



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Sistema para el Tratamiento de Variables en una Sala de Estimulación Sensorial

TRABAJO FIN DE MÁSTER

MASTER UNIVERSITARIO EN INDUSTRIA 4.0

Autor: **PABLO SAURA RÓDENAS**
Director: **JOSE ALFONSO VERA REPULLO**

Cartagena, 11/09/2023



Universidad
Politécnica
de Cartagena

Índice

1	Introducción	7
1.1	Motivación	7
1.2	Objetivos	8
1.3	Organización de la memoria	9
2	Estado del arte	12
2.1	Salas de estimulación sensorial	12
2.2	Sistemas de control de salas Snoezelen	16
2.2.1	DOIT Sensory Control	16
2.2.2	Eneso Sense	17
2.2.3	Handycat	17
2.2.4	BJ Adaptaciones	18
2.3	Plataformas de automatización de viviendas o edificios	18
2.4	Plataformas open-source para el desarrollo domótico	20
3	Desarrollo de un sistema de control domótico	23
3.1	Estado previo de la instalación y qué queremos controlar	23
3.2	Selección de hardware	¡Error! Marcador no definido.
3.3	Instalación de sistema operativo domótico	29
3.3.1	Infraestructura de red	29
3.3.2	Integraciones	31
3.3.3	Complementos	32
3.3.4	Configuración de dispositivos	33
3.3.5	Interfaz gráfica	44
3.3.6	Escenarios	47
3.3.7	Datos de salud	52
3.3.8	Programaciones horarias:	61
3.3.9	Sistema Multimedia	62
3.3.10	Integración de dispositivos externo	63
3.3.11	Asistente	63
3.3.12	Monitorización con telegram	64
4	Conclusiones	67
5	Trabajos futuros	68
6	Bibliografía	71
7	Anexo	73
7.1	Documentación código creación escenarios	73

Índice de Figuras

Figura 1.1.Imagen real sala Snoezelen.....	8
Figura 2.1.Tubo de burbujas	13
Figura 2.2.Fibras Ópticas.....	13
Figura 2.3.Proyector Aceite.....	14
Figura 2.4.Paneles interactivos	14
Figura 2.5.Difusor aromas	14
Figura 2.6.Cama vibratoria.....	15
Figura 2.7.Luz UV	15
Figura 2.8.Elementos táctiles	15
Figura 2.9.Elementos musicales	15
Figura 2.10.Panel DOIT Console	16
Figura 2.11.Eneso Sense	17
Figura 2.12.Handycat IPad	18
Figura 2.13.Communicator 5 BJ Adaptaciones	18
Figura 3.1. LED luz blanca.....	24
Figura 3.2. Panel luz UV	24
Figura 3.3. Downlight regulable	24
Figura 3.4. Proyector aceite	25
Figura 3.5. Proyector colores	25
Figura 3.6. Bola discoteca	25
Figura 3.7. Cama movimientos	26
Figura 3.8. Columna burbujas.....	26
Figura 3.9. Difusor de aromas.....	26
Figura 3.10. Cielo estrellado	27
Figura 3.11. Mazo de fibras ópticas	27
Figura 3.12. Panel táctil	27
Figura 3.13. Cama de bolas.....	28
Figura 3.14. Columpio para equilibrio	28
Figura 3.15. Mesa de dibujo.....	28
Figura 3.16. Pulsador inteligente Sonoff Touch.....	36
Figura 3.17. Dimmer inteligente Shelly	37
Figura 3.18. Placa ESP8266 4 relés	40
Figura 3.19. Pantalla principal Sala.....	45
Figura 3.20. Pantalla de mandos	45
Figura 3.21. Ventana de programación horaria	46

Figura 3.22. Pestaña Música	46
Figura 3.23. Ventana de selección de fuente de música	46
Figura 3.24. Ventana desarrollador.....	47
Figura 3.25. Pantalla escenarios.....	47
Figura 3.26. Caja Escenarios.....	48
Figura 3.27. Selector ayundante	48
Figura 3.28. Listado ayudantes.....	49
Figura 3.29. Flujo Nuevo Escenario	49
Figura 3.30. Nodo de creación de path.....	50
Figura 3.31. Flujo de log de horas de grabación	50
Figura 3.32. Flujo de grabación de eventos	50
Figura 3.33. Flujo de actualización de ficheros	51
Figura 3.34. Flujo eliminación ficheros.....	51
Figura 3.35. Flujo de reproducción de ficheros	51
Figura 3.36. Depuración flujo análisis servicios Mi Band 3.....	53
Figura 3.37. Captura servicios nRF Connect	54
Figura 3.38. Flujo lector de pulsaciones.....	54
Figura 3.39. Listado de entidades de Fitbit	57
Figura 3.40. Captura Fitbit con aplicación	58
Figura 3.41. Captura actualización aplicación Fit2Fit	59
Figura 3.42. Captura Google Fit.....	59
Figura 3.43. Captura servicios MQTT Explorer	60
Figura 3.44. Valor entidades Mi Band 3	61
Figura 3.45. Flujo de exportación de datos históricos de pulsera	61
Figura 3.46. Captura ventana de programación horaria	62
Figura 3.47. Captura asistente Home Assistant	64
Figura 3.48. Comando start bot Telegram.....	65
Figura 3.48. Comando ping bot Telegram.....	65
Figura 3.49. Comando statusSala bot Telegram	65
Figura 5.1. Captura ChatGPT creando automatizaciones	69

Índice de Tablas

Tabla 1-1. Listado de tareas	10
------------------------------------	----

1 Introducción

Una sala multisensorial es una estancia dotada de elementos que proporcionan estímulos sobrecargados y controlados, ya sea mediante el empleo del sonido, olor, tacto, temperatura u otros, que pretenden conseguir una mayor respuesta del paciente, de manera que pueda alcanzar una mejor comprensión del medio y de sí mismo a través de la exploración en un lugar que invita a la relajación de una manera segura, además de poder analizar los avances de manera más detenida y concreta.

Los humanos percibimos el mundo por lo que hemos vivido y experimentado como decía *Aristóteles* “*Nada está en la inteligencia que primero no haya pasado por los sentidos*”. No le falta verdad en que la concepción del mundo por una persona ciega o sorda seguro que es muy distinta a alguien que puede disfrutar de estos sentidos.

Este tipo de salas creadas por dos psicólogos neerlandeses fueron creadas en los años 70's como terapia para individuos con discapacidad severa, y ya está ya muy implantado a lo largo del mundo. En la actualidad, existen multitud de empresas que realizan instalaciones de salas multisensoriales llave en mano o venden los elementos por independiente.

Además, el concepto sala multisensorial o Snoezelen, que significaría algo como oler y relajar, se sigue manteniendo gracias a la Asociación de Estimulación Sensorial y Snoezelen, (ISNA España).

El público de estos espacios va desde niños a adultos con distintos ritmos de aprendizaje, personas con autismo, parálisis cerebrales, demencia, trastornos mentales, etc.

1.1 Motivación

La motivación principal para este proyecto es la de poner en marcha un sistema inteligente para el control de la sala de estimulación sensorial, de ahora en adelante EMS situada en el centro CDIAT Virgen de la Caridad de Cartagena, que permita al terapeuta centrarse en la respuesta al estímulo del paciente y no en la creación del estímulo, que en muchos casos supondría dejar al paciente sin supervisión, sobre todo si se trata de un bebé.



Figura 1.1. Imagen real sala Snoezelen

En el desarrollo de este proyecto se pretende seleccionar y adaptar una interfaz que sea amigable y fácil de usar, y que a la vez permita una fácil actualización, que permita que el funcionamiento de una sala con tantos beneficios para el paciente permanezca en desuso.

Aprovechando que dicha sala es fruto de un proyecto social y que no se cuenta con presupuesto para mejorar constantemente, se trata de crear un sistema en el que todo sea integrable y todo a un bajo coste, además de fácilmente ampliable.

1.2 Objetivos

El objetivo principal de este trabajo es la puesta en marcha del hardware y software que permita controlar los elementos de estimulación sensorial de manera fácil, e incluso crear determinados escenarios en base a las necesidades de cada ejercicio.

- Analizar las posibilidades de la sala de estimulación sensorial y la aplicación de sistemas inteligentes a ella.
- Diseñar un sistema que sea fácilmente integrable y que tenga muchas posibilidades de actualización.
- Integrar utilizando dispositivos inteligentes el mayor número posible de elementos de la sala sin entorpecer a la manera tradicional de funcionamiento.
- Disponer de una plataforma de control en tiempo real de la sala, que permita crear rutinas de trabajo.

- Adaptar el hardware de la sala a nuevas tecnologías utilizando para ello software *open source* y dispositivos fáciles de integrar.
- Exportación de datos de uso de la estancia, así como de estado del paciente.
- Crear una unidad de control que sea más integrada y estable, y que permita solucionar pequeñas averías de manera rápida, e incluso, la realización de mejoras por parte de alumnado de la universidad.

1.3 Cronograma

El presente proyecto se ha llevado a cabo durante un periodo de unos 7 meses dentro de los cuales ha habido diferentes interrupciones y para el que se ha podido trabajar una media de unas 2-4h semanales. Es por ello su extensión en el tiempo.

A continuación, se muestra un desglose de las tareas más importantes llevadas a cabo para el desarrollo del mismo junto con una estimación en días de lo que realmente ha costado a razón de como hemos dicho, unas 2-4h al día, sumando así una cantidad en torno a las 180h de trabajo junto a un diagrama Gantt que pretende mostrar el avance del proyecto en el tiempo.

Tabla 1-1. Listado de tareas

Tareas	Fecha de inicio	Fecha final	Días
Reunión Inicial	10-oct-22	11-oct-22	1
Recolección de información	31-oct-22	25-nov-22	5
Adquisición de elementos para la sala	14-nov-22	10-dic-22	2
Análisis de alternativas	21-nov-22	20-dic-22	1
Instalación en entorno de pruebas	01-dic-22	22-dic-22	2
Configuración de persianas	09-ene-23	20-ene-23	1
Configuración de enchufes	10-ene-23	20-ene-23	1
Configuración de Volumio y Broadlink	12-ene-23	20-ene-23	1
Creación interfaz 3D y 2D	01-feb-23	17-feb-23	8
Pruebas de integración con relojes inteligentes	20-feb-23	15-mar-23	5
Creación de herramienta de escenarios	15-mar-23	03-abr-23	5
Creación herramienta monitorización pulsaciones	10-abr-23	28-abr-23	2
Exportación de datos de escenarios y pulsera	24-abr-23	28-abr-23	2
Manual de uso Instalación	10-abr-23	28-abr-23	4
Memoria de Trabajo de Fin de Estudios	22-abr-23	02-jun-23	20

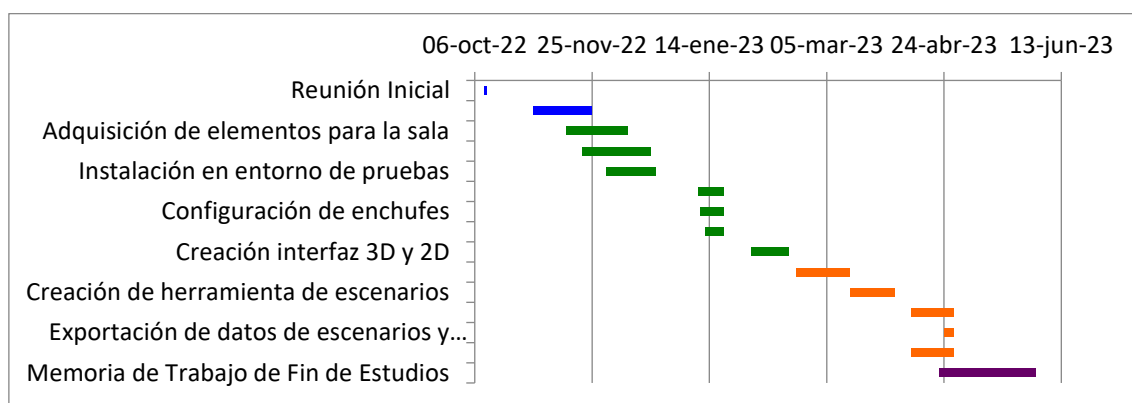


Figura 1.2. Cronograma de actividades

1.4 Organización de la memoria

La memoria constará de los siguientes capítulos:

- Estado del Arte. En este apartado veremos un resumen general de las distintas opciones tecnológicas para automatizar una estancia, además de analizar el avance actual de las salas de estimulación con sus usos y componentes.
- Desarrollo del sistema de control domótico. Siguiendo el orden natural, tras el análisis se llevará a cabo la puesta en marcha del sistema ideado junto con las configuraciones seleccionadas.
- Conclusiones y trabajos futuros. Por último, se analizará la herramienta desarrollada y se buscarán posibles cambios, mejoras o ideas para futuras integraciones al sistema.

2 Estado del arte

2.1 Salas de estimulación sensorial

Después de la introducción llevada a cabo en el apartado **¡Error! No se encuentra el origen de la referencia.** de esta memoria, se ha llevado a cabo un proceso de análisis más profundo en lo que son estas salas, su función, objetivos y qué beneficios tienen.

Una sala de estimulación o *Snoezelen*, es un espacio con un diseño determinado que permite a los niños condiciones especiales para la estimulación de uno o varios de sus sentidos de manera individualizada. El diseño tiene la función de generarles seguridad y tranquilidad, permitiéndoles conocer nuevas sensaciones bajo el control del terapeuta que siempre deberá estar presente.

Los principales beneficios de estas salas son:

- Mejora la relajación física y mental y disminuye el estrés.
- Asimilación de conceptos causa-efecto
- Mejora de la creatividad e interacción social
- Promueve la exploración del entorno por la reactivación de los sentidos
- Aumento de la concentración y focalización
- Mejora del estado de alerta y coordinación
- Mejora las conductas agresivas

Aunque cada sala es casi personalizada y tendrá distintos elementos, dentro de las salas de estimulación sensorial se diferencian tres tipos desde el punto de vista de los materiales y objetivos principalmente:

- Salas blancas: es la más común. Permite alcanzar la relajación y la estimulación por medio de la espontaneidad.
- Salas negras: más orientadas al aprendizaje causa-efecto y la búsqueda de la sorpresa y el movimiento.
- Salas de Aventuras: los elementos facilitan la psicomotricidad y la actividad perceptivo motora.
- Salas Acuáticas: se trata de un espacio propio con una piscina. Los estímulos principales son el sonido, la luz e incluso hidroterapia.

También se pueden diferenciar las salas desde el punto de vista de la actividad que pueden ser:

- Activas: el sujeto entra en un ambiente que le estimula

- **Pasivas:** se busca que el usuario participe y sea consciente de los efectos que conllevan su actuación con el entorno.

Tras una rápida visión por los tipos de salas, sus beneficios, objetivos y para quien están diseñadas, vamos a analizar los componentes de estimulación que normalmente las componen:

- **Tubos de burbujas:** los tubos de burbujas, normalmente colocados cerca de un espejo son un fuerte estímulo visual por las burbujas de colores y elementos que pasan entre los dos tubos concéntricos y vibratorio proveniente del funcionamiento de la bomba. Disponen de una caja que permite cambiar el color de la luz.



Figura 2.1. Tubo de burbujas

- **Fibras ópticas:** consiste en una fuente de luz que ilumina y hace brillar a la vez que cambia el color de las fibras ópticas, lo que supone una estimulación visual y táctil mediante ejercicios de psicomotricidad.



Figura 2.2. Fibras Ópticas

- **Proyector de imágenes:** potencian el reconocimiento de personas o patrones, además de fomentar la comunicación.



Figura 2.3. Proyector Aceite

- **Paneles interactivos:** convierten la voz o el movimiento en imagen, permitiendo así prácticas con la voz o con el ritmo.



Figura 2.4. Paneles interactivos

- **Aromas:** permiten estimular las emociones y los recuerdos, asociando olores a actividades y analizando la reacción ante el estímulo olfativo.



Figura 2.5. Difusor aromas

- **Camas, pufs o sofás vibratorios:** provoca estímulos vibratorios y vestibulares, buscando la reacción ante distintos ruidos, el ritmo o el cambio de volumen.



Figura 2.6.Cama vibratoria

- **Luz ultravioleta:** permite captar la atención con diferentes objetos, además de descubrir formas, texturas y colores.



Figura 2.7.Luz UV

- **Elementos táctiles:** permiten realizar una sesión de relajación y probar diferentes texturas, así como hacer masajes.



Figura 2.8.Elementos táctiles

- **Elementos musicales:** la música también ayuda a la relajación y a la estimulación auditiva.



Figura 2.9.Elementos musicales

Todos estos elementos y muchos más tienen la función principal de permitir al terapeuta una serie de respuestas mediante la observación del ritmo respiratorio y el movimiento que realiza para detectar posibles anomalías en alguno de los sentidos.

No siempre se llevan a cabo estas sesiones con un terapeuta si no que estas salas también tienen la versatilidad de actuar como entorno lúdico, potenciando sensaciones agradables y autonomía.

2.2 Sistemas de control de salas Snoezelen

La variedad de salas sensoriales existentes podría ser lo amplio que se quisiera ya que existen multitud de empresas que venden material escolar preparado para salas multisensoriales, y otras que directamente te personalizan la sala e instalan todo lo necesario. Sin embargo, no todas venden un sistema de control integrado de los elementos que la componen, y vamos a analizar los más destacados en España de manera rápida:

2.2.1 DOIT Sensory Control

La compañía DOIT tiene un gran catálogo de productos para salas de estimulación sensorial, entre los que destaca la DOIT Sensory Console [1] que es una placa similar a Raspberry que gestiona los elementos de la sala, permitiendo interactuar con ellos y realizar actividades cognitivas. Esta consola dispone de una Tablet con un software propio que permite conectarse a distintos elementos DOIT con el DOIT Adapter para poder manipular los distintos dispositivos. Además de controlar la sala y poder crear escenas, también permite conectarse a internet y crear cuentos vivenciales.

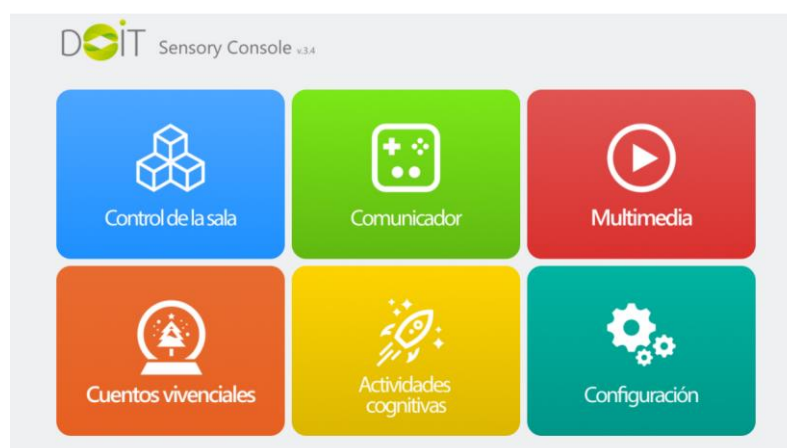


Figura 2.10. Panel DOIT Console

2.2.2 Eneso Sense

La empresa Eneso, también fabricante y distribuidora de material para salas de estimulación sensorial también dispone del Eneso Sense [2], que es un software propio para el control de dichas salas. Ofrecen tres opciones de sala Sala Lite Sense, Classic Sense y Teatro Sensorial, aunque se puede personalizar completamente el sistema. Promete un control total de la sala de los elementos que ellos distribuyen.



Figura 2.11. Eneso Sense

2.2.3 Handycat

Es otra empresa pionera en dispositivos de estimulación sensorial que dispone de multitud de elementos y muy variados. Dispone de la aplicación EMS Genie [3] que puede controlar utilizando un iPad, muchos de los elementos a través de los dispositivos de control que ellos mismos suministran.



Figura 2.12.Handycat IPad

2.2.4 BJ Adaptaciones

Esta empresa es la que más despunta a nivel tecnológico, vendiendo múltiples innovadoras opciones como proyecciones interactivas, grandes pantallas y un software propio con muchas opciones como es el Sistema SHX [4], que permite que la sala se convierta en todo un espacio temático con solo pulsar un botón. Al igual que DOIT, los dispositivos que proporcionan se pueden integrar dentro del sistema. Estas salas están más preparadas para un público adulto o mayor en las que el usuario también realiza el control de la sala y no solo el terapeuta. Incluye control por voz y con la mirada, o a través de la Tablet Communicator 5, y también realiza proyectos completamente a medida.

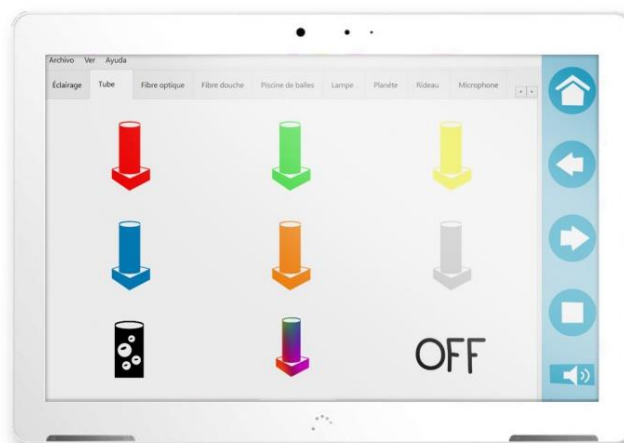


Figura 2.13.Communicator 5 BJ Adaptaciones

2.3 Plataformas de automatización de viviendas o edificios

Conforme avanzan los tiempos cada vez es más normal ver edificios inteligentes, en especial edificios empresariales, centros comerciales, etc. Estas instalaciones necesitan de un control de energía, control de clima, control de aguas, accesos, y muchas cosas más. Dado que la envergadura de este proyecto se sitúa más bien cerca de una vivienda ya que es únicamente una sala, vamos a optar por comparar sistemas domóticos para casas.

Dentro del terreno comercial, encontramos tecnologías propias como:

- **Control4:** esta empresa ofrece software de control que se conecta a dispositivos de Control4 o de otras empresas compatibles. Existe una tabla de compatibilidades donde ver las opciones [5].

- Loxone: esta empresa ofrece además del hardware de control unos miniservidores que se comunican a través de Loxone Link con entradas y salidas digitales, además de múltiples actuadores y relés [6].
- ABB free home: esta marca ofrece un conjunto de soluciones como pulsadores de escena, pantallas, estaciones meteorológicas, detectores de presencia que pueden ser integrados dentro del sistema, aunque no permite añadir elementos externos [7].

También existen tecnologías fuertemente afianzadas y ofrecidas por multitud de marcas como es KNX y que no dependen de una marca:

“KNX España” [8] es una Asociación sin ánimo de lucro que aglutina empresas y entidades que fabrican, comercializan, integran, instalan o de cualquier otra forma promueven productos, soluciones y servicios basados en el estándar mundial KNX, un protocolo de comunicación desarrollado para el control y la automatización de viviendas y edificios.” Existen en el mercado multitud de dispositivos compatibles con KNX, que resultarán mucho más fiables y abiertos que otras opciones, aunque quizás el precio no ayude tanto.

Si bajamos mucho el nivel con respecto a las opciones anteriores, tenemos opciones muy básicas a nivel usuario para automatizaciones básicas y baratas ofrecidas por fabricantes que dependen 100% de la nube como son [9]:

- SmartThings es un conjunto de hardware y software perteneciente a Samsung, aunque soporta algunos externos, que podremos controlar desde su aplicación, permitiendo realizar algunas automatizaciones básicas. Dentro de este grupo podría ser la más potente para un usuario aficionado que quiere incluir elementos inteligentes en casa, aunque como contrapartida no es la que mejor se integra con los altavoces de la propia marca.
- Google Home de similar índole a la anterior encontramos la alternativa de Google, si bien dispone de una perspectiva bonita que puede trabajar sin necesidad de altavoz inteligente, es altamente recomendable. La aplicación está bastante cargada de opciones y multitud de equipos conectables.
- Alexa se encuentra en una gran batalla con Google Assistant por ser el rey de los asistentes virtuales. A nivel de aplicación, Alexa está orientado a trabajar con el altavoz inteligente. Tiene menos opciones que Google Home pero el asistente es más potente.

- ***Apple HomeKit***: Apple no iba a ser menos que el resto. Funciona bastante bien con los dispositivos inteligentes compatibles con este sistema, aunque el abanico es más limitado.
- ***Tuya/Smart Life***: son la gran alternativa a vivir tras las marcas y conseguir un precio competitivo, sacrificando un poco la calidad de los equipos. Tuya agrupa multitud de fabricantes y puede vincular casi cualquier dispositivo.

Esta lista podría ser interminable si contamos que casi cada fabricante dispone de una aplicación, como, por ejemplo, *Xiaomi* dispone de equipos *MI*, *Aqara*, *Dreame*, etc, controlables desde la app de *Xiaomi Home* que permite integrar varios dispositivos. Existen también algunas que incluso pueden trabajar con cámaras IP o incluso con pasarelas KNX IP como *Houseinhand KNX*.

Para resumir, se han visto algunas aplicaciones que permiten integrar una casa domótica, pero, ¿qué pasa cuando la cantidad de dispositivos sea mucho mayor o el tipo de dispositivos que queremos vincular sea completamente abierto y pueda trabajar en local?

En el siguiente apartado se van a abordar dos sistemas de control *open source* potentes que están a la altura e incluso superan a muchas soluciones comerciales, además de permitir integrar todo tipo de dispositivos y de permitir trabajar completamente en local. Disponen de una comunidad bastante amplia, muchas actualizaciones con multitud de nuevas opciones y nuevas funcionalidades, aunque para ello se tenga que perder un poco de estabilidad, ya que las actualizaciones pasan a ser bastante habituales.

2.4 Plataformas *open-source* para el desarrollo domótico

OpenHAB (*Open Home Automation Bus*) [10] es una plataforma de automatización del hogar de código abierto y altamente personalizable. La plataforma es compatible con una amplia variedad de dispositivos domésticos inteligentes, sistemas de automatización de hogares y tecnologías de IoT (Internet de las cosas) y ofrece una forma centralizada de controlar y automatizarlos.

OpenHAB se basa en el modelo de objetos de la tecnología OSGi (*Open Service Gateway initiative*) para permitir la integración de diferentes dispositivos y tecnologías. Además, la plataforma es altamente personalizable y se puede adaptar a las necesidades y preferencias individuales de cada usuario.

OpenHAB también ofrece una interfaz de usuario basada en la web y una aplicación móvil para que los usuarios puedan controlar y monitorear sus dispositivos inteligentes desde cualquier lugar en cualquier momento. Además, la plataforma es altamente escalable, lo que significa que los usuarios pueden agregar nuevos dispositivos y tecnologías a medida que los necesiten.

En resumen, *OpenHAB* es una plataforma de automatización del hogar de código abierto y altamente personalizable que ofrece una forma centralizada de controlar y automatizar una amplia variedad de dispositivos domésticos inteligentes y tecnologías de IoT, todo ello a través de una interfaz de usuario basada en la web y una aplicación móvil. *OpenHAB* tiene más experiencia y opciones hasta el momento, aunque la comunidad de *Home Assistant* está trabajando mucho para actualizar los componentes a las últimas versiones y poder disfrutar de infinidad de herramientas de personalización.

De igual manera que *OpenHAB*, *Home Assistant* se trata de otra herramienta del hogar de código abierto y con una gran comunidad que permite a los usuarios controlar una gran variedad de dispositivos y servicios del hogar [11].

Home Assistant está basado principalmente en *Python*, y se basa en componentes para ir ampliando la compatibilidad con los diferentes sistemas.

En resumen, *Home Assistant* es una plataforma de automatización del hogar de código abierto que permite controlar y automatizar una amplia variedad de dispositivos y servicios de IoT del hogar. La plataforma es fácil de integrar, altamente personalizable, segura y ofrece una interfaz de usuario intuitiva para que los usuarios puedan controlar y monitorear sus dispositivos desde cualquier lugar.

2.5 Proyectos similares

Antes de cerrar este apartado 2 acerca del estado del arte, se ha querido incluir un apartado que ha servido de guía del proyecto realizado. Sobre los niños que pasan por la sala se realiza un pequeño seguimiento acerca de sus avances, y se consideró que si se pudiese someter a los niños a algunas pruebas o escenarios y analizar su respuesta se podría tener un seguimiento mas profundo.

Algunas de las compañías anteriormente mencionadas ya incluyen algo similar a los escenarios, aunque no puedes modificar las acciones a lo largo del tiempo. Otro proyecto de fin de estudios que realiza esta tarea es el del desarrollo de una aplicación Android para hacer esto [12], aunque eso implica tener otro dispositivo y otra aplicación

para gestionarlos. Es por esto que con las limitaciones que dispone la interfaz, se ha creado una herramienta que se integra completamente en el sistema domótico.

Respecto al tema del seguimiento y monitorización, se han tomado en cuenta trabajos que integran una pulsera inteligente de bajo coste [13] dentro de una aplicación o sistema que nos permitiese leer los datos y posteriormente exportarlos sin depender de la aplicación propia y que podemos analizar aquellos que provienen de los escenarios.

Por último, y con el avance de las tecnologías cada vez mas nos encontramos con empresas que ofrecen su soporte junto con su software lo que aumenta la adaptabilidad del software y el soporte a errores, por ello se ha querido incluir esta funcionalidad.

3 Desarrollo del sistema de control domótico

3.1 Estado previo de la instalación

La instalación en la que se va a realizar el proyecto está situada en un centro de apoyo al alumno con discapacidad, en concreto, en una sala de estimulación sensorial especialmente preparada para estimular de manera óptica, acústica y táctica las sensaciones en niños, que por sus condiciones físico-psicológicas no pueden o tienen la misma capacidad que un niño en sus plenas cualidades.

En sus inicios, esta sala contaba con el sistema por defecto que la empresa que distribuye e instala el material dispone, es decir, un mando para controlar algunas cosas, u otras que cuentan con un interruptor de conexión, que funcionaban por radio frecuencia.

La sala recibió una modificación para adaptarla al sistema domótico *Home Assistant*, incluyendo con ella la motorización de las persianas y la inclusión de un Tablet de control. Debido a la complejidad del sistema en sus inicios y el desconocimiento por parte del personal provocó que rápidamente el sistema dejase de actuar como ayuda al personal dejando la sala en desuso.

Finalmente, y con la intención de corregir y adaptar aún más el sistema, se decide volver a comenzar de cero con la instalación, preparándola tanto a nivel de hardware como de software para un funcionamiento más estable, más integrado y con control remoto que permita solucionar pequeñas averías de manera rápida, e incluso, la realización de mejoras por parte de alumnado de la universidad.

Con esta mejora también se busca el objetivo de analizar el comportamiento del paciente durante la sesión ante los diferentes estímulos. Para ello, se incorpora una pulsera de seguimiento y un sistema de escenas que permitirán el post-proceso por parte del profesional de la respuesta al estímulo con su número de pasos, pulsaciones, etc.

3.2 Hardware disponible

En primer lugar, se va a llevar a cabo una breve descripción de los elementos que conforman la sala.

- Iluminación principal:
 - Iluminación con paneles LED 60x60 con luz blanca fría.



Figura 3.1. LED luz blanca

- Iluminación UV con paneles LED 60x60 con ultravioleta.



Figura 3.2. Panel luz UV

- Iluminación mediante downlights regulables

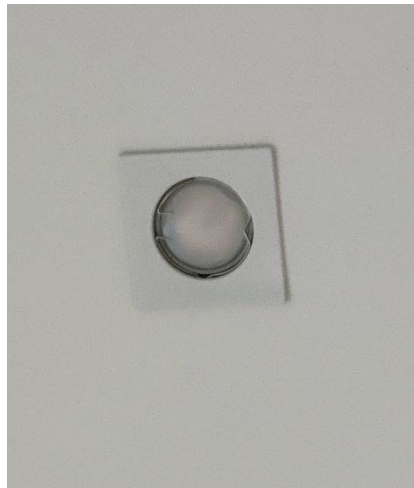


Figura 3.3. Downlight regulable

- Elementos de control:
 - Proyector de aceite



Figura 3.4. Proyector aceite

- Proyector de colores



Figura 3.5. Proyector colores

- Bola de discoteca



Figura 3.6. Bola discoteca

- Cama con movimiento, termostato y subwoofer



Figura 3.7. Cama masajes y movimientos

- Columna de burbujas



Figura 3.8. Columna burbujas

- Difusor de aromas



Figura 3.9. Difusor de aromas

- Estrellas



Figura 3.10. Cielo estrellado

- Mazo de fibras



Figura 3.11. Mazo de fibras ópticas

- Elementos táctiles



Figura 3.12. Panel táctil

- Cama de bolas



Figura 3.13. Cama de bolas

- Colupio



Figura 3.14. Colupio para equilibrio

- Mesa de dibujo



Figura 3.15. Mesa de dibujo

- Pufs
- Otros

Se va a dividir en dos partes el hardware, ya que por un lado tenemos los elementos de la sala a los que incluiremos un hardware para el control, y el servidor que hace de gestor de todos ellos.

Respecto al hardware donde residirá el servidor doméstico, se contaba con una Raspberry Pi 3B+ donde había un sistema del que no se aprovechó nada, y se valoraron otras opciones, como la de actualizar a un modelo superior e incluso cambiar a un NUC, pero resulto ser innecesario ampliar el hardware para las necesidades actuales de la sala.

Se valora también en este punto la posibilidad de ampliar a un SSD (*solid state disk*) debido a las limitaciones de lectura/Escritura de una SD, pero gracias al sistema de *backups* en *Google Drive* se valora innecesaria la inversión.

Respecto al hardware que va a hacer inteligente y controlar cada uno de los elementos de la sala que pasarán a ser controlables, se van a ir analizando conforme se desarrolle su integración software para facilitar la lectura de dicha memoria.

3.3 Instalación de sistema operativo doméstico

3.3.1 Infraestructura de red

Con la intención de separar la red doméstica de la sala del resto de la red de la empresa, se decidió poner un *router* intermedio que permita aislarla en caso necesario, además de permitir un control más severo de la instalación, ya que no hay posibilidad de tener el control del *router* de la empresa proveedora.

Este *router* se establecerá en una IP fija fuera del rango de IP dinámicas en el *router* de la empresa proveedora y hará de puente para los dispositivos, que los conectará en un rango distinto.

Al instalar este *router* en medio, podemos decidir a qué equipos y en que rango ponemos IP fija, principalmente para el servidor doméstico, evitando el uso de *DNS* locales, además de guardar en la lista de *MAC's* de equipos conectados una IP predefinida (tabla *ARP*), lo que aportará una mayor robustez a la red, especialmente para equipos de bajo consumo.

Al *router* previo se le realiza un proceso de apertura de puertos contra el servidor doméstico, abriendo los puertos 443 para accesos a través de *SSL*, 80 para *http* y 8123 para el acceso a la interfaz de *Home Assistant*. Esto finalmente acaba cerrado por las reglas de seguridad informática de la empresa.

3.3.2 Instalación de Home Assistant sobre Raspberry Pi

Existen varias posibilidades para la instalación de *Home Assistant*, ya sea instalando el sistema operativo propio llamado *Hass OS* o mediante *Docker*. Aunque en la actualidad *Docker* es bastante estable, se prefiere el modelo de instalación de sistema operativo por que nos aporta más seguridad y estabilidad, además que, alejamos aun más la tecnología de la sala a la operación real, es decir, que, en caso de fallos, complica el tiempo de su puesta a punto.

Por defecto, la Raspberry Pi incluía un transformador para alimentarla que proporcionaba 2A, que en caso de tareas usando Node Red y algunas otras aplicaciones se quedaba algo justo, por lo que se decidió ampliarlo.

La instalación se llevo a cabo usando la herramienta *Raspberry Pi Imager* [14] que nos permite formatear e instalar el sistema operativo de *Home Assistant* al igual que muchas otras imágenes para Raspberry simplemente seleccionando el sistema operativo a instalar y seleccionando la SD en donde lo queremos incorporar.

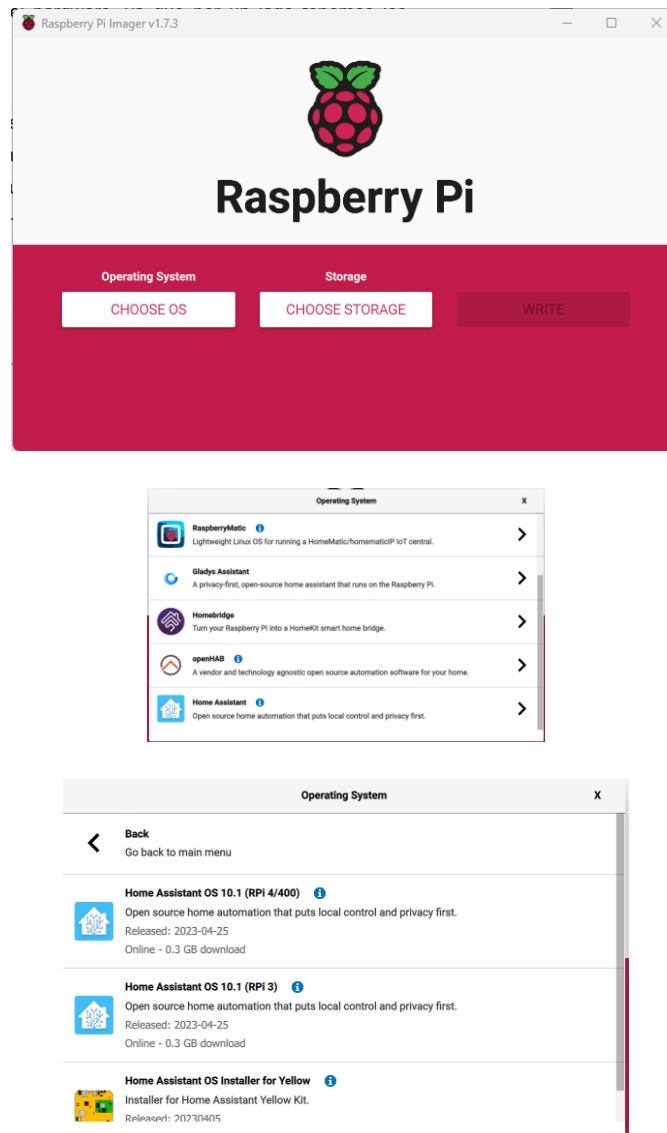


Figura 3.16. Raspberry Pi Imager

Una vez terminado este proceso, insertamos la SD en la Raspberry y conectamos el cable Ethernet a nuestra Raspberry al igual que el cable de alimentación.

Después de unos minutos, podemos hacer una petición a la *url* al *dns* de *Home Assistant* en el puerto 8123 (<http://homeassistant.local:8123/>) y nos aparecerá la ventana de configuración de nombre, ubicación, etc.

Una vez finalizadas los datos personales, se accede a la interfaz por defecto de *Home Assistant* tras instalar.

En los siguientes apartados, vamos a analizar las integraciones con los distintos dispositivos de la sala, así como los complementos y funcionalidades aportadas al sistema.

3.3.3 Integraciones

Una de las partes más destacables del Sistema Domótico seleccionado es que presenta una gran cantidad de integraciones de manera oficial, además de las que la propia comunidad desarrolla y pone a disposición del resto de la comunidad.

En nuestro caso, se han utilizado para poner en marcha el sistema:

- 1) *Volumio*: con la intención de aprovechar el potencial acústico de la sala, beneficioso para el paciente, se instaló un servidor con la herramienta *Volumio* a modo de reproductor de música. Este sistema permite reproducir música contenida en la memoria propia del servidor, por un medio externo, e incluso reproducir desde *Youtube Music*, aunque esta integración no es oficial. También nos permite emitir desde la aplicación de *Youtube* a *Volumio* desde un dispositivo que este en la misma red de la sala mediante casting.
- 2) *Tasmota* [15]: los dispositivos que se conectan a la nube directamente para su control muchas veces presentan inestabilidad si la conexión falla. Para ello, la mayoría de los equipos de la sala llevan consigo el firmware de *Tasmota* que nos permitirá controlar los equipos en local. Hasta hace un par de años, la integración con *Tasmota* se realizaba por MQTT y en algunos casos era inestable, pero con la nueva integración ha mejorado considerablemente su rendimiento hasta quedarse muy cerca de *ESPHome*, que en la actualidad es mantenida por *Nabu Casa*, la empresa propietaria de *Home Assistant*.

Dentro de esta integración estarán incluidas las persianas, los enchufes inteligentes y los interruptores de luz.

- 3) *Broadlink*: en la sala había algunos equipos como el amplificador o el propio aire que son controlables por su botonera, o mediante infrarrojos. Se utilizará un equipo de *Broadlink*, en concreto el *RM3 mini*, para aprender y

posteriormente lanzar los comandos desde *Home Assistant* con esta integración.

- 4) Tuya: esta integración permite añadir los dispositivos tuya de la aplicación a *Home Assistant*. Su uso está ligado a la pasarela de *Zigbee*, que nos devolverá el valor del sensor de temperatura y humedad.

3.3.4 Complementos

Los complementos son plugin que se instalan sobre *Home Assistant* y que permiten ampliar la funcionalidad, ya sea con aplicaciones, como puede ser *Node Red* o meras herramientas como un *Terminal* y *SSH*. Se encuentran instalados:

- 1) File editor: una de las ventajas de los sistemas domóticos de código abierto es que tendrás mucha versatilidad, y gran capacidad para integrar casi cualquier cosa que te plantees, pero para ello, hay que pasar por la tediosa fase de aprendizaje de sus documentos de configuración, que en caso de *Home Assistant* tienen formato *yaml*. Cada vez es menos necesaria esta fase, pero en muchos casos será necesaria.
- 2) Terminal & SSH: esta herramienta nos permite configurar un canal *ssh* contra el servidor de *Home Assistant*, además de poder acceder a ficheros ocultos a File Editor. La principal función en este proyecto es la de comprobar los archivos del volumen compartido con el *Docker* encubierto de Node Red, o permitir hacer *ssh*, aunque únicamente se pueda desde la red local.
- 3) DuckDNS: esta herramienta finalmente no va a tener uso en el proyecto final debido a las limitaciones que nos ofrece el proveedor de servicios de internet. Su función principal es la de evitar poner la IP pública de la red, para en su lugar poner una URL como la siguiente <https://astusems.duckdns.org/>. Este complemento también incluye de serie el complemento *Let's Encrypt* que emite un conjunto de clave pública y privada para poder visitar la página de forma segura.
- 4) Spotify Connect: esta herramienta nos permite mostrar al servidor de *Home Assistant* como un equipo *Cast*, es decir, poder emitir desde la aplicación de Spotify y que se reproduzca por el sistema de audio de la sala. La única limitación que presenta es que el usuario que emite debe de tener una cuenta Premium.
- 5) MQTT: algunos equipos son firmware viejo de *Tasmota* o con otro firmware aun trabajan con MQTT directamente. Es por ello, que se instala un servidor de MQTT para poder realizar el control de estos equipos, además de abrir una puerta para la ampliación de equipos que necesiten de este protocolo de

comunicación como aquellos equipos *Zigbee* que usan pasarela de código abierto.

- 6) *Zigbee2mqtt*: como acabo de mencionar, los equipos *Zigbee* que funcionan a través de una pasarela integrable en *Home Assistant*, la mejor manera de comunicarse con ellos es a través de este complemento que transformará la información recibida a través de *Zigbee* al complemento MQTT. Este módulo está en desuso ya que la pasarela que se dispone es de la marca *Tuya* y no es *flasheable* ya que incorpora un hardware propio de la marca, lo que lo hace difícil de instalarle un firmware libre.
- 7) *Tor*: el navegador *Tor* es conocido como un navegador completamente anónimo. Este navegador tiene brechas de seguridad si no se utiliza con cuidado, ya que a través de él se puede acceder a la *Deep web*. Sin embargo, para evitar esto se emite desde el complemento un par de claves asimétricas que permiten que el tráfico entre *Home Assistant* y el cliente remoto vayan cifrados. Se trata de la única alternativa para superar las barreras del firewall y conseguir una conexión remota cifrada contra el sistema. Para evitar ataques en caso de que pudiera haberlos, se desarrolla un *bot de telegram* que recibirá mensajes de unos *CHAT_ID* determinados y que permitirán el arranque y parada de este complemento restringido al tiempo de soporte.

3.3.5 Configuración de dispositivos

Home Assistant posee un potente sistema de auto detección de integraciones compatibles, de esta manera si compramos un dispositivo de *Xiaomi*, o un dongle *Sonoff Zigbee*, o un equipo de *Broadlink*, veremos que nos sugiere agregar este equipo como integración a la plataforma.

Dichas integraciones en la mayoría de las veces necesitan de un poco de configuración, que en la mayoría de los casos será bastante sencilla como la IP del dispositivo o la cuenta y contraseña, pero normalmente se tiene una documentación bastante legible al respecto.

En este apartado se definirá sin detallar estrictamente el proceso de configuración de cada uno de los dispositivos:

3.3.5.1 Configuración de las persianas

Las cuatro persianas con las que cuenta la sala disponen de un motor de persiana colocado en un momento posterior a la instalación de la propia sala. Para su control, disponen de una caja universal de superficie que contiene un interruptor táctil de 2 canales de la marca *Sonoff*, también llamado *Sonoff T1* [16]. Para permitir un mejor

control de las persianas, además de poder personalizarlas e incluir bloqueos se realiza un cambio de su software cargándolos con Tasmota (esto solo es posible si disponen de un Soc ESP, dígase ESP8266, ESP32, etc) con un adaptador USB a serial TTL, conectando el cable de 3.3V, GND, TX y RX y GPIO se conecta a GND para ponerlo en modo programación. Hay información de las ubicaciones de estos pines en el PCB en la página web oficial de Tasmota [16].

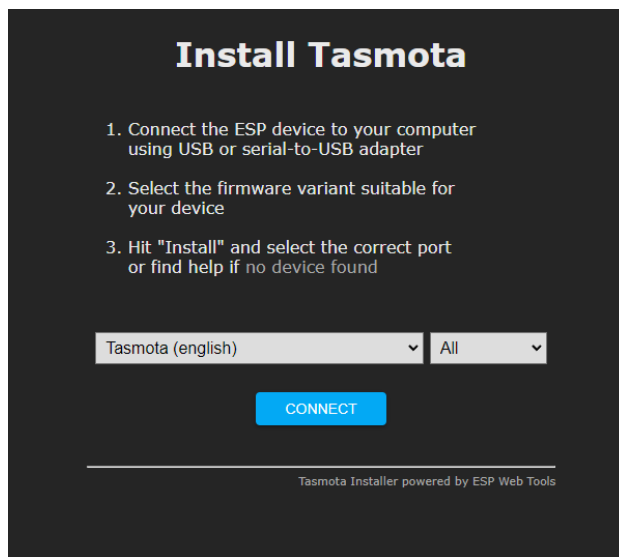


Figura 3.17. Tasmota instalador web

Para instalar *Tasmota* se ha empleado el instalador web cuyo aspecto es el de la figura previa, que facilita mucho el trabajo. Una vez correctamente colocado, tendremos que realizar las siguientes configuraciones conectándonos al punto de acceso que aporta el dispositivo.

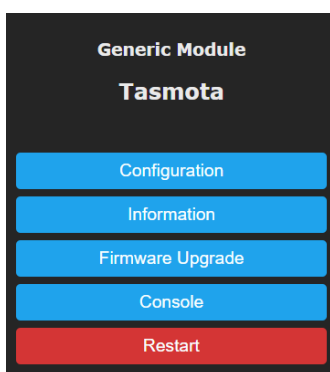


Figura 3.18. Tasmota página principal

Configurar el WiFi del módulo, la dirección, usuario y contraseña del servidor MQTT, y las siguientes configuraciones de consola:

- 1) Habilitar modo persiana con `SetOption80 1`
- 2) Activamos `ShutterRelay1 1` para indicar que actúan como par de relés

- 3) Activamos ahora el bloqueo mutuo para evitar cortocircuitos con `Interlock 1,2`
- 4) Situar la persiana cerrada y lanzar el comando `ShutterSetClose`
- 5) Indicar el tiempo de apertura y cierre con `ShutterCloseDuration1 XX` y `ShutterOpenDuration1 XX`
- 6) Si vemos que esa iteración no ha funcionado, tendremos que ajustar los tiempos de apertura y cierre.
- 7) En el caso de las persianas de la sala, tendremos tiempos de apertura según la persiana que oscilan los 16s y 20s de bajada/subida.

3.3.5.2 Configuración del mando remoto:

- 1) En primer lugar, tendremos que configurar el dispositivo desde la aplicación de *Broadlink* ingresando para ello la cuenta creada a tales efectos en la plataforma tal y como indica el manual de instrucciones propio del dispositivo.
- 2) Una vez configurado y conectado el equipo a nuestra red, en los ajustes del dispositivo tendremos la *IP*, que es el dato que necesitamos en la integración para hacer la conexión. Para evitar que esto sea un problema, se le ha asignado en la base de datos de direcciones *MAC* del *router* una *IP* predefinida, para que en caso de cambiar de *IP* no haya problemas con la integración.
- 3) Para configurar los códigos IR [17], iremos a las herramientas de desarrollador y buscaremos *Learn Command* dentro de los servicios. Seleccionamos el dispositivo *remote* y le asignamos un nombre al dispositivo (si queremos asignar los códigos por dispositivo) y en el apartado de data una lista con los comandos uno a uno que queremos grabar. Una vez llamemos al servicio, nos irán apareciendo notificaciones en *Home Assistant* para que vayamos presionando las distintas teclas delante del dispositivo *Broadlink*. Se encenderá una luz en el dispositivo que confirmará que ha recibido el comando.
- 4) Para reproducir el mismo comando aprendido, tendremos que seleccionar el servicio *Send Commands* con los datos de dispositivo y comando y llamar al servicio para que lo envíe. Para agilizar esta tarea, se han creado una secuencia de scripts para el Amplificador y el Aire con sus códigos que realice directamente esta llamada al servicio. Se puede encontrar más información acerca de este proceso en la página web oficial de la integración de *Home Assistant* con *Broadlink*.

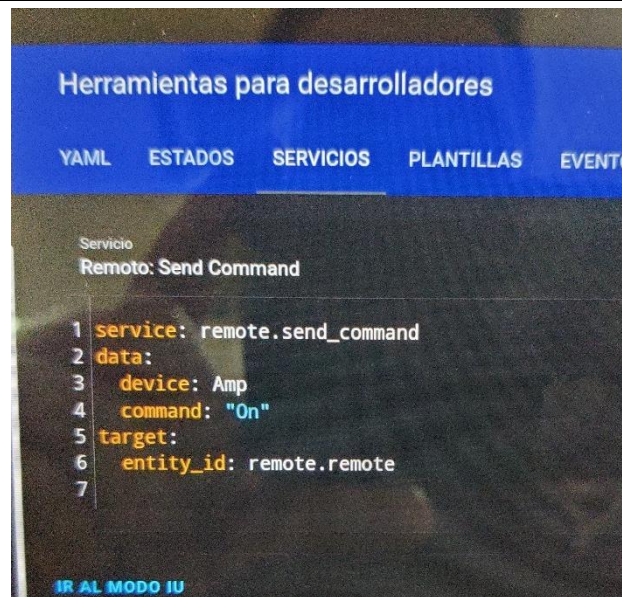


Figura 3.19. Broadlink enviar comando IR

3.3.5.3 Configuración de la iluminación:

La iluminación de la sala está compuesta por tres fuentes de luz principalmente, sin tener en cuenta la iluminación propia de los dispositivos de la sala:

- 1) Luz fluorescente: la luz fluorescente se controla a través del canal derecho de un interruptor táctil de dos canales de *Sonoff* como el que se utiliza para las persianas, situado a la entrada de la sala y como el de la imagen posterior. El interruptor sustituye el interruptor original, y al que hubo que pasar un neutro de la instalación para el funcionamiento del interruptor en actualizaciones previas de la sala. Para poder hacer el control de este dispositivo desde *Home Assistant* se ha abierto y *flasheado* con *Tasmota* [18], y se ha conectado al WiFi de la sala, configurado el cliente MQTT y ajustado el nombre del equipo.



Figura 3.20. Pulsador inteligente Sonoff Touch

- 2) Luz paneles LED: los paneles de luz LED de 60x60 instalados en el techo también actúan de fuente de luz cuando hace falta una mayor iluminación.

Para controlar este tipo de iluminación se ha utilizado el canal izquierdo del interruptor táctil de *Sonoff* del que hemos hablado en la luz fluorescente. La programación del dispositivo ha sido la comentada en el apartado de iluminación fluorescente.

- 3) Luz ojos de buey: la sala dispone de una iluminación tipo ojo de buey en tono más cálido regulable en brillo al que se le ha acoplado un *dimmer* de Shelly [19]. Este tipo de iluminación conserva un pulsador conectado directamente al *dimmer* que se encuentra situado a la izquierda del comentado en la iluminación previa.



Figura 3.21. Dimmer inteligente Shelly

Para realizar la configuración de este dispositivo y poder conectarlo a la red basta con encenderlo, y buscar el punto de acceso creado por el dispositivo, que tiene la forma *shellyswitch-xxxxxx*, cuyas x suelen ser el modelo o número de serie. Nos conectamos a la red y si no somos direccionados automáticamente, podemos introducir la dirección 192.168.33.1 en el navegador para ir a la web de configuración propia del regulador. Accederemos al menú “Internet & Security” para configurar la conexión WiFi como cliente de la red de la sala introduciendo el SSID y la credencial de acceso. Para poder trabajar con el regulador será necesario crear una cuenta en shelly cloud, aunque no sea necesaria para funcionar puesto que este dispositivo puede funcionar en red local.

Una vez creada la cuenta, accedemos a la pestaña de habitaciones y *Discovered Devices*. Hacemos clic sobre el dispositivo y nos pedirá algunos datos como nombre para el equipo, habitación donde se ubicará, etc. Una vez rellenados todos los campos, le damos y guardamos y marcamos que no queremos depender de la nube.

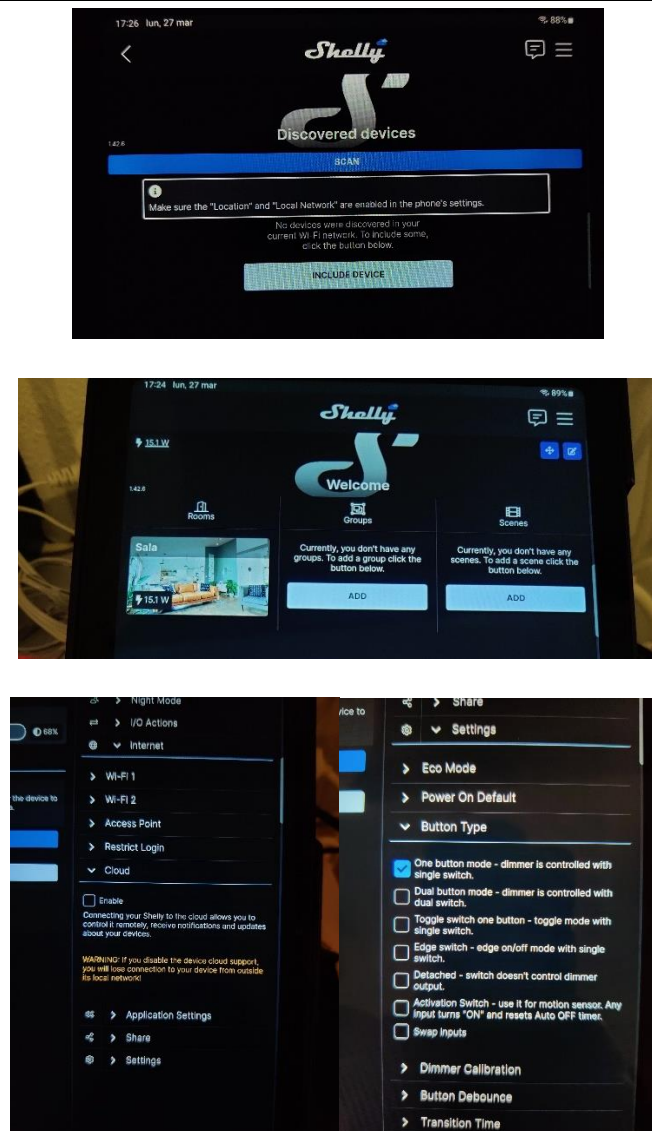


Figura 3.22. Capturas configuración Shelly Dimmer

Para vincular este dispositivo a *Home Assistant* será necesario configurar la integración con Shelly. En caso de que no haya sido detectada la integración automáticamente (esto será más cómodo ya que no tenemos que buscar la IP asociada al dispositivo Shelly), buscamos en la lista de integraciones y seleccionamos Shelly. La integración nos pedirá que incluyamos la IP, que en nuestro caso será relativamente fija, ya que hemos asociado determinadas IP por MAC en el *router*. Si todo ha discurrido sin errores, ya podemos controlar la iluminación y modificar el brillo desde HA.

3.3.5.4 Configuración de los enchufes

Los enchufes inteligentes que se han instalado en la sala con intención de controlar las estrellas, los proyectores, el micrófono con luz y el tubo de burbujas, son el *Sonoff Basic* y el enchufe inteligente S26R2 [20]. Estos enchufes tienen la función de poder aportar características inteligentes a los componentes que ya forman parte de la

sala y que no tenían un comportamiento inteligente. Ambos dispositivos se quieren poder controlar en local, por lo que es necesario para ello cargarles el firmware de Tasmota, que como ya hemos comentado por encima en el apartado de {control de persianas}, es un firmware alternativo para dispositivos ESP32 y ESP8266. Para realizar el flasheo de estos dispositivos se ha seguido la guía oficial de Tasmota y para la configuración la página dedicada a este dispositivo: link ambos enchufes, pero consiste principalmente en conectar usando los pines VCC, GND, TX y RX y conectar GPIO0 a GND para poner en modo programación. Una vez cargados, tendremos que ir al Menú WiFi para conectarlos a la red, al menú MQTT para conectarlo al servidor MQTT que corre en *Home Assistant*, y configurar el módulo.

Para poder auto detectar los dispositivos sin la integración de Tasmota bastará con habilitar el `SetOption19 1` para activar la auto detección. Los dispositivos con una versión moderna de Tasmota, se añadirán automáticamente con la integración, y no necesitan habilitar esta opción.

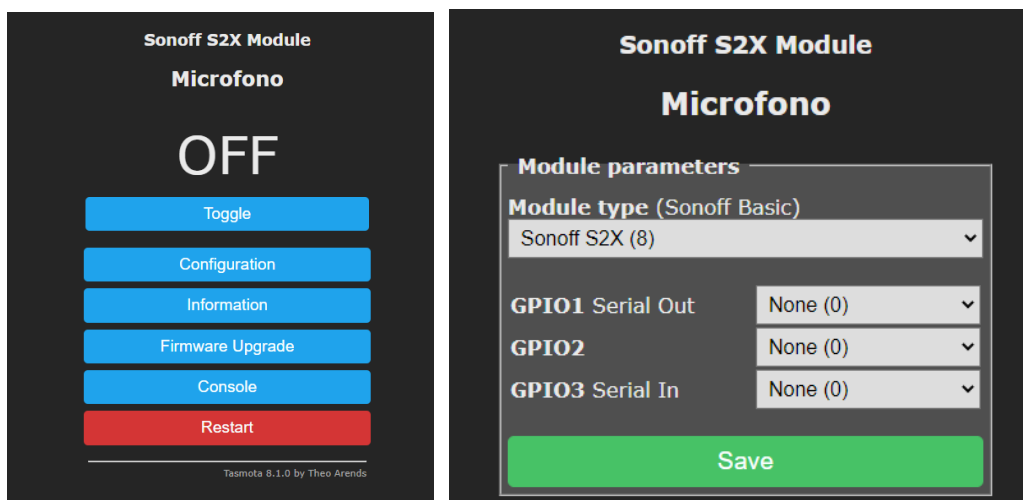


Figura 3.23. Capturas WebUI Tasmota

3.3.5.5 Integración de columna de burbujas

La columna de burbujas dispone de una placa de control que no es accesible debido a las pérdidas de agua, por lo que se ha tenido que optar por controlar el color de las luces como si se fuera un niño en la sala, esto es, mediante la pulsación de los botones de color iluminados que tenemos en la caja de control. Para llevar a cabo este proceso, se ha adquirido una placa con un ESP8266 con 4 relés y que permita alimentarla en continua de 7-30VDC fabricada por DIYMORE [21]. Esto es, porque la caja internamente funciona con corriente continua a 12V para los pulsadores. Se pasará un cable desde cada pulsador de modo que se cierren los relés simulando una pulsación cuando queremos cambiar de color a modo de contactos libres de tensión.



Figura 3.24. Placa ESP8266 4 relés

Para llevar a cabo una integración conjunta se ha elegido Tasmota como el firmware que realizará la activación de los relés de la columna de burbujas. Para ello, se va a hacer uso de la función *PulseTime* que nos permite enviar un flanco de duración terminada al relé deseado.

Como ya se ha comentado previamente, la configuración de los dispositivos Tasmota se realiza mediante la conexión a su punto de acceso y abriendo el servidor que ofrecen. Desde él, configuraremos los nombres, el módulo (con ayuda de los *templates de tasmota*) y conectándolo a la red de la sala y configurando el cliente MQTT. Este dispositivo se conectará utilizando la integración nativa de Tasmota para *Home Assistant*.

Además de la configuración por defecto, tendremos que añadir algunas configuraciones para poder hacer funcionar la pulsación:

- 1) Configurar el `SwitchModeX` de cada uno de los relés a 13. Es decir, que cuando le damos tensión se encenderá y se apagará cuando termine la función *PulseTime*.
- 2) Configurar el *PulseTime* de cada uno de los relés en torno a 1 segundo, es decir `PulsetimeX 10`, ya que funciona en intervalos de 0.1s, excepto para números grandes que se suma 100 al valor en segundos que queremos esperar.
- 3) Desde la interfaz de *Home Assistant* probar a lanzar un encendido de alguno de los colores para comprobar que todo funciona correctamente.

En la integración de Tasmota para *Home Assistant* dispondremos de los cuatro relés que dispone. En caso de no pulsar ninguno, por la propia programación del dispositivo continuará con un bucle cambiando colores.

3.3.5.6 Integración con Volumio

Volumio [22] es un software de reproducción de audio de alta calidad diseñado para transformar un PC como Raspberry Pi en un reproductor de música de alta fidelidad.

Volumio es un sistema operativo completo basado en Linux que se ejecuta en una SBC y ofrece una interfaz de usuario web fácil de usar para reproducir música desde diferentes fuentes, como archivos de música almacenados localmente, servicios de música en línea y dispositivos de almacenamiento en red. *Volumio* es especialmente popular entre los audiófilos y entusiastas de la música que buscan una forma rentable de disfrutar de la música en alta fidelidad en su hogar. La plataforma admite una amplia variedad de formatos de audio, incluidos FLAC, WAV, MP3 y DSD, y utiliza tecnologías de audio avanzadas como *resampling* de alta calidad, ecualización y corrección de fase para garantizar una calidad de audio óptima.

Para instalar *volumio* en la Raspberry, se puede realizar con *Pi Imager* o con *Balena Etcher*. Una vez instalado, nos pedirá que nos conectemos al punto de acceso que crea para configurarlo y conectarlo a la red.

Una vez que está correctamente configurado, tenemos que pasar a integrarlo en *Home Assistant*. *Volumio* dispone de aplicación Android por lo que es perfectamente posible conectarnos directamente desde la tabla de la sala, pero utilizaremos la integración con *Home Assistant*. La integración nos pedirá la IP donde está *volumio* que será fijada como estática en la tabla de MAC's a las que le asigna siempre la misma IP el servidor de DHCP del *router*. La integración nos permite controlar la música, además de seleccionar canciones de la biblioteca para reproducir.

Una vez instalado, instalamos la app de *Volumio Control* para que sea más fácil el acceso.

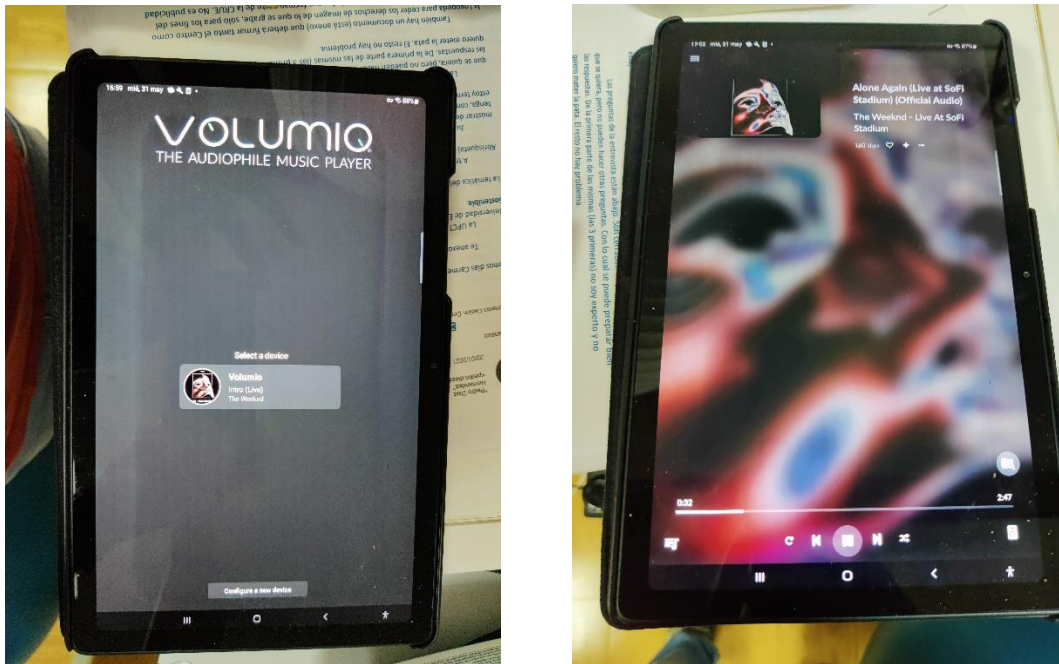


Figura 3.25. Capturas volumio

Se ha configurado también la extensión no oficial de *Youtube Music* [23] que permite vincular la cuenta y reproducir las canciones que tengamos en la biblioteca de youtube seleccionándola desde la Tablet en *Home Assistant* siempre y cuando la hayamos metido antes en la librería o esté en las listas de la carpeta Explorar. Esto será mejorado en futuras versiones del complemento.

Para poder realizar casting desde Spotify, de igual manera, para cuentas premium se ha instalado también Spotify Connect para poder emitir desde la propia aplicación de Spotify al hilo musical de la sala.

Volumio también ha sido configurado para poder emitir sonido desde la aplicación de Youtube de algún equipo conectado a la red de la sala, utilizando para ello la opción de casting, ya que, debido al desconocimiento de uso por parte del personal de la sala, el sistema de audio estaba sin uso, posibilitando así que cualquiera pueda poner música.

Para evitar que el personal tenga que configurar el amplificador para que la salida de Jack funcione directamente, se crea una automatización para que cuando el estado de volumio sea reproduciendo, active el encendido del amplificador y selecciona la entrada AUX, que es la salida de Jack de la Raspberry que ejecuta Volumio.

3.3.5.7 Integración del difusor de aromas:

El difusor de aromas que había por defecto en la sala no disponía de ningún control inteligente, por lo que en la previa instalación y adecuación de la sala se incorporó un humidificador con aromas inteligente modelo Cecotec Pure Aroma [24].

Este dispositivo venía de serie con el firmware tuya, y al disponer de un módulo ESP, se procedió a su flasheo con Tasmota, empleando para ello tuya convert [25]. *Tuya-convert* es un mecanismo que permite flashear dispositivos con tuya haciendo de enlace entre la nube y el dispositivo, usando una Raspberry de por medio.

Tras el flasheo, se procede a incluir el dispositivo Tasmota dentro de la red WiFi y se configura el cliente MQTT. La autodetección de entidades no funcionaba correctamente con el *template* que hay que cargarle al dispositivo para que funcione correctamente, por lo que se optó por configurar las entidades desde el fichero de configuración de *Home Assistant configuration.yaml*. Para ello, se incluye el siguiente código:

```
switch:
  - platform: mqtt
    unique_id: swith.difusor
    name: "Difusor"
    command_topic: "cmdnd/<TOPIC>/POWER1"
    state_topic: "stat/<TOPIC>/POWER1"
    payload_on: "ON"
    payload_off: "OFF"
    retain: false
    optimistic: false
    availability_topic: "tele/<TOPIC>/LWT"
    payload_available: "Online"
    payload_not_available: "Offline"

binary_sensor:
  - platform: mqtt
    unique_id: binary.nivel_difusor
    name: "Nivel agua difusor"
    state_topic: "stat/<TOPIC>/POWER4"
    payload_on: "ON"
    payload_off: "OFF"
    availability_topic: "tele/<TOPIC>/LWT"
    payload_available: "Online"
    payload_not_available: "Offline"

light:
  - platform: mqtt
    unique_id: light.luz_difusor
    name: "Luz del humidificador"
    command_topic: "cmdnd/<TOPIC>/POWER2"
    state_topic: "tele/<TOPIC>/STATE"
    state_value_template: "{% if value_json.Light is defined %}{{ value_json.Light }}{% endif %}"
    payload_on: "ON"
    payload_off: "OFF"
    effect_command_topic: "cmdnd/<TOPIC>/EVENT"
    effect_state_topic: "tele/<TOPIC>/STATE"
    effect_value_template: "{% if value_json.Light is defined %}{{ value_json.LightMode }}{% endif %}"
    effect_list:
      - rgb_cycle
      - color
    availability_topic: "tele/<TOPIC>/LWT"
    payload_available: "Online"
    payload_not_available: "Offline"
    qos: 1
    retain: false
    rgb_command_topic: "cmdnd/<TOPIC>/TUYASEND3"
    rgb_command_template: "{% set brightness = state_attr('light.diffuser_lamp', 'brightness') | int %}{{ '0x02x02x02x02xxxxff6464' | format(red, green, blue) }}"
    rgb_value_template: "{% if value_json.TuyaReceived is defined and value_json['TuyaReceived'].DpId == 8 %}{{ (value_json['TuyaReceived'].Type3Data[8:2])int("
    rgb_state_topic: "tele/<TOPIC>/RESULT"
```

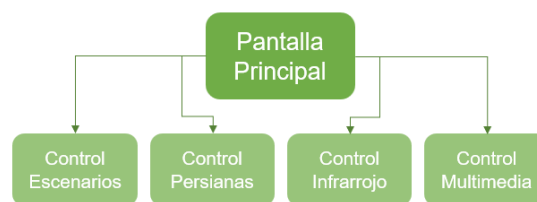
Figura 3.26. Código difusor de aromas

Para que esto surja efecto, se reinicia HA para que recargue la configuración. Tras el reinicio, tendremos las entidades propias del dispositivo que se pueden probar.

3.3.6 Interfaz gráfica

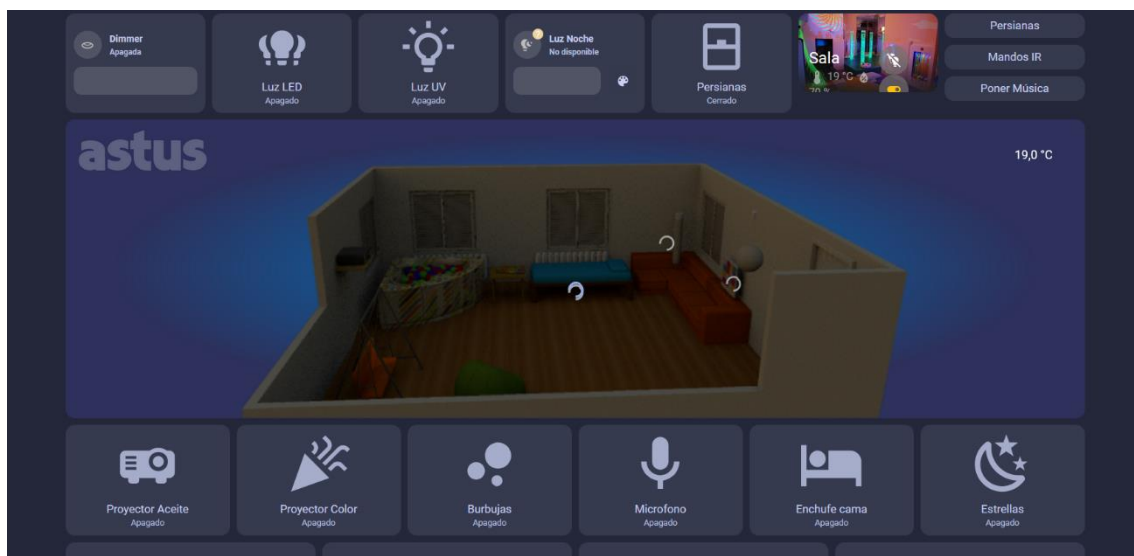
La interfaz gráfica debe de ser el punto fuerte de este sistema ya que debe permitir conocer el estado de la sala, además de un rápido y eficaz control de la misma.

La interfaz desarrollada ha sido dividida en una pantalla principal con la que controlar los elementos principales de la sala, que dispone de 4 subvistas que contienen más información de cada uno de los sistemas siguiendo el siguiente esquema:



Se describen las pantallas a continuación:

- 1) **Pantalla principal**: la pantalla se encuentra separada en 4 partes. La parte superior permitirá el encendido de luces, junto con una segunda parte que nos permitirá navegar a otras pantallas y apagar la sala en bloque. La tercera parte será una imagen dinámica de la sala en 3D, que nos informará del estado de los elementos de la sala. Por último, la barra inferior nos permite encender y apagar cosas independientes como la columna de burbujas o el proyector led de acetonas.



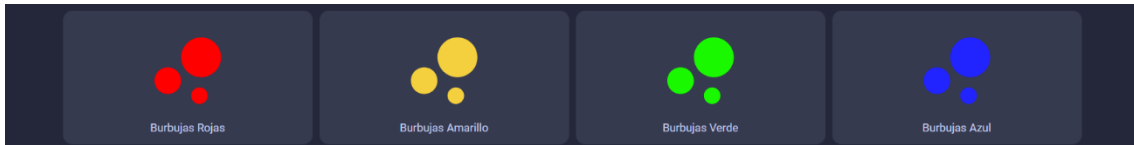


Figura 3.27. Pantalla principal Sala

- 2) Pantalla de mando remoto: la pantalla estará separada en dos partes a izquierda y derecha, estando a la izquierda los controles del amplificador y a la derecha las del aire. La sección del aire contiene una tarjeta que nos permite realizar un calendario de encendido/apagado del aire.

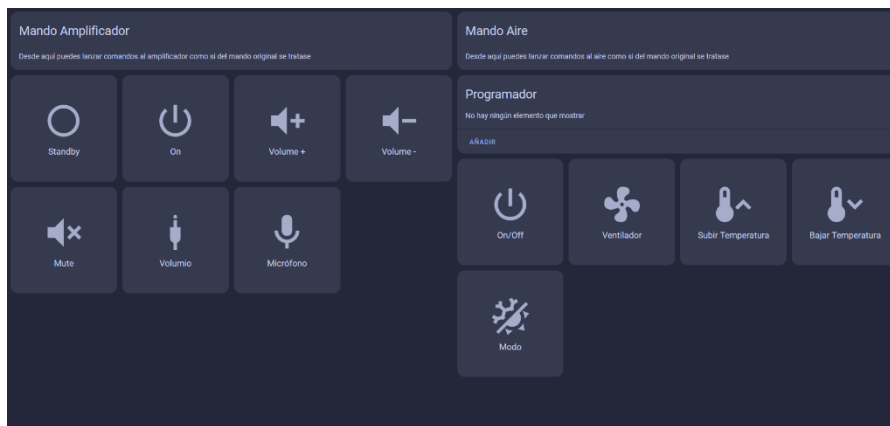


Figura 3.28. Pantalla de mandos

La programación funciona con la interfaz siguiente que permite realizar calendario semanal, diario, e incluso días personalizados. En concreto, la siguiente captura activa a las 8 y a las 16 el botón de encendido de los aires, lo que provoca el encendido y apagado. Es posible cambiar las horas deslizando las barritas blancas, e incluso crear más secciones de cambio, pudiendo implementar una automatización con la temperatura.

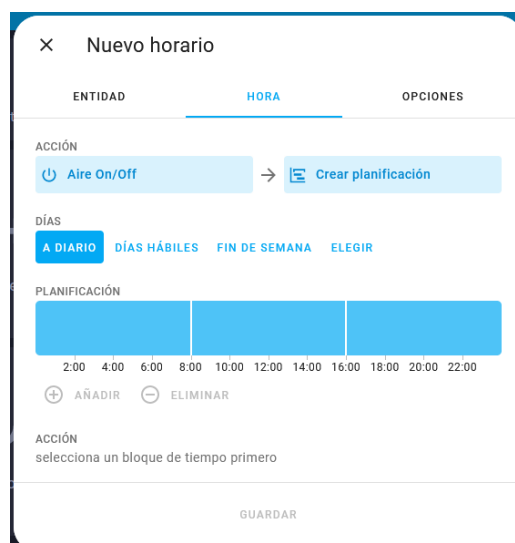
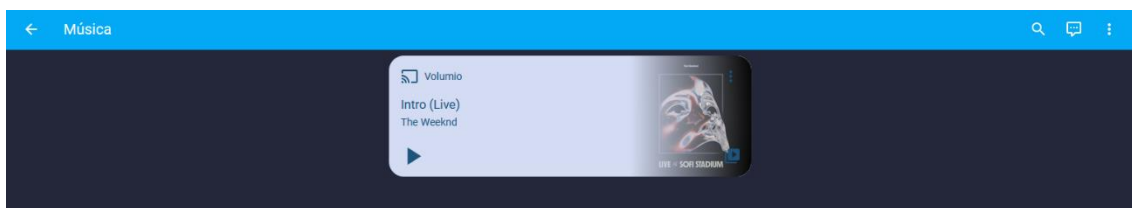


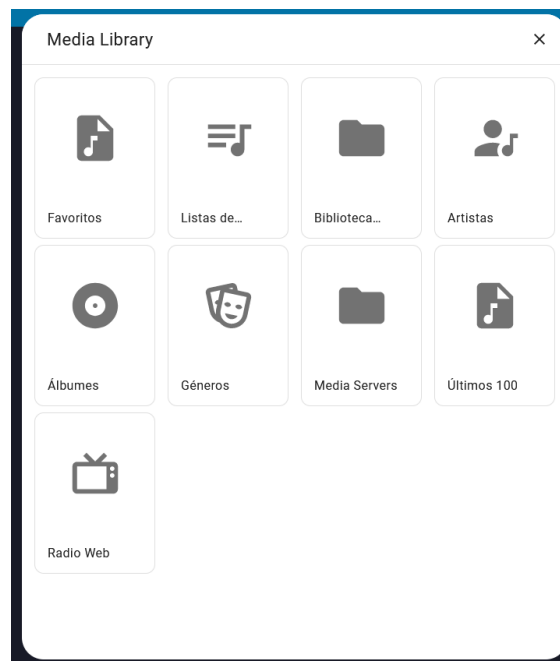
Figura 3.29. Ventana de programación horaria

Podemos encontrar más información al respecto sobre esta tarjeta en su página de desarrollo oficial [26] y [27].

- 3) **Pantalla de multimedia:** dicha pantalla dispondrá de un reproductor que controlará la reproducción de Volumio, además de permitir lanzar desde la librería de Youtube Music siempre y cuando esté correctamente autenticado y las canciones se encuentren previamente en nuestra librería o en su defecto, seleccionarla desde explorar.

*Figura 3.30. Pestaña Música*

Desde esta misma tarjeta podemos elegir la fuente desde la cual queremos escuchar música:

*Figura 3.31. Ventana de selección de fuente de música*

Y muchas otras configuraciones. Podemos acceder a la página de desarrollo oficial para encontrar más información del funcionamiento [28].

- 4) **Pantalla de configuración:** esta pantalla será utilizada únicamente por el desarrollador y contendrá herramientas como el reinicio automático del

sistema configurando una hora de reinicio. Puesto que son pantallas de prueba que permanecen en el *background*, no se tienen en cuenta para esta memoria.

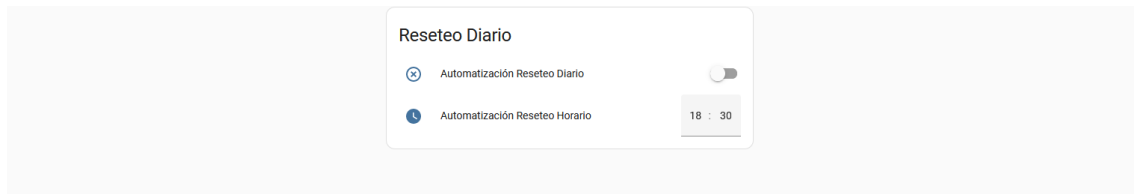


Figura 3.32. Ventana desarrollador

- 5) **Pantalla de escenarios**: dicha pantalla contendrá los controles necesarios para poder crear, grabar, reproducir y borrar escenarios, así como la exportación de los datos ofrecidos por la pulsera de actividad para su posterior análisis. Se verá con más profundidad en el apartado 3.3.7.

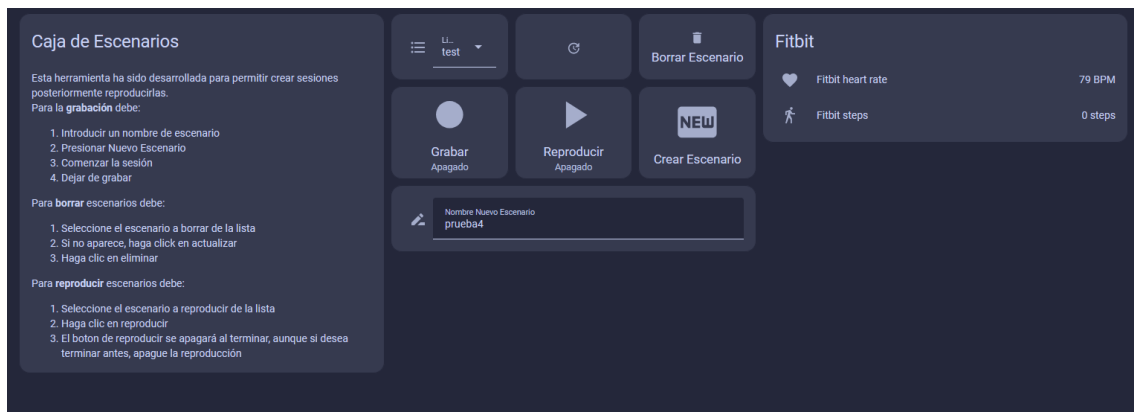


Figura 3.33. Pantalla escenarios

3.3.7 Escenarios

La herramienta de escenarios ha sido desarrollada utilizando la herramienta Node Red, y permite registrar y reproducir grabaciones de los cambios que se han dado durante la grabación.

Esta herramienta funciona introduciendo un nombre de escenario haciendo clic en Nuevo Escenario, que creará el nuevo fichero con la cabecera necesaria para identificar los campos. Cuando activemos el botón de grabación, los cambios de estado que vayan ocurriendo en enchufes, persianas, luces, reproductor y script serán almacenados hasta que paremos el botón de grabación. Este archivo no se borrará a menos que desde la interfaz se seleccione y se presione en borrar escenario.

Respecto a las entidades que se reproducirán se encuentran los elementos *switch*, con funciones de encendido apagado, *cover*, con funciones de apertura y cierre, *light*, con funciones de encendido y apagado y *script* con función de lanzamiento.

La pantalla desde la cual se realiza el control de los escenarios, que posee una descripción a la izquierda de su funcionamiento es la siguiente:

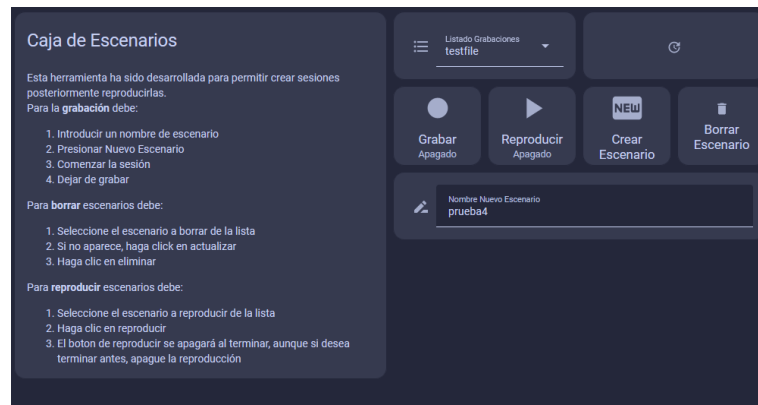


Figura 3.34. Caja Escenarios

Para poder crear los controles que lanzan los diferentes flujos de Node Red, nos hemos servido de los ayudantes de *Home Assistant*, con los que crearemos una entidad de texto para el nombre del escenario, un selector de escenarios disponibles, dos entidades interruptor para controlar la grabación y reproducción, y un botón para actualizar y otro para eliminar los escenarios disponibles en la ruta. Para crearlos, iremos a integraciones, la pestaña Ayudantes y se crean los ayudantes de la lista siguiente:

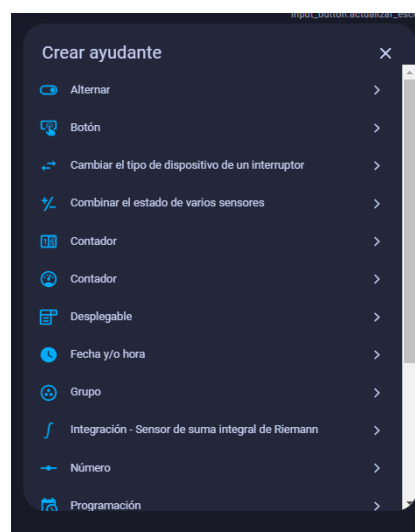


Figura 3.35. Selector ayudante

● Grabar	input_boolean.grabando	Alternar
▶ Reproducir	input_boolean.reproducir	Alternar
🔄 Actualizar Escenarios	input_button.actualizar_escenarios	Botón
🗑️ Borrar Escenario	input_button.borrar_escenario	Botón
📄 Crear Escenario	input_button.crear_escenario	Botón
🕒 Automatización Reseteo Horario	input_datetime.automatizacion_reseteo_horario	Fecha y/o hora
☰ Listado Grabaciones	input_select.listado_grabaciones	Desplegable
✍️ Nombre Escenario	input_text.nombre_escenario	Texto

Figura 3.36. Listado ayudantes

Una vez creados los ayudantes, ahora llevaremos a cabo la creación de los flujos. *Home Assistant* ofrece dos tipos de nodos distintos que nos hubiesen permitido exportar los diferentes estados alcanzados por cada una de las entidades, como son *get history* o *event state*. La principal ventaja de eventos con respecto a *get history* es que al analizar uno a uno, podemos tratar cada uno de los eventos del registro como una posición en un array, filtrar de manera fácil, además de no depender de si el final ha sido marcado o si queremos añadir más cosas al escenario, es decir, hacer una pausa en lugar de una parada. El propósito de este trabajo no es la descripción de la integración, por lo que en caso de necesitar más información podemos revisar la documentación incluida en cada uno de los nodos en Node Red, o acudir a la página de documentación [29].

Continuando con la programación asociada al guardado de estados, el flujo actual presenta una parte en funcionamiento, y una parte con herramientas o métodos descatalogados entre el que se encuentra el previamente comentado acerca de recibir los históricos. En la parte en funcionamiento se pueden diferenciar cuatro partes en función de la tarea a realizar por el flujo:

1. Nuevo Escenario

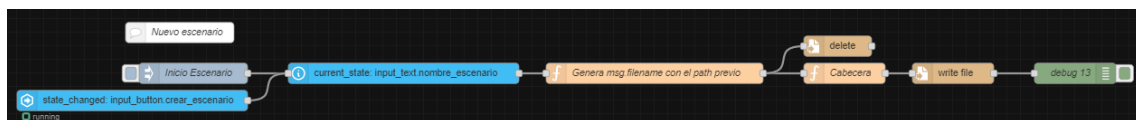


Figura 3.37. Flujo Nuevo Escenario

Esta sección recibe un *start* por parte del usuario o cuando se presiona en la entidad *crear_escenario*, que leerá el nombre de escenario en el campo de texto con entidad *nombre_escenario*, que pasará por un nodo función que generará la ruta al escenario como vemos a continuación:

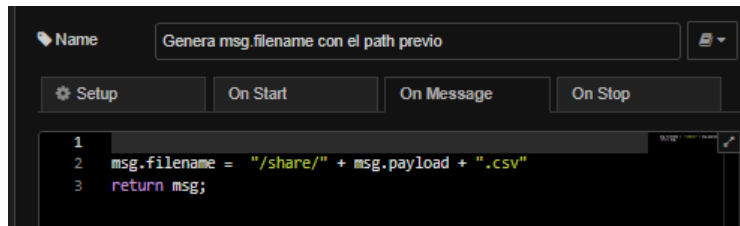


Figura 3.38. Nodo de creación de path

El porqué de usar esa ruta viene de que el *Docker Node Red* se comunica con la máquina host con el volumen compartido `/share`. A continuación, borrará el fichero anterior (si existe), agregará la cabecera al archivo, y por último, escribirá el archivo.

2. Grabación de hora de inicio y fin

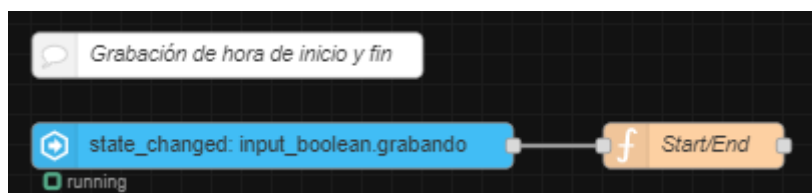


Figura 3.39. Flujo de log de horas de grabación

Esta sección tiene la misión de analizar los estados de la entidad `grabando`, y almacenar su valor de inicio y fin como variables de flujo que serán utilizadas para calcular el desfase con respecto a la hora de inicio.

3. Grabación de eventos

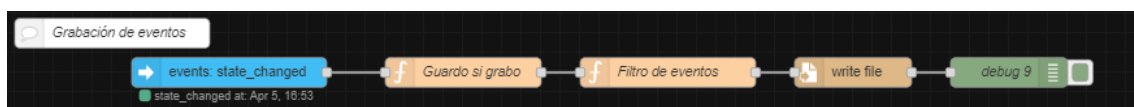


Figura 3.40. Flujo de grabación de eventos

Es una de las partes más importantes. Esta sección estará a la escucha de todos los eventos de la sala. A modo de filtro, se comprobará si pasa a la siguiente fase únicamente si tengo la grabación activada, que será rechazada en caso contrario.

Tras pasar el filtro de grabación, se analiza la entidad y se comprueba que el estado haya cambiado y si es de los tipos mencionados anteriormente. En caso favorable, se escribirá en el fichero seleccionado por el usuario como una línea al final.

4. Listado de ficheros y actualización de fichero

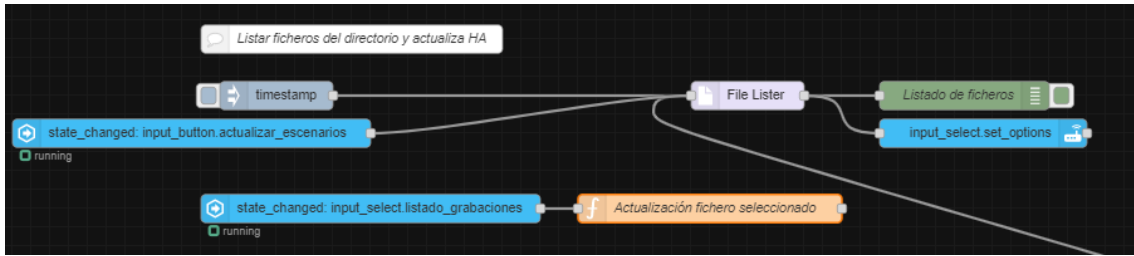


Figura 3.41. Flujo de actualización de ficheros

Esta sección se va a encargar de utilizando el nodo File Lister, que tendrá que ser instalado con *node-red-contrib-fs*, se encargará de listar los ficheros que acaban en “.csv” de la ruta */share*, ya sea por un clic de usuario en Node Red o desde la entidad *actualizar_escenarios*, para que se actualice el desplegable de escenarios disponibles.

La segunda parte, se encarga de actualizar en variable de flujo el nombre del fichero seleccionado para poder ser utilizado por el resto de los nodos del flujo cada vez que este cambia.

5. Borrado de ficheros

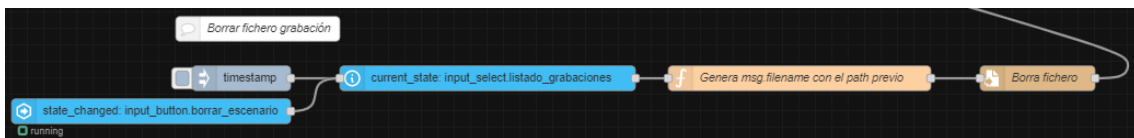


Figura 3.42. Flujo eliminación ficheros

Esta sección se va a encargar de que cuando el usuario haga clic en borrar escenario, se leerá el fichero seleccionado en la entidad *listado_grabaciones*, generará la ruta al fichero y mandará eliminarlo. Tras su eliminación, llamará a actualizar la lista de los escenarios disponibles para evitar seleccionar contenido vacío.

6. Reproducción de ficheros

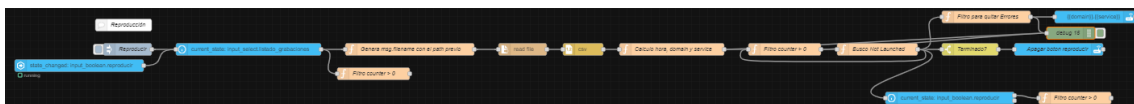


Figura 3.43. Flujo de reproducción de ficheros

Esta parte es probablemente la más compleja y extensa, y su principal función es la de reproducir las entidades en el tiempo calculado a partir de la fecha de inicio de grabación al alcanzar el *timestamp*.

Este flujo se iniciará al clicar en reproducir escenario, que recogerá el fichero seleccionado, que por un lado actualizará una variable de flujo que controlará el estado de la reproducción y por otro lado construirá la ruta hasta el archivo que estamos

buscando. Posteriormente, leerá y organizará este archivo como un fichero de objeto de Node Red, algo similar a un *JSON*.

A continuación, va a extraer el dominio, es decir el tipo de objeto (luz, persiana, enchufe...), el nombre de la entidad, la cantidad de elementos por reproducir, el servicio (por ejemplo, para encender una luz, usaremos *turn_on*), y se calculará la hora a la que tiene que lanzarse esta acción en función de la hora de inicio de la reproducción.

El flujo continuará por un comprobador de que la entidad sigue con la reproducción activa, para en caso contrario, pararla, o en caso de que no queden acciones por hacer, termine la reproducción. Después, recorrerá todos los elementos buscando elementos sin lanzar. Si esta sin lanzar, comprueba si es su hora y en caso contrario, sigue con el siguiente. Si da la casualidad de que tiene una acción por lanzar, comprobaré si sigo en modo reproducción, comprobaré si he terminado para apagar la entidad de reproducción y lanza la acción en *Home Assistant*.

3.3.8 Datos de salud

Otra principal novedad que se incorpora al proyecto es la posibilidad de monitorizar el comportamiento del alumno de la sala, pudiendo almacenar y posteriormente tratar datos relacionados con los pasos realizados, calorías quemadas y ritmo cardiaco durante la sesión, pudiendo así, analizar a posteriori determinados patrones de actividad ante determinados estímulos.

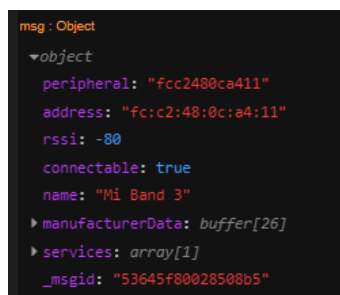
Para poder desarrollar esta herramienta se ha tratado de poner en marcha una Xiaomi Mi band 3 y una pulsera *Fitbit Sense* para ver posibilidades a distintos precios.

Esta herramienta permite desde la interfaz de *Home Assistant* seleccionar un intervalo de fechas y realizar una exportación de los registros asociados a las entidades que nos ofrece la pulsera.

En primer lugar, se trató de utilizar la Mi band 3 ya que su versatilidad y precio puede aportar una ventaja extra. El problema de esta pulsera es que los servicios a los que permite acceder utilizando un servidor GATT (sistema de funcionamiento de bluetooth Low Energy o BLE). La única alternativa a este problema es la de utilizar una aplicación que previamente haya realizado ingeniería inversa para descubrir el significado de la información que ofrecen cada uno de los servicios, para los que previamente hay que sincronizarse con la pulsera.

Se intentó mediante el empleo de herramientas *bash* y *Node Red* intentar acceder a los servicios emitidos por la Mi Band 3. Para ello, se desarrolló un flujo en Node Red que hacía un escaneo de los dispositivos de la red. El escáner bluetooth no viene por defecto con la instalación, sino que hay que ir al apartado de añadir paletas e instalar el módulo *node-red-contrib-noble-bluetooth* [30]. Puede acudir a la documentación oficial en caso de no saber instalar un módulo:

Tras muchas iteraciones de búsqueda de dispositivos, finalmente encontramos la Mi Band 3, que contiene un array con 1 servicio, que es “fee0”, que contendrá información.



```
msg - Object
└─ object
  peripheral: "fcc2480ca411"
  address: "fc:c2:48:0c:a4:11"
  rssi: -80
  connectable: true
  name: "Mi Band 3"
  ▶ manufacturerData: buffer[26]
  ▶ services: array[1]
  _msgid: "53645f80028508b5"
```

Figura 3.44. Depuración flujo análisis servicios Mi Band 3

Acorde a la documentación, deberíamos obtener la característica “2a37” del servicio “180d”, que únicamente estará disponible en caso de conectarse.

Se desarrolla un flujo en Node Red para conectarnos a la pulsera e intentar recibir ya que con la aplicación nRF Connect [31] no somos capaces de leer esa característica.

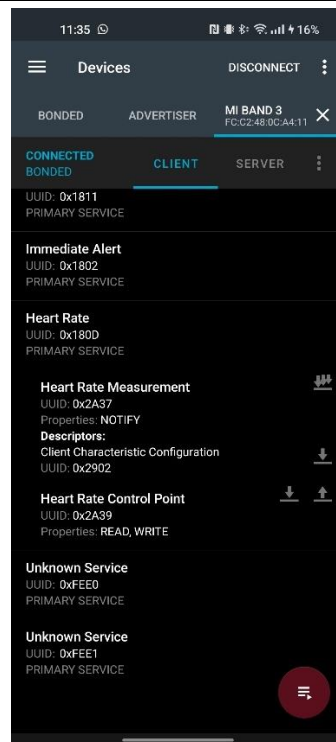


Figura 3.45. Captura servicios nRF Connect

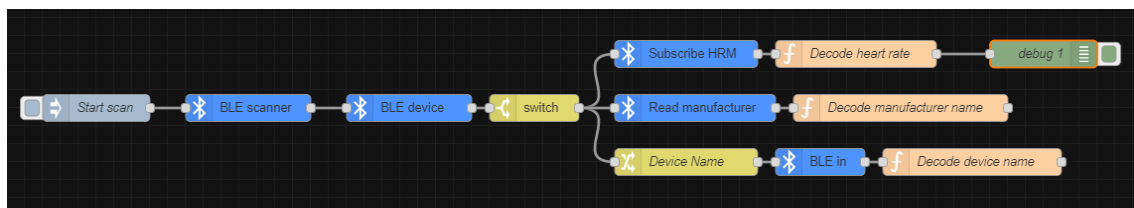


Figura 3.46. Flujo lector de pulsaciones

Para evitar tener que hacer este proceso, lo que implicaría un conocimiento avanzado de BLE, hemos optado por utilizar una aplicación que haga el trabajo por nosotros. Esta aplicación es un proyecto publicado en *github* [32] que nos permite hacer lo mismo que hace la aplicación Zepp, pero de manera *open source* y donde los datos residen completamente en local. El único problema de esta aplicación es que publica los resultados en la ruta local del dispositivo Android:

`/storage/emulated/0/Android/data/nodomain.freeyourgadget.gadgetbridge/files`

En esta ubicación tendremos los archivos de log con la información previamente solicitada. Estos datos no residen de manera automática en la ruta mencionada, sino que hay que hacer clic en un botón en el apartado de administración de datos de la aplicación o en su lugar configurar la exportación automática.

El proceso automático de envío de datos parece arduo y peligroso, por lo que se decide descartar esta opción, ya que tendríamos que crear un servicio del tipo *Tasker*

que enviase automáticamente este archivo a algún lugar, etc, lo que implica una gran inestabilidad.

Como alternativa al sistema anterior, se descubre que la aplicación *notify & fitness* permite que en su opción de pago exportar los datos desde la aplicación de Xiaomi oficial a *Google Fit*, y mediante una integración no oficial de la comunidad poder acceder a los datos en *Home Assistant*. Esta alternativa presenta el problema de que estamos tras una instalación subvencionada, por lo que pagar una suscripción para poder exportar los datos no es una opción viable.

Se decide dejar de lado la *Mi Band 3* y optar en este caso por la pulsera *Fitbit Sense* ya que dispone de una integración oficial con *Fitbit* lo que permite tener los datos directamente en nuestra mano. Las dos únicas limitaciones que impone este sistema es que las peticiones han de ser de máximo de 150 a la hora, lo cual no es un problema, además de que las peticiones han de realizarse a través de SSL.

Como se ha mencionado en el apartado 3.3.1 de infraestructura, en los inicios de la instalación se planteó poder acceder en remoto por parte del personal para inspeccionar, poder mantener y controlar la sala desde remoto con el problema principal de que el proveedor de internet tiene el control de la instalación y por seguridad cierra determinados puertos del *router* y no permite al personal de la instalación poder hacer modificaciones en el firewall para ello.

Como solución a este problema, se opta por instalar la aplicación de *Fitbit* en la Tablet de la sala, y con ayuda de la aplicación libre *FitToFit* [33] poder coger los datos de la pulsera *Fitbit* y poder pasarlos a *Google Fit* para posteriormente integrar *Google fit* y poder acceder a los datos. Cabe destacar que esta aplicación permite automatizar este proceso pudiendo seleccionar intervalos desde media hora hasta 1 semana, por lo que la única limitación que presentara este sistema es que para poder exportar los datos de actividad habrá que esperar un rato tras la finalización de la sesión o en su lugar abrir la aplicación y hacer clic en el botón de exportar y seleccionando las fechas que queremos actualizar.

Al principio la aplicación de *Fitbit* para la Tablet no permite conectar de manera correcta la pulsera, lo que puede dar aspecto de que no se esta sincronizando, pero luego se puede apreciar que cada determinado tiempo la pulsera se va conectando y vemos que la fecha de última sincronización de la pulsera va cambiando y que los datos de la pulsera se van actualizando a *Google Fit*. Esto es debido a una falta de permisos por la aplicación *Fitbit*. Al dar acceso completo a dispositivos cercanos y ubicación, la conexión funcionará correctamente.

Se procederá a instalar la integración de Google Fit proveniente de HACS, que es la comunidad de *Home Assistant*, y seguir los pasos para poder tener los datos de Google Fit directamente en la aplicación o poder acceder a ellos desde HA.

Hace un tiempo esta integración de la comunidad funcionaba copiando el token ofrecido por *Google Actions*, que exportaba un archivo de token, pero Google recientemente ha actualizado sus criterios de autorización, por lo que será necesario ejecutar un script en Python junto con unos previos paquetes de requerimientos, que nos exportarán el archivo oculto “.*google.fit.token*” que tendremos que pegar en la carpeta *config*, necesario para el correcto funcionamiento.

El enlace siguiente contiene las instrucciones, y ha sido la única integración en funcionar tras la actualización de Google.

<https://github.com/IvanVojtko/googlefit-homeassistant>

Además de la generación de las credenciales, tendremos que añadir lo siguiente al *configuration.yaml*, donde el nombre es personalizable:

```
sensor:  
  - platform: google_fit  
    name: Fitbit
```

Esta integración crea un conjunto de entidades de tipo sensor.*Fitbit.** como las siguientes:
















<input type="checkbox"/>		Fitbit blood pressure DIA	sensor.fitbit_blood_pressure_dia	Sensor
<input type="checkbox"/>		Fitbit blood pressure SYS	sensor.fitbit_blood_pressure_sys	Sensor
<input type="checkbox"/>		Fitbit BMR	sensor.fitbit_bmr	Sensor
<input type="checkbox"/>		Fitbit calories	sensor.fitbit_calories	Sensor
<input type="checkbox"/>		Fitbit distance	sensor.fitbit_distance	Sensor
<input type="checkbox"/>		Fitbit heart rate	sensor.fitbit_heart_rate	Sensor
<input type="checkbox"/>		Fitbit height	sensor.fitbit_height	Sensor
<input type="checkbox"/>		Fitbit hydratation	sensor.fitbit_hydratation	Sensor
<input type="checkbox"/>		Fitbit move time	sensor.fitbit_move_time	Sensor
<input type="checkbox"/>		Fitbit nutrition	sensor.fitbit_nutrition	Sensor
<input type="checkbox"/>		Fitbit oxygen	sensor.fitbit_oxygen	Sensor
<input type="checkbox"/>		Fitbit resting heart rate	sensor.fitbit_resting_heart_rate	Sensor
<input type="checkbox"/>		Fitbit sleep	sensor.fitbit_sleep	Sensor
<input type="checkbox"/>		Fitbit steps	sensor.fitbit_steps	Sensor
<input type="checkbox"/>		Fitbit weight	sensor.fitbit_weight	Sensor

Figura 3.47. Listado de entidades de Fitbit

De las cuales no todas vienen de la pulsera, si no que provienen de la aplicación en la cual podemos añadir de manera manual o sincronizar con otros dispositivos *Fitbit*, para posteriormente mostrar los datos en *Home Assistant*.

A modo de comprobación, podemos ver como al actualizar los datos desde la *Fitbit*, estos se sincronizan correctamente con la aplicación de *Fitbit*, y al transferir los datos usando la aplicación Fit2fit de manera manual, o dejando que automáticamente los sincronice, podemos ver los datos directamente en la integración de Google Fit para *Home Assistant*.

Analizando el día 29, podemos ver lo siguiente:

En la aplicación de *Fitbit*.

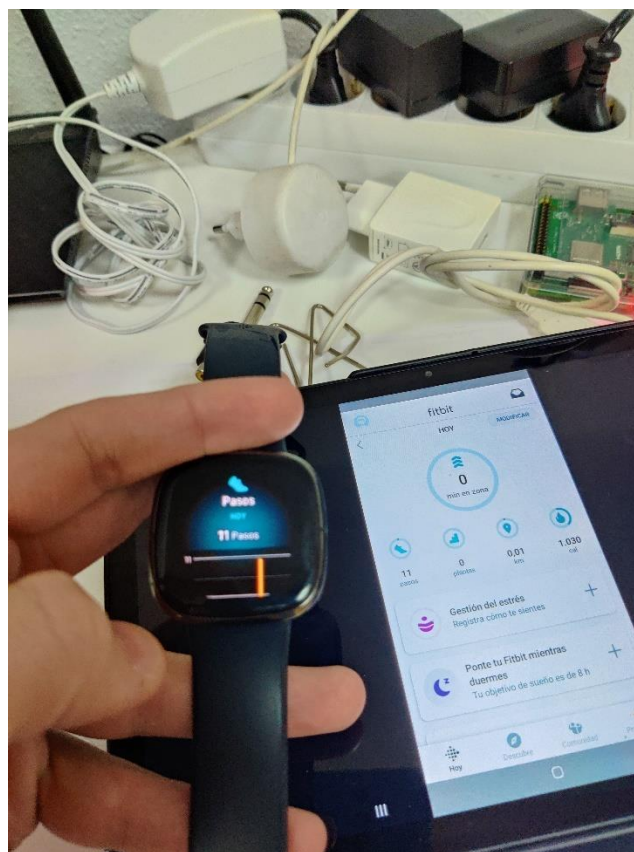
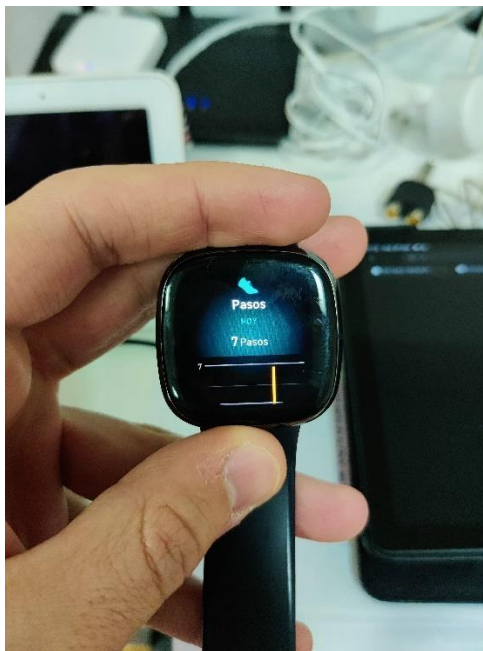


Figura 3.48. Captura Fitbit con aplicación

En la aplicación FitToFit:

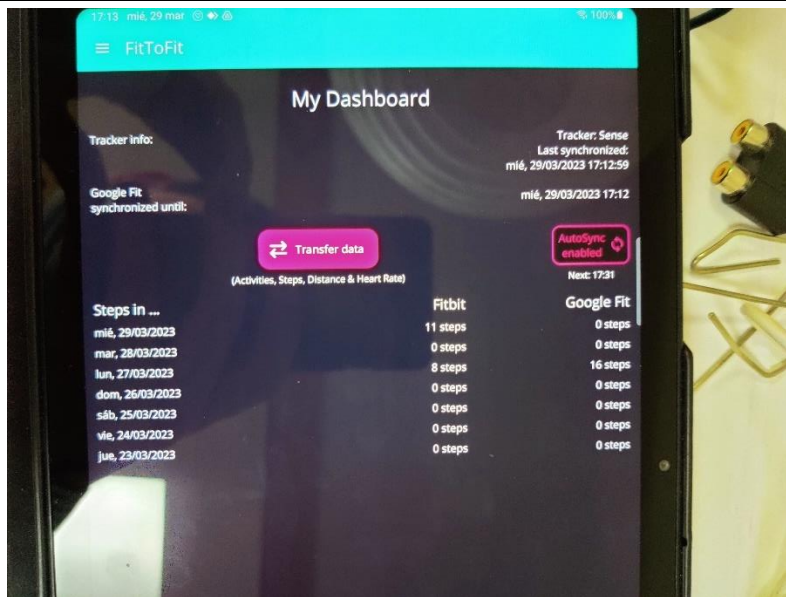


Figura 3.49. Captura actualización aplicación Fit2Fit

En la aplicación Google Fit:



Figura 3.50. Captura Google Fit

En *Home Assistant* los datos no se vinculan si no se introducen en el momento de hacerlos, es decir, si actualizamos en el día, los que llevemos los actualiza, pero si pasa al día siguiente y se resetean volveremos a tener 0. Debido al modo ahorro del

dispositivo, aunque permitamos a la aplicación de sincronización una actualización automática, no se lleva siempre a cabo, por lo que muchas veces los datos no coincidirán y otros muchos la pulsera estará sin carga. Visto esta situación, se opta por dejar el desarrollo realizado, y poder acceder a los datos de Google Fit directamente desde la aplicación, ya que la exportación de datos a nivel de históricos solo tendría sentido si se actualizase y moviese todos los días.

Dados los problemas que presenta este tipo de enlace, sobre todo por los métodos de ahorro de batería disponibles en los dispositivos Android, que detienen mucha de la actividad en segundo plano de las aplicaciones, resulta difícil mantener actualizada la aplicación de *Fitbit*, que se haga secuencialmente unas actualizaciones de datos a Google Fit y que las actualizaciones de Google Fit representen un momento cercano en el pasado.

Como solución a los problemas previos, se plantea utilizar OpenGatewayMQTT [34], que es una plataforma de código abierto que se utiliza para conectar dispositivos Bluetooth Low Energy (BLE) a una red MQTT. Permite la comunicación bidireccional entre los dispositivos BLE y la red MQTT, lo que permite el intercambio de datos y la automatización de dispositivos.

OpenMQTTGateway BLE es compatible con una amplia variedad de dispositivos BLE, incluidos sensores, medidores de energía, dispositivos de iluminación, interruptores y otros dispositivos inteligentes, entre los que destaca la pulsera Mi Band de Xiaomi. Además, al estar basado en MQTT, es fácil de integrar.

Para llevar a cabo la instalación nos servimos del instalador web, que usa el API de Web Serial, para permitirnos programar el microcontrolador ESP32 directamente al igual que la instalación de Tasmota. Siguiendo la documentación, lo conectaremos a la red, y a nuestro servidor MQTT montado en la Raspberry con Home Assistant, y buscaremos entre los dispositivos disponibles la Mi Band con ayuda del software MQTT Explorer.

```
{
  "id": "FC:C2:48:0C:A4:11",
  "name": "Mi Band 3",
  - "rssi": -71,
  ~+ "rssi": -75,
  "brand": "Xiaomi/Amazfit",
  "model": "Mi Band/Smart Watch",
  "model_id": "MB/SW",
  ~ "steps": 237,
  - "act_bpm": 33,
  ~+ "act_bpm": 103,
```

Figura 3.51. Captura servicios MQTT Explorer

Una vez localizada, se comprueban que los valores sean válidos y estables, y si fuese necesario, se puede crear un flujo de Node Red que vaya leyendo, aunque con el propio complemento de MQTT, se puede leer directamente los valores de pasos y pulsaciones.

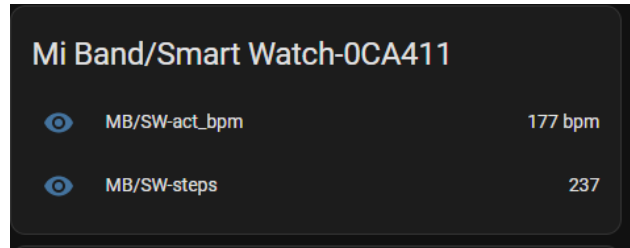


Figura 3.52. Valor entidades Mi Band 3

Ha de mencionarse que en determinados casos en los que la pulsera no mide las pulsaciones (por ejemplo, no llevo la pulsera en la muñeca), estas presentan unos valores un tanto altos en vez de dar un valor nulo.

Para esta sección y con la intención de exportar los datos de las pulsaciones para posteriormente analizarlas, se ha creado un flujo en Node Red que comparte los datos recogidos en los intervalos seleccionados mediante un correo a la dirección adjuntada.

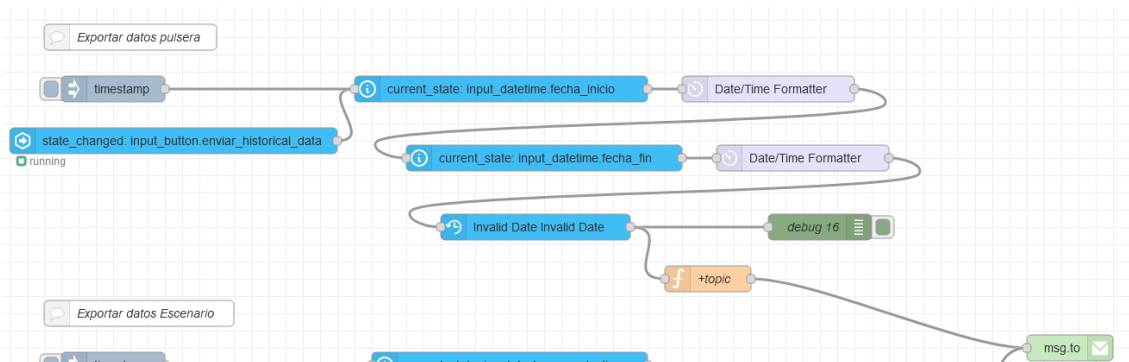


Figura 3.53. Flujo de exportación de datos históricos de pulsera

Se puede consultar más información acerca de este flujo en el 7.1, en el que se documenta más a fondo el código.

3.3.9 Programaciones horarias

Se considera bastante útil y necesaria la posibilidad de poder incorporar acciones programadas sobre las entidades con las que cuenta el sistema. Para ello, se lleva a cabo la instalación del componente *Scheduler Component*, que con ayuda de la tarjeta *Scheduler card* nos permite realizar diferentes acciones con distintos tipos de

repeticiones, desde diarias a totalmente personalizadas. Esta tarjeta se puede cargar desde cualquiera de las tarjetas que nos ofrece la interfaz de *Home Assistant*, disponiendo de una en la ventana de sensores infrarrojos.

En este caso, su uso principal es para encender y apagar el aire, aunque es completamente personalizable.

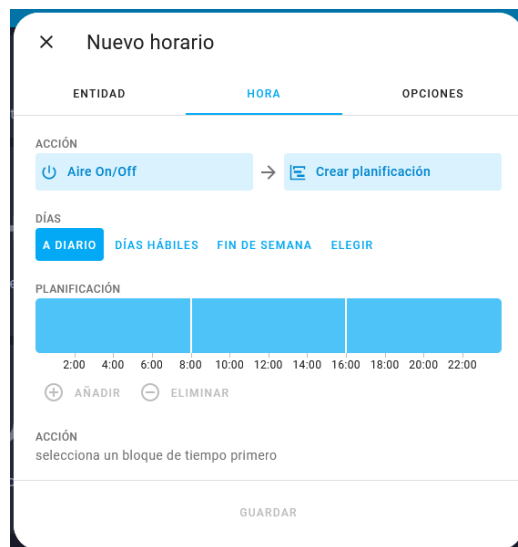


Figura 3.54. Captura ventana de programación horaria

3.3.10 Sistema Multimedia

El sistema multimedia con el que cuenta la sala, es bastante más avanzado y de mejor calidad que el que puede ofrecernos un altavoz inteligente o cualquier otro altavoz bluetooth, por lo que se decide aprovechar la infraestructura con la que contamos en el momento. Se instala Volumio sobre una Raspberry, que nos permitirá reproducir música de muchas fuentes directamente al amplificador de la sala, o en su defecto, música directa de youtube o Spotify mediante cast de música (en caso de contar con una cuenta premium en el servicio de streaming).

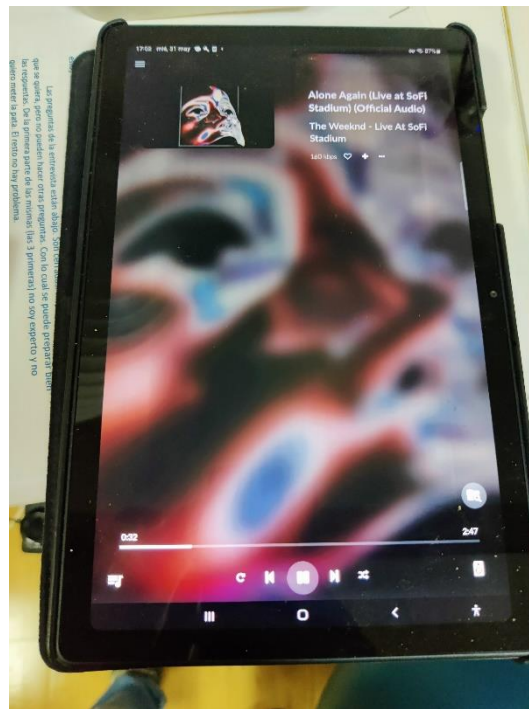


Figura 3.55. Captura Volumio

3.3.11 Integración con dispositivos externos

En muchos casos, los dispositivos inteligentes, ya sea por la calidad de su fabricación, o por el precio, pueden provenir de países en los que la electrónica suele ser más barata. Se ha querido contar en este servidor con la presencia de Tuya, que, si bien no es la mejor nube para un hogar inteligente, quizás es de las que más dispositivos soporta, por lo que disfrutar de una integración que facilita el acceso a algunos elementos es un factor a favor.

Por otro lado, un estándar que está creciendo poco a poco es Matter. Una de las ventajas de *Home Assistant* es que desde casi su inicio cuenta con un servidor de Matter y posibilidad de Thread, que son dos estándares en plena ebullición en el sector de la domótica pero que por su fase de desarrollo aún no se encuentran completamente implementados en los hogares inteligentes, aunque será cuestión de tiempo.

3.3.12 Asistente

Desde la actualización de febrero de 2023.1, *Home Assistant* cuenta con un asistente de texto que puede ayudarnos a cambiar el estado de determinadas entidades, además de controlar la mayoría de ellas. En la actualidad, no cuenta con muchas

acciones, pero puede ayudarnos a construir acciones básicas del día a día. Para poder acceder a él, en la pantalla principal clicamos en el botón de mensajes.



Figura 3.56. Icono apertura asistente Home Assistant

Se muestra captura de un caso práctico en el que le pedí encender la luz de mi salón haciéndolo correctamente, aunque como podemos ver en el mensaje siguiente, no es capaz de procesar acciones no establecidas por la gente de HA previamente.

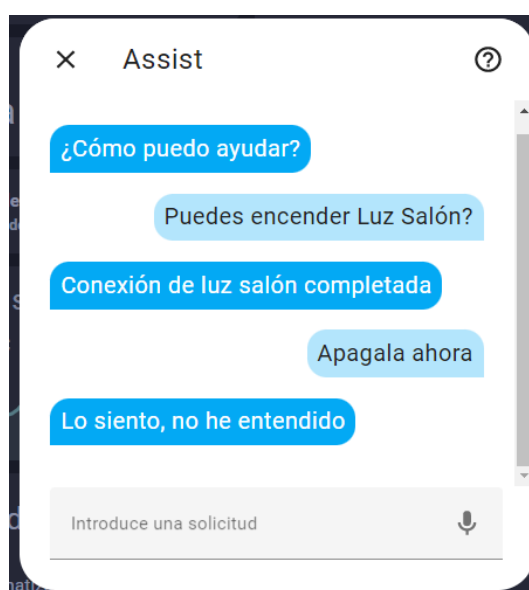


Figura 3.57. Captura asistente Home Assistant

3.3.13 Monitorización con telegram

Conociendo la limitación de acceso vía red a la instalación de la sala, y con la intención de poder controlar de manera puntual y analizar el comportamiento se ha desarrollado un *bot* de telegram cuya función principal es la de aportar al desarrollador o usuario avanzado de algunas herramientas de mantenimiento de la sala siempre y cuando el chat con el desarrollado este incluido en la configuración del mismo.

El *bot* desarrollado permite los siguientes comandos, de entre los cuales solo los 3 primeros tendrán una respuesta con información, ya que el resto serán meros mensajes informando de que se ha recibido la acción.

- `/start`: este comando nos devuelve el mensaje `commands` a la vez que una botonera con las opciones disponibles para el bot, es decir, los comandos posibles.

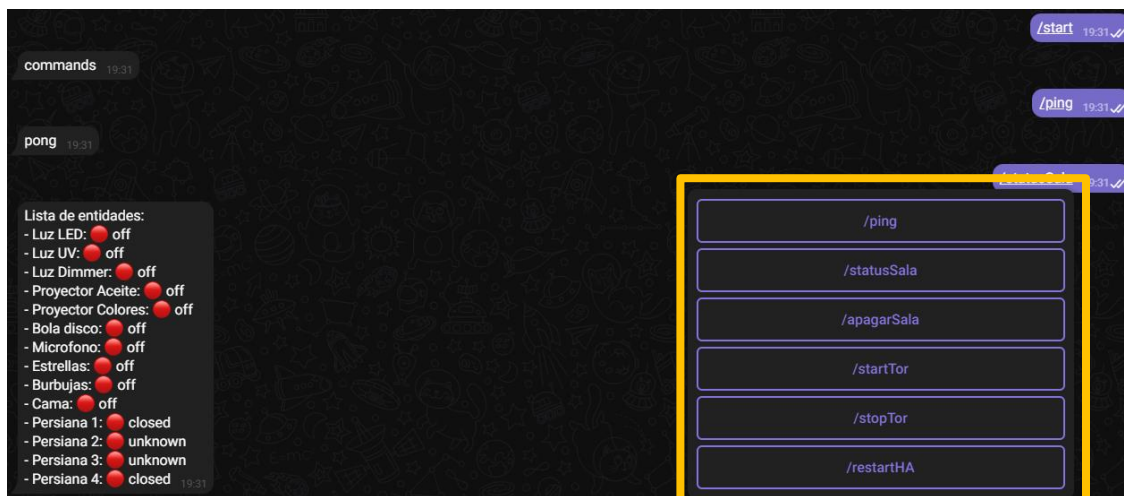


Figura 3.58. Comando start bot Telegram

- `/ping`: el ping es un comando que nos permite analizar si tenemos comunicación con la sala. Al lanzarlo, si la sala esta activa y conectada a internet nos devolverá el mensaje `pong`.



Figura 3.59. Comando ping bot Telegram

- `/statusSala`: este comando nos devolverá un listado con las entidades más importantes de la sala y el estado de cada una de ellas para saber el estado actual de la sala estemos donde estemos. Esta opción es realmente útil si por un descuido, hemos dejado cosas en la sala encendidas.

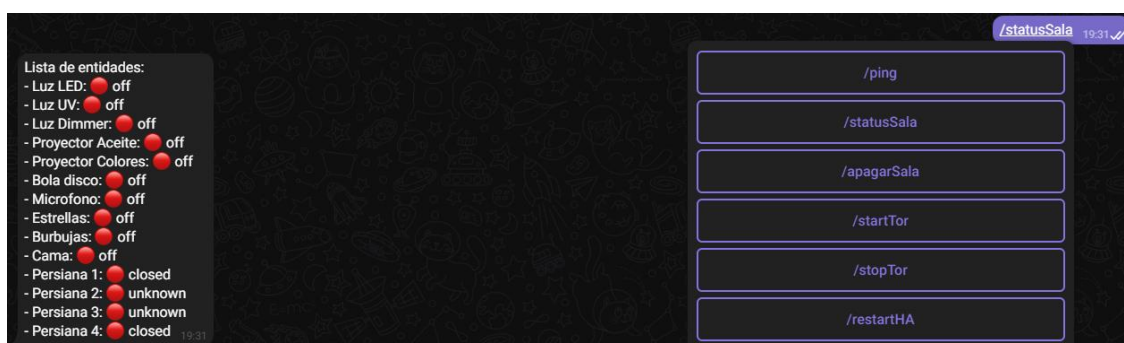


Figura 3.60. Comando statusSala bot Telegram

- `/apagarSala`: este comando nos permitirá apagar los dispositivos que pertenecen a la zona `sala`.

- */startTor*: este comando permite inicializar el complemento de Tor, lo que permite al desarrollador conectarse en remoto a la sala si se sabe la url y clave privada.
- */stopTor*: cuando el complemento Tor no está en uso, por seguridad es útil apagarlo. Este comando nos permite apagar el complemento.
- */restartHA* este comando como su nombre indica manda un comando de reinicio al sistema operativo Hass OS.

4 Conclusiones

Desde el principio se quiso montar un sistema dotándolo de muchas posibilidades, así como integración con altavoz o mando que dotaran al personal de más poder sobre la instalación, aunque se ha visto con el trascurso del proyecto que el sistema debe simplificarse a lo que realmente es necesario, ya que los constantes cambios necesarios para evadir los problemas de red con el exterior provocaban inestabilidad en el sistema, y así, familiarizar al personal con la instalación y sus elementos y no incluir elementos que no aporten.

Se ha tratado de adaptar la interfaz visual que existía previamente por una mejorada que actúe a modo de unidad de control desde la cual se puedan realizar diferentes acciones casi con un máximo de dos toques en pantalla.

También se han llevado a cabo trabajos para poder realizar una monitorización y exportación en escenarios de las diferentes acciones tomadas por el profesional de la sala, y su posterior reproducción si se desea.

Para acompañar a la monitorización de la sala, se ha instalado una pulsera de actividad *Mi Band 3*, que monitorizará los pasos y pulsaciones con una tasa que ronda el segundo, y que podrá exportarse de manera similar a los escenarios para posteriormente unirlos e integrarlos con la intención de analizar más a fondo la respuesta del alumno a estímulos mediante un postproceso que pudiese ser fruto de un futuro proyecto de Inteligencia Artificial.

Esta herramienta también acerca la posibilidad de añadir más sensores a la instalación que hagan una tarea similar, además de una mejor vía de mejora y monitorización.

5 Trabajos futuros

La principal rama de desarrollo futuro para este trabajo es la captación de datos que se van exportando por parte del servidor doméstico, que puedan ser utilizados para mejorar el ambiente la interactividad con el usuario. Se planteo utilizar la integración de *The Silly Home Cloud* [35] que es un *addon* que permite conectarnos a la nube y calcular determinadas rutinas entre nuestros dispositivos en base a los eventos pasados. Sin embargo, el *addon* no se encuentra completamente desarrollado y aun no es completamente estable y tiene varias limitaciones de acceso a la base de datos de *Home Assistant*, por lo que se ha querido dejar atrás, debido a que por tener pocos datos y sensores no creaba automatizaciones.

En las fases tempranas de este desarrollo, la aplicación contaba con una pantalla con el asistente de Google que se integraba con la aplicación de domótica permitiendo lanzar acciones y rutinas, lo que facilita el uso de la sala. Sin embargo, Google no permite la integración si la instalación no contiene un cifrado SSL para acceder a su servidor.

Podría ser interesante para siguientes versiones construir un altavoz con micrófono que no dependa de ningún fabricante y permita interactuar con el asistente incorporado de *Home Assistant* “*Hey Ada!*”, el cual tendrá que ser acompañado por otro asistente que traduzca a texto todas las peticiones, disponible actualmente en la versión de pago. Desde febrero de 2023, *Home Assistant* cuenta con un asistente a texto que ayuda a interactuar con las entidades, aunque aún no está muy avanzado como hemos podido ver en el apartado 3.3.12 del asistente. Esta mejora también es más interesante cuando el desarrollo del asistente incorporado este más avanzado.

Se ha querido añadir no como trabajo futuro si no como ayuda para el integrador, o el titular que use la sala, de poder aprovechar los avances en *bots* de inteligencia artificial y poder pedirle que nos ayude con las creaciones de determinadas automatizaciones, como, por ejemplo, encender una determinada luz si hoy es día de semana y la hora es de 9-11 am.

Se muestra el ejemplo anterior preguntándole a uno de ellos como ChatGPT:

EL Puedes ayudarme a crear una automatización en Home Assistant que encienda la luz del comedor si hoy es día de semana y son entre las 9 y las 11 am

¡Por supuesto que puedo ayudarte! Para crear una automatización en Home Assistant que encienda la luz del comedor si hoy es un día de la semana y son entre las 9 y las 11 am, sigue estos pasos:

1. Abre tu archivo de configuración de Home Assistant (generalmente llamado `configuration.yaml`) y agrega la siguiente sección al final del archivo:

```
makefile Copy code
automation:
```

2. Debajo de la sección `automation`, agrega una nueva automatización con un nombre descriptivo, como "Encender luz del comedor en días de semana por la mañana". También agrega un gatillo que se activará cuando sea un día de la semana y sean entre las 9 y las 11 am. Puedes hacerlo de la siguiente manera:

```
yaml Copy code
- alias: Encender luz del comedor en días de semana por la mañana
  trigger:
    platform: time
    at: '9:00:00'
  condition:
    - condition: time
      weekday:
        - mon
        - tue
        - wed
        - thu
        - fri
  action:
    - service: light.turn_on
      entity_id: light.comedor
```

En este ejemplo, la automatización se activará a las 9:00 am de lunes a viernes. Si deseas ajustar el rango horario, modifica el valor en `at` para que se ajuste a tus necesidades.

3. Guarda el archivo de configuración y reinicia Home Assistant para que tome en cuenta los cambios.

Con estos pasos, tu automatización debería estar lista para funcionar. Asegúrate de que el nombre de la entidad de la luz del comedor en Home Assistant sea `light.comedor` y que la entidad sea compatible con el servicio `light.turn_on`.

Figura 5.1. Captura ChatGPT creando automatizaciones

6 Bibliografía

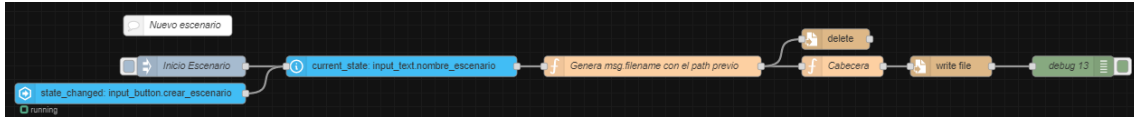
- [1] «Home», *DOIT – Salas multisensoriales*, 1 de abril de 2019. <https://doitmultisensorial.com/en/> (accedido 27 de mayo de 2023).
- [2] «Eneso Sense, tecnología para salas de estimulación multisensorial», *Eneso*, 25 de julio de 2017. <https://www.eneso.es/blog/eneso-sense-tecnologia-salas-estimulacion-multisensorial/> (accedido 19 de marzo de 2023).
- [3] «Handycat | Sensory Healthcare», *Handycat*. <https://handycat.com/es/> (accedido 27 de mayo de 2023).
- [4] «Sistema SHX: ¡Una nueva forma de vivir las Salas Multisensoriales! - Qinera». <https://qinera.com/es/42-sistema-shx> (accedido 19 de marzo de 2023).
- [5] «Home Automation and Smart Home Systems | Control4». <https://es.control4.com/> (accedido 27 de mayo de 2023).
- [6] «Loxone. Create Automation.», *ESES Loxone*. <https://www.loxone.com/eses/> (accedido 19 de marzo de 2023).
- [7] «ABB-free@home® - Automatización de viviendas y edificios». <https://new.abb.com/low-voltage/es/productos/automatizacion-viviendas-edificios/oferta-de-productos/abb-freeathome> (accedido 27 de mayo de 2023).
- [8] «KNX for End Customers – KNX Association [Official website]». https://www.knx.org/knx-es/para-su-vivienda/beneficios/end-customers/index.php?gad=1&gclid=CjwKCAjw1MajBhAcEiwAagW9MS8NNPy-jstMWEcj1S7A_VecLh2Y4VqKZxvUk-qhb2Zgbh0SqZ-xhoClyUQAvD_BwE (accedido 27 de mayo de 2023).
- [9] «The 6 best home automation systems of 2022», *ZDNET*. <https://www.zdnet.com/home-and-office/smart-home/best-home-automation-system/> (accedido 19 de marzo de 2023).
- [10] «openHAB». <https://www.openhab.org/> (accedido 27 de mayo de 2023).
- [11] H. Assistant, «Home Assistant», *Home Assistant*. <https://www.home-assistant.io/> (accedido 27 de mayo de 2023).
- [12] L. F. Torres, «Diseño y desarrollo de un control inteligente para una sala de estimulación multisensorial de bajo coste».
- [13] «Monitorizacion_de_usuarios_con_pulseras_y_relojes_Parrilla_Navarro_Francisco.pdf». Accedido: 10 de abril de 2023. [En línea]. Disponible en: https://rua.ua.es/dspace/bitstream/10045/109281/1/Monitorizacion_de_usuarios_con_pulseras_y_relojes_Parrilla_Navarro_Francisco.pdf
- [14] H. Assistant, «Raspberry Pi», *Home Assistant*. <https://www.home-assistant.io/installation/raspberrypi/> (accedido 27 de mayo de 2023).
- [15] «Tasmota Documentation - Tasmota». <https://tasmota.github.io/docs/> (accedido 27 de mayo de 2023).
- [16] «Sonoff T1 - Tasmota». <https://tasmota.github.io/docs/devices/Sonoff-T1/> (accedido 30 de abril de 2023).
- [17] H. Assistant, «Broadlink», *Home Assistant*. <https://www.home-assistant.io/integrations/broadlink/> (accedido 30 de abril de 2023).
- [18] «Install Tasmota». <https://tasmota.github.io/install/> (accedido 30 de abril de 2023).
- [19] «Relé Shelly Dimer 2 - Shellyspain | Distribuidor oficial en España», *info@shellyspain.com*. <https://shellyspain.com/rele-shelly-dimer-2.html> (accedido 27 de mayo de 2023).
- [20] «Sonoff S26 Smart Socket - Tasmota». <https://tasmota.github.io/docs/devices/Sonoff-S26-Smart-Socket/> (accedido 27 de mayo de 2023).
- [21] «ESP-12F 5V/7-30V/220V 4 Channel Relay Board (ESP12F_Relay_X4) Configuration for Tasmota». https://templates.blakadder.com/ESP12F_Relay_X4.html (accedido 27 de mayo de 2023).

- [22] «Volumio - The Music Player», *Volumio*, 6 de julio de 2021. <https://volumio.com/en/> (accedido 27 de mayo de 2023).
- [23] P. Kan, «YouTube Music plugin for Volumio». 1 de enero de 2023. Accedido: 29 de mayo de 2023. [En línea]. Disponible en: <https://github.com/patrickkfkkan/volumio-ytmusic>
- [24] «Cecotec Pure Aroma 300 Yang Humidificador Ultrasónico y Difusor de Aroma | PcComponentes.com». <https://www.pccomponentes.com/cecotec-pure-aroma-300-yang-humidificador-ultrasonico-y-difusor-de-aroma> (accedido 27 de mayo de 2023).
- [25] «Tuya Convert - Tasmota». <https://tasmota.github.io/docs/Tuya-Convert/> (accedido 27 de mayo de 2023).
- [26] N. Faber, «scheduler-card». 6 de abril de 2023. Accedido: 10 de abril de 2023. [En línea]. Disponible en: <https://github.com/nielsfaber/scheduler-card>
- [27] N. Faber, «scheduler-component». 24 de mayo de 2023. Accedido: 27 de mayo de 2023. [En línea]. Disponible en: <https://github.com/nielsfaber/scheduler-component>
- [28] K. Kihlström, «Mini Media Player». 25 de mayo de 2023. Accedido: 27 de mayo de 2023. [En línea]. Disponible en: <https://github.com/kalkih/mini-media-player>
- [29] «Home Assistant Community Add-on: Node-RED». Home Assistant Community Add-ons, 25 de mayo de 2023. Accedido: 27 de mayo de 2023. [En línea]. Disponible en: <https://github.com/hassio-addons/addon-node-red>
- [30] «node-red-contrib-noble-bluetooth». <http://flows.nodered.org/node/node-red-contrib-noble-bluetooth> (accedido 28 de mayo de 2023).
- [31] «nRF Connect for Mobile - Apps en Google Play». https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=es_419&gl=US&pli=1 (accedido 28 de mayo de 2023).
- [32] «Gadgetbridge». Freeyourgadget, 26 de mayo de 2023. Accedido: 28 de mayo de 2023. [En línea]. Disponible en: <https://github.com/Freeyourgadget/Gadgetbridge>
- [33] «FitToFit - Apps on Google Play». https://play.google.com/store/apps/details?id=fitapp.fittofit&hl=en_US (accedido 28 de mayo de 2023).
- [34] «OpenMQTTGateway v1.5.1». <https://docs.openmqttgateway.com/> (accedido 28 de mayo de 2023).
- [35] «The Silly Home». <https://thesillyhome.ai/> (accedido 28 de mayo de 2023).

7 Anexo

7.1 Documentación código creación escenarios

1. Nuevo Escenario



Esta sección recibe un *start* por parte del usuario o cuando se presiona en la entidad *crear_escenario*, que leerá el nombre de escenario en el campo de texto con entidad *nombre_escenario*, que pasará por un nodo función que generará la ruta al escenario como vemos a continuación:

```

1
2 msg.filename = "/share/" + msg.payload + ".csv"
3 return msg;

```

El porqué de usar esa ruta viene de que el *Docker Node Red* se comunica con la máquina host con el volumen compartido */share*. A continuación, borrará el fichero anterior (si existe), agregará la cabecera al archivo con la función siguiente:

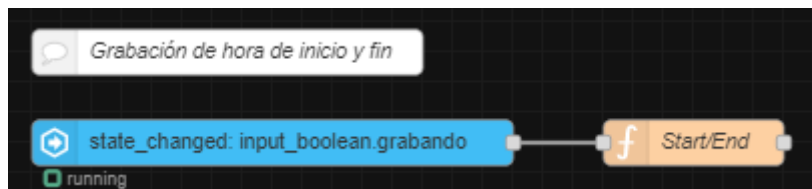
```

1 //msg.payload = "Inicio del escenario con fecha " + msg.startScene + "\n{}";
2 msg.payload = "ACTION_TIME, SCENE_STARTTIME, TIME_DIFF, ENTITY_ID, NEW_STATE, ATTRIBUTES"
3 return msg;

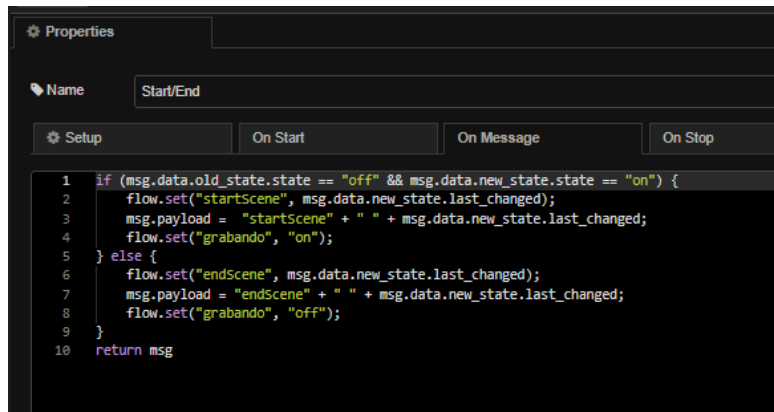
```

Por último, escribirá el archivo con el nombre y la ruta previamente creado.

2. Grabación de hora de inicio y fin

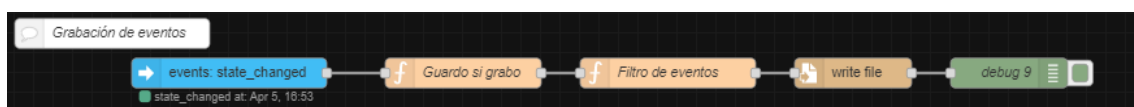


Esta sección tiene la misión de analizar los estados de la entidad *grabando*, y almacenar su valor de inicio y fin como variables de flujo que serán utilizadas para calcular el desfase con respecto a la hora de inicio. Las guardas utilizando el método *flow.set* cuando estas cambian de valor *off* a *on*:

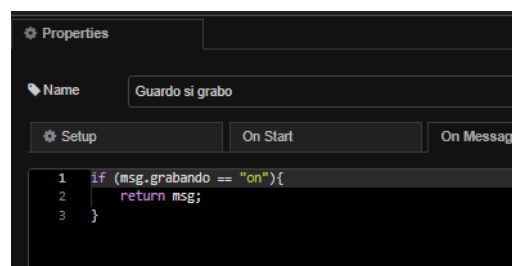


```
1 if (msg.data.old_state.state == "off" && msg.data.new_state.state == "on") {
2   flow.set("startScene", msg.data.new_state.last_changed);
3   msg.payload = "startScene" + " " + msg.data.new_state.last_changed;
4   flow.set("grabando", "on");
5 } else {
6   flow.set("endScene", msg.data.new_state.last_changed);
7   msg.payload = "endScene" + " " + msg.data.new_state.last_changed;
8   flow.set("grabando", "off");
9 }
10 return msg
```

3. Grabación de eventos



Es una de las partes más importantes. Esta sección estará a la escucha de todos los eventos de la sala. A modo de filtro, se comprobará si pasa a la siguiente fase únicamente si tengo la grabación activada, que será rechazada en caso contrario.



```
1 if (msg.grabando == "on"){
2   return msg;
3 }
```

Tras pasar el filtro de grabación, se analiza la entidad y se comprueba que el estado haya cambiado y si es de los tipos light, switch, cover, script, remote, media_player, sensor.temp, sensor.hum y no es no disponible. Almacena la entidad, el nuevo estado, la hora del log y la de inicio de escenario, los atributos (que al pasar a formato texto se perderán, pero no se ha trabajado en ello por desuso) y la diferencia entre el inicio y la hora de lanzamiento. Añade también el nombre del fichero con la ruta proveniente del módulo previo.

A continuación, muestra un log y borra lo innecesario.

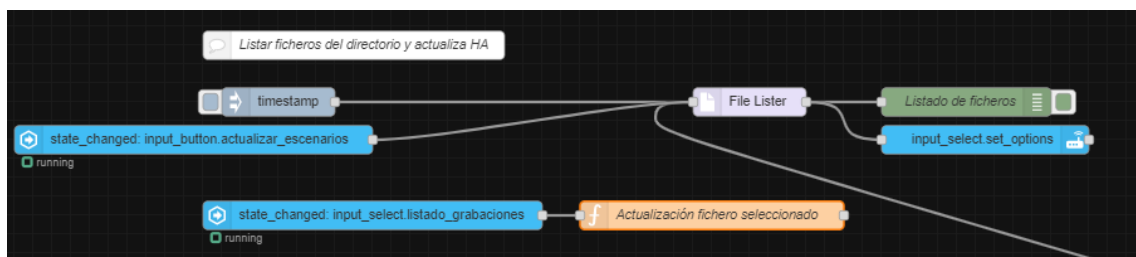
```

1  if (msg.payload.event.old_state.state == msg.payload.event.new_state.state){
2
3  } else {
4    if ((msg.payload.entity_id.startsWith("light.") || msg.payload.entity_id.startsWith("switch.") || msg.payload.entity_id.startsWith("cover.") ||
5      msg.payload.entity_id.startsWith("script.") || msg.payload.entity_id.startsWith("remote.") || msg.payload.entity_id.startsWith("media_player.") ||
6      msg.payload.entity_id.startsWith("sensor.temp" || msg.payload.entity_id.startsWith("sensor.hum")) && (msg.payload != "unavailable"))){
7      var log_entity = msg.payload.entity_id
8      var log_new_state = msg.payload.event.new_state.state
9      var log_time = new Date(msg.payload.time_fired)
10     var log_startScene = new Date(msg.startDate)
11     var log_attributes = msg.payload.event.new_state.attributes
12     var diff = log_time.getTime() - log_startScene.getTime();
13     var filename = msg.filename
14     msg.filename = "/share/" + filename
15     delete msg.topic
16     delete msg.payload.event_type
17     delete msg.payload.entity_id
18     delete msg.payload.event
19     delete msg.payload.origin
20     delete msg.payload.time_fired
21     delete msg.payload.context
22     msg.payload = log_time + "," + log_startScene + "," + diff + "," + log_entity + "," + log_new_state + "," + log_attributes;
23     return msg;
24   }
25 }

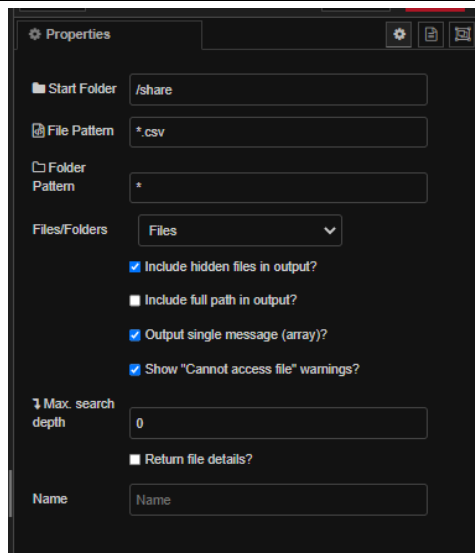
```

En caso favorable, se escribirá en el fichero seleccionado por el usuario como una línea al final.

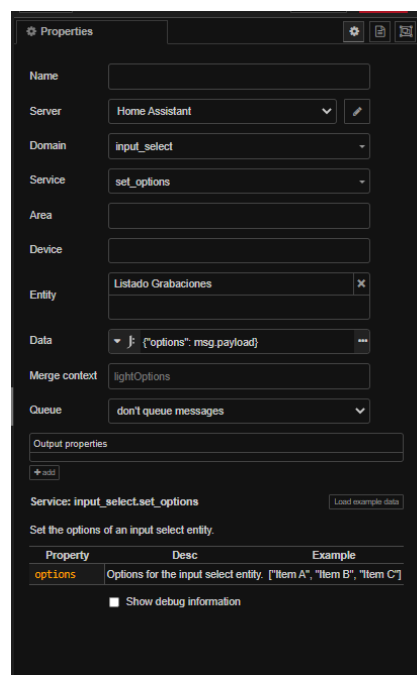
4. Listado de ficheros y actualización de fichero



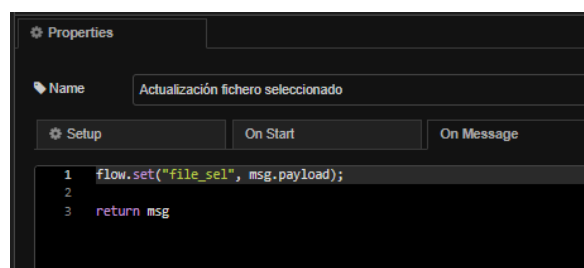
Esta sección se va a encargar de utilizando el nodo File Lister, que tendrá que ser instalado con node-red-contrib-fs, se encargará de listar los ficheros que acaban en “.csv” de la ruta /share, ya sea por un clic de usuario en Node Red o desde la entidad actualizar_escenarios, para que se actualice el desplegable de escenarios disponibles. La configuración de File Lister , ya que solo tendremos que tener en cuenta la ruta, la extensión de archivo y que la salida se adapte a lo que necesitamos:



La salida de este nodo, la muestra por pantalla, además de llamar al método `set_options` para actualizar las opciones de la entidad:



La segunda parte, se encarga de actualizar en variable de flujo el nombre del fichero seleccionado para poder ser utilizado por el resto de los nodos del flujo cada vez que este cambia, usando el método `set`:

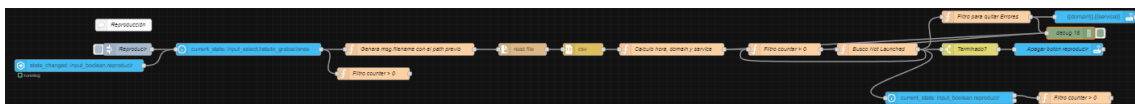


5. Borrado de ficheros



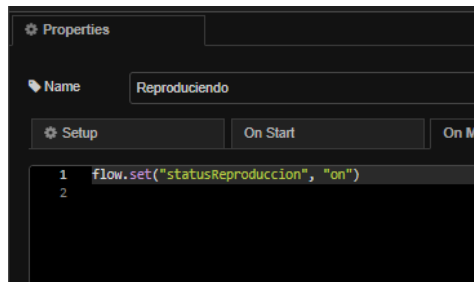
Esta sección se va a encargar de que cuando el usuario haga clic en borrar escenario, se leerá el fichero seleccionado en la entidad listado_grabaciones, generará la ruta al fichero y mandará eliminarlo. Tras su eliminación, llamará a actualizar la lista de los escenarios disponibles para evitar seleccionar contenido vacío. El funcionamiento del método de generar el path del archivo ya lo hemos visto en la sección de Nuevo Escenario.

6. Reproducción de ficheros

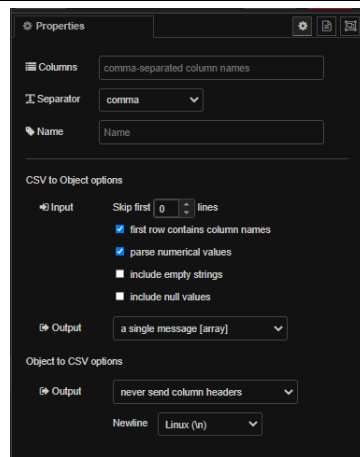


Esta parte es probablemente la más completo de los métodos que ofrece el sistema de grabación y reproducción.

En primer lugar espera que el usuario active la grabación, y usando el nombre de archivo seleccionado por el usuario en la interfaz de HA crea el path del archivo, además de cambiar el estado actual a reproduciendo:



Después, trata de leer el archivo configurando la lectura para que tras pasarla por el nodo csv, tengamos todos los registros de grabación como un array con las propiedades:



El hilo continua con un nodo que calcula el dominio y el servicio en función del tipo de entidad, además de calcular la hora a la que debería de lanzarse con respecto al inicio del escenario. Esta función también guarda en counter un indicador de la cantidad de acciones por lanzar para tener un control de la ejecución:

```
Properties
Name: Calculo hora, domain y service
Setup On Start On Message On Stop
1 //Primerio voy a recorrer todas y voy a calcular lo necesario
2 var fechaStart = new Date()
3 flow.set("fechaStartPlay", fechaStart)
4 msg.PlayStartTime = fechaStart;
5 var numero_acciones = msg.payload.length
6 msg.counter = numero_acciones
7 if(msg.payload.length > 0){
8   for (var i = 0; i < msg.payload.length; i++) {
9     //var playScenarioTime = msg.playScenarioTime;
10    // crea un nuevo objeto Date
11
12    var timediff = new Date(msg.payload[i].TIME_DIFF);
13    var actiontime = fechaStart.getTime() + timediff.getTime();
14
15    //Calculo la hora de la acción
16    msg.payload[i].PLAY_TIME = actiontime;
17
18    msg.payload[i].STATUS = "Not launched"
19
20    //Vemos que pasa si es tipo switch
21    if (msg.payload[i].ENTITY_ID.startsWith('switch.')){
22      msg.payload[i].DOMAIN = "switch"
23      if (msg.payload[i].NEW_STATE == "on"){
24        msg.payload[i].SERVICE = "turn_on"
25      } else if (msg.payload[i].NEW_STATE == "off"){
26        msg.payload[i].SERVICE = "turn_off"
27      } else{
28        msg.payload[i].SERVICE = "*****Unknow switch service"
29      }
30    }
31
32    //Vemos que pasa si es tipo light
33    if (msg.payload[i].ENTITY_ID.startsWith('light.')) {
34      msg.payload[i].DOMAIN = "light"
35      if (msg.payload[i].NEW_STATE == "on") {
36        msg.payload[i].SERVICE = "turn_on"
37      } else if (msg.payload[i].NEW_STATE == "off") {
38        msg.payload[i].SERVICE = "turn_off"
39      } else {
40        msg.payload[i].SERVICE = "*****Unknow light service"
41      }
42    }
43    //Atributos
44
45    //Vemos que pasa si es tipo script
46    if (msg.payload[i].ENTITY_ID.startsWith('script.')) {
47      msg.payload[i].DOMAIN = "script"
48      msg.payload[i].SERVICE = msg.payload[i].ENTITY_ID.split('.')[1]
49    }
50  }
51 }
```

El conjunto esta comprobando continuamente si continuo reproduciendo, es decir, no he parado la reproducción y si quedan elementos por lanzar:

```

1  var reproducir = flow.statusReproduccion
2  if (msg.counter > 0 && reproducir == "on"){
3      return msg;
4  }
5  return msg;
6

```

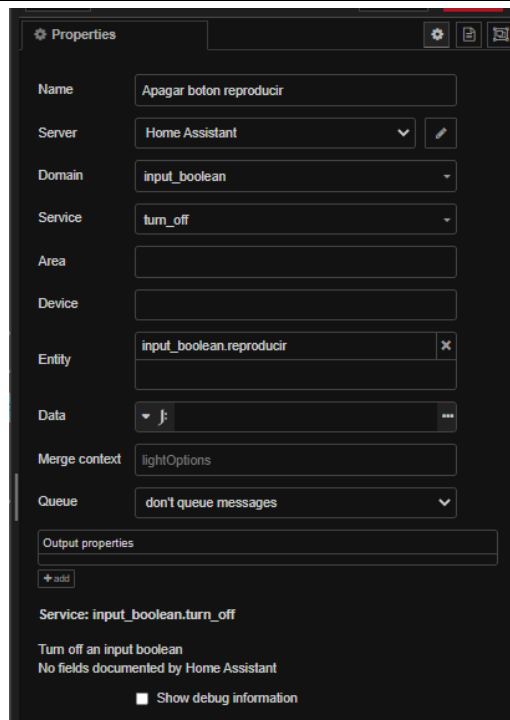
Si las condiciones previas se han cumplido, busca aquellos que aún están por lanzar, y si ha llegado su momento, lo lanza y lo cambia a lanzado y disminuye el contador:

```

1
2 //Ahora recorro todos buscando que esten lanzados
3
4 var i = msg.payload.length - msg.counter
5 while (msg.counter > 0) {
6     if (msg.payload[i].STATUS == "Not Launched"){
7         //Comprobamos si es su momento
8         var now = new Date();
9         var msg_time = new Date(msg.payload[i].PLAY_TIME);
10        var resta = msg_time.getTime() - now.getTime()
11        msg.resta = resta;
12        if(resta <= 0){
13            msg.domain = msg.payload[i].DOMAIN
14            msg.service = msg.payload[i].SERVICE
15            msg.entity = msg.payload[i].ENTITY_ID
16            msg.payload[i].STATUS = "Launched"
17            msg.counter = msg.counter - 1;
18            i = i+1;
19            return msg;
20        }
21        else{
22            delete msg.domain
23            delete msg.service
24            delete msg.entity
25            return msg;
26        }
27    }
28 }
29

```

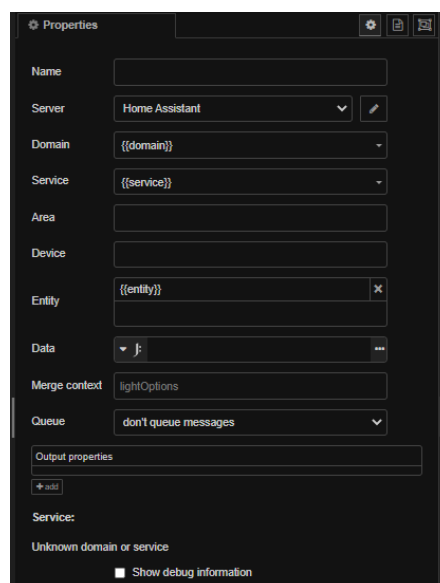
Los que ha ido disminuyendo y ha marcado como lanzados, salen del nodo y tropiezan con un nodo selector que comprobará si ese era el último por lanzar. Si lo era, lanza el apagado del interruptor de reproducción:



En caso de no haber llegado al final, recirculará y pasará a comprobar si sigue reproduciendo, y el comando por lanzar pasa por un filtro para evitar errores si tengo el dominio vacío (), y lanzará el nodo de llamar a un servicio de HA con el dominio y servicio previamente calculado ():

The screenshot shows a 'Filter' node configuration with the name 'Filtro para quitar Errores'. It has 'Setup' and 'On Start' tabs. The code in the 'Setup' tab is:

```
1 if (msg.domain){
2   return msg;
3 }
4
```

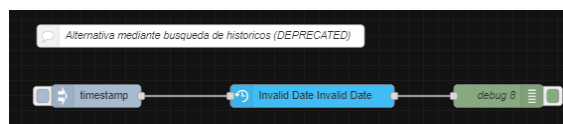


Además de esto, se activa un nodo que comprueba si sigo teniendo el botón de reproducción activo y el usuario no ha parado la reproducción a mitad, que es la que se comprueba en el nodo “Filtro counter>0”.

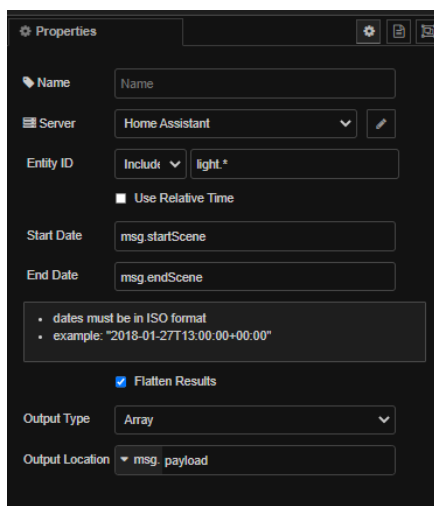
```

1 var statusRepr = msg.payload
2 if(statusRepr == "off"){
3     flow.set("statusReproduccion", statusRepr)
4 }
5
    
```

7. Búsqueda de históricos



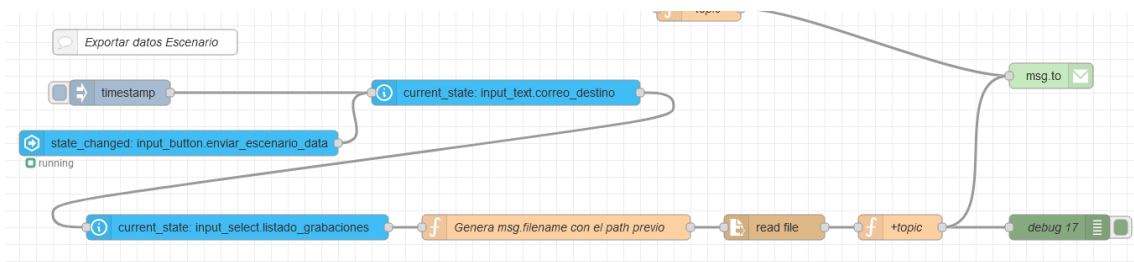
Esta sección está en desuso ya que no muestra los cambios de color o brillo, si no únicamente los cambios de estado, por lo que se descartó en su inicio. Aun así, es destacable su comentario. Este nodo funciona introduciendo una fecha de inicio y fin, y en este caso busca las entidades de tipo luz entre esas fechas, y lo exporta como un array para cada uno de los registros.



8. Exportación de datos de escenario y pulsera

Esta sección se encarga de abrir el fichero de escenario seleccionado y enviar el texto que contiene a través de correo cuando se pulse en el botón de enviar datos de escenarios. Para ello, es necesario previamente crear una credencial de aplicación de

Google y tener activa previamente la verificación en dos pasos. Existe más información al respecto en el apartado de ayuda de dicho módulo.



Abre el fichero donde hemos guardado el escenario con la lista de acciones realizadas, le actualiza el topic con el nombre del archivo seleccionado y lo incluye en el topic para que te llegue en el asunto el nombre del escenario.

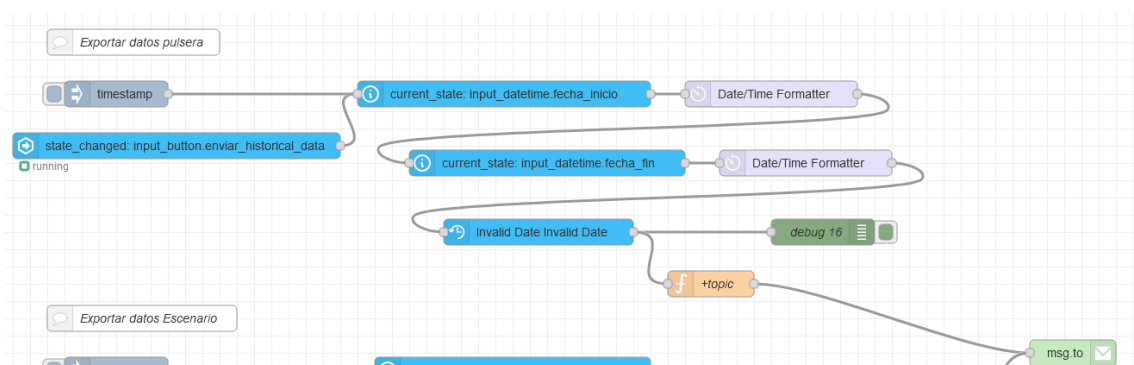
Name

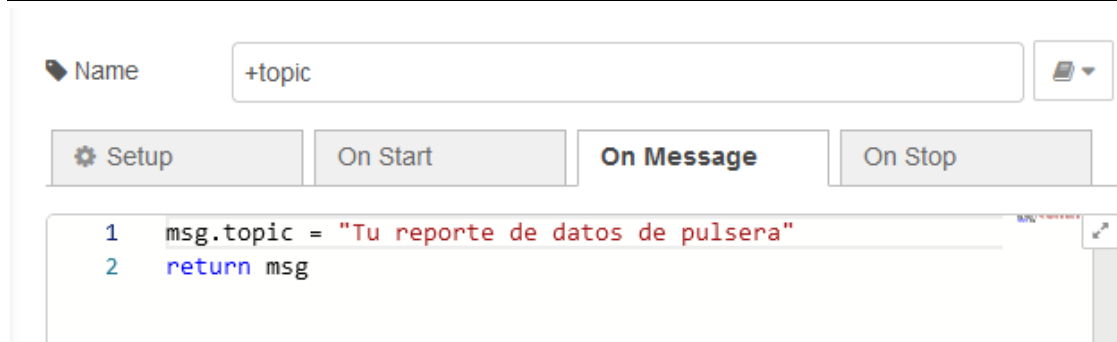
Setup On Start **On Message** On Stop

```

1 msg.topic = "Tu escenario "+msg.scenario
2 return msg
    
```

De manera similar, cuando se pulse en el botón de enviar datos de pulsera y previamente se hayan introducido en fecha y hora los instantes inicio y fin, que serán formateados para que el nodo de *Home Assistant* no de fallos, le añadirá el topic y de igual manera enviará el correo con los datos solicitados de las entidades de pulsaciones leídas de la pulsera:





Name

Setup On Start **On Message** On Stop

```
1 msg.topic = "Tu reporte de datos de pulsera"  
2 return msg
```

7.2 Manual de uso de la sala

SALA DE ESTIMULACIÓN MULTISENSORIAL

Manual de Uso

V1.0

CDIAT VIRGEN DE LA CARIDAD

astus

1 INTRODUCCIÓN

El propósito principal de este documento es la de ayudar al profesional con el uso de los distintos dispositivos que han sido integrados en la domotización de la sala.

Cabe mencionar que este manual incluirá rutinas de operación que pueden realizarse de distintas maneras, y que en caso de encontrar algún comportamiento anormal pueden solicitar ayuda al respecto.

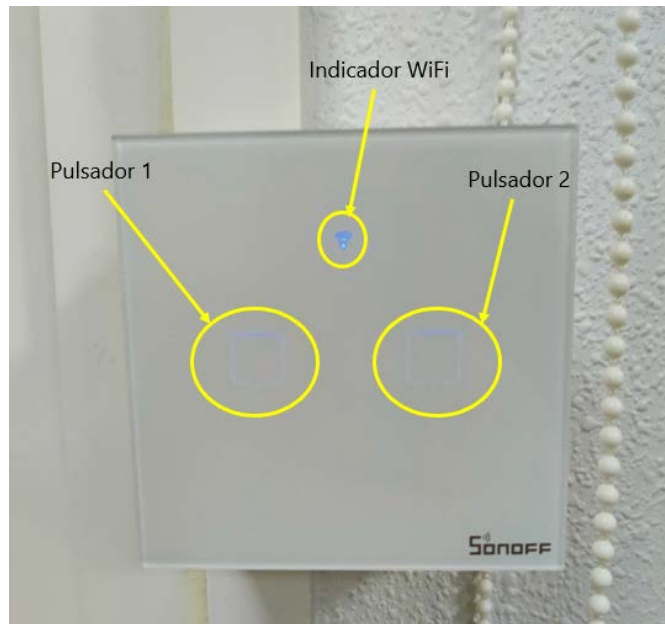
2 INTERFACES DE CONTROL

En este apartado se van a enumerar las distintas interfaces de control de las que disponemos en la sala:

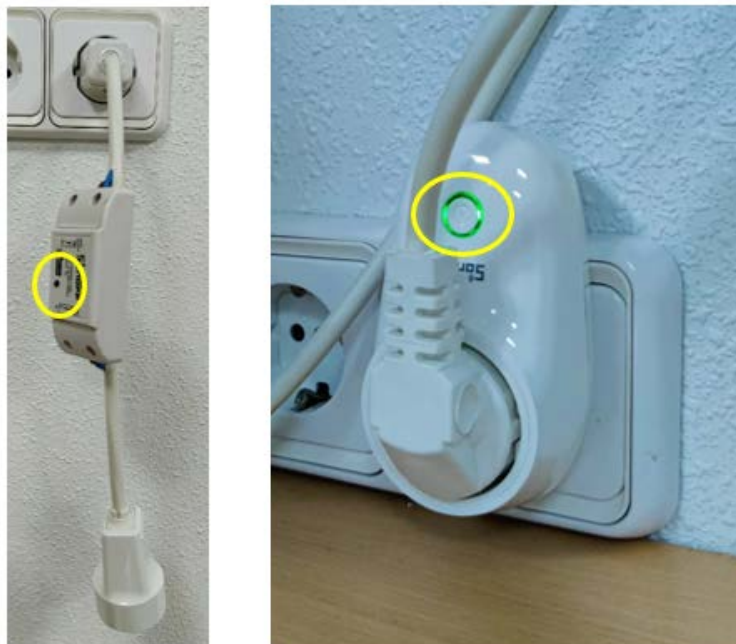
- a) *Pulsadores convencionales*: la sala dispone de un pulsador normal conectado al regulador de los ojos de buey, que alterna el estado de los mismos, y está situado cerca del panel de colores:



- b) *Pulsadores de pared inteligentes*: los paneles Sonoff como el de la entrada disponen de dos pulsadores táctiles que permiten cambiar el estado de las luces o lo que haya conectado. Con una pulsación, alternamos el estado. En el caso de las persianas será algo parecido, aunque al cerrarse o abrirse, se desconectará automáticamente. Estos paneles llevan un indicador WiFi. Un parpadeo indica que el dispositivo no se puede conectar a la red, o que se ha iniciado en modo de configuración ocasionado por pulsar el botón 1 durante mas de 10 segundos. En este caso, consulte el procedimiento de actuación.



- c) *Pulsadores en los elementos de control:* algunos dispositivos inteligentes de la sala incluyen un botón que permite alternar el estado del dispositivo como el de la siguiente imagen. Al pulsar, se alternará el estado del dispositivo. En el caso de los enchufes de la derecha, además del sonido de la activación del relé, veremos un cambio de color en el botón, que indica el estado.



- d) *Tablet u otro dispositivo inteligente:* cuando decimos la Tablet, hablamos de la Tablet cuyo propósito particular es poder controlar la sala, aunque cualquier dispositivo **conectado a la red de la sala** (SALA_EMS) y con usuario y contraseña podrá tener acceso. En el caso concreto de la Tablet, entramos utilizando la aplicación, aunque es posible el acceso introduciendo la siguiente url en el navegador e introduciendo usuario y contraseña. Normalmente si

accedes con el mismo dispositivo y no cierras sesión suele guardarte el usuario para evitar ponerlo todo el rato, aunque esta opción se puede desactivar:

<http://192.168.2.100:8123/>

Una vez listadas las interfaces disponibles para el control de la sala, se va a proceder a explicar cómo utilizar la Tablet de control.

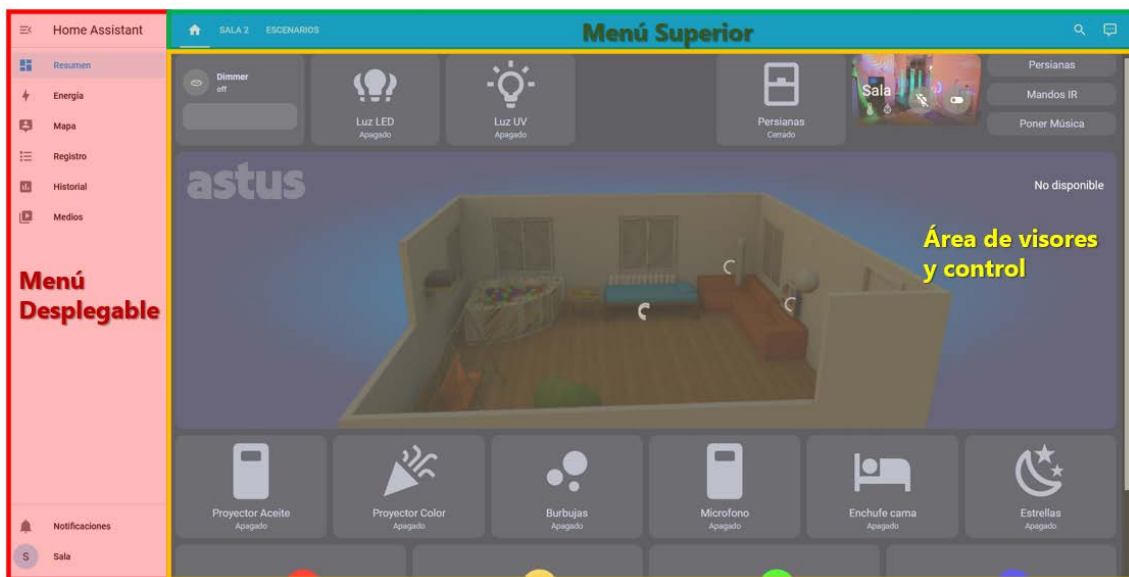
3 INTERFAZ SOFTWARE DE CONTROL

Como ya se ha comentado previamente, la sala dispone de una Tablet que se conecta a la red WiFi de la sala (SALA_EMS) para poder controlar los elementos que se encuentran en ella. Cuando se abre la aplicación *Home Assistant* puede pedirnos en algún momento la contraseña, entrando con el siguiente usuario:

Usuario: sala

Contraseña: 1234

Una vez dentro, la interfaz dispone de un menú desplegable a izquierda, un menú superior y lo que es la zona de visores y control:



- *Menú desplegable*: dispone de una serie de pestañas básicas que nos permiten acceder al historial, medios, registro, ajustes de usuario y notificaciones.
- *Menú superior*: nos permite navegar entre una serie de pestañas predefinidas. Dispondrá de 3 ventanas, la primera la principal desde donde se controlará toda la sala, la segunda que será similar a la primera, pero en una disposición de lista, y, por último, la ventana de escenarios que nos permitirá crear y manipular escenarios.
- *Área de visores y control*: reflejarán la vista seleccionada en el menú desplegable y superior. La disposición de ventanas será la siguiente:



Ahora se va a proceder a analizar cada una de las ventanas:

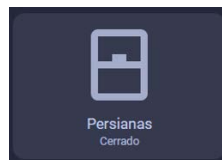
- **Home:**

Está separada en 7 partes y es la pantalla principal, la que más información contiene. Se listan y se explican a continuación cada una de las partes:

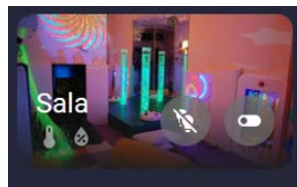
- a) *Control Iluminación:* permite alternar el estado de la luz LED, la ultravioleta, así como de alternar y modificar el brillo de Dimmer mediante deslizamiento de la barra. Estos botones cambiarán a iluminado cuando dicha luz esté encendida



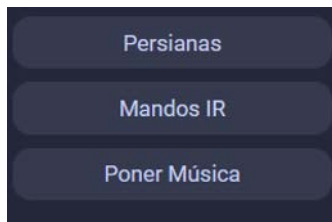
- b) *Control Persianas:* es un botón que permite abrir y cerrar todas las persianas de golpe. Cuando una de ellas esté abierta, se denotará como estado abierto. Para un control más exhaustivo, vaya al panel de navegación, a la pestaña de "Persianas".



- c) *Control Global:* este icono nos permite un control global mediante el apagado de todos los dispositivos de golpe y toda la iluminación de golpe. También nos da información de temperatura de la sala.



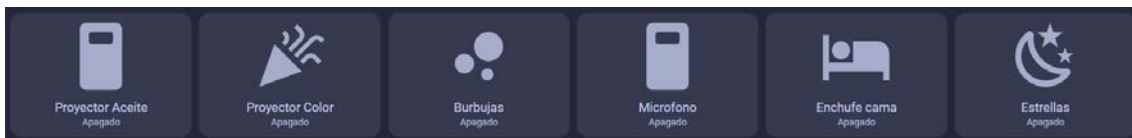
- d) *Navegación:* nos permite ir a las subvistas Persianas, Mandos IR y Poner Música que veremos más adelante.



- e) *Zona de supervisión:* permite visualizar el estado de la sala, ya que irá variando la imagen de fondo para asemejarse al estado actual de la sala. Es principalmente útil cuando no tenemos la sala delante.



- f) *Zona de Control Rápido*: esta zona permite activar y desactivar los principales elementos de la sala como son el proyector de láminas, el de colores con la bola, las burbujas, el micrófono, el enchufe de la cama y las estrellas. Se dispone de una guía de nombres de equipo con lo que representa en la realidad en el Anexo I del documento.

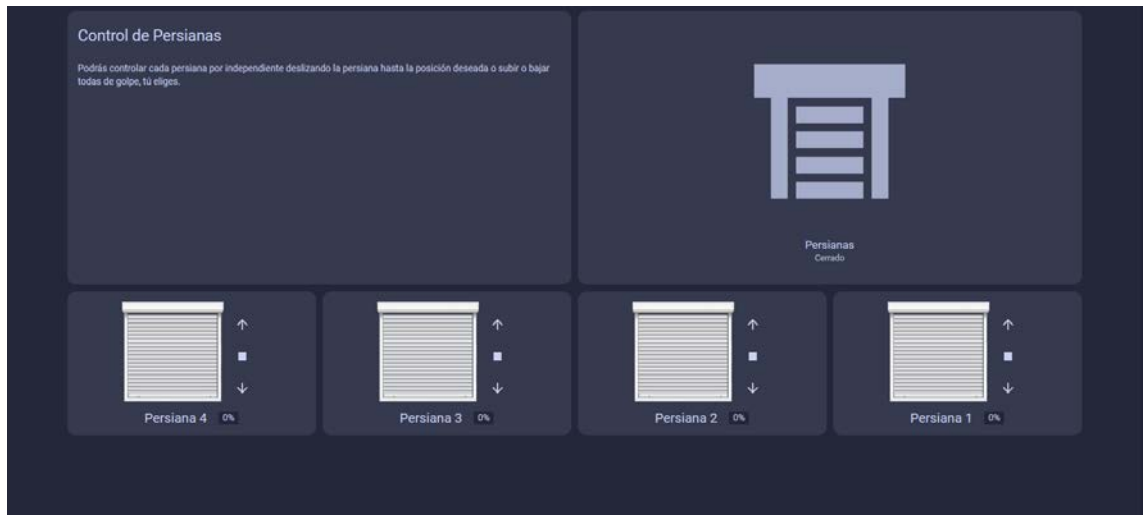


- g) *Cambio de color de burbujas*: esta zona permite únicamente cuando en enchufe de burbujas esta activo, lanzar comandos de variación de color de las burbujas. Si se esta un tiempo sin mandar un comando, el color escogido volverá a ser aleatorio.



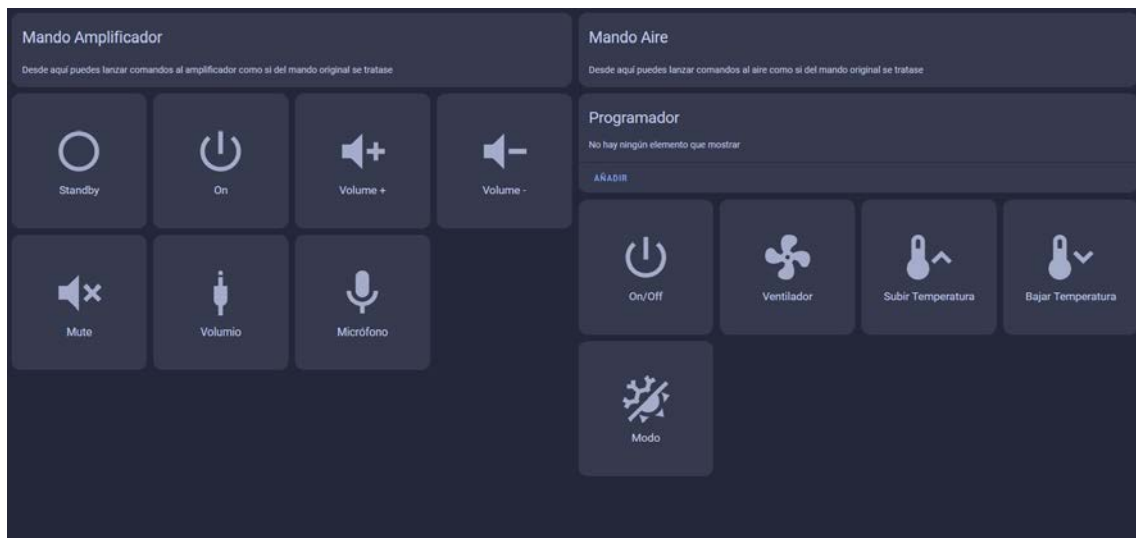
- **Persianas**

Como su nombre indica, esta subvista permite un control más concreto de cada una de las persianas y su nivel mediante el desplazamiento hasta la zona solicitada. También disponemos en grande del botón para cambiar todas de golpe o darle a los botones de comando de la persiana individualmente.

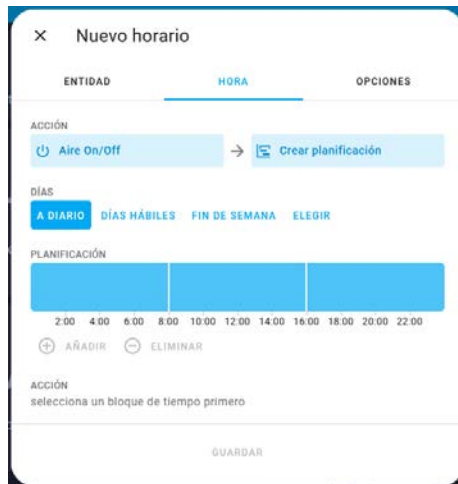


- **Mandos**

Esta subvista permite un control de los elementos hasta el momento que funcionaban con IR, además de un módulo en el centro a la derecha que permite configurar de manera horaria cuando se va a activar el aire.

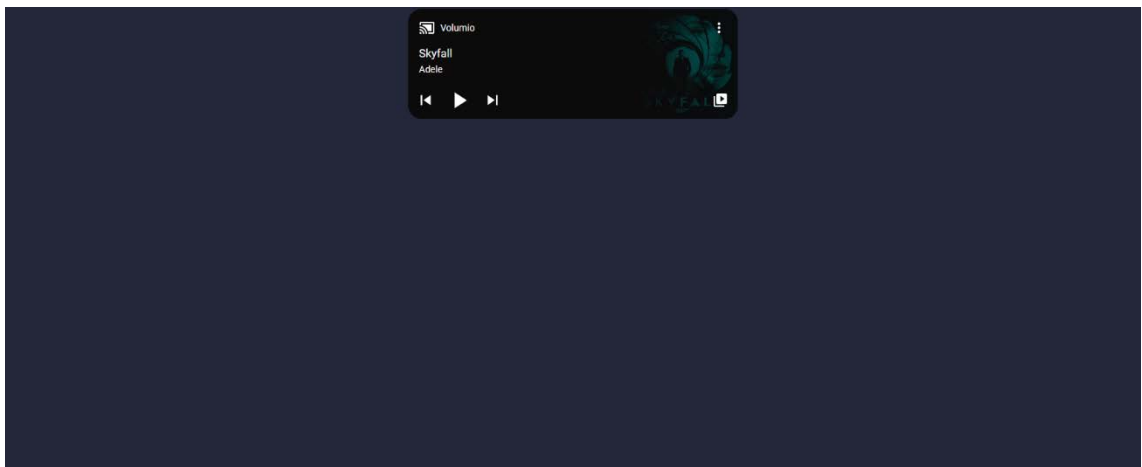


La ventana de programador permite desplazar las horas y cambiar la acción. Diferentes acciones y condiciones son configurables.

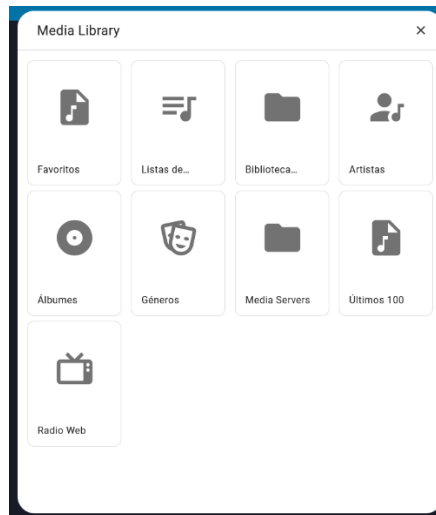


- **Música**

Esta ventana tiene una utilidad muy concreta y es la del control remoto de *Volumio*, que es el gestor de contenido multimedia, que también se puede hacer con la aplicación de *Volumio Control* instalada en la Tablet, o directamente enviar contenido desde *Youtube* mediante compartición de contenido como si de un dispositivo *Chromecast* se tratase.



Esta ventana nos permite seleccionar de la librería que contenido queremos escuchar en caso de que tengamos algo de música copiada o en un USB o en Youtube Music.

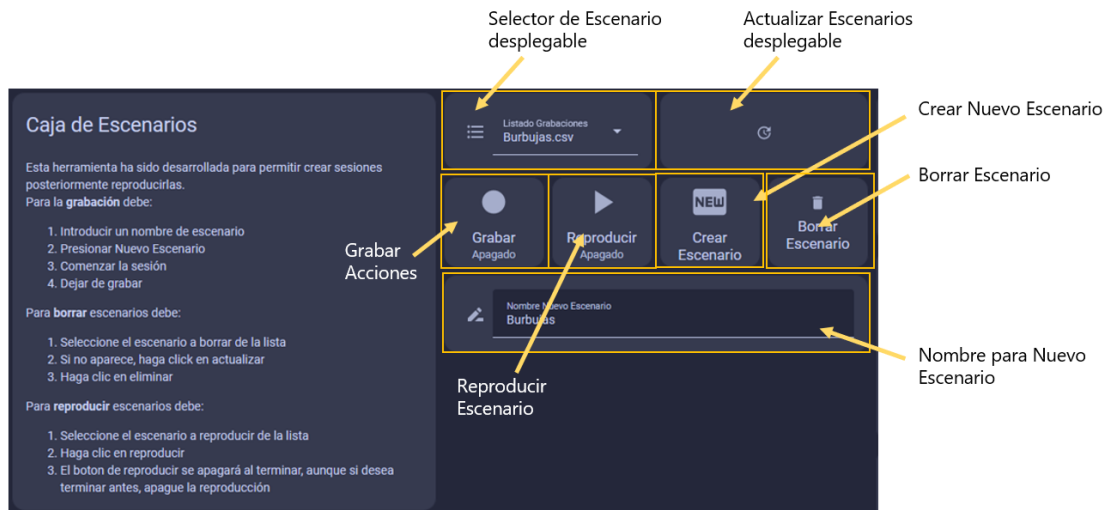


- **Escenarios:**

Esta pestaña esta separada en dos partes principales, una es la gestión de escenarios, otra la supervisión de entidades propias de la pulsera de actividad y por último un apartado de exportación de datos.



- a) *Zona de Escenarios:* esta zona contempla la generación de escenarios. Dispone de una guía a la izquierda que puede ayudar en su uso.



Esta herramienta permite crear un escenario con el nombre introducido, borrarlo y grabar acciones sobre el y luego reproducirlas. Se permite parar de grabar y luego seguir añadiendo acciones. Almacenará acciones sobre luces, enchufes, scripts tipo acciones sobre infrarrojos y algunos sensores.

- b) *Zona Monitorización*: la cual incluye dos secciones separando los datos provenientes de la pulsera Fitbit y la Mi Band 3. Aunque permanezca presente, únicamente se seguirá utilizando para el desarrollo y exportación la información de la pulsera de Xiaomi por su facilidad y precio. Almacenará información de pasos, frecuencia cardíaca y si esta conectada por Bluetooth o no.

Fitbit		
Fitbit heart rate		86 BPM
Fitbit steps		0 steps
Mi Band		
MB/SW-act_bpm		80 BPM
MB/SW-steps		0 steps
MB/SW-tracker		Desconocido

- c) *Zona de Exportación*: al igual que la Caja de Escenarios, la de reporte llevará una pequeña guía de uso, aunque principalmente deberemos seleccionar un correo de destino y escenario y hacer clic en Enviar Históricos de Escenario, y para exportar los históricos de la pulsera, una fecha de inicio y otra de fin, el correo de destino y hacer clic sobre Enviar Históricos de Pulsera para recibir un correo con la información solicitada.

Envío de Información Pulsera y Escenarios



Correo Destino
test@gmail.com



Enviar Históricos de Escenario seleccionado

PULSA



Fecha Inicio
15/5/2023



0 : 00



Fecha Fin
15/5/2023



0 : 00



Enviar Históricos de Pulsera

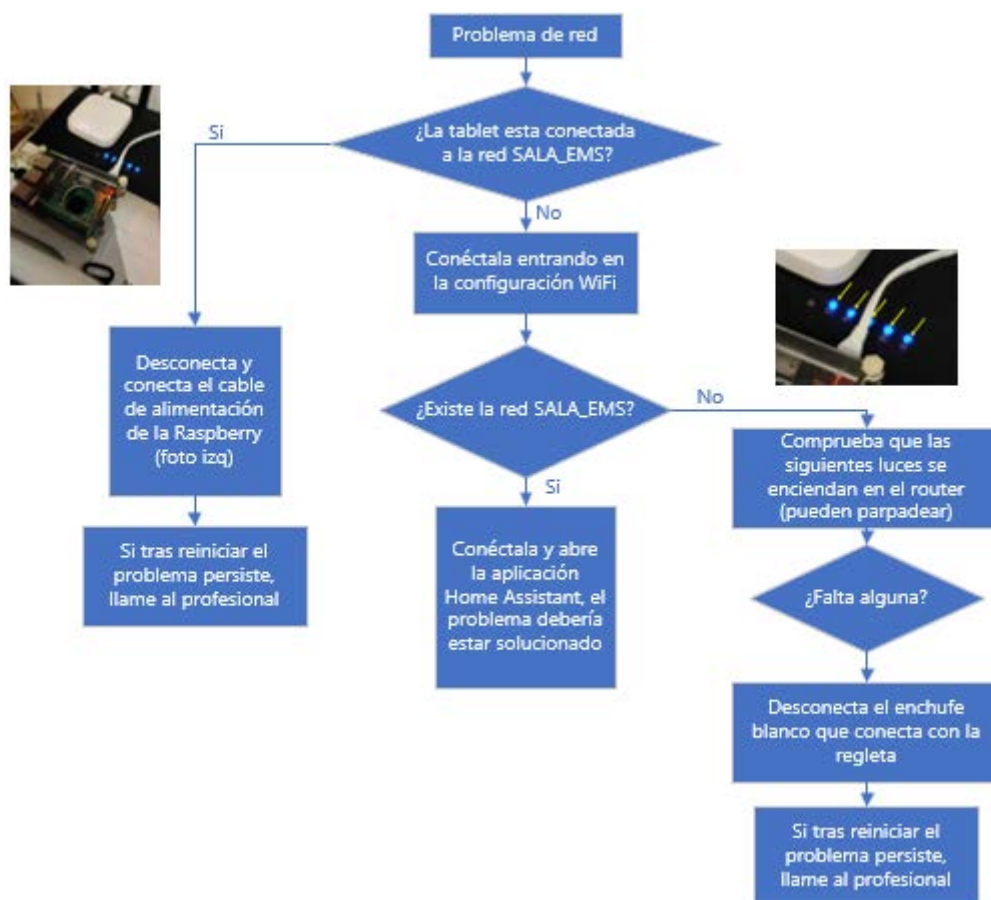
PULSA

4 TROUBLESHOOTING:

En este apartado se han querido reflejar algunos errores que pueden ocurrir y la manera de corregirlos. Antes de comenzar se debe indicar que cada vez que se hace un reinicio de la instalación puede tardar de 2-5 min en estar todo en marcha y listo, por lo que, si tras reiniciar la Tablet tenemos un problema de red en los siguientes 5 min, es completamente normal ya que el sistema se esta reestableciendo.

4.1 Error de comunicación de la tablet:

Este error es de los más comunes y sus causas pueden ser muy diversas. En primer lugar, debemos asegurarnos que la Tablet esta conectada a la red WiFi SALA_EMS. Siga el siguiente flujograma:



4.2 Error comunicación de alguno de los elementos:

También puede ocurrir que alguno de los dispositivos pierda la comunicación. Lo primero que deberemos de comprobar es si el problema es a nivel local (es decir, el resto de cosas funcionan) o si es solo este dispositivo.

Si el resto de dispositivos no funcionan, deberemos comprobar que existe la red SALA_EMS y seguir los pasos del diagrama anterior a partir de la pregunta ¿Existe la red SALA_EMS?

En el caso de que sea local, lo ideal es desconectar el dispositivo durante 10 segundos de la corriente para que se reinicie. En caso de ser algún elemento tipo estrellas a las que no se puede acceder, se recomienda abrir el cuadro eléctrico y bajar el térmico de iluminación, y posteriormente volver a subirlo. Si el problema persiste tras el reinicio, contactar con el profesional.

4.3 Fallo librería de Youtube Music en Volumio:

En algunos casos la integración que posee Volumio para visualizar directamente la librería de *Youtube Music* se desconecta sin avisar debido a que no es oficial y no esta muy depurada. Para solventar los problemas ocasionados podemos acceder a la aplicación de *Volumio*, ir a complementos e iniciar sesión de nuevo en *Youtube Music*.

5 NOMENCLATURA DE DISPOSITIVOS

El propósito de dicho apartado es la de reflejar al usuario el nombre asignado a los dispositivos físicos de la sala. Se presentan las siguientes imágenes comentadas:



