, y para no tener unos cambios poco notables en la bombilla por ser muy pequeños. En caso de superar el valor de 255 haciendo demasiadas pulsaciones, el valor se mantendría en 255, al ser el máximo del espacio de color RGB, ya que no tendría sentido aumentar aún más ese valor. De manera

equivalente se puede reducir el valor de un color manteniendo presionado el botón inferior, y el código sigue la misma lógica que el del aumento del valor, con la diferencia que el límite para disminuir el color en este caso, obviamente, es el 0, ya que introducir valores negativos en el espacio de color RGB no tendría sentido. Asimismo, ya que la automatización requiere del uso de todo el mando prácticamente, decidí usar el único botón libre, el botón superior, para establecer el color de la bombilla en blanco, es decir un color de iluminación común, haciendo por tanto que en esta automatización el mando únicamente sirva como un control del color de la luz LED, por lo que en una instalación real haría falta otro sensor que encienda esa luz individualmente, ya que aunque técnicamente para cambiar la luz de la bombilla se llama al servicio de encender la luz, como está condicionado a que la luz este encendida ya antes, lo único que hará realmente será cambiar el color, no encenderla. El código que aumenta y disminuye el valor RGB se detalla en los anexos 7.4 y 7.5 .



Figura 38. Código en Node-Red para cambiar el color de la bombilla seleccionando la parte del espacio de color RGB que se desea cambiar.

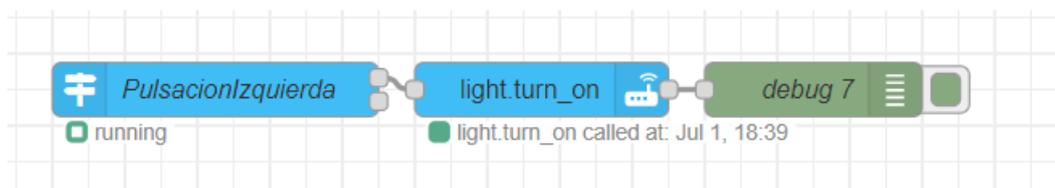


Figura 39. Código en Node-Red que permite reestablecer el color de la bombilla a blanco.

3.3.2. Automatizaciones con el Mando Rodret

En el caso del mando Rodret incluí menos automatizaciones ya que las opciones que permite al ser un mando de dos botones solo son menores. Igualmente siguiendo la lógica del mando anterior en este caso lo use para controlar la luz LED que está conectada directamente a la red, esto permite tener la luz apagada a pesar de que el interruptor al que está conectada este permitiendo el paso de corriente.

La primera de las automatizaciones que programe fue para poder cambiar el estado de la luz, al igual que en la otra existe también la posibilidad de que un botón encienda y otro apague, pero para poder aprovechar tanto la bombilla como el mando preferí que fuese el botón superior únicamente el que cambiase el estado de la bombilla, permitiendo usar más acciones para distintas automatizaciones.

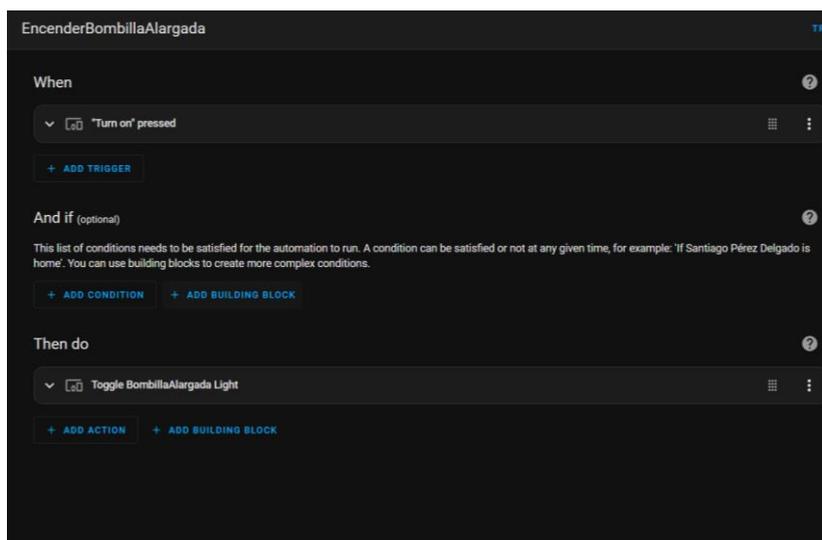


Figura 40. Código con el que se cambia el estado de la bombilla.

Posteriormente para cambiar el brillo con este mando es más tedioso, ya que no dispone de los botones que dispone el otro, por tanto, queriendo hacer que el aumento y disminución del brillo fuesen contrarios entre si opte porque sea el mantener pulsado el botón superior el que aumente el brillo de la bombilla y el botón inferior el que disminuya el brillo de la bombilla.

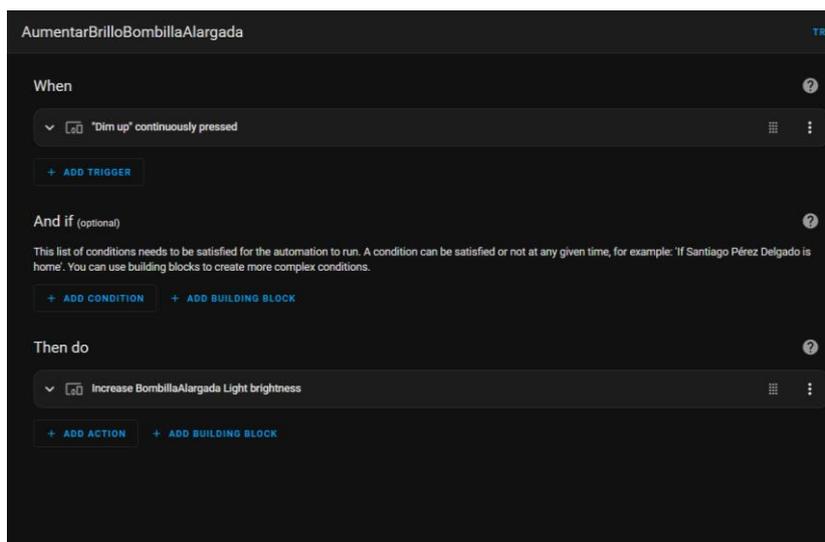


Figura 41. Código con el que se aumenta el brillo de la bombilla.

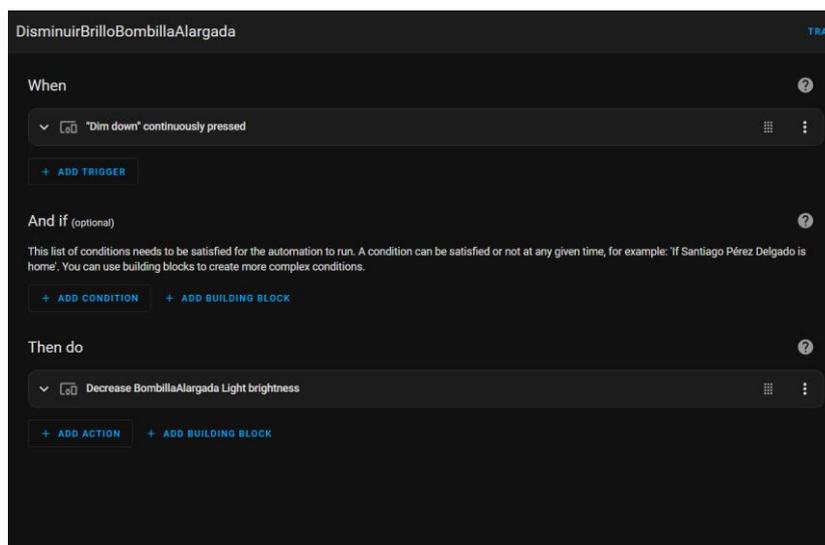


Figura 42. Código con el que se disminuye el brillo de la bombilla.

Finalmente, la última acción que quedaba por poder implementar era la de pulsar el botón inferior, el cual usé para devolver el valor de brillo a un valor concreto, en este caso el máximo de la bombilla. Ya que además esta bombilla es más sencilla, por lo que no se puede cambiar si quiera el color de esta, únicamente el brillo y la temperatura de la luz, y esta última solo podría establecerse a un valor concreto con las automatizaciones básicas de Home Assistant.

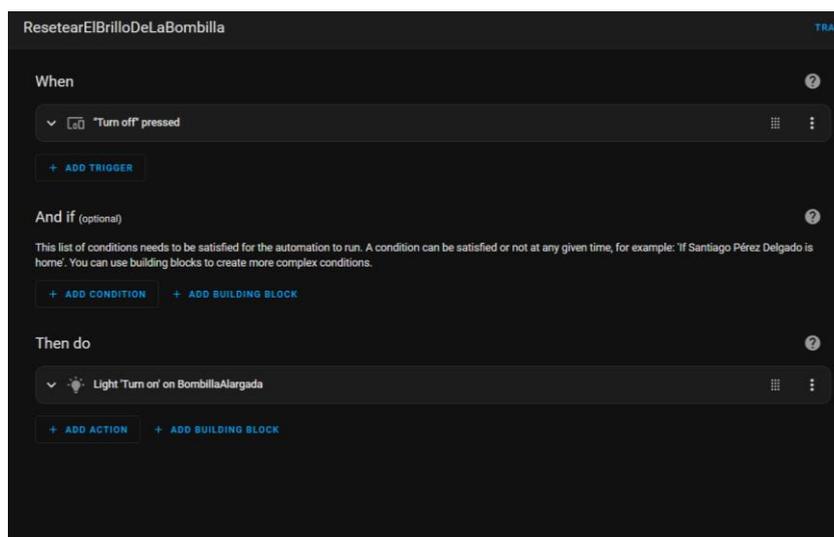


Figura 43. Código con el que se reestablece el brillo de la bombilla.

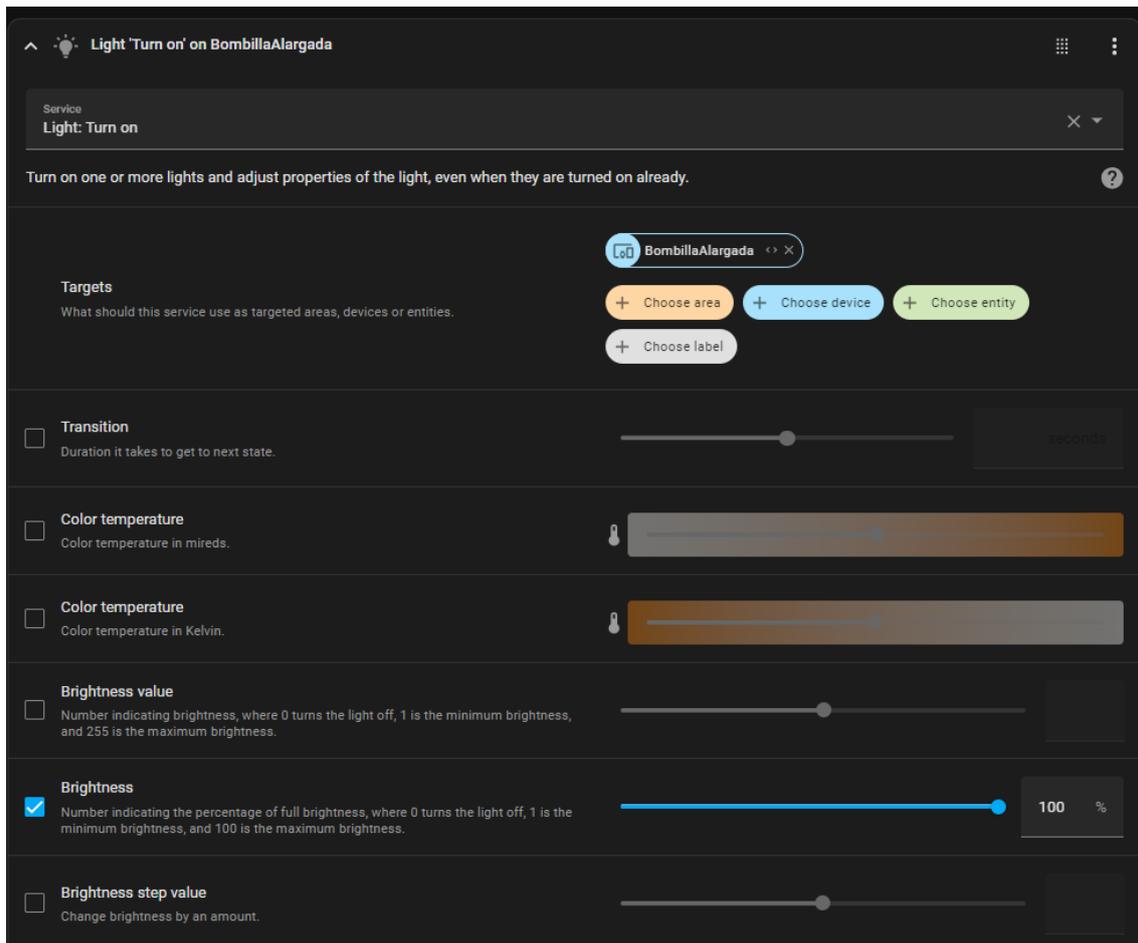


Figura 44. Detalle de la llamada a servicio con el que se reestablece el brillo de la bombilla.

Además de estas automatizaciones, dado que tenía otra bombilla LED decidí probar, a parte de estas automatizaciones hacer varias acciones distintas en esta otra bombilla LED, para ver exactamente todas las posibilidades que tiene una bombilla. Por ejemplo, con pulsaciones en el botón superior o inferior respectivamente se podría establecer el cambio de la temperatura de la bombilla, al mínimo o al máximo, ya que en un punto medio realmente no se nota mucho cambio.

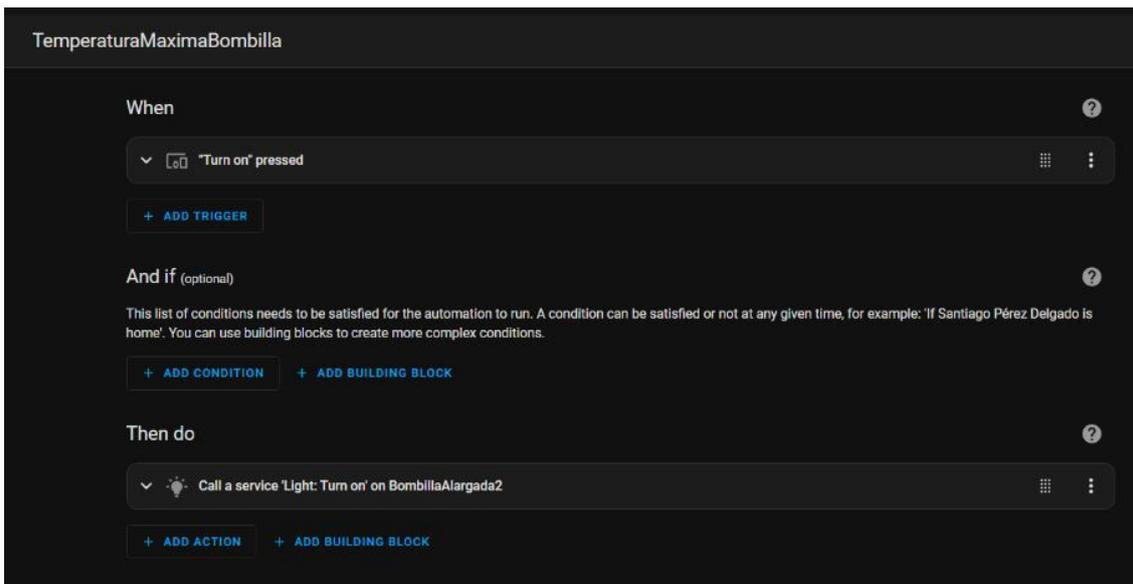


Figura 45. Código para establecer la temperatura de la bombilla al máximo.

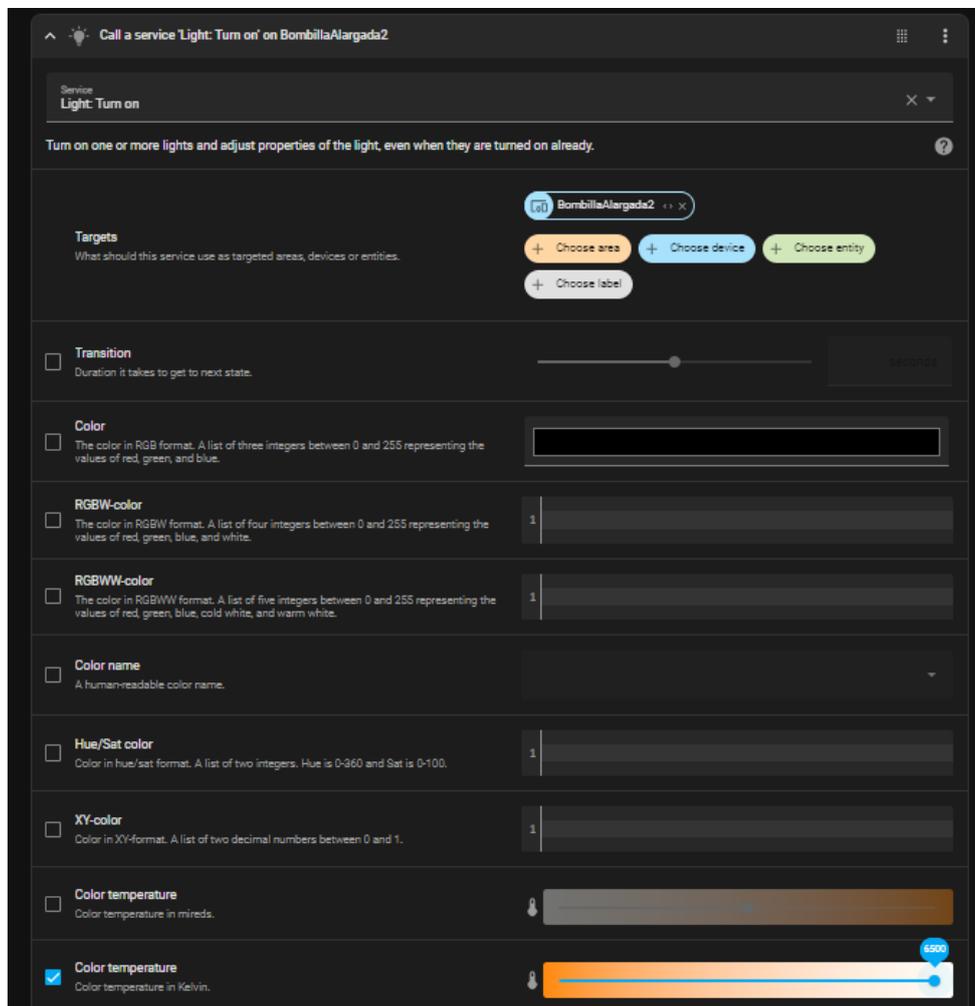


Figura 46. Detalle de la llamada al servicio de encender la luz donde se ve que la temperatura de la luz esta al máximo.

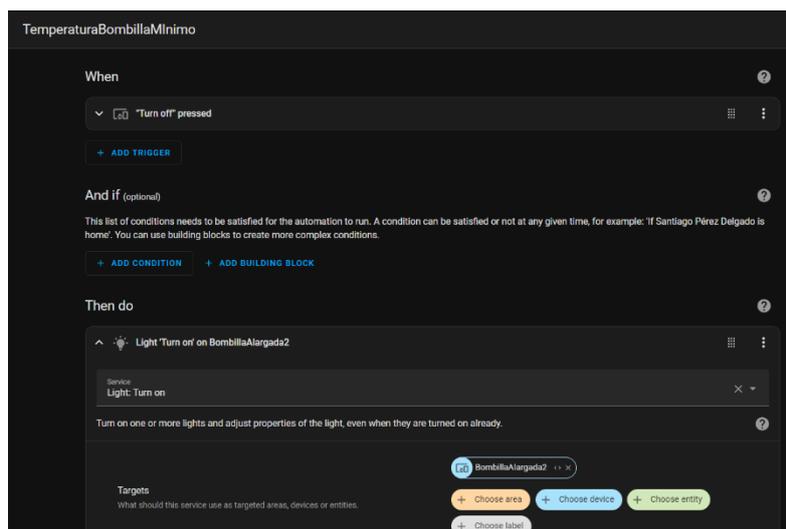


Figura 47. Código para establecer la temperatura de la bombilla al mínimo.

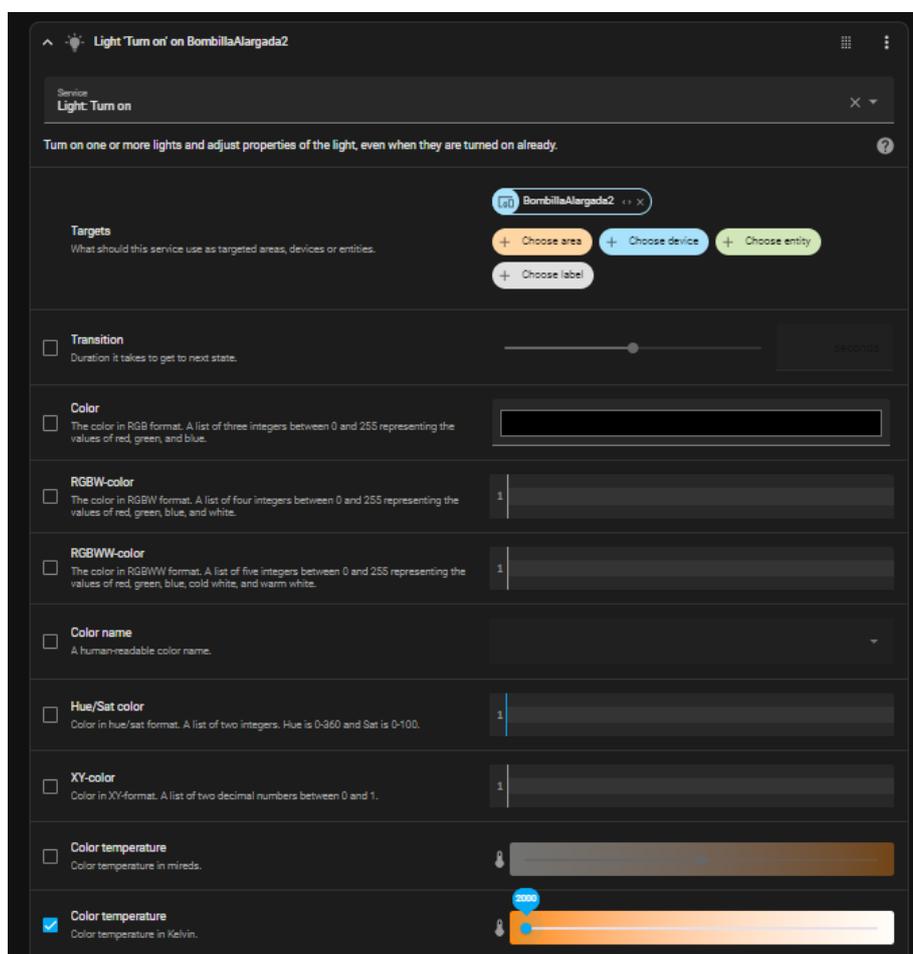


Figura 48. Detalle de la llamada al servicio de encender la luz donde se ve que la temperatura de la luz esta al mínimo.

De igual manera también se puede hacer que la bombilla parpadee, lo que esto solo se mantiene un tiempo concreto, pero mediante el Node-Red se podría forzar a que esta parpadee constantemente hasta que demos la orden de que esta pare de parpadear, en vez de hacerlo durante un tiempo concreto que viene dado por el Home Assistant.

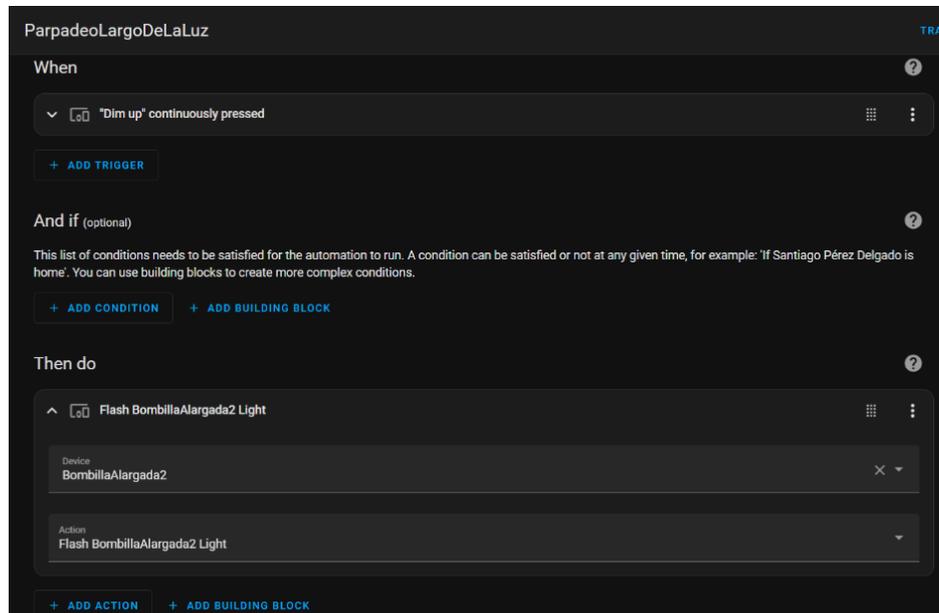


Figura 49. Código con el que se puede hacer que la bombilla parpadee.

En este caso el parpadeo esta hecho directamente como una acción de la bombilla, pero llamando al servicio de encender la bombilla también se puede hacer que esta parpadee, pudiendo elegir además si se desea un parpadeo largo o corto.

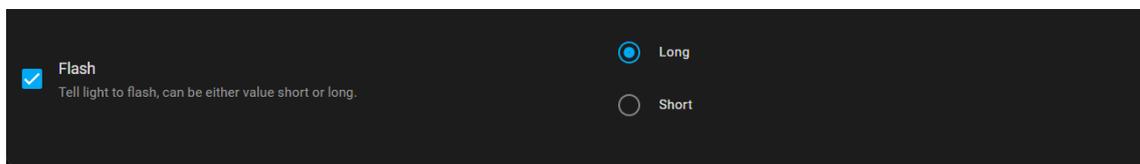


Figura 50. Detalle de la elección del parpadeo y sus modos en la llamada al servicio de encender la bombilla.

3.3.3. Automatización Con el Sensor de Puertas y Ventanas

En el caso del sensor de puertas y ventanas la automatización se vuelve más sencilla, pues este solo tiene dos estados, o está abierto, o está cerrado. Por lo que con este sensor controle el conmutador con dos interruptores, de tal manera que al abrir el sensor el conmutador se enciende, y al cerrarse el conmutador se apaga. Es preciso anotar que esta automatización simularía por ejemplo el caso de una puerta, para que al abrirse se encendiese la luz de la sala a la que se accede y al cerrarse se apague directamente. En caso de que se quisiese establecer un tiempo antes de que se apague

la luz se puede hacer en el propio código, ya que todas las condiciones pueden forzarse a que se evalúe si están activas un tiempo determinado. También podría hacerse que la acción se retrase simplemente añadiendo un bloque de “Delay” que hay en el propio Home Assistant, dependiendo de cómo se quiera realizar la automatización.

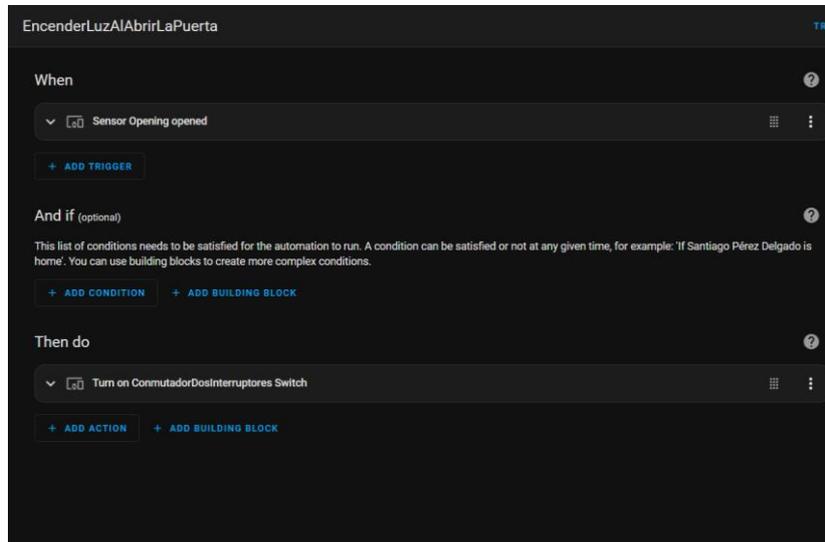


Figura 51. Código con el que es conmutador con dos interruptores se enciende al abrirse el sensor.

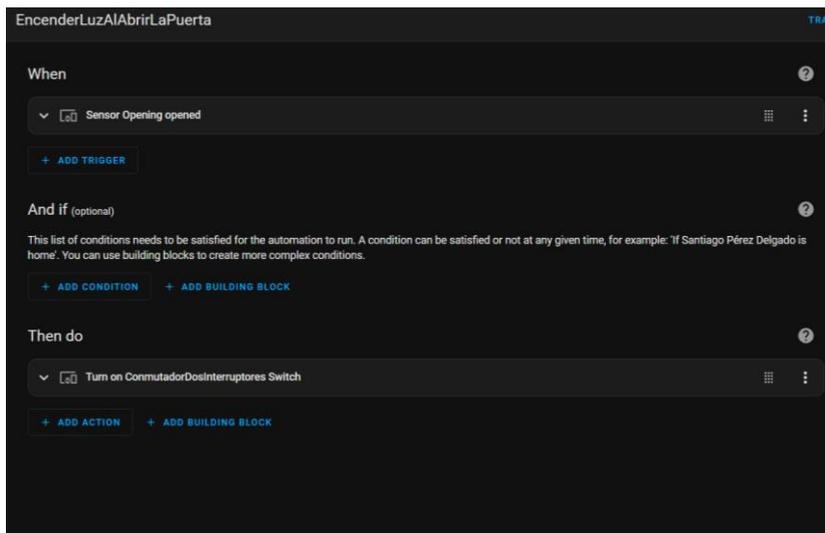


Figura 52. Código con el que es conmutador con dos interruptores se apaga al cerrarse el sensor.

3.3.4. Automatización con Comunicaciones

Además de las automatizaciones previamente explicadas también es posible incorporar en las automatizaciones distintas formas de comunicación entre el propio

Home Assistant y el usuario, principalmente por si este no se encuentra delante del ordenador donde se ejecuta Home Assistant viendo el estado de los dispositivos. Dentro del Home Assistant básico existe un gran abanico de posibilidades para establecer conexión con el usuario, los que implementé en mi caso son la conexión mediante un bot de Telegram y el uso de la propia aplicación del Home Assistant.

Para la conexión mediante Telegram primero requieres de crear un bot, siguiendo los pasos dados en la propia página de Home Assistant^[9], donde mediante otros bots existentes dentro de la aplicación de Telegram que te permiten, primero crear un bot y el segundo te permite obtener tu chat ID. Con el bot creado y conociendo tu ID de chat lo introduces en el archivo configuration.yaml del Home Assistant, para que este sepa que bot ha de usar y con que usuarios puede comunicarse.

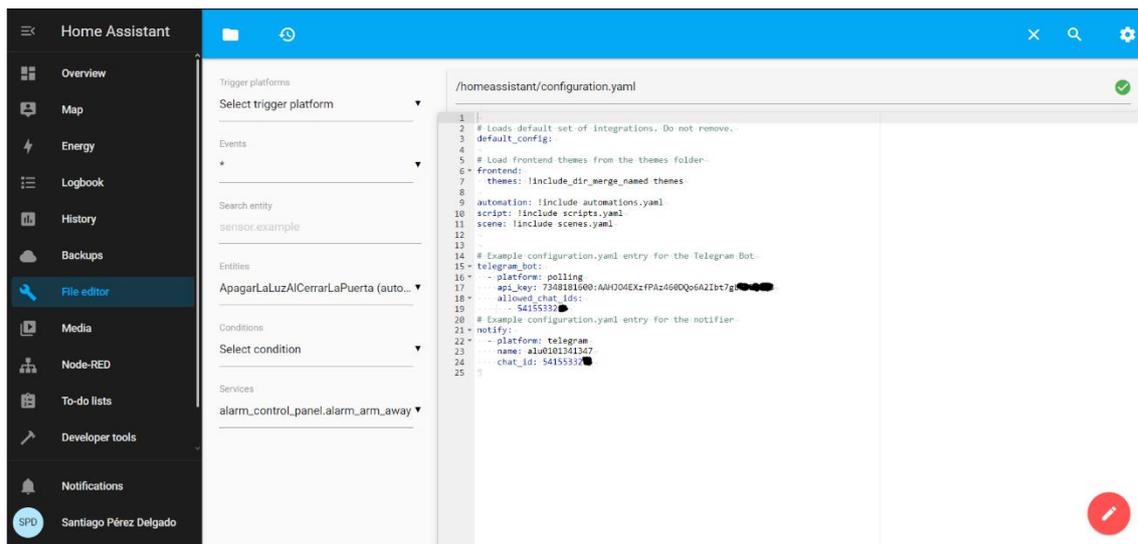


Figura 53. Archivo configuration.yaml con la inclusión de la integración de Telegram.

De esta manera el bot de Telegram podrá enviar mensajes a los usuarios cuyo chat ID este registrado en el archivo de configuración. Esto es útil, por ejemplo, para situaciones en las que la luz de una sala que no se usa durante un largo tiempo se mantiene encendida más de lo normal, pudiendo avisar de que se ha quedado encendida, así como para avisar cuando los mandos y el sensor de puertas y ventanas tienen poca batería, para que sepas que dentro de poco tendrás que cambiarle las pilas. Además, como se puede ver en el código de la figura 57, se puede escoger el receptor del mensaje en caso de que hayan autorizados varios usuarios dentro del mismo hogar.



Figura 54 Mensaje del bot enviado para comprobar que la integración funcionaba

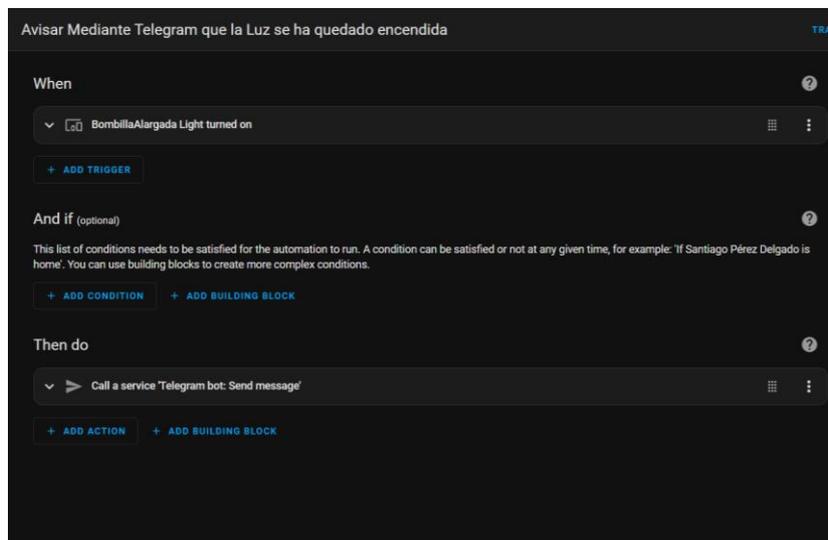


Figura 55 Mensaje del bot enviado para comprobar que la integración funcionaba.

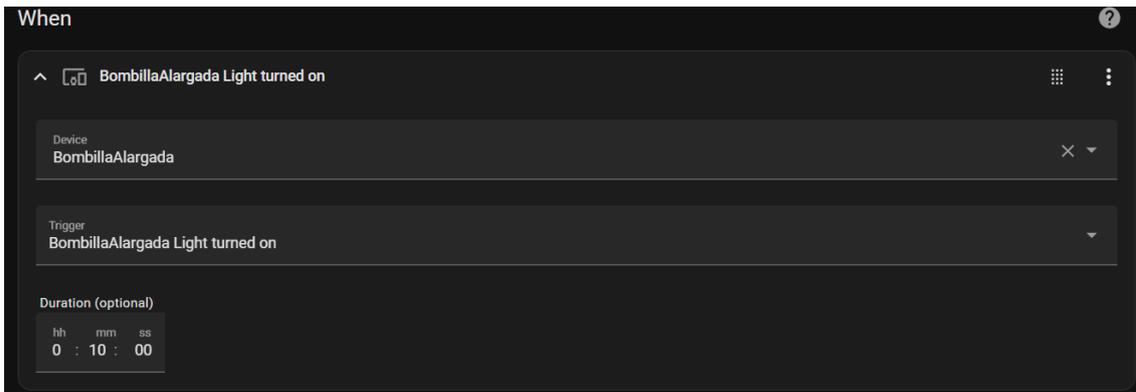


Figura 56. Detalle del disparador, donde se ve que está condicionado a que la bombilla este encendida diez minutos.

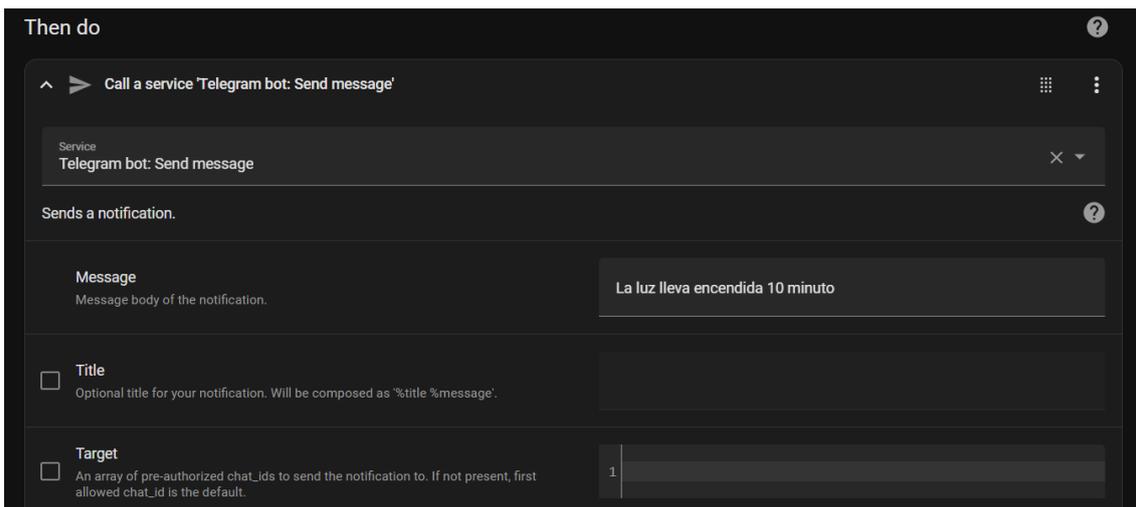


Figura 57. Detalle de la llamada al servicio de envío de mensaje por Telegram donde se ve el mensaje que se enviara pasados los diez minutos.

A parte del bot de Telegram también existe la posibilidad de comunicarte con el dispositivo móvil del usuario mediante la propia aplicación de Home Assistant, disponible en el App Store de iOS y la Play Store de Android. Al abrir la aplicación directamente se pondrá a buscar el servidor en la red, razón por la que probe la aplicación en mi casa, ya que la red Wi-Fi de la universidad no es la misma que la conexión LAN. Una vez encuentra el servidor en la red, establece conexión con este y ya puedes manejar todo el hogar de Home Assistant desde el móvil, teniendo un menú exactamente igual que el menú en el ordenador.

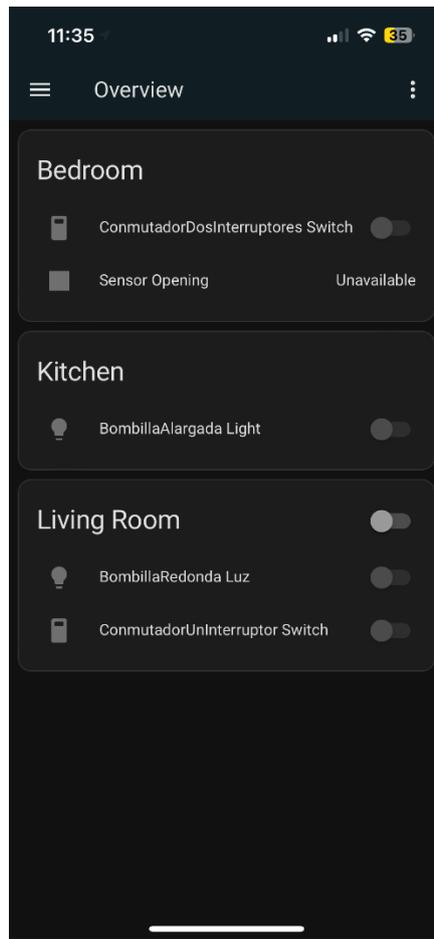


Figura 58. Vista de Home Assistant en la aplicación móvil.

En este caso se ve que todos los dispositivos no están disponibles porque para poder conectar la aplicación móvil a la instalación del ordenador se necesita poder encontrar el servidor en la red, cosa que en la sala de la universidad donde he realizado gran parte de este trabajo me era imposible.

Cuando has conectado el móvil al servidor directamente te aparece una integración del dispositivo móvil, por lo que ya puedes establecer en una automatización que se notifique mediante la aplicación móvil, pudiendo cambiar el mensaje, el título y el subtítulo del mensaje, por lo que para distintas automatizaciones se pueden enviar distintos mensajes con el propio servicio de notificaciones de Home Assistant.

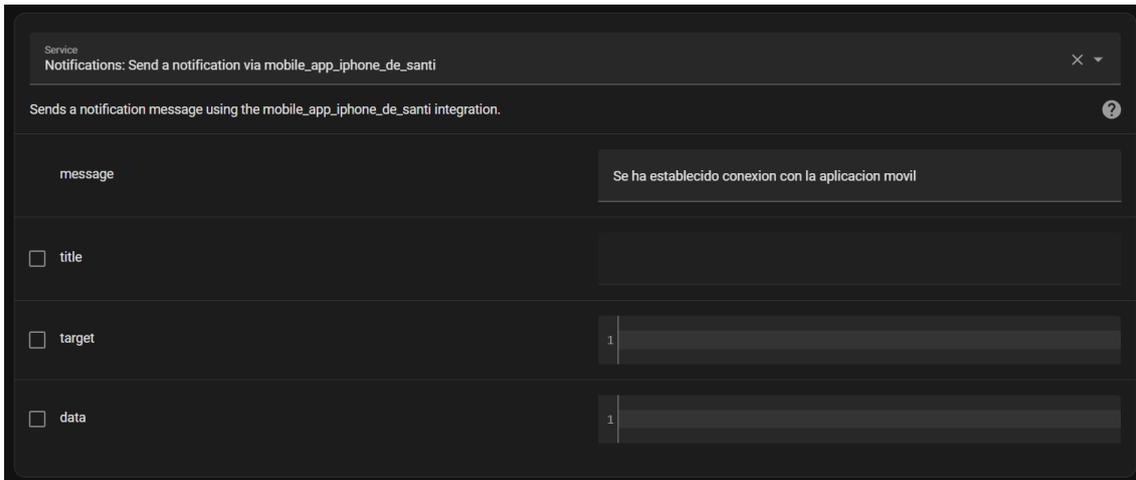


Figura 59. Código de llamada de servicio para comprobar que la integración de la aplicación móvil funcionaba.

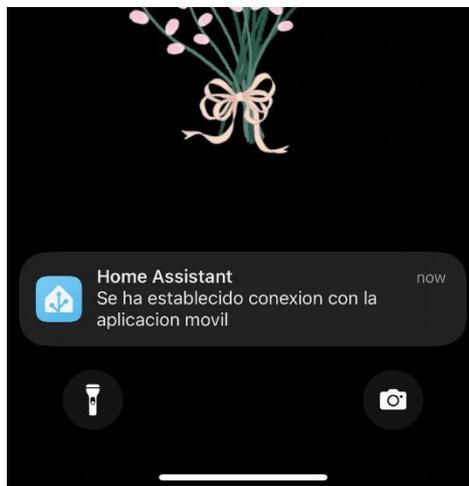


Figura 60. Notificación de Home Assistant enviada para comprobar que la integración funcionaba

Después de haber probado en mi móvil personal, un dispositivo iOS, también probe a realizar la integración con un móvil con un sistema operativo Android, para comprobar que esta funcionase y en efecto esta integración funciona correctamente también en dispositivos con sistemas operativos Android.

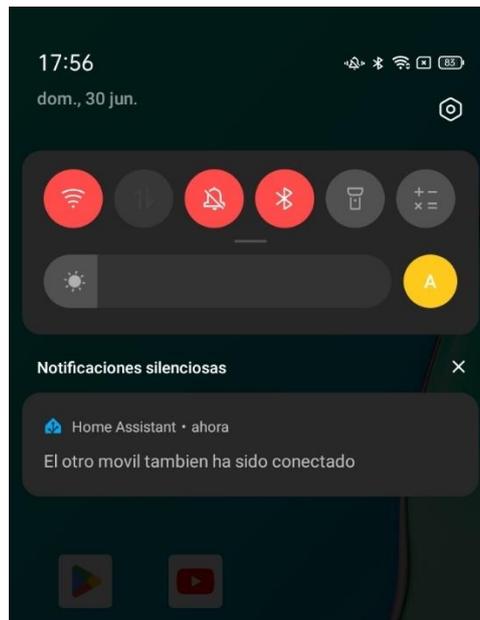


Figura 61. Notificación de la aplicación de Home Assistant en dispositivo Android.

Además, aunque en este caso no lo he implementado, se puede añadir a la notificación una imagen, video en directo e incluso cambiar el sonido que hace al saltar la notificación, de tal manera que permite identificar que la notificación es del Home Assistant.

3.4. Estudio Rango Mandos y Sensor

Finalmente, procedí a realizar el estudio del rango de los mandos a distancia y el sensor de puertas y ventanas, con la ayuda de mi tutor Alberto Hamilton. La manera en que realizamos el estudio fue primeramente realizando pulsaciones en el mando hasta que en una cierta distancia el Home Assistant ya no detectase las pulsaciones. Estas distancias no pudimos medirlas de manera exacta, ya que al ser grandes la posibilidad de usar un metro era imposible, y no disponíamos de otro material con el que medir distancias de manera exacta. Aunque en la información sobre los mandos dice que estos tienen un rango de 10 metros, nosotros, dentro de las limitaciones que teníamos, pudimos ver que, sin obstáculos de por medio, los mandos tienen un rango en línea recta de unos 30 metros. Además, ya que este estudio se hizo dentro de la universidad probamos la conexión desde otros pisos hacia las maquetas y el coordinador y ahí sí que vimos que el Home Assistant no detectaba las pulsaciones. Posteriormente, probamos si este rango podría ampliarse desplazando la maqueta que contenía la bombilla a través de la que se conectaron los mandos, por lo que nuestra suposición era que si alejábamos la maqueta del coordinador y nos alejábamos aún más con los mandos igualmente la señal podría llegar al coordinador, ya que la señal sería repetida por la bombilla. Esta prueba probó ser fructífera, pues pudimos aumentar el rango considerablemente, solamente alejando la maqueta unos diez metros, las pulsaciones del mando se

detectaban a una distancia mayor que antes, a unos cuarenta metros del coordinador, con obstáculos de por medio, y sin fallos en la detección de las pulsaciones.

3.5. Instalación de rasgos personalizados en Home Assistant

Como ya expliqué previamente los quirks, o rasgos a partir de ahora, son un tipo de archivos del propio Home Assistant que asignan a cada dispositivo sus entidades y acciones posibles. Puede ocurrir que ciertos dispositivos no tengan asignados unos rasgos concretos en el propio Home Assistant, por lo que a la hora de conectarlos mediante la integración no realizaran todas las acciones que tienen que o podrían realizar. Para ello existe la posibilidad de usar rasgos personalizados para ciertos dispositivos y aunque no acabe usándolo en este trabajo, como investigue sobre ello y en un cierto momento pensé que tendría que aplicarlos, quiero resaltarlo como una posibilidad más del Home Assistant, pues es realmente interesante a la hora de realizar integraciones.

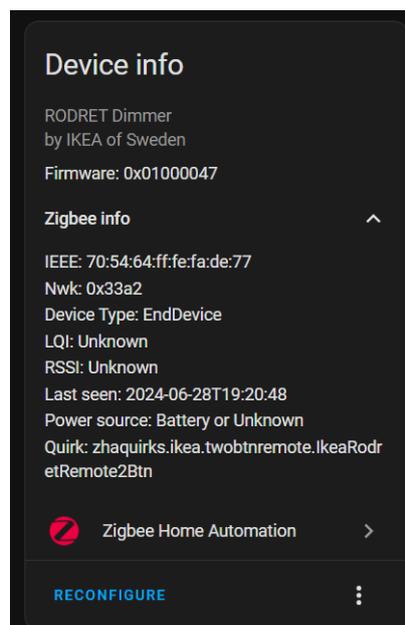


Figura 62. Información del mando Rodret, donde se ve la dirección del archivo de rasgos propio.

Para poder realizar la asignación de un archivo de rasgos personalizado en un dispositivo, primero habrá que permitir que se usen rasgos personalizados mediante el archivo de configuración, así como indicarle la dirección de estos. También será necesario tener los archivos de rasgos personalizados que se vayan a usar en esta carpeta antes de reiniciar el Home Assistant para hacer efectiva la configuración, de manera que así se puedan evitar errores en la configuración.

```

1 ~
2 # Loads default set of integrations. Do not remove.~
3 default_config:~
4 ~
5 # Load frontend themes from the themes folder~
6 frontend:~
7   themes: !include_dir_merge_named themes~
8 ~
9 automation: !include automations.yaml~
10 script: !include scripts.yaml~
11 scene: !include scenes.yaml~
12 ~
13 ~
14 # Example configuration.yaml entry for the Telegram Bot~
15 telegram_bot:~
16   platform: polling~
17   api_key: 7348181600:AAHJ04EXzfPAz460DQo6A2Ibt7gL79~
18   allowed_chat_ids:~
19     - 5415533~
20 # Example configuration.yaml entry for the notifier~
21 notify:~
22   platform: telegram~
23   name: alu0101341347~
24   chat_id: 5415533~
25 zha:~
26   database_path: /config/zigbee.db~
27   enable_quirks: true~
28   custom_quirks_path: /config/zha_quirks/|

```

Figura 63. Archivo de configuración con la inclusión del código para poder incluir rasgos personalizados.

Una vez creados los archivos y la carpeta y reiniciado el Home Assistant el rasgo personalizado debería aplicarse al reintegrar el dispositivo. Además, también resulta interesante ver en que constan estos archivos de Python. Estos archivos definen de manera precisa todas las acciones que puede realizar un dispositivo, definiendo de manera concreta todas sus entradas salidas y entidades como se puede ver en los múltiples archivos de rasgos disponibles en GitHub^[10].

4. Conclusiones y líneas abiertas

Una vez realizadas las automatizaciones e integraciones y vistas las posibilidades que ofrece tanto el software como el hardware usado en el trabajo es necesario valorar la efectividad de las herramientas usadas, así como las posibilidades que ofrece a parte de los elementos usados.

4.1. Conclusiones

La aplicación de Home Assistant como software de domótica ha demostrado ser en este trabajo un fácilmente adaptable y sencillo. El hecho de que permita complejizar las integraciones y automatizaciones dependiendo del nivel de conocimiento previo de la persona lo hace atractivo al facilitar el acceso a domótica. Además, el hecho de permitir distintas integraciones, a parte de la usada en este trabajo, lo hace increíblemente flexible. No sólo permite estas distintas integraciones, si no que estas pueden interactuar entre ellas, como pasó en el uso de la aplicación en este trabajo, por lo que sí se dispusiese de distintas integraciones en un hogar no habría problema para manejarlas. El uso de las automatizaciones también es limpio y sencillo, haciéndolo muy accesible, aunque con opciones para poder volverlo más complejo si así se quiere. Aunque es posible que a la hora de que un usuario sin conocimientos quiera aplicar automatizaciones más complejas pueda sentirse abrumado dado que el salto de lo más sencillo a lo más complejo parece ser muy amplio. Aun así, hay multitud de foros con guías de cómo realizar los procesos más complejos, por lo que si se desea se puede realizar estas automatizaciones e integraciones más complejas, como ya he explicado anteriormente. El problema que podría tener el uso de la instalación usada en este trabajo es que, aunque los dispositivos ZigBee usados no dependen por sí mismos de la red Wi-Fi, el uso del Home Assistant como controlador domótico sí. Esto podría producir que si se cae la conexión en el hogar la integración de ZigBee pudiera fallar.

ZigBee Home Automation como integración y por el uso de sus dispositivos también ha resultado ser una aplicación de la domótica favorable. Entre las facilidades que ofrece el Home Assistant y la manera de crear las redes mediante ZigBee hace que establecer la red sea un proceso sencillo y sin complicaciones una vez realizada la integración del conmutador. El hecho de que el rango de conexión sea amplio por lo demostrado en el estudio de rangos permite que si una instalación con dispositivos ZigBee tiene suficientes dispositivos y estos están bien distribuidos la conexión es muy improbable que falle. Además, buena parte de estos dispositivos disponen de múltiples cualidades que permiten una integración de estos mucho más agradable permitiendo modificar una gran variedad de cualidades de estos. Únicamente podría haber problemas con esta integración con algo que pude experimentar más de una vez, ya que para hacer el trabajo conectaba y desconectaba el coordinador, y es que a veces daba error al hacer la configuración de la integración al establecer la red fallaba. El error que se producía era porque no era capaz de detectar el coordinador en el puerto USB al no poder abrir el puerto serie, por lo que desconozco si este error proviene de haber conectado el coordinador con el ordenador recién encendido o si tarda en detectar el

coordinador al restablecer la red. Igualmente, este error se producirá únicamente en caso de que la red completa se cayese desconectando totalmente el Home Assistant, por lo que no debería ser común. Además, después de un cierto tiempo este error se resuelve solo, por lo que no debería suponer un gran problema igualmente.

4.2. Conclusions

The application of Home Assistant as home automation software has proven to be easily adaptable and simple in this work. The fact that it makes integrations and automations more complex depending on the person's level of prior knowledge makes it attractive by facilitating access to home automation. In addition, the fact that it allows different integrations, apart from the one used in this work, makes it incredibly flexible. Not only does it allow these different integrations, but they can interact with each other, as a step in the use of the application in this work, so if different integrations were available in a home there would be no problem to manage them. The use of automations is also clean and simple, making it very accessible, although with options to make it more complex if you want. Although it is possible that when a user without knowledge wants to apply more complex automations, they may feel overwhelmed since the leap from the simplest to the most complex seems to be very wide. Even so, there are many forums with guides on how to carry out the most complex processes, so if you wish, you can carry out these more complex automations and integrations, as I have already explained above. The problem that the use of the installation used in this work could have been that although the ZigBee devices used do not depend on the Wi-Fi network by themselves, the use of the Home Assistant as a home automation controller does. This could mean that if the connection in the home is dropped, the ZigBee integration could fail.

ZigBee Home Automation as an integration and by the use of its devices has also turned out to be a favorable application of home automation. Between the facilities offered by the Home Assistant and the way to create networks using ZigBee, it makes establishing the network a simple and hassle-free process once the switch integration is done. The fact that the connection range is wide as demonstrated in the range study means that if an installation with ZigBee devices has enough devices and these are well distributed, the connection is very unlikely to fail. In addition, many of these devices have multiple qualities that allow a much more pleasant integration of these allowing a wide variety of qualities to be modified. There could only be problems with this integration with something that I could experience more than once, since to do the work I connected and disconnected the coordinator, and sometimes it gave an error when making the configuration of the integration when establishing the network failed. The error that occurred was because it was not able to detect the coordinator in the USB port when it could not open the serial port, so I do not know if this error comes from having connected the coordinator with the newly turned-on computer or if it takes time for the coordinator to detect when resetting the network. Likewise, this error will only occur if the entire network goes down by completely disconnecting the Home Assistant, so it should not be common. Also, after a certain time this error resolves itself, so it should not be a big problem anyway.

4.3. Líneas Abiertas

Debido a que el Home Assistant es un software de gran profundidad en este trabajo obviamente no se ha cubierto todas las posibilidades que tiene este. La posibilidad de realizar varias configuraciones manuales es real, aunque en este caso no fue necesario. Gracias a la amplia comunidad que tiene Home Assistant es una posibilidad que puede resultar algo más fácil que si tuviese que hacerse desde cero e intentando corregir los errores que pudiesen surgir. Ya que existen multitud de códigos que permiten implementaciones precisas para distintos dispositivos y que no están implementados aun en Home Assistant. Otra posibilidad que estudiar es su instalación en otros dispositivos, en la propia página de Home Assistant clasifican la instalación en Windows y otros sistemas operativos como una de las más complicadas, por lo que estudiar el cómo se instala en otros dispositivos y que posibilidades ofrece podría resultar interesante. Como último punto que considero que podría resultar de interés, se encuentra la posibilidad de cruzar distintas integraciones, pues el uso de la aplicación móvil, por ejemplo, se considera una integración aparte, así como la valoración de la meteorología que tiene el propio Home Assistant. Por esto a lo mejor usar ZigBee Home Automation con otras integraciones que permitan otras posibilidades que el ZigBee Home Automation no tiene podría resultar interesante.

A parte de las posibilidades del propio Home Assistant, el ZigBee Home Automation resulta muy explotable como integración, en este trabajo se ha visto la integración de distintos tipos de sensores, luces y conmutadores, pero este dispone de muchas otras clases dentro de la integración, por lo que podría resultar interesante estudiarlas, especialmente veo interesantes las clases de números, que permite controlar entradas y salidas analógicas, como por ejemplo para estudiar la calidad del aire, los sensores, ya que el usado es un sensor binario y el resto de entidades sensores no se usan prácticamente, como puede ser el caso de las baterías, y el llamado lock, cerradura en inglés, que te permite ver el estado de una cerradura, abrirla o cerrarla.

5.Presupuesto

El presupuesto para la realización de este trabajo seria:

CONCEPTO	PRECIO	UNIDADES	SUBTOTAL	TOTAL
COSTES MATERIALES				119,29 €
COMPONENTES ELECTRÓNICOS				119,29 €
SonOff ZigBee 3.0 Dongle Plus	33,99 €	1	33,99 €	33,99 €
SonOff ZBMINIL 2	19,98 €	2	39,96 €	39,96 €
Sensor para puertas y ventanas	2,99 €	1	2,99 €	2,99 €
Bombilla LED LIVARNOLUX	3,39 €	1	3,39 €	3,39 €
Bombilla LED Tradfri 806 lm	11,99 €	1	11,99 €	11,99 €
Bombilla LED Tradfri 470 lm	9,99 €	1	9,99 €	9,99 €
Mando Strybar	9,99 €	1	9,99 €	9,99 €
Mando Rodret	6,99 €	1	6,99 €	6,99 €
COSTES DE EJECUCIÓN				6.000,00 €
TRABAJO DE INGENIERÍA				6.000,00 €
Horas de trabajo ingeniero técnico industrial	20,00 €	300	6.000,00 €	6.000,00 €
CONCEPTO			TOTAL	
COSTES MATERIALES			119,29 €	
COSTES DE EJECUCIÓN			6.000,00 €	
TOTAL COSTES			6.119,29 €	
GASTOS GENERALES (G.G.)			13%	795,51 €
BENEFICIO INDUSTRIAL (B.I.)			6%	367,16 €
SUMA DE G.G. Y B.I.			1.162,67 €	
COSTE ESTIMADO			7.281,96 €	
I.G.I.C			7%	509,74 €
COSTE TOTAL DEL PROYECTO			7.791,69 €	
EL COSTE TOTAL DEL PROYECTO ES DE SIETE MIL SETECIENTOS NOVENTA Y UN EUROS CON SESENTA Y NUEVE CÉNTIMOS				

Tabla 1. Presupuesto de realización del trabajo.

A parte se puede adaptar este presupuesto para conocer el coste que podría tener la instalación de un sistema así en un hogar. Para esto tomare como ejemplo mi propia casa, y a la hora de adaptar el presupuesto, existe la posibilidad de realizar la instalación únicamente poniendo las bombillas LED, ya que se pueden apagar o encender aun recibiendo corriente, o instalando conmutadores inteligentes junto a los interruptores para controlar las luces, en caso de querer mantener bombillas comunes. En este caso se ha realizado el presupuesto únicamente de la instalación teniendo en cuente mi propia casa, con el número de bombillas e interruptores que hay actualmente, y poniendo un mando por habitación por comodidad.

CONCEPTO	PRECIO	UNIDADES	SUBTOTAL	TOTAL
COSTES MATERIALES				271,73 €
COMPONENTES ELECTRÓNICOS				271,73 €
SonOff ZigBee 3.0 Dongle Plus	33,99 €	1	33,99 €	33,99 €
Sensor para puertas y ventanas	2,99 €	1	2,99 €	2,99 €
Bombilla LED Tradfri 470 lm	9,99 €	17	169,83 €	169,83 €
Mando Strybar	9,99 €	3	29,97 €	29,97 €
Mando Rodret	6,99 €	5	34,95 €	34,95 €

Tabla 2. Presupuesto para la instalación en un hogar común, instalando las bombillas inteligentes.

CONCEPTO	PRECIO	UNIDADES	SUBTOTAL	TOTAL
COSTES MATERIALES				380,60 €
COMPONENTES ELECTRÓNICOS				380,60 €
SonOff ZigBee 3.0 Dongle Plus	33,99 €	1	33,99 €	33,99 €
SonOff ZBMINIL 2	19,98 €	13	259,74 €	259,74 €
Sensor para puertas y ventanas	2,99 €	1	2,99 €	2,99 €
Mando Rodret	6,99 €	12	83,88 €	83,88 €

Tabla 3. Presupuesto para la instalación en un hogar común, instalando los conmutadores inteligentes.

En el caso del presupuesto de las bombillas se han escogido distintos mandos dependiendo de la habitación, pues hay habitaciones donde hay múltiples bombillas y algunos que se iluminan con una única bombilla. En el caso de los conmutadores, como estos controlarían cada uno un interruptor para apagar o encender no hará falta el mando más complejo si no el de dos botones.

6.Referencias

[1] Home Assistant Community <https://community.home-assistant.io> . Accedido por ultima vez el 8 de juio de 2024.

[2] GitHub – zigpy/zigpy: Library implementing a ZigBee Stack
<https://github.com/zigpy/zigpy> . Accedido por última vez el 4 de julio del 2024.

[3] Installation – Home Assistant <https://www.home-assistant.io/installation/> .
Accedido por última vez el 2 de julio de 2024.

[4] Windows – Home Assistant <https://www.home-assistant.io/installation/windows> . Accedido por última vez el 2 de julio de 2024.

[5] Windows – Home Assistant <https://www.home-assistant.io/installation/windows> . Accedido por última vez el 2 de julio de 2024.

[6] ZigBee Home Automation – Home Assistant <https://www.home-assistant.io/integrations/zha/> . Accedido por última vez el 2 de julio de 2024.

[7] IKEA STYRBAR V3 by MattWestb · Pull Request #3144 · zigpy/zha-devices · GitHub
<https://github.com/zigpy/zha-device-handlers/pull/3144> . Accedido por última vez el 29 de junio de 2024.

[8] Can add IKEA RODRET Wireless dimmer to ZigBee ZHA, but cannot see button press in logbook? – Configuration/ZigBee – Home Assistant Community
<https://community.home-assistant.io/t/can-add-ikea-rodret-wireless-dimmer-to-zigbee-zha-but-cannot-see-button-press-in-logbook/673338/7> . Accedido por última vez el 3 de julio de 2024.

[9] Telegram – Home Assistant <https://www.home-assistant.io/integrations/telegram/> . Accedido por última vez el 29 de junio de 2024.

[10] zigpy/zha-device-handlers: ZHA device handlers bridge the functionality gap created when manufacturers deviate from the ZCL specification, habdling deviations and exceptions by parsing custom messages to and from Zigbee devices
<https://github.com/zigpy/zha-device-handlers> . Accedido por última vez el 8 de julio de 2024.

7. Anexos

Código utilizado en los bloques de funciones de Node-Red para usar el mando Styrbar para cambiar el color de la bombilla LED.

7.1 Código para el cambio de color rojo

```
1 msg.seleccioncolor=0  
2 return msg;
```

7.2 Código para el cambio de color verde

```
1 msg.seleccioncolor=1  
2 return msg;
```

7.3 Código para el cambio de color azul

```
1 msg.seleccioncolor=2  
2 return msg;
```

7.4 Código para aumentar el valor del color escogido

```
1 var color= msg.color  
2 var i = flow.get('seleccioncolor')  
3 color[i]=color[i]+51;  
4 if(color[i]>255){  
5     color[i]=255;  
6 }  
7 msg.color=color  
8 msg.i = i  
9 return msg;
```

7.5 Código para disminuir el valor del color escogido

```
1 var color= msg.color  
2 var i = flow.get('seleccioncolor')  
3 color[i]=color[i]-51;  
4 if(color[i]<0){  
5     color[i]=0;  
6 }  
7 msg.color=color  
8 msg.i = i  
9 return msg;
```