

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA
DE TELECOMUNICACIÓN
TRABAJO FIN DE MÁSTER**

**CONTROL Y MONITORIZACIÓN DE
INFRAESTRUCTURA ENERGÉTICA EN
LOS CAMPUS DE LA UPM**

FRANCISCO JAVIER NIETO ORTIZ

2020

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

Título: CONTROL Y MONITORIZACIÓN DE INFRAESTRUCTURA ENERGÉTICA EN LOS CAMPUS DE LA UPM

Autor: D. FRANCISCO JAVIER NIETO ORTIZ

Tutor: D. JOSE IGNACIO MORENO NOVELLA

Departamento: DEPARTAMENTO DE INGENIERÍA DE SISTEMAS TELEMÁTICOS

MIEMBROS DEL TRIBUNAL

Presidente: D.

Vocal: D.

Secretario: D.

Suplente: D.

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:
.....

Madrid, a de de 2020

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA
DE TELECOMUNICACIÓN
TRABAJO FIN DE MÁSTER**

**CONTROL Y MONITORIZACIÓN DE
INFRAESTRUCTURA ENERGÉTICA EN LOS
CAMPUS DE LA UPM**

FRANCISCO JAVIER NIETO ORTIZ

2020

AGRADECIMIENTOS

A mi madre, por ser la persona que me ha escuchado todas las mañanas y que, a pesar de la distancia, ha sabido enseñarme el camino.

A mi padre, por ser un referente, ejemplo a seguir y apoyarme en todo lo que hago. Nos esperan muchos momentos juntos y este ha sido uno más de ellos.

A mis hermanas, por apoyarme en los momentos más débiles y estar ahí siempre.

A mis abuelos, sé que estarían muy orgullosos de haber podido leer este trabajo.

A mi tutor, por ser una de las personas más importantes que he tenido en la universidad, por haberme ayudado y dado las fuerzas para lograr mis objetivos.

A mis amigos, por siempre haberme ayudado en todo. Puedo sentirme muy afortunado de tenerlos.

RESUMEN

Uno de los principales problemas a los que se enfrentan todos los habitantes del planeta es el cambio climático. La producción y el consumo de energía, junto al transporte, son la principal causa de la emisión de gases de efecto invernadero. Los países son muy conscientes de ello y es por esta razón por la que tratan de tomar una serie de medidas para poder proteger el planeta. En 2015 la ONU se reunió para impulsar el desarrollo sostenible con unos objetivos claros para el 2030.

Vivimos en una sociedad en la que cada día estamos conectados a más dispositivos. En España según el Informe Digital de 2019 realizado por We Are Social, el 93% de la población está conectada a Internet y un 87% de la población dispone de un Smartphone. Todo esto lleva a que aparezcan nuevas necesidades en el estilo de vida de los usuarios y por eso, cada vez es más frecuente la aparición de dispositivos cotidianos con conexión a la red.

Este es el origen del Internet de las cosas (IoT), que hace referencia a dispositivos conectados y a cómo el usuario saca partido de ellos. Los dispositivos de IoT nos sirven para poder transmitir y recoger datos, como puede ser a través del uso de sensores. Además, son muy útiles para enviar información sobre actuadores sin la necesidad de intervención humana a un bajo coste.

Este proyecto nace por la necesidad de recopilar información sobre el uso del consumo energético en la Universidad y a través del análisis de estos, intentar aplicar políticas para mejorar la eficiencia energética de las infraestructuras. Por ello, el objetivo de este estudio es diseñar, desarrollar e implementar una plataforma que sirva para el control y monitorización de infraestructuras energéticas dentro de los campus universitarios de la UPM, utilizando sensores y actuadores de bajo coste para controlar y medir el consumo energético. Para ello, tras realizar un diseño de la arquitectura general del sistema se ha realizado un despliegue en dos fases. Inicialmente se ha montado en un entorno de laboratorio, donde se han configurado estos dispositivos y se han realizado una serie de pruebas de cara a una su posterior instalación. Posteriormente, se han llevado a un entorno en producción como es la Escuela Técnica Superior de Telecomunicaciones de la UPM.

El trabajo se ha desarrollado dentro de la ETSIT para el control y monitorización de los alumbrados en los edificios A y C. Se han tomado datos sobre el consumo energético para después tomar decisiones sobre ellos en forma de políticas que permitan una mayor eficiencia. Estas políticas se basan en aplicar automatizaciones en los dispositivos instalados y evitar que en horas donde no haya actividad dentro de la universidad, se produzca un mal uso de la energía eléctrica.

Finalmente, la UPM dentro del Marco de la Estrategia de Sostenibilidad ha aprobado el proyecto ReS2+U (Responsables, Sostenibles, Sociales y más Universitarios), que es una iniciativa para contribuir al logro de los Objetivos de Desarrollo Sostenible de la ONU. Este trabajo pretende lograr cumplir con los objetivos del área de monitorización. En este ámbito este trabajo ha sido seleccionado con el premio de una de las Becas que otorga la UPM en su programa de Campus Sostenible a mejor TFM.

SUMMARY

Climate change is one of the main challenges that every human being on earth has to face nowadays. Energy generation and consumption, together with the use of transportation means, bring the largest amount of greenhouse gas emissions to the globe. The awareness of most of the countries is flagrant and for this reason, they strive for taking actions to preserve the planet. In 2015, UN organization met to boost sustainable development with numerous objectives for 2030.

These days, people live in a society where connected devices are continuously increasing its presence. For instance, in Spain and according to the Digital Survey made by ‘We are Social’, around 93% of the population has an Internet connection and the 87% owns a smartphone. This situation makes incoming necessities to arise in the daily routine of inhabitants and therefore, day by day is even more frequent the emergence of new mundane devices that have connection to the network.

This is the origin of Internet of things (IoT), which refers to connected devices and how the user draw on them. IoT devices are extremely useful for transmission and recollection of data through sensors, as well as being worthwhile, at a low cost, for transmitting the information collected from actuators without the necessity of human intervention.

This project come into existence due to the necessity of gathering data about the energy consumption at University and using this information to establish new procedures to enhance energy efficiency of the infrastructure. For this reason, the aim of this study is to design, develop and implement a platform that is able to control and monitor energetic infrastructures within UPM college campus, using low cost sensors and actuators. Due to this reason and after developing the general system architecture design, a two-phase initial deployment has been made. Initially, a laboratory setting was created, where the devices were configured and then several tests were passed prior to their installations. Subsequently, the devices were taken to a final production environment such as ETSIT-UPM is.

The project has focused on ETSIT campus with the purpose of controlling and monitoring the illumination of buildings A and C. This will lead to different conclusions about energy consumption, which will became into better efficiency policies thanks to the analysis of the data from the connected devices. These policies are based on applying automation processes to the installed sensors and actuators in order to avoid an inappropriate use of electric power when there is no activity inside a building.

Finally yet importantly, it is of huge interest to highlight that UPM, within the Sustainability and Strategy Framework, holds the ReS2+U project with the aim of fulfilling the Sustainable Development Objectives embraced by NU. This work claims to complete all the area-monitoring goals. Besides, the project has been honoured with one of the Internships that UPM grants for the best Master Final Thesis within the campus-sustainable program.

PALABRAS CLAVE

Eficiencia energética, Sostenibilidad, Internet de las cosas, IoT, MQTT, Monitorización, Automatización, Consumo, Sensores, Home Assistant, Objetos conectados

KEYWORDS

Energy efficiency, Sustainability, Internet of Things, IoT, MQTT, Monitoring, Automation, Energy consumption, Sensors, Home Assistant, Connected Devices.

ÍNDICE DEL CONTENIDO

1. INTRODUCCIÓN Y OBJETIVOS.....	1
1.1 INTRODUCCIÓN	1
1.2 OBJETIVOS DEL PROYECTO	3
1.3. FASES REALIZADAS	3
1.4. ENTORNO DE DESARROLLO	4
1.5. ORGANIZACIÓN DE LA MEMORIA	4
2. ESTADO DEL ARTE.....	6
2.1 PROTOCOLOS, ESTÁNDARES Y TECNOLOGÍAS DE IoT	6
2.1.1 TECNOLOGÍAS Y PROTOCOLOS DE COMUNICACIÓN/TRANSPORTE	6
2.1.2 PROTOCOLOS DE DATOS	9
2.2 PLATAFORMAS DE DISPOSITIVOS IOT	12
2.2.1 HOME ASSISTANT	12
2.2.2 OPENHAB	13
2.2.3 DOMOTICZ	14
3. DISEÑO Y DESARROLLO DEL SISTEMA	16
3.1 ARQUITECTURA DEL SISTEMA DE CONTROL Y MONITORIZACIÓN DE INFRAESTRUCTURAS ENERGÉTICAS	16
3.2 DISPOSITIVOS	19
3.3 FIRMWARE Y FLASHEO DE DISPOSITIVOS.	23
3.4 INSTALACIÓN DE HOME ASSISTANT	29
3.5 INSTALACIÓN DE MOSQUITTO BROKER	32
3.6 ARCHIVOS DE CONFIGURACIÓN YAML	33
3.7 INTEGRACIÓN DE DISPOSITIVOS	34
3.8 INTERFAZ GRÁFICA DE HOME ASSISTANT	39
3.9 AUTOMATIZACIONES	48
3.10 VISUALIZACIÓN DEL SISTEMA	50
4. VALIDACIÓN Y EVALUACIÓN.....	53
4.1 VALIDACIÓN EN LABORATORIO	53
4.2 VALIDACIÓN DEL PILOTO	56
4.3 EVALUACIÓN DE LA PLATAFORMA	62
4.4 RECOMENDACIONES	63
5. CONCLUSIONES	64
5.1 CONCLUSIONES	64
5.2 TRABAJOS FUTUROS	64
6. BIBLIOGRAFÍA	65

ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES.....	67
A.1 INTRODUCCIÓN	67
A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO	67
A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS	67
A.4 CONCLUSIONES	67
ANEXO B: PRESUPUESTO ECONÓMICO.....	68
ANEXO C: GUÍA PARA CAMBIAR EL FIRMWARE A TASMOTA EN DISPOSITIVOS SONOFF.....	70
C.1 INTRODUCCIÓN Y OBJETIVOS	70
C.2 HARDWARE NECESARIO	70
C.3 CONSIDERACIONES PREVIAS	71
C.4 INSTALACIÓN TASMOTA	74
ANEXO D. GUÍA DE INSTALACIÓN DE HOME ASSISTANT Y SERVIDOR MQTT MOSQUITTO.....	79
D.1 HARDWARE NECESARIO	79
D.2 SOFTWARE NECESARIO	79
D.3 INSTALACIÓN DE HOME ASSISTANT	80
ANEXO E. GUÍA DE ACCESO A LA PLATAFORMA HOME ASSISTANT PARA PERSONAL DE LA ETSIT	86
E.1 ACCESO A LA PLATAFORMA	86
E.2 NAVEGACIÓN POR LA PLATAFORMA	86
E.3 FUNCIONALIDADES DE LA PLATAFORMA	87
E.4 ESTADO DE LOS DISPOSITIVOS	89

ÍNDICE DE FIGURAS

Figura 1 Logotipo del Proyecto RES2+U.....	1
Figura 2 Smarts Cities e IoT [4].....	2
Figura 3 Interconexión de dispositivos con Bluetooth Low Energy [5].	6
Figura 4 Tecnología ZigBee [5].....	7
Figura 5 Estructura de la red Sigfox [6].....	9
Figura 6 Funcionamiento MQTT Broker [7].....	10
Figura 7 Interfaz de usuario de Home Assistant.....	13
Figura 8 Interfaz de usuario openHab.....	14
Figura 9 Interfaz de Domoticz.....	15
Figura 10 Arquitectura general del sistema.....	16
Figura 11 Arquitectura del sistema en laboratorio.....	17
Figura 12 Arquitectura final del sistema de monitorización en la ETSIT.....	18
Figura 13 Sonoff Wifi Smart Switch.....	19
Figura 14 Sonoff Pow R2.....	20
Figura 15 Sonoff TH16 y sensor Si702.....	20
Figura 16 Sonoff 4CH.....	21
Figura 17 Sonoff S20/S26.....	22
Figura 18 Sonoff DIY Smart Switch.....	23
Figura 19 Sonoff Mini DIY Smart Switch.....	23
Figura 20 Interconexión entre FTDI y Sonoff Smart Switch.....	26
Figura 21 Placa Sonoff Switch antes de ser conectada.....	26
Figura 22 Proceso de Flasheo con PyFlasher.....	27
Figura 23 Punto de acceso creado por Sonoff después del flasheo.....	27
Figura 24 Página de configuración Wifi de Tasmota.....	28
Figura 25 Consola Termite.....	29
Figura 26 Configuración del módulo Sonoff.....	29
Figura 27 Página de un Router Movistar para ver las direcciones IP.....	31
Figura 28 Página de inicio de Home Assistant.....	31
Figura 29 Instalación Mosquitto.....	32
Figura 30 Configuración de puertos en Mosquitto.....	33
Figura 31 Archivos YAML Home Assistant.....	33
Figura 32 Menú de configuración de MQTT de Tasmota.....	34
Figura 33 Configuración Sonoff Switch en configuration.yaml.....	36
Figura 34 Consola de Tasmota.....	38
Figura 35 Entidades de un dispositivo.....	39
Figura 36 Vistas de Home Assistant.....	39
Figura 37 Página de selección de tarjetas de la interfaz de Home Assistant.....	41
Figura 38 Configuración de tarjeta Entidades.....	42
Figura 39 Estado del dispositivo.....	42
Figura 40 Ubicación de red de Hassio.....	43
Figura 41 Vista con una tarjeta de imagen y diferentes entidades.....	44
Figura 42 Gráfica de consumo de Home Assistant.....	45
Figura 43 Consumo de los dispositivos Sonoff Pow.....	46
Figura 44 Página para crear un Token de acceso en GitHub.....	51
Figura 45 HACS de Home Assistant.....	52
Figura 46 Pantallazo de Termite que muestra el error de conexión de red a ServiciosUPM.....	54
Figura 47 Edificio A en Home Assistant.....	58
Figura 48 Edificio C en Home Assistant.....	58
Figura 49 Medidas de potencia de Home Assistant.....	60
Figura 50 Problema de conectividad en los dispositivos.....	61

ÍNDICE DE TABLAS

Tabla 1 Comparación Tecnologías para comunicaciones de corto/medio alcance.....	8
Tabla 2 Comparación Tecnologías para comunicaciones de largo alcance	9
Tabla 3 Comparación de protocolos para transmisión de datos [9].	12
Tabla 4 Interconexión entre FTDI y Sonoff.....	25
Tabla 5 Tabla de funcionalidades de Sonoff Switch y Pow.....	35
Tabla 6 Medidas de potencia sin aplicar políticas de automatización.	57
Tabla 7 Medidas de potencia después de aplicar políticas de automatización.....	59

ACRÓNIMOS

API	Application Programming Interface
AFH	Adaptive Frequency Hopping
BLE	Bluetooth Low Energy
CSS	Chip Spread Spectrum
C2C	Consumer to Consumer
DCPS	Data Centric Publish Subscribe
DDS	Data Distribution Service
DLRL	Data Local Reconstruction Layer
D2D	Device to Device
ETSIT	Escuela Técnica Superior de Ingenieros de Telecomunicación
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
IP	Internet Protocol
LAN	Local Area Network
LORA	Long Range
LORAWAN	Long Range Wide Area Network
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
ONU	Organización de Naciones Unidas
OTP	One Time Password
RES2+U	Responsables, Sostenibles y Universitarios
RFID	Radio Frequency Identification
SASL	Simple Authentication and Security Layer
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer security
UDP	User Datagram Protocol
UPM	Universidad Politécnica de Madrid
URI	Uniform resource identifier
VPN	Virtual Private Network
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

1. INTRODUCCIÓN Y OBJETIVOS

1.1 INTRODUCCIÓN

El cambio climático es un fenómeno global que afecta a la vida de las personas con consecuencias como el aumento de las temperaturas, aumento de fenómenos meteorológicos o incluso la propagación de enfermedades tropicales. Las ciudades y la vida urbana son el principal emisor de gases de efecto invernadero. Se estima que el 75% de las emisiones globales de CO2 son debido a la producción y uso de energía, y el transporte dentro de ellas. [1]

La Asamblea General de la ONU en septiembre de 2015 se reunió para establecer una serie de objetivos de cara al 2030. Estos objetivos tienen como fin metas para asegurar la prosperidad y proteger el planeta. Se tratan en concreto de 17 objetivos con 169 metas que abarcan diferentes ámbitos dentro las esferas económicas, sociales y ambientales. Con el siguiente trabajo trataremos de cumplir con algunos de los objetivos que se promovieron en esta cumbre, en concreto, nos centraremos en los siguientes:

- Objetivo 9 (Industria, innovación e infraestructuras): Para poder lograr un desarrollo sostenible es necesario hacer inversiones en el ámbito tecnológico y mejorar las infraestructuras actuales. En este proyecto trataremos de abordar a través de dispositivos IoT la eficiencia de los recursos energéticos en la Escuela Técnica Superior de telecomunicaciones de la Universidad Politécnica de Madrid.
- Objetivo 11 (Ciudades y comunidades sostenibles): En la Universidad somos una comunidad y tenemos el deber de llevar a cabo estos objetivos y metas. Los estudiantes somos la base de esta comunidad y, por tanto, somos los encargados de que esta siga prosperando y creciendo de manera eficiente y sostenible.
- Objetivo 12 (Producción y consumo responsable): en la actualidad el uso de recursos del planeta está aumentando considerablemente. Cuando hablamos de consumo y producción sostenible nos referimos al uso eficiente de los recursos, se trata de hacer más con menos y fomentar la reducción de estos. Con este proyecto queremos mejorar la eficiencia energética de la escuela y reducir el consumo energético de la misma.

En este trabajo se abordarán todos estos objetivos para desarrollar un sistema de monitorización energética de la ETSIT y que sirva de referencia al resto de escuelas de la UPM. En relación con este aspecto, la UPM dentro del Marco de la Estrategia de Sostenibilidad cuenta con el proyecto ReS2+U para contribuir al logro de los Objetivos de Desarrollo Sostenible de la ONU.



Figura 1 Logotipo del Proyecto RES2+U.

El uso del Internet de las cosas ha traído consigo varios desafíos, entre los que destaca la seguridad y la privacidad. Como hemos hablado anteriormente estos dispositivos te permiten recopilar datos constantemente y tomar decisiones en función de estos, lo que conlleva el riesgo de que al estar conectados a Internet pueden sufrir vulnerabilidades y enviar esta información a personas no autorizadas. La información se almacena en un servidor, que en muchos de los casos no se tiene acceso y puede conllevar problemas de seguridad y privacidad.

1.2 OBJETIVOS DEL PROYECTO

Tras haber hablado en el apartado anterior de los objetivos que tiene la organización de Naciones Unidas para el desarrollo sostenible en el 2030 y el aumento del uso de dispositivos IoT en la vida cotidiana, nos surge la necesidad de desarrollar un sistema para mejorar la eficiencia energética a través de dispositivos de bajo coste de IoT. Este sistema lo desarrollaremos en el entorno de la universidad para poder mejorar las infraestructuras energéticas. Se desarrollará un piloto en la ETSIT que permita el control y actuación sobre el alumbrado de los edificios A y C.

El objetivo final de este proyecto es el de diseñar, desarrollar y validar una plataforma adecuada para el control y monitorización de infraestructuras energéticas que sirva de referencia dentro de los campus universitarios de la UPM.

Para poder llevar a cabo este objetivo final se han de desarrollar los siguientes objetivos específicos:

- Realizar un estudio de las diferentes tecnologías, estándares y protocolos para la comunicación/transporte y transmisión de datos en el ámbito del internet de las cosas.
- Realizar un estudio de las diferentes plataformas para implementar los dispositivos IoT.
- Diseñar una arquitectura para soportar un sistema de control y monitorización de infraestructuras energéticas.
- Desarrollo de la arquitectura en un entorno de laboratorio.
- Creación de un piloto en un entorno en producción ubicados en los edificios A y C de la ETSIT.
- Evaluación de aspectos de eficiencia energética en el piloto.

1.3. FASES REALIZADAS

El proyecto se desarrollará en las siguientes fases:

Fase 1: Estudio de dispositivos IoT, protocolo MQTT y plataformas para la monitorización de estos dispositivos.

Fase 2: Diseño y desarrollo de la plataforma. Se desarrollará un prototipo para pruebas en laboratorio y posteriormente se volcará en un servidor para alojar los dispositivos de la ETSIT.

Fase 3: Preparación de la red e instalación de los dispositivos. Se estudiará el comportamiento de estos dispositivos en la Red “Servicios UPM”, se integrarán y se instalarán en diferentes zonas de la Escuela.

Fase 4: Monitorización de la infraestructura energética sin ningún tipo de política o automatización para hacer un estudio de consumo energético de la misma.

Fase 5: Aplicación de políticas y automatizaciones para mejorar el consumo energético y la eficiencia de la escuela.

1.4. ENTORNO DE DESARROLLO

En este apartado se van a describir los elementos Hardware que se han utilizado para el desarrollo del trabajo:

- Raspberry Pi 4 Model B + Carcasa+ Disipadores térmicos + Tarjeta de memoria de 32 GB categoría 10 + Cable alimentación 2.5 A.
- Controladora para transferencia de datos en serie de un solo chip. FTDI FT232RL USB to TTL Serial Converter Adapter Module 5V and 3.3V.
- Cables para conectar controladora con dispositivos. Jumper Wire Cables de Puente hembra a hembra.
- Pines para soldar en los sensores y actuadores. Aussel 20 piezas 2.54mm Breakaway PCB Junta 40 Pin macho.
- Soldador JBC 40 ST y estaño.
- Sensores:
 - 10 Switch Smart Sonoff.
 - 5 Sonoff Pow.
 - 10 Sonoff Pow R2.
 - 5 Sonoff TH16 + 5 Sensores de temperatura SI7021.
 - 4 Sonoff 4CH.
 - 5 Sonoff S20.
- Punto de acceso: TP LINK MR3020.
- Cable NYM-J con Recubrimiento (3 Cables de 1,5 mm², 10 m), Color Gris.
- Cable Ethernet.

Además, el software que se ha utilizado para el diseño del sistema ha sido los siguientes:

- Sistema Operativo Windows 10.
- Navegador Google Chrome e Internet Explorer.
- Balenaetcher.
- Sublime Text.
- Tasmota 5.14.0
- Termite.
- Hassos 3-2.12
- Win32DiskImager.

1.5. ORGANIZACIÓN DE LA MEMORIA

La organización de la memoria queda estructurada en las partes que serán presentadas a continuación de forma resumida:

- **Parte 1. Introducción y objetivos.**
Justificación de la necesidad de un sistema para la monitorización y el control de infraestructuras energéticas de la ETSIT. Presentación de las fases del proyecto y de la organización de la memoria.
- **Parte 2. Estado del arte.**
Estudio de las diferentes tecnologías, estándares y protocolos para la comunicación/transporte y transmisión de datos en el ámbito del internet de las cosas. Estudio de las diferentes plataformas para implementar los dispositivos IoT

- **Parte 3. Desarrollo de la aplicación.**
Especificación de todos los procesos llevados a cabo para diseñar una plataforma con Home Assistant para integrar los dispositivos de la marca Sonoff.
- **Parte 4. Validación del sistema.**
Presentación de las distintas pruebas realizadas al sistema en la fase de laboratorio y en su posterior despliegue en la ETSIT.
- **Parte 5. Evaluación del sistema.**
Exposición de los aspectos positivos y negativos del trabajo, así como una serie de recomendaciones como consecuencia del desarrollo del trabajo.
- **Parte 6. Conclusiones y trabajo futuro.**
Exposición de las conclusiones del trabajo, así como la definición de unas posibles líneas de trabajo futuro.
- **Parte 7. Bibliografía.**
Especificación de la documentación utilizada para la realización del trabajo.
- **Anexo A. Aspectos éticos, económicos, sociales y ambientales.**
Desarrollo de los principales impactos éticos, económico, sociales y ambientales de la memoria.
- **Anexo B. Presupuesto económico.**
Análisis económico del trabajo.
- **Anexo C. Guía para cambiar el firmware a Tasmota en dispositivos Sonoff.**
Desarrollo de una guía paso a paso para poder modificar el firmware de los dispositivos.
- **Anexo D. Guía de instalación de home assistant y servidor mqtt mosquito.**
Desarrollo de una guía para la instalación de Home Assistant y mosquito broker.
- **Anexo E. Guía de acceso a la plataforma home assistant para personal de la ETSIT.**
Desarrollo de un manual de uso de la aplicación desarrollada para el personal de la escuela que será el que utilice la aplicación.

2. ESTADO DEL ARTE

2.1 PROTOCOLOS, ESTÁNDARES Y TECNOLOGÍAS DE IoT

En este apartado realizaré un análisis de los diferentes protocolos, estándares y tecnologías de comunicación que existen entre dispositivos IoT. Para realizar el análisis vamos a dividirlos en dos capas básicas: tecnologías y protocolos de Comunicación/Transporte y de datos.

2.1.1 TECNOLOGÍAS Y PROTOCOLOS DE COMUNICACIÓN/TRANSPORTE

Existen múltiples tecnologías para establecer la comunicación entre dispositivos IoT de forma inalámbrica. Dentro de estas tecnologías se van a analizar las más importantes y voy a separar aquellas que permiten usar redes de corto/medio alcance y largo alcance. Para las redes de corto/medio alcance los más importantes son los siguientes:

- Bluetooth:** muchos dispositivos de “Internet de las cosas” están utilizando redes de corto alcance para conectarse con los usuarios. Debido al auge de dispositivos de conexión inalámbrica de bajo consumo se ha extendido el uso de Bluetooth Low Energy (BLE). La principal diferencia con el Bluetooth convencional es que con menos potencia posee un rango de alcance considerable, muy similar al convencional. Opera en la banda ISM de 2.4 GHz y tiene un tiempo de conexión de pocos milisegundos. Una de las características más importantes de este protocolo es que no se procesan datos, únicamente los recoge y es por esta razón, por la que los dispositivos con una batería pueden estar activos al menos 5 años. En cuanto a la seguridad de BLE, se utiliza un cifrado de 128 bits y autenticación. Utiliza en transmisión de datos el salto de frecuencias adaptables (AFH) con corrección de errores FEC, lo que permite la corrección en transmisión sin necesitar reenvíos.

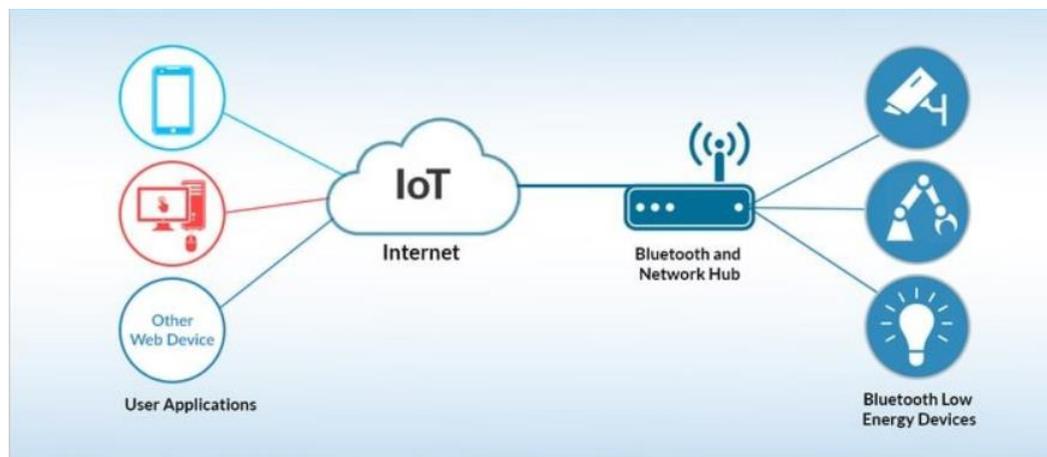


Figura 3 Interconexión de dispositivos con Bluetooth Low Energy [5].

- Wifi:** el uso de la conexión Wifi es el más común dentro de los entornos domésticos a través de una red local o LAN (Local Area Network). El estándar más utilizado es el 802.11n que permite un alcance de hasta 100 metros, aunque la cobertura se limita a 10-35 metros. Ofrece una rápida transferencia de datos y puede procesar grandes cantidades de datos. Utiliza canales en 2,4 GHz o 5 GHz. Una desventaja es que el uso de esta tecnología conlleva un alto consumo de energía para determinadas aplicaciones de IoT. Un aspecto muy importante dentro de esta tecnología es que los

fabricantes han podido integrar estos chips de forma sencilla en los dispositivos y sensores de IoT.

- **RFID:** es la tecnología que utiliza ondas de radio para permitir la comunicación de transmisión de datos entre los sistemas lector y aquellos que tienen una etiqueta especial adherida o grabada. En IoT esta tecnología se utiliza para que los objetos del día a día se comuniquen entre si y el servidor, para informar del estado de los dispositivos. Una de las principales características de los protocolos de conectividad RFID IoT es que no necesitan de energía. Utilizan campos electromagnéticos con ayuda de tecnología inalámbrica, se utiliza en redes de radiofrecuencia de corto alcance, entorno a los 10 cm, pudiendo llegar a los 200 mm.
- **ZigBee:** es una especificación creada para redes inalámbricas basada en el estándar IEEE 802.15.4 que incluye un diseño de software y hardware para WSN (Red de sensores inalámbricos). Destaca porque requiere de alta confiabilidad, bajo coste, escalabilidad y baja velocidad de datos. ZigBee proporciona un consumo y una eficiencia muy bajos. Habilita redes a gran escala, lo que lo convierte en un solución para la industria muy eficaz, pues generalmente se transfieren a pequeñas tasas entre el hogar o un edificio. Opera en la banda de 2.4 GHz y tiene una velocidad de 250 kbps. Tiene un alcance de 10 a 75 metros. A pesar de trabajar en la misma banda que Bluetooth o Wifi no se ve afectado por ellas debido a la baja transmisión de datos. Una de las características más importantes es su seguridad, utiliza el estándar AES128 para cifrar y autenticación de los paquetes.



Figura 4 Tecnología ZigBee [5].

Una vez realizado un pequeño análisis de las características más importantes, en cuanto a seguridad, alcance, banda de frecuencias, tasa binaria y consumo los he recogido de forma simplificada en la siguiente tabla:

	BLUETOOTH	WIFI	RFID IoT	ZIGBEE
Frecuencia de trabajo	2.4 GHz	2.4 GHz	125-134 KHz	2.4GHz 868/915MHz
Alcance	10-30 m	30-100 m	10-20 cm	30-75 m
Velocidad Binaria	1Mbps	100+ Mbps	200 bps /1kbps	20/40/250 Kbps
Consumo de Potencia	Bajo	Alto	Bajo	Bajo
Introducción al mercado	Alto	Alto	Bajo	Bajo
Dispositivos	8	50-200	1	255-65k
Coste	Bajo	Bajo	Alto	Alto

Tabla 1 Comparación Tecnologías para comunicaciones de corto/medio alcance.

En los últimos años se han desarrollado un nuevo tipo de red de medio y largo alcance para el IoT. Después de analizar las tecnologías para comunicaciones de corto alcance, desarrollaré un análisis para las comunicaciones de largo alcance donde las más importantes son las siguientes:

- **LoRa:** utiliza un tipo de modulación en radiofrecuencia llamada Chip Spread Spectrum CSS que permite establecer comunicaciones de largo alcance de entre 10 a 20 km y que es de gran utilidad para dispositivos de IoT. Se trata de una tecnología muy económica y fácil de implementar de bajo consumo. En cuanto a la comunicación, LoRa puede utilizar transmisión bidireccional de forma que un módulo transmisor puede hacer de receptor en cualquier momento. Los mensajes que se transmiten no pueden durar más de 5 segundos en el aire y destaca su resistencia frente a posibles interferencias. Utiliza cifrado AES-128 extremo a extremo. El protocolo que se utiliza en una capa superior es LoRaWAN que determina aspectos en cuanto a capacidad, seguridad y transferencia de datos. Un aspecto importante es que LoRa funciona bien cuando los dispositivos están en movimiento.
- **NB-IoT:** es una tecnología de largo alcance de baja potencia que se utiliza para conectar grandes cantidades de dispositivos IoT y que transmiten pocas cantidades de datos. Se caracteriza por permitir una cobertura de hasta 15 km en espacios abiertos y de 2 km en entornos urbanos. Su factor diferencial es que el espectro entra dentro del rango del LTE por lo que su despliegue está garantizado. Tiene tiempos de respuesta más rápidos que LoRa y mejor calidad de servicios. Un aspecto particular es que esta tecnología es más adecuada para dispositivos que están en una ubicación fija.
- **Sigfox:** es un protocolo desarrollado para el uso de aplicaciones M2M, que envían datos de baja velocidad. La velocidad de bits está comprendida entre 10-1000 bps. La frecuencia de trabajo es de 900 MHz. En zonas rurales tiene un alcance de 30 a 50 km y en las zonas urbanas de 3-10 km. Se caracteriza por utilizar una banda muy estrecha que le permite obtener gran penetración incluso en suelo urbano. La red Sigfox está formado por una estructura de antenas y estaciones bases repartidas por más de 60 países dando una amplia cobertura en estos. En su

seguridad utilizan un método de autenticación entre dispositivos extremo a extremo. Las estaciones bases están conectadas a Sigfox Cloud a través de un enlace punto a punto con una VPN privada y virtualizada en servidores privados.

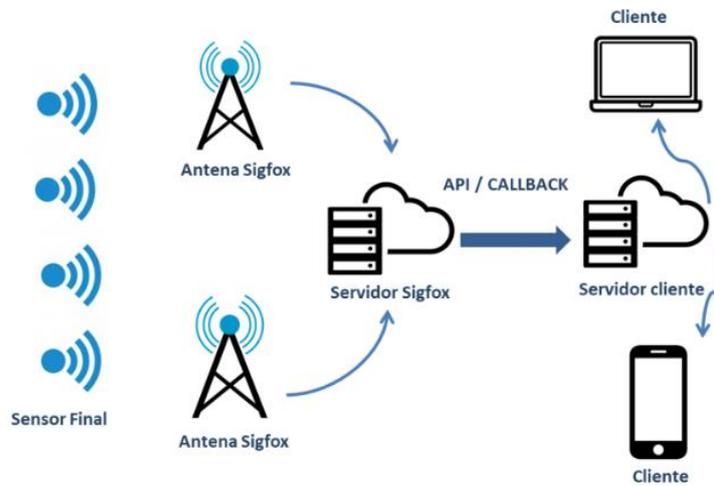


Figura 5 Estructura de la red Sigfox [6].

Finalmente recogemos los datos de forma simplificada en la siguiente tabla:

	LoRa	Sigfox	NB-IoT
Frecuencia de trabajo	433,868 y 915MHz	868,915 MHz	700-800 MHz
Alcance	20 Km	40 Km	10 Km
Velocidad Binaria	50-300 Kbps	100 bps	200 Kbps
Consumo de Potencia	Bajo	Alto	Bajo
Introducción al mercado	Bajo	Alto	Bajo
Coste	Medio	Medio	Alto

Tabla 2 Comparación Tecnologías para comunicaciones de largo alcance

En este proyecto el entorno de aplicación se basa en el control y monitorización de dispositivos eléctricos en interiores de edificios. Por tanto, se utilizan tecnologías de corto/medio alcance y como se detallará en el capítulo 3 concretamente será Wifi.

2.1.2 PROTOCOLOS DE DATOS

En la siguiente parte del trabajo analizaré los diferentes protocolos que existen para la transmisión de datos en entornos de IoT. Los más importantes son los siguientes.

- MQTT:** Transporte de telemetría de Message Queue Server es un protocolo de red de código abierto que se basa en una comunicación M2M. El protocolo se ejecuta sobre TCP/IP. Su modelo de transporte de mensajes consiste en Publicación/Suscripción. Es uno de los protocolos más importantes de IoT debido a su sencillez y ligereza. Por esta razón permite tener conectados muchos dispositivos ya que consume muy poco ancho de banda y apenas recursos en los dispositivos. Esto se traduce en un menor consumo de energía y lo hace ideal para dispositivos de escasa potencia. El protocolo incluye tres componentes: Subscriber, Publisher y Broker. Tanto los subscriptores

como los publicadores son clientes de MQTT y deben estar suscritos a un “topic” que hacen que sea posible establecer la comunicación. La arquitectura de red de MQTT es en estrella y el nodo central hace de servidor o “broker” con capacidad para 1000 clientes. El broker se encarga de gestionar la red y transmitir los mensajes. La comunicación puede ser de uno a uno, o de uno a muchos y se establece en estos tres pasos:

1. Cliente se conecta con el bróker, tanto suscriptores como publicadores. Para ello se subscriben a un “Topic”. Los clientes no se conocen entre ellos, de esta manera unos publican la información y otros la reciben.
2. El cliente publica los mensajes en un tema. Este tema es gestionado por el bróker.
3. El bróker envía los mensajes a todos los clientes que están suscritos a ese tema.

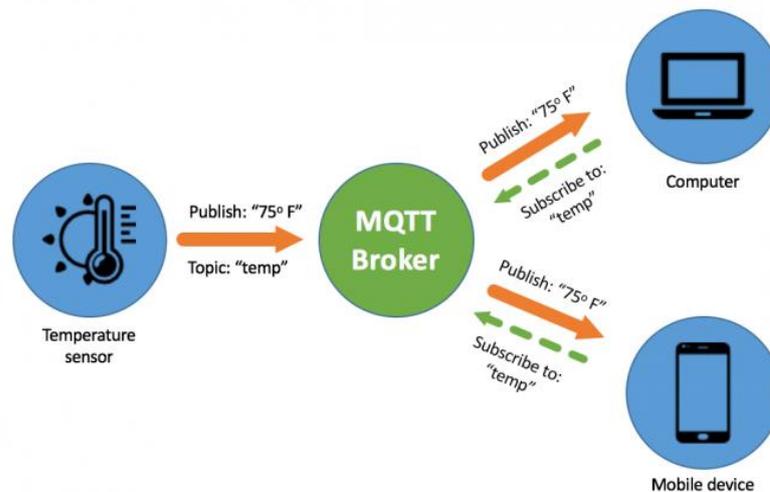


Figura 6 Funcionamiento MQTT Broker [7]

Existen gran cantidad de brokers MQTT con diferentes características, ventajas e inconvenientes. A continuación, haremos un análisis de los más importantes:

- **Mosquitto:** es un servidor de mensajes de código abierto que implementa el protocolo MQTT. Utiliza un método liviano para llevar a cabo la mensajería de utilizando un modelo de Publicación/Subscripción. Es ideal para servidores de baja potencia, muy adecuado para la instalación en Raspberry Pi
- **Mosca:** es un bróker MQTT de código abierto para Node.js. Se puede utilizar de forma independiente o en cualquier proyecto de Node.js
- **HBMQTT:** es un bróker desarrollado en Python que proporciona una API sencilla basada en rutinas y facilita la escritura de aplicaciones muy concurrentes. Su seguridad es SSL sobre TCP y Websocket. Su instalación es a través de un entorno virtual y no directamente en Python.
- **ActiveMQ:** es un bróker de mensajería de código abierto sobre Apache que implementa la especificación de Java Message Service.
- **RabbitMQ:** es un bróker open source que está altamente implementado, es ligero y fácil de instalar. Es compatible con múltiples protocolos de mensajería y se puede ejecutar en diferentes sistemas operativos.
- **EMQTT:** es un agente de mensajes MQTT distribuido, masivamente escalable y altamente extensible escrito en Erlang/OTP.
- **HiveMQ CE:** es un bróker de MQTT de código abierto basado en Java totalmente compatible con MQTT y es ideal para desarrolladores que necesitan instalar un bróker MQTT en aplicaciones Java.

- **MQTTnet:** es una biblioteca .NET de alto rendimiento para comunicaciones basadas en MQTT. [8]

En cuanto a la seguridad de MQTT, se hace a través de los siguientes mecanismos:

-Autenticación usuario/contraseña

-Seguridad SSL/TLS

El puerto estándar es el 1883 para la comunicación no cifrada y el 8883 para la comunicación mediante SSL/TLS. Debido a que el protocolo MQTT pretende ser un protocolo para dispositivos con recursos limitado, el mecanismo de SSL/TLS puede que no sea una opción y lo que se utilice sea una autenticación de usuario y contraseña de texto claro que el cliente envía al servidor como parte de la secuencia de paquetes CONNECT/CONNACK.

- **CoAP:** Advanced Message Queuing Protocol es un protocolo que se utiliza para redes restringidas y con dispositivos que tengan recursos limitados. Está diseñado para traducir fácilmente a HTTP y poder integrarlo en la web de forma simplificada. Debido a la baja sobrecarga y simplicidad son ideales para los dispositivos de Internet de las cosas. Minimiza la complejidad del encabezado, es compatible con URI y soporta el descubrimiento de recursos. Utiliza el protocolo UDP para la implementación de datos.
- **DDS.** Servicio de distribución de datos para sistemas en tiempo real. Es un estándar para comunicaciones M2M de alto rendimiento, ampliable y en tiempo real. Se pueden enviar datos en dispositivos de baja huella y en plataformas en la nube. Su arquitectura describe dos niveles de interfaces:
 - **DCPS:** que tiene el fin de un reparto de la información de forma eficiente.
 - **DLRL:** que permite una integración simple en la capa de aplicaciones.

Las ventajas principales de este estándar son su arquitectura flexible y adaptable, eficiencia, escalabilidad, calidad del servicio y que es independiente de la plataforma en que se implemente.

- **XMPP:** Extensible Messaging and Presence Protocol es un protocolo abierto, escalable y descentralizado basado en XML diseñado para la mensajería instantánea. Es ideal para comunicaciones en tiempo real y los servidores que utiliza pueden estar en la red pública XMPP con seguridad SASL y TLS. Permiten el tráfico TCP utilizado por HTTP.
- **AMQP:** es un protocolo para aplicaciones distribuidas que soporta comunicaciones punto a punto y del tipo publicación/subscripción. Está diseñado para soportar una amplia variedad de aplicaciones y patrones de comunicación. Proporciona comunicación controlada por flujo y garantiza la entrega de mensajes al menos una vez. Autenticación y cifrado SASL y TLS sobre TCP.

Una vez descritos los protocolos más importantes en el ámbito del internet de las cosas para la transmisión de datos, dejamos alguna de las características más importantes en la siguiente tabla:

	Transporte	Modelo	Ámbito de aplicación	Conocimiento del contenido	Datos principales	Seguridad	Prioridad de los datos	Tolerancia a fallos
AMQP	TCP/IP	Intercambio de mensajes punto a punto	D2D D2C C2C	Ninguno	Codificados	TLS	Ninguno	Específica de la implementación
CoAP	UDP/IP	Petición/Respuesta (REST)	D2D	Ninguno	Codificados	DTLS	Ninguno	Descentralizado
DDS	UDP/IP (unicast + mcast) TCP/IP	Publicación/Suscripción Petición/Respuesta	D2D D2C C2C	Enrutamiento basado en el contenido, consultas	Declarados codificados	TLS, DTLS, DDS	Prioridades de transporte	Descentralizado
MQTT	TCP/IP	Publicación/Suscripción	D2C	Ninguno	No definidos	TLS	Ninguno	El nodo central (broker) es el punto único de fallo (SPoF)

Tabla 3 Comparación de protocolos para transmisión de datos [9].

El protocolo más extendido en tecnologías de IoT es MQTT y en este trabajo se utilizará MQTT con la implementación del Broker Mosquitto. Se detallará en el capítulo 3.

2.2 PLATAFORMAS DE DISPOSITIVOS IOT

En el siguiente apartado se van a analizar algunas de las diferentes plataformas que existen actualmente para el control y la monitorización de dispositivos IoT. Para realizar el análisis, se van a analizar las siguientes plataformas.

2.2.1 HOME ASSISTANT

Home assistant es una plataforma de código abierto y gratuita que te permite la automatización del control de dispositivos del hogar y se ejecuta sobre Python 3. Esta plataforma nace como un proyecto destinado a desarrolladores y expertos en tecnologías. Debido al crecimiento de usuarios que disponen de dispositivos IoT, se ha incrementado la preocupación por la seguridad, y ha hecho que esta plataforma se adapte a cualquier tipo de usuario, independientemente de su conocimiento previo, y haya aumentado exponencialmente el número de miembros que la utilizan para diseñar sus soluciones con diferentes tipos de dispositivos IoT.

Home assistant es el software que gestiona el sistema y viene dentro de Hass.io, que es una distribución de Linux creada para funcionar de la mejor manera en un Hardware específico bajo Docker, por lo general una Raspberry Pi e incluye un sistema de Add-Ons que hace que sea más fácil instalar una serie de componentes que se utilizarán para agregar nuestros dispositivos IoT.

Para conseguir estar al alcance de muchos usuarios, tiene un editor de interfaz gráfica de usuario que proporciona todas las características básicas. Dispone de muchas funcionalidades como es el soporte a WebSocket que permite la reacción en tiempo real para que cada sensor se actualice sin necesidad de recargar la página o mostrar las automatizaciones por pantalla y editarlas.

Una de las principales características que tiene es que permite la automatización del hogar sin la nube, lo que da como resultado que no es necesario una conexión a Internet. La plataforma Home assistant se instala sobre una Raspberry Pi, de forma que te permite controlar todos los dispositivos desde tu propio Hardware y no de manera independiente para cada dispositivo y fabricante.

Otra de las principales ventajas es que te permite la integración de cientos de dispositivos y que estos interactúen entre si gracias a la interfaz gráfica que se ha hablado anteriormente o a través del código de configuración. Además, los usuarios pueden actualizar el sistema de forma muy sencilla en la interfaz gráfica. Lo que permite que este actualizado y se corrijen los errores que pueda haber en la plataforma.

Una ventaja muy importante a la hora de utilizar Home assistant es que dispone una página con una gran documentación acerca de todas las características y posibilidades del sistema, con un amplio número de ejemplos del código y con las guías para su preparación. Además, cuenta con un gran foro donde muchos usuarios comparten sus conocimientos y solucionan los problemas de otros usuarios.

Home assistant utiliza el lenguaje YAML para la configuración. Es un lenguaje muy potente que le permite realizar cualquier tipo de integración que no esten disponibles a través de la interfza de usuario. Se basa en colecciones de bloques y asignaciones que contienen clave-valor.

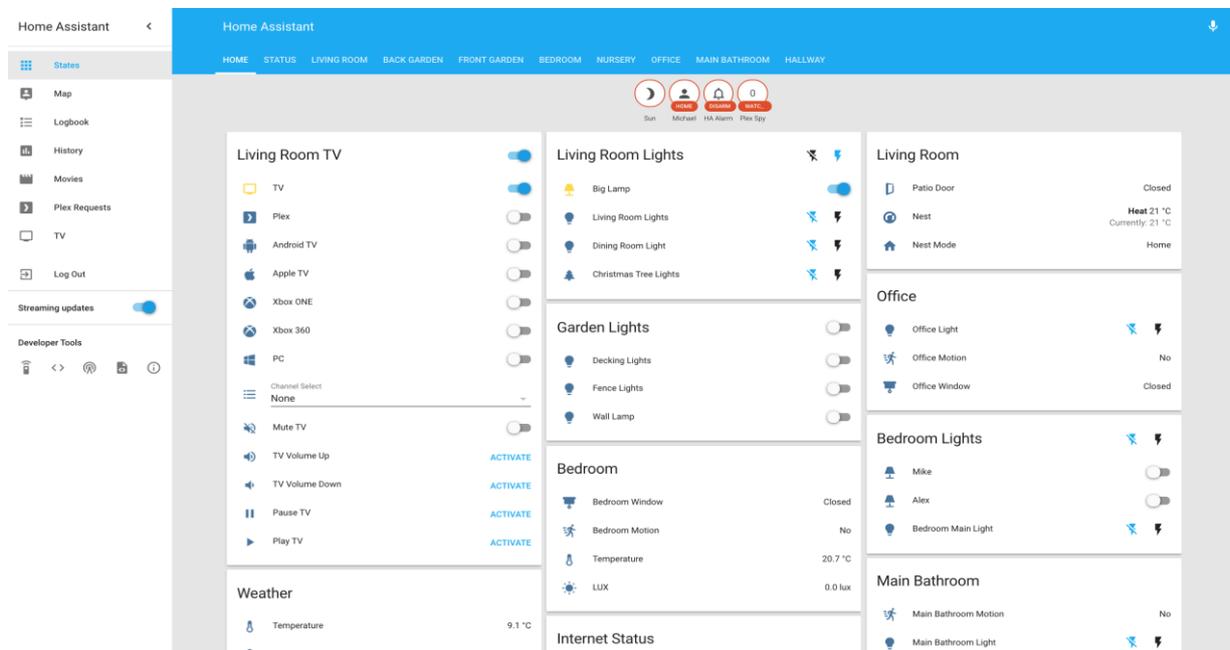


Figura 7 Interfaz de usuario de Home Assistant

2.2.2 OPENHAB

La plataforma openHAB es un software para la automatización del hogar que se basa en código abierto escrito en Java. Este software se caracteriza por ser independiente de proveedores y la tecnología que se quiera utilizar para los dispositivos del hogar. Admite en su arquitectura alrededor de 200 tecnologías y miles de dispositivos.

Una de las grandes ventajas de este software es que el servidor se puede ejecutar en todas las plataformas (Linux, MacOS, Windows, Raspberry Pi...). Al igual que Home Assistant no necesita de conexión a internet y se ejecuta en su propio Hardware. Por tanto, evita un problema de seguridad al mantener almacenados todos los datos de sus dispositivos en su propio Hardware.

La plataforma esta basada en el framework de Eclipse SmartHome y utiliza para crear un entorno de ejecución OSGi que es un software que se extiende a través de add-ons. El lenguaje de programación de openHAB está basado en scripts sobre Xbase. Este lenguaje no es fácil de manejar y puede volverse bastante complejo en algunos casos.

En cuanto a la interfaz de usuario no cubre todos los aspectos y es más compleja que la de Home Assistant ya que se han de utilizar archivos textuales configurables. Es adaptable para varios tamaños de pantallas y se actualiza en vivo.

El proceso de instalación es de entre 20 y 40 minutos y es muy fácil de manejar. La configuración de la interfaz no es compatible con todas las funcionalidades y la desventaja principal es que se hace en dos lugares diferentes, la interfaz y los archivos de configuración.

Al igual que Home Assistant dispone de una comunidad muy activa y que te permite resolver cualquier tipo de problema a la hora de realizar la implementación de cualquier dispositivo en su plataforma. Además, disponen de toda la documentación en su Web.

El ritmo de actualización de la plataforma es bastante inferior al de Home Assistant, pero esto es consecuencia de que el ritmo de aprobación de las actualizaciones es muy lento por su rigurosidad. Esto hace que las nuevas versiones de la plataforma sean muy estables.

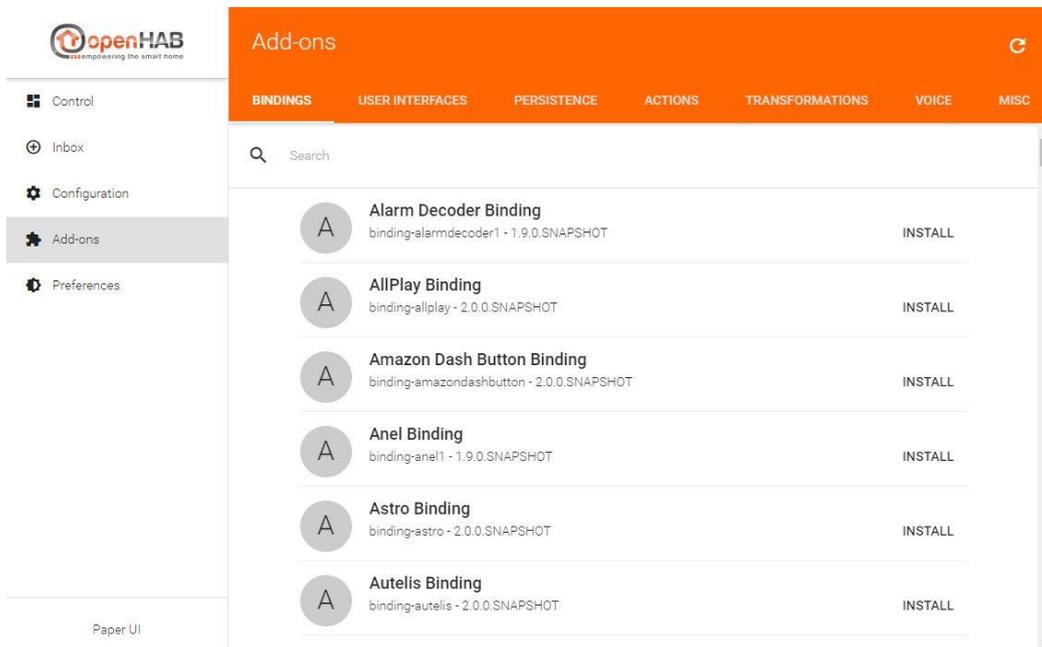


Figura 8 Interfaz de usuario openHab.

2.2.3 DOMOTICZ

Es otra de las grandes plataformas para los sistemas de automatización del hogar. Está diseñado para operar en varios sistemas operativos y ofrece una amplia posibilidad de características. Su configuración se realiza principalmente a través de la interfaz web de tipo HTML 5 que está disponible en todos los navegadores. Te permite el uso de complementos para ampliar las funcionalidades de la plataforma. Su lenguaje de programación es C++ e implementa un servidor web propio.

La instalación es más compleja que las dos anteriores plataformas, ya que se ha de instalar Raspbian sobre una Raspberry Pi e instalarlo desde ahí con conexión a internet. Su configuración se realiza desde la interfaz de usuario, pero esta no es muy intuitiva. Una de las principales desventajas que tiene es que está limitada en el número de dispositivos compatibles y funcionalidades. Con respecto a las dos anteriores se está quedando atrás y no utiliza todos los protocolos actuales del internet de las cosas.

La documentación que se dispone de esta plataforma es desactualizada y es difícil de encontrar soluciones para las configuraciones. Su nivel de desarrollo al tener menos usuarios es bastante menor y no te permite llegar tan lejos como Home Assistant u openHAB.

Esta plataforma no tiene ventajas con respecto a las dos anteriores, se trata de una plataforma muy estable pero desactualizada y que no es compatible con todos los dispositivos y protocolo de internet de las cosas.

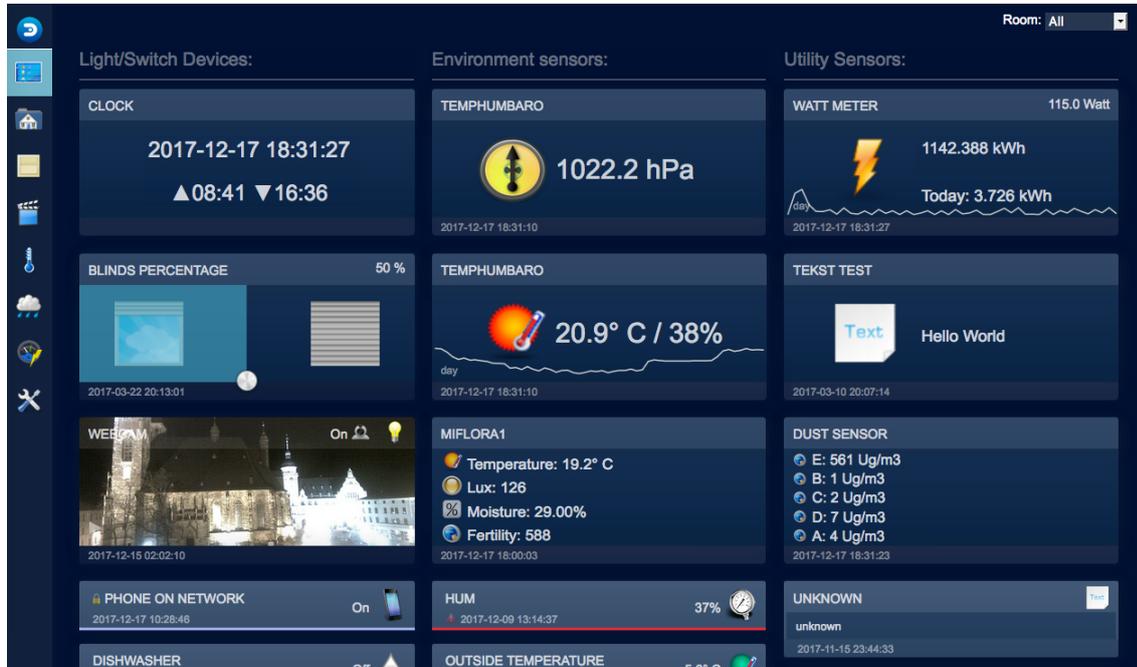


Figura 9 Interfaz de Domoticz.

3. DISEÑO Y DESARROLLO DEL SISTEMA

Una vez realizado el estado del arte sobre las tecnologías existentes y plataformas para dispositivos de IoT voy a realizar el diseño inicial para crear una plataforma de monitorización de dispositivos de IoT. Posteriormente realizar las pruebas necsarias en el laboratorio con los dispositivos integrados en esta plataforma y poder desplegar un piloto.

3.1 ARQUITECTURA DEL SISTEMA DE CONTROL Y MONITORIZACIÓN DE INFRAESTRUCTURAS ENERGÉTICAS

La arquitectura general del sistema tras el análisis realizado en el estado del arte debe de tener las siguientes características:

- Sensores y actuadores alimentados por la red eléctrica y conectados a Internet a través de Wifi.
- Un servidor al que estos sensores enviarán la información obtenida. Además, el servidor puede enviar información de estado a los actuadores.
- Los usuarios se podrán conectar al servidor a través de los proveedores de Internet, ver en tiempo real la información de los sensores y controlar los actuadores.

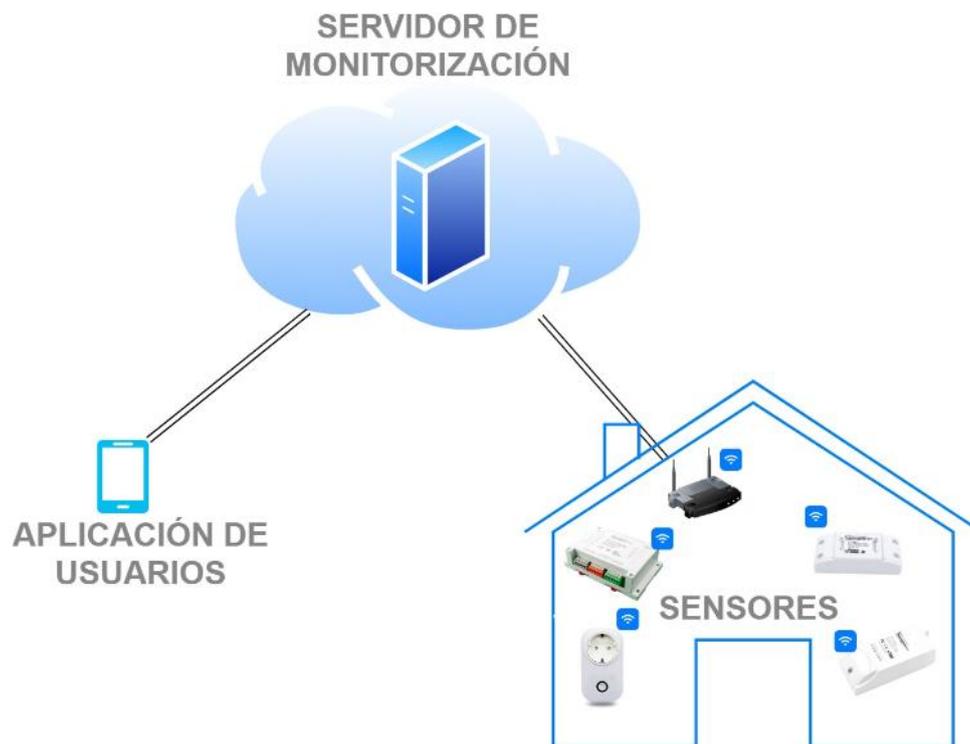


Figura 10 Arquitectura general del sistema.

Esta será la arquitectura de nuestro sistema, pero para poder llevar a cabo esta implementación será necesario los siguientes pasos que se irán detallando en profundidad en los siguientes capítulos de la memoria.

- Cambiaremos el Firmware que traen los dispositivos para poder controlar los dispositivos en nuestro servidor. Para ello los flashearemos e instalaremos Tasmota en ellos.
- Instalaremos Home Assistant en una Raspberry Pi. Dentro de Home Assistant tendremos un broker a través del cual enviaremos los mensajes a nuestros dispositivos.
- Habrá que configurar los dispositivos para que apunten a nuestro servidor de mosquito.

Se desarrollará en dos fases y en dos entornos diferentes. Por un lado, tendremos un entorno de laboratorio donde haremos la primera instalación y veremos los posibles errores de conexión y estabilidad. Conectaremos los equipos a elementos que tengamos en el laboratorio, como son luces, equipos de aire acondicionado, impresoras, calefactores... Dispondremos en un punto de acceso propio, un equipo de TP-LINK conectado a la red del Departamento de Ingeniería de Sistemas Telemáticos.

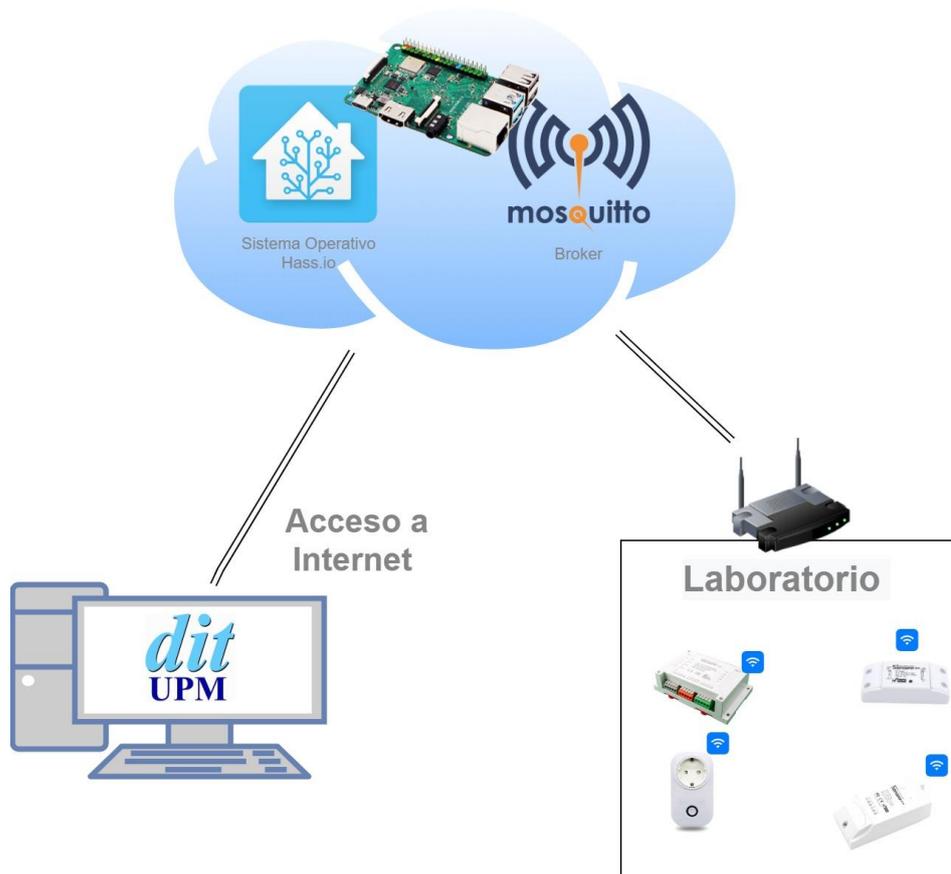


Figura 11 Arquitectura del sistema en laboratorio.

Por otro lado, estará un entorno de producción diferenciado donde tendremos otra Raspberry Pi colocada sobre un Rack en paralelo a la del laboratorio y dará servicio a las luminarias del Edificios A y C. Los equipos se conectarán con una Wifi existente llamada "ServiciosUPM". Se crearán usuarios para que puedan acceder a la plataforma, pero solo a las interfaces de los edificios. En paralelo tendremos un sistema para monitorizar el gasto energético y estudiar las posibles políticas para hacer el sistema más eficiente. Finalmente, al ser un sistema en producción se han creado guías para los vigilantes y conserjes de la ETSIT.

El diseño final en la ETSIT del sistema quedará de la siguiente forma.

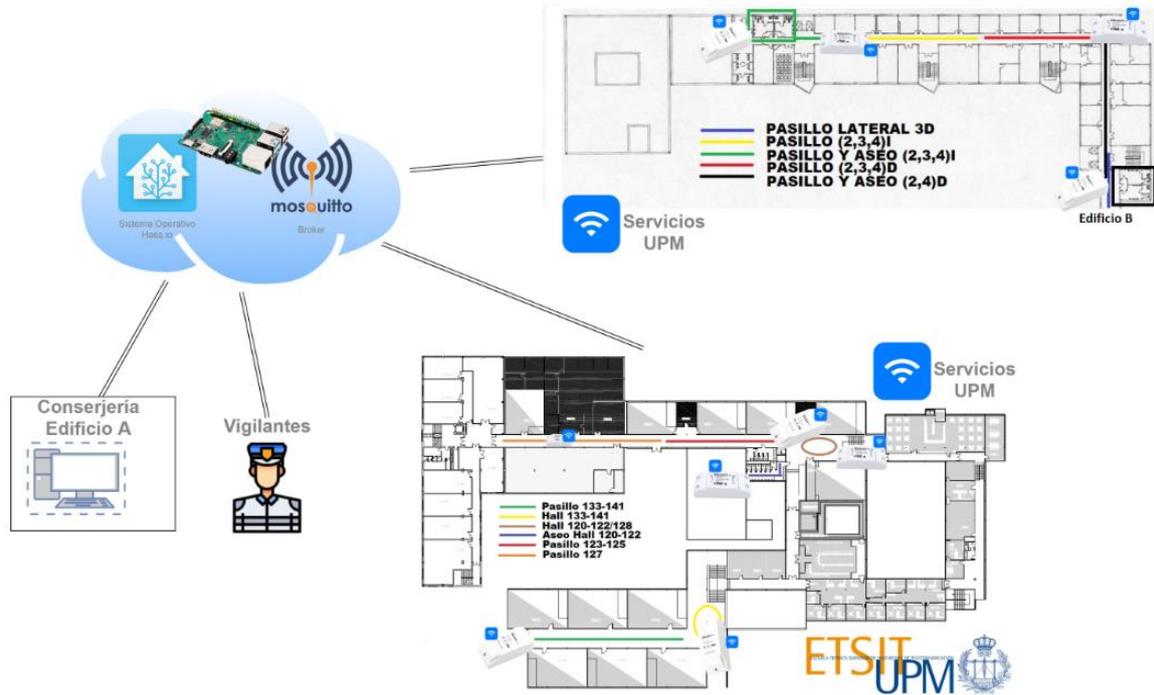


Figura 12 Arquitectura final del sistema de monitorización en la ETSIT.

3.2 DISPOSITIVOS

Los dispositivos que se van a utilizar son de la marca Sonoff. Actualmente son de los más vendidos del mercado y son de bajo coste. Estos dispositivos están basados en el chip ESP8266 y te permiten cambiar el firmware por uno alternativo para poderlo integrar en tu propia plataforma y así no utilizar eWeLink que es la aplicación que hay que instalar por defecto a la hora de comprarlos.

Como se habló en la introducción el objetivo principal será poder monitorizar y controlar luminarias. A pesar de que los dispositivos elegidos sean los más ideales para el encendido y apagado de las luminarias, pueden ser utilizados para muchos otros ámbitos. A continuación se detallan los dispositivos elegidos, con sus características principales:

- **Sonoff Wifi Smart Switch:** se trata de un interruptor para encendido y apagado de dispositivos eléctricos. Tiene como cableado de entrada, el neutro y la fase, que ha de ser conectado al dispositivo de forma manual, se detallará más adelante como hacerlo en el apartado de Instalación de los dispositivos. Su rango de voltaje va de 90 a 250 V en corriente alterna de 50/60 Hz. Soporta una corriente máxima de 10 A y una potencia máxima de 2200 W. Para el enlace de comunicaciones utiliza el estándar inalámbrico de 802.11 b/g/n y utiliza seguridad de WPA-PSK/WPA2-PSK. Además, dispone de un interruptor para poder realizar el switch de forma manual en caso de que se pierda la conexión remota con el dispositivo. Para alimentar el dispositivo su entrada está conectada a la red eléctrica y en la salida a todo aquello que queramos controlar de forma remota, como son luces, enchufes, electrodomésticos, etc. Una vez instalado te permite ver en tiempo real su estado, es decir, si está en “ON” deja pasar la corriente, o en estado “OFF” el switch no deja pasar corriente.

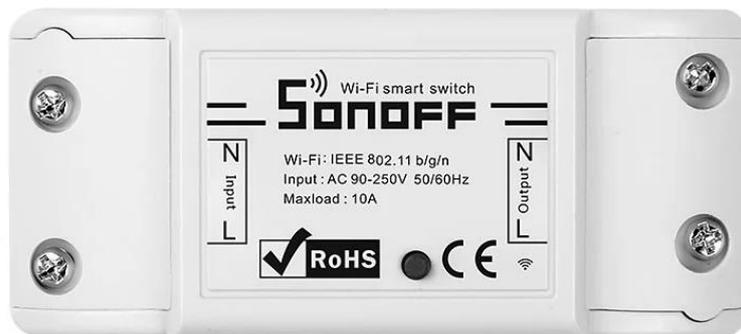


Figura 13 Sonoff Wifi Smart Switch.

- **Sonoff Pow/Pow R2:** este interruptor Wifi de 15 A te permite monitorear la energía en tiempo real, además de encender y apagar los dispositivos como hace el Sonoff Switch. Una de las características principales es la capacidad de ofrecer un historial de consumo. Este te ofrece la posibilidad de ver en tiempo real la potencia, corriente y tensión en la que está trabajando aquello que conectemos al dispositivo. De esta manera se puede saber el consumo de energía diaria y el coste total de esa energía. Además de ofrecerte los datos históricos completos del día, mes y total desde que se pone en producción. La ventaja que te ofrece esto es además de saber el consumo, poder fijar un umbral para protección frente a sobrecargas, es decir, si el dispositivo llega a un parámetro de tensión, corriente o potencia hacer que se apague automáticamente. La potencia máxima admisible de este interruptor es de 3500 W. Los parámetros de seguridad y comunicación son los mismos que el del Sonoff Switch.

y también dispone de un interruptor en caso de que ocurra algún problema de conexión y se tenga que realizar el encendido o apagado de forma manual. Actualmente se disponen de dos versiones de este interruptor, Sonoff Pow y Sonoff Pow R2. La diferencia entre estas dos versiones son a nivel de firmware que trae instalado los dispositivos por defecto. En este trabajo ese aspecto no es relevante, ya que se va a realizar un cambio de firmware y de esta forma disponer de las mismas funcionalidades en ambos dispositivos que se detallara más adelante en la memoria. Otro aspecto a diferenciar con el dispositivo descrito en el punto anterior es que para la alimentación del dispositivo, además del neutro y la fase, tiene línea de tierra pudiendo así conectar equipos de más potencia y con protección frente a posibles derivaciones en los dispositivos. Una vez instalado al igual que el anterior te permite saber el estado del dispositivo ON/OFF y los parámetros descritos al inicio.



Figura 14 Sonoff Pow R2.

- **Sonoff TH16/SI7021:** este dispositivo además de ser un interruptor de encendido y apagado como los anteriores, te permite conectar un sensor para monitorizar la temperatura y humedad en tiempo real. Así se pueden establecer reglas para el encendido y apagado a cierta temperatura, lo cual es muy útil a la hora de conectar equipos de aire acondicionado/calefacción que quieres que se activen o desactiven a una determinada temperatura. Su corriente de trabajo es de hasta 15 A y soporta una potencia máxima de 3500W. Al igual que los anteriores dispone de un interruptor manual para pérdidas de conexión de red. Así como los mismos estándares de comunicación y seguridad. En todo momento se verá el estado del dispositivo, así como la humedad y la temperatura que proporciona el sensor.



Figura 15 Sonoff TH16 y sensor Si702.

- **Sonoff 4CH Wifi:** este interruptor es ideal para aplicaciones industriales, su funcionamiento es a partir de una línea de corriente de entrada, puede controlar el encendido y apagado de cuatro líneas de salida. Este dispositivo se puede integrar fácilmente en un cuadro eléctrico ya que trae los enganches necesarios para ello. Para explicar mejor el funcionamiento, voy a exponer un ejemplo práctico. Se instalaría el interruptor en un cuadro eléctrico con una línea de entrada, las salidas son líneas de corriente conectadas a equipos de aire acondicionado que van a cuatro habitaciones diferentes, de esta forma con un único equipo se puede controlar los equipos de estas habitaciones de forma independiente. La gran ventaja es que, si con los dispositivos descritos anteriormente se necesitaba un equipo por cada habitación, con este interruptor puedes controlar el encendido y apagado de las tres habitaciones. Además, la diferencia que tiene con los anteriores con relación a problemas de conexión es que dispone de un interruptor manual para cada línea que hay conectada pudiendo encender y apagar cada una de ellas de forma manual. Este dispositivo permite soportar hasta 10 A por línea con un máximo de potencia de cada una de ellas de 2200W. El total de las cuatro líneas no puede superar los 16 A y una potencia máxima de 3500W.



Figura 16 Sonoff 4CH.

- **Sonoff S20/Sonoff S26:** se trata de dos enchufes inteligentes que son fácil de conectar a cualquier dispositivo que dispongo de conexión a la red de eléctrica con clavijas del estándar UE (Tipo F/E). Son muy útiles ya que la instalación de estos dispositivos es mucho más sencilla que todos los dispositivos descritos anteriormente, no requieren de cableado eléctrico si no que se conectan directamente a cualquier enchufe de pared para posteriormente conectar sobre este cualquier dispositivo y poder controlarlo remotamente. En cuanto a la diferencia entre los dos dispositivos del enunciado, no hay ninguna en ningún aspecto funcional del dispositivo. Ambos interruptores, tanto el S20 como el S26, soportan dispositivos conectados de hasta 2200W de potencia con corriente de 10A. Sin embargo la diferencia principal es a nivel físico del dispositivo, el enchufe S20 es mucho más grande que el S26 el cual es más pequeño y fácil de usar en enchufes que comparten espacio con otros transformadores. Esta diferencia influye a la hora de poder cambiar el firmware del dispositivo, pues en el S20 trae los agujeros necesarios en su placa base para soldar los pines y conectarse con él desde un ordenador para poder manipularlo, mientras que en el S26 el espacio

en a placa es más pequeño y no dispone de los agujeros necesarios para poder conectarte con el dispositivo. Todo este proceso será explicado en detalle más adelante. Otro aspecto relevante de ambos dispositivos es que traen un botón manual con un led de color verde para ver en el estado que están y cambiarlo en caso de que haya problemas de conectividad en la red Wifi. Estos dispositivos al igual que los otros te van a permitir ver el estado de cualquier dispositivo conectados a ellos en todo momento, y en relación con su seguridad y comunicación son exactamente iguales todos los aspectos.



Figura 17 Sonoff S20/S26.

- **Sonoff DIY Smart Switch RFR3:** se trata de la nueva generación de Sonoff Wifi Smart Switch, la diferencia principal es la nueva funcionalidad del modo DIY que permite el control total del dispositivo desde la LAN sin necesidad de tener conexión a Internet. Para ello trae unos pines en la placa base sobre los que habría que colocar un jumper en los pines que incluye OTA SW. Otro aspecto importante que lo diferencia es el tamaño del dispositivo ya que es mucho más pequeño que el Switch Basic. Cabe destacar la posibilidad de ser controlado por radiofrecuencia, para ello con un mando como el Wireless 4 Buttons Push Cover Remote Controller podemos controlar el dispositivo en caso de no tenerlo conectado a la red Wifi o de que hubiera cualquier tipo de problema de conexión. En resumen, es el mismo interruptor que se ha descrito al inicio de este apartado, realiza la acción de encendido y apagado sobre cualquier dispositivo conectado a él, pero trae la posibilidad con su firmware instalado por defecto de fábrica, de poder trabajar en LAN sin necesidad de conexión a Internet del dispositivo. Para ello se ha de instalar ewelink que es la aplicación que proporciona el fabricante a los usuarios para poder trabajar con los dispositivos de la marca Sonoff. Además, tiene la posibilidad de ser controlado por un mando de RF sin necesidad de utilizar la aplicación. Otro punto que se ha mejorado es el aspecto de los interruptores siendo ahora más pequeños y con un diseño más elegante. Al igual que el Switch Basic trae un interruptor manual y un led para indicar el estado en el que se encuentra. Finalmente, su corriente máxima admisible es de 10A con una potencia máxima de 2200W.



Figura 18 Sonoff DIY Smart Switch.

- **Sonoff DIY Mini Smart Switch:** es una de las versiones de interruptores más pequeñas de los dispositivos de Sonoff, esto se debe a que esta pensado para introducir en un interruptor de pared de uso tradicional. Lo que esto permite es saber el estado en que se encuentra en todo momento las luces y en caso de que no se hayan apagado hacerlo remotamente. Una opción que incluye es que trae un conector para poder sincronizar dos interruptores conectados a un dispositivo, es decir, en habitaciones grandes o espacios donde haya varios interruptores para el encendido y apagado se puede conectar a un único Sonoff Mini. Otra opción es que tiene un cable que sirve como el adaptador para conectar el interruptor basculante. Tiene una corriente máxima de 10A y una potencia máxima de 2200W. [10]



Figura 19 Sonoff Mini DIY Smart Switch.

3.3 FIRMWARE Y FLASHEO DE DISPOSITIVOS.

Todos estos dispositivos que se han descrito anteriormente traen una placa universal de bajo coste basada en el chip ESP8266. Este procesador de placas IoT ha integrado micro-MCU de 32 bits Tensilica L106 de bajo consumo de energía. Tiene 1 Mb de memoria que te permite escribir código sobre él. Todo esto ha sido desarrollado por la empresa china Itead que vende todo tipo de productos enfocados en los hogares inteligentes, domotica y automatización del hogar. Por un lado, nos ofrecen estos productos finales con una plataforma en la nube para poder darles soporte. Esta plataforma es la que hemos mencionado anteriormente llamada ewelink donde se pueden integrar fácilmente todos los dispositivos que se han descrito en el punto anterior y poder controlarlos desde la nube que nos ofrecen de forma gratuita. Sin embargo, esto puede suponer un problema para muchos usuarios, ya que estas enviando información a un servidor que ponen a disposición ellos pero el usuario final no tiene ningún control sobre este. Los principales problemas que supone esto son los siguientes:

- Se envía información confidencial acerca de parametros de tu conexión. Como pueden ser SSID y Password. Cada vez que se conecta un dispositivo, a través de la aplicación móvil se actualizan estos parametros en el dispositivo.
- Se envía información acerca de tus movimientos, consumo o incluso camaras de videovigilancia. Esta información si el servidor que ellos tienen sufre alguna vulnerabilidad puede ser robada facilmente.
- Si en algún momento ocurre algún fallo de conexión con su servidor no se tiene la capacidad de reentrarlo o solucionar el problema hasta que ellos lo hagan. Sin saber el tiempo en que lo pueden solucionar.
- El rendimiento que se le puede sacar a los dispositivos está limitado a las funcionalidades que ellos te proporcionan en su aplicación.
- Solo te permiten sincronizar los dispositivos del mismo fabricante, es decir, solo podras tener en la aplicación los dispositivos de la marca Sonoff

Sin embargo, el punto fuerte de este fabricante es que da al usuario la opción de modificar el firmware para que podamos cargar cualquier tipo de programa en el dispositivo. Existen muchos firmwares, tanto como dispositivos pero de los que se dispone para el ESP8266 son:

ESPEasy

Fue el primer firmware alternativo, nació en 2015. Fue diseñado por Let's Control IT y su código está disponible en GitHub. Marcó el inicio de los firmwares alternativos y demostró la potencia y el control que era capaz de alcanzarse con los dispositivos que hasta estaban limitados a las funcionalidades que daba el fabricante.

Tasmota

Su primera versión fue la 1.0.5 en marzo de 2016 de la mano de Theo Arends y su código está disponible en GitHub. Integra en una misma imagen todos los dispositivos y es seleccionable una vez accedes al dispositivo sobre el que está. Esto ocasiona un mayor código, al tenerlo todo disponible. El que lo tenga todo integrado lo hace más versátil, ya que puedes especificarles que sensores tienes conectados en cada GPIO.

ESPurna

Es el más joven de los tres. Al igual que los otros su código también está disponible en GitHub. Su creador es Xose Perez y es asiduo en eventos sobre IOT. Existe una imagen personalizada para cada dispositivo, con lo que es flashear los dispositivos y empezar a usarlo [11].

A continuación, se va a explicar a rasgos generales como cambiar el firmware de un dispositivo Sonoff de fábrica a Tasmota. He realizado una guía completa paso a paso que se puede ver en al final de este documento en un anexo. Para poder cambiar el firmware de un dispositivo se necesitan los siguientes elementos físicos antes de empezar:

- **FTDI FT232RL USB:** se trata de una controladora que nos permite la transferencia de datos en serie de un solo chip. De esta forma nos podemos comunicar con los dispositivos Sonoff que no poseen una interfaz USB-Serial integrada en el dispositivo. Puede funcionar con potencia de 3.3V o 5V, en nuestro caso debemos colocar el jumper sobre el FTDI para trabajar en 3.3V
- **Soldador + Estaño + Pin:** se necesitarán soldar a la placa base de los Sonoff los pines para poder conectarlos a la controladora y mandarle la información. Se necesitan 4 pines macho-macho para cada placa base que modifiquemos.

- Cables para conexiones tipo Dupont hembra-hembra, para la conexión entre el FTDI y el dispositivo Sonoff que queremos modificar.
- Cable USB tipo B mini para conectar el FTDI con el ordenador desde donde le mandaremos los datos a los dispositivos.

Una vez que tenemos preparados todos los elementos físicos para poder modificar el firmware comenzamos con el proceso para poder realizar este cambio. De todos los Sonoff que hemos descrito anteriormente se pueden modificar más fácilmente al traer los agujeros para los pines en su placa base los siguientes dispositivos:

- Sonoff Smart Switch.
- Sonoff POW/POW R2.
- Sonoff 4CH.
- Sonoff S20.

El resto de dispositivos también se pueden modificar pero a la hora de realizar la soldadura hay que tener más experiencia realizando este tipo de soldaduras, aún así en el anexo estará indicado la forma de hacerlo.

Así pues, la manera de proceder para los que hemos mencionado que tiene la ranura correspondiente es la siguiente.

1) Soldar los pines a la placa base del dispositivo. Los Sonoff traen agujeros para soldar los pines sobre la placa base y hay que identificar el orificio correspondiente a 3.3V, RX, TX y GND. Dependiendo del dispositivo puede aparecer indicado o no. También destacar que es diferente para cada tipo de dispositivo la posición donde se encuentran. Lo más importante es que la conexión entre el FTDI y el Sonoff debe de quedar de la siguiente forma:

Pin FTDI	Pin ESP8266
3,3v	3,3 v
Tx	Rx
Rx	Tx
Gnd	Gnd

Tabla 4 Interconexión entre FTDI y Sonoff.

En la siguiente imagen se puede ver como quedaría la interconexión entre la placa de un Sonoff Smart Switch y el FTDI. Lo más importante es conectar el TX del FTDI con el RX del Sonoff y el RX del FTDI con el TX del Sonoff. Además se puede apreciar como hay que colocar el jumper en el FTDI para trabajar a 3.3V.

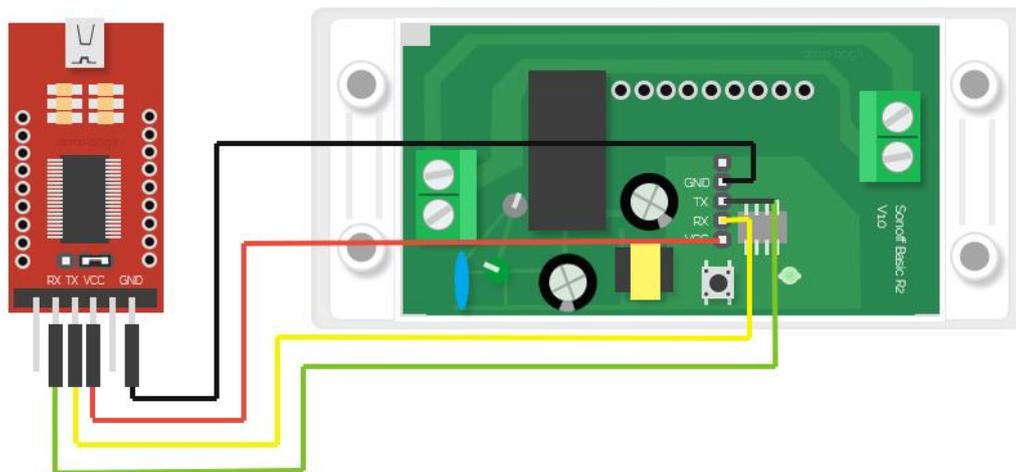


Figura 20 Interconexión entre FTDI y Sonoff Smart Switch.

2) Conectar el dispositivo con el ordenador. Es muy importante que antes de conectar el USB al ordenador hay que mantener presionado el GPIO0 que es el switch que trae el sonoff para realizar el cambio de estado ON/OFF de forma manual en caso de que ocurra algun problema de conexión con él.

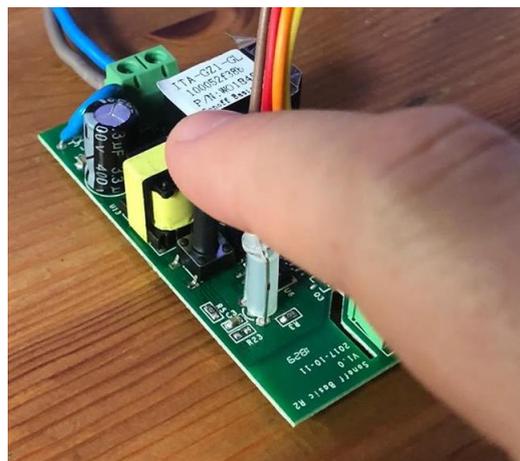


Figura 21 Placa Sonoff Switch antes de ser conectada.

3) Realizar el flasheo. Para realizar el cambio de firmware necesitaremos irnos a la página de Github de Tamota [12] y descargarnos la última versión del archivo binario de Tasmota.bin que queremos mandar a nuestro Sonoff. Para realizar el Flasheo necesitaremos utilizar una herramienta software para proceder de la forma más sencilla. Existen varias y son muy intuitivas, yo he utilizado las dos siguientes y el resultado ha sido satisfactorio en ambos casos, solo es necesario instalar una de las dos.

- Tasmota PyFlasher: herramienta de flasheo diseñada para tasmota. Es muy sencilla de utilizar e instalar a través del ejecutable .exe que encontraras en la página de GitHub. Es la que recomiendo para realizar el flasheo por primera vez [13]
- Esptool.py: esta herramienta al igual que la anterior es muy sencilla de utilizar, pero su instalación es algo más compleja ya que requiere tener instalado previamente una versión de Python 2.7 o 3.4 [14]

Una vez que hemos instalado alguna de estas herramientas, hay que realizar el flasheo. Para ello hay que seleccionar el puerto serie donde hemos conectado previamente el Sonoff presionando el botón de Switch, seleccionar la imagen que hemos descargado de la última versión de Tasmota y realizar el Flash hasta que se completa el 100%. Durante este proceso el software lo que hace es eliminar el firmware antiguo y guardar la versión de Tasmota que hemos descargado.

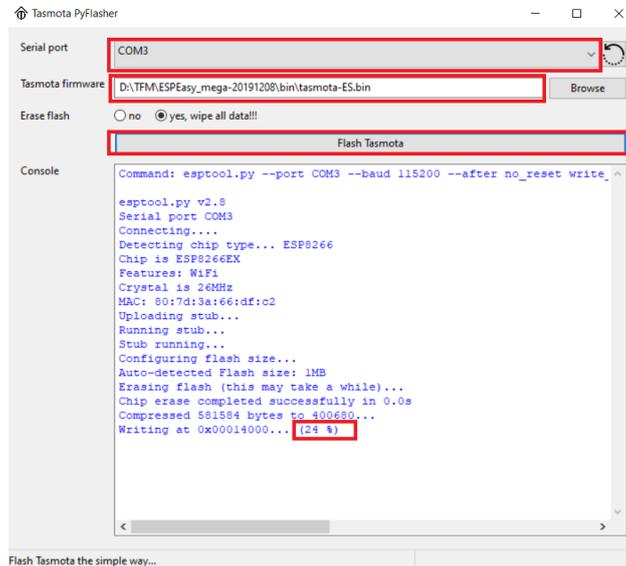


Figura 22 Proceso de Flasheo con PyFlasher.

Una vez que se ha completado hay que desconectar el dispositivo y volverlo a conectar sin presionar nada.

4) Introducir los parámetros de la red Wifi que queremos conectar al dispositivo. Para ello hay dos formas de hacerlo.

- Tasmota ofrece un punto de acceso inalámbrico de configuración de Wifi una vez que se ha flasheado y conectado al ordenador o fuente de alimentación. Por tanto, una vez conectado a la alimentación hay que buscar un AP llamado tasmota-XXXX y conectarse.

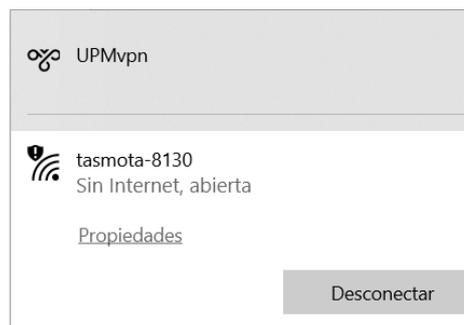


Figura 23 Punto de acceso creado por Sonoff después del flasheo.

Después de que se haya conectado al AP lo normal es que nos redirija a una página para poder configurar los parámetros que deseamos. En caso de no se haya redirigido podemos abrir en el navegador esta página accediendo a la dirección <http://192.168.4.1>. En esta página se puede hacer que Tasmota busque redes disponibles o ingresar manualmente los parámetros de SSID y Password de la red Wifi. También es recomendable introducir otra red Wifi alternativa para que se conecte en caso de problemas de conexión con la que se ha asignado en primer lugar.

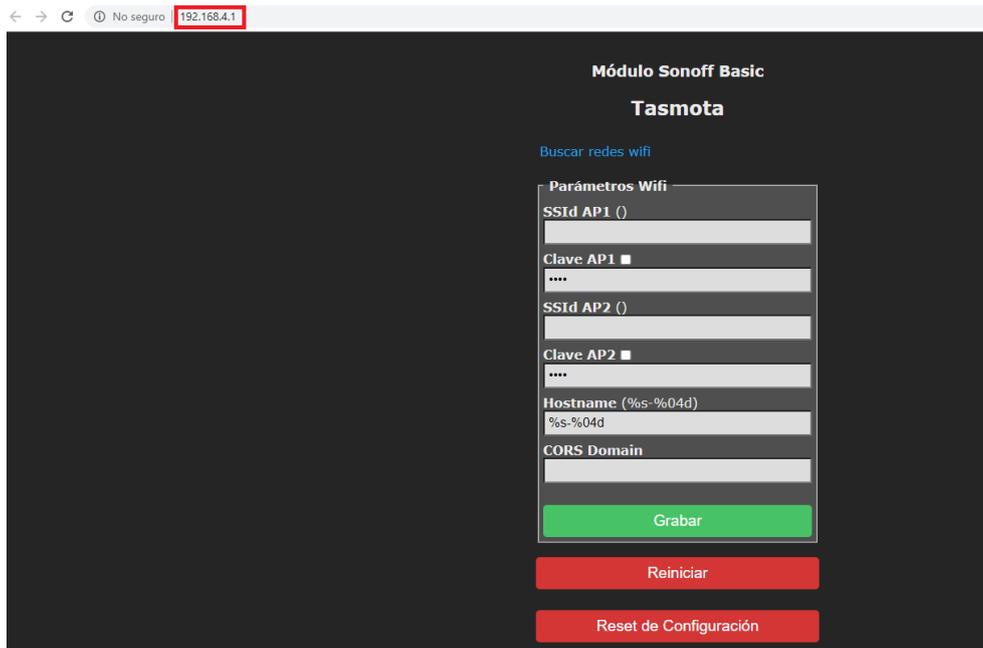


Figura 24 Página de configuración Wifi de Tasmota.

- La otra opción y la más rápida es la de utilizar un Terminal Serie. En mi caso he utilizado Termite [15], desde su página Web te puedes descargar el ejecutable y es muy fácil de instalar. Con esta herramienta software podremos conectarnos al dispositivo a través del USB que utilizamos para realizar el Flasheo utilizando comandos para enviarle los parámetros de la red wifi de la siguiente forma:

```
ssid1 <yourssid>; password1 <your_password>; ssid2 <your_ssid2>;  
password2 <your_password>
```

El dispositivo se reiniciará y se conectará a la red Wifi que le hemos asignado. Aunque el otro método es mucho más intuitivo por la interfaz grafica que nos ofrece, una de las ventajas que tenemos al utilizar Termite es que en su consola podremos ver la dirección IP que se le ha asignado al módulo Sonoff. Con el otro método habría que meterse en el Router y ver la dirección IP que se le ha asignado al dispositivo. Es importante saber esta dirección IP que se le ha asignado ya que para poder modificar cualquier parámetro de Tasmota y tener más control sobre el dispositivo hay que acceder de la forma <http://X.X.X.X> donde tendremos que introducir la dirección IP asignada.

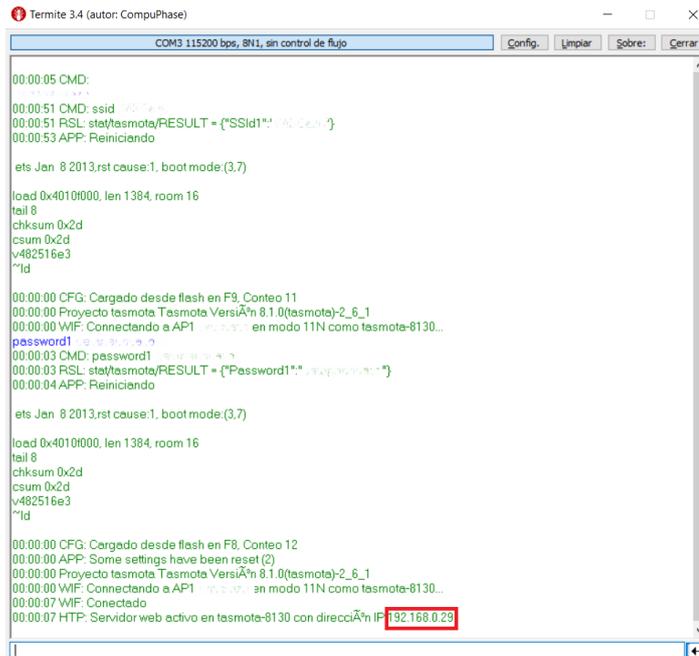


Figura 25 Consola Termite.

5) Por último, hay que seleccionar el tipo de módulo Sonoff que tenemos. Para ello en la página de inicio de Tasmota hay que acceder a **Configuración- Configuración del Módulo** y seleccionar el dispositivo que estamos configurando.

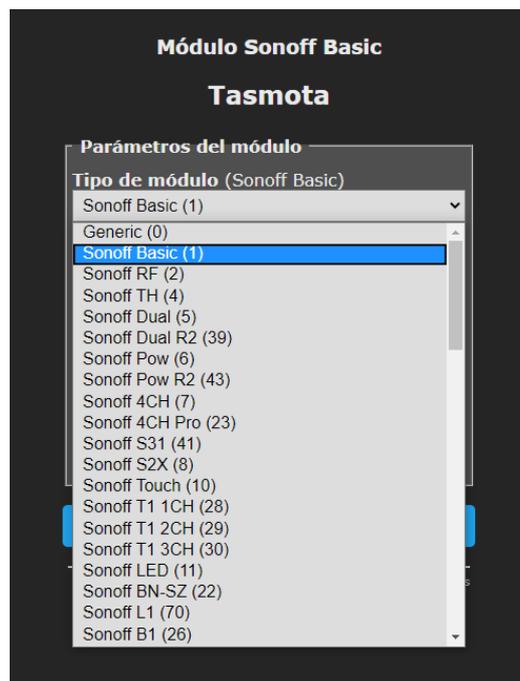


Figura 26 Configuración del módulo Sonoff.

3.4 INSTALACIÓN DE HOME ASSISTANT

En el estado del arte se ha realizado una comparativa de las plataformas actuales que existen en el mercado de los dispositivos IoT. Para este trabajo se va a utilizar Home Assistant ya que tiene gran flexibilidad para alojar cualquier tipo dispositivo, no solo de la marca Sonoff que son los que se van a utilizar para el sistema que se está diseñando, prácticamente cualquier tipo de dispositivo que hay en el mercado. Además, es una plataforma que está en continuo

cambio y con una gran comunidad para cualquier tipo de ayuda que se necesite. Otro aspecto que es de los más importante es que posee una de las interfaces gráficas más fáciles de utilizar y que te permite añadir los dispositivos sin necesidad de tener muchos conocimientos acerca del lenguaje de programación que se utiliza en la plataforma. Home assistant usa el formato YAML, al comienzo de su desarrollo significaba Yet Another Markup Language y ahora es un acrónimo recursivo que significa YAML Ain't Markup Language. Se trata de un lenguaje inspirado en Python, tiene una sintaxis relativamente sencilla para que sea fácilmente mapeable a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel. Utiliza espacios en blanco y no tabulaciones para delimitar las secciones de código. Este lenguaje lo emplearemos más adelante para ver como integrar los dispositivos a la plataforma, crear automatizaciones o personalizar la interfaz de usuario, aunque como ya se ha dicho no hay que tener un conocimiento muy avanzado para poder realizar integraciones sencillas.

A continuación, se va a explicar cómo instalar Home Assistant, que será nuestro servidor en el sistema que se está diseñando, para albergar a los dispositivos Sonoff. Al igual que en el punto anterior se ha realizado una guía completa paso a paso en los anexos. En esta parte de la memoria se comentarán los aspectos más relevantes a la hora de la instalación de Home Assistant y mosquito broker.

Antes de comenzar la instalación se han de tener preparados los siguiente elementos Hardware:

- Una Raspberry Pi 4 Model B + Fuente de alimentación + Carcasa será donde arrancaremos nuestro servidor Home Assistant.
- Una tarjeta de memoria Micro SD Card. Se recomienda una tarjeta Class 2 de al menos 32 GB donde guardaremos la imagen de Home assistant. Se recomienda tener dos tarjetas de memoria, ya que una de las ventajas que tenemos al usar una Raspberry Pi es que podemos cambiar de SO fácilmente cambiando la Micro SD, por eso es recomendable tener dos tarjetas y en una de ellas tener una copia de seguridad que se realice con una frecuencia de al menos 10 días. Durante un sobrecalentamiento en el desarrollo de este trabajo, la tarjeta se vio afectada y tuvimos que cambiar la tarjeta de memoria con la imagen de una copia de seguridad.
- Un cable ethernet.

Una vez que tenemos todos los elementos preparados los paso que hay que dar son los que se detallan a continuación:

1) Hay que descargar la imagen correspondiente con la versión de Raspberry Pi que tenemos de la página oficial de Home Assistant [16]. Una vez descargada debemos grabar esta imagen en la Tarjeta Micro SD con el software BalenaEtcher [17] que formateará la memoria y grabará la imagen que hará arrancar Home Assistant.

2) Introducimos la SD dentro de la Raspberry Pi. Procedemos a conectar el cable de alimentación y un cable Ethernet. Una vez iniciada, pasado unos segundos se le ha asignado una dirección IP por DHCP automáticamente. Para averiguar cuál es la dirección asignada, se puede acceder al router y buscar "hassio". En mi caso para realizar la instalación he utilizado un router de Movistar al que se puede acceder desde la dirección <http://192.168.1.1/> y ahí puedo visualizar que dirección IP se le ha asignado a la Raspberry Pi. Otra opción es a través de la dirección MAC asignarle una dirección IP estática a la Raspberry.

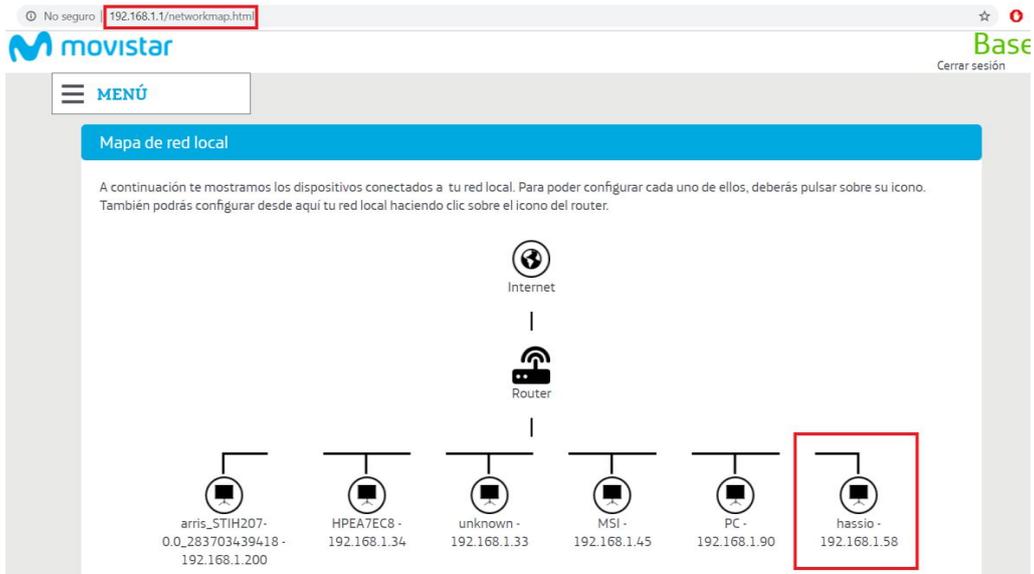


Figura 27 Página de un Router Movistar para ver las direcciones IP.

3) Una vez que sabemos la dirección IP del dispositivo podremos acceder a él de la siguiente forma `http://Z.Z.Z.Z:8123`. Habría que sustituir Z.Z.Z.Z por la que se haya asignado en la red que se quiera conectar. Tardará 20 minutos aproximadamente en descargar la última versión de Hassio y pasado ese tiempo ya tendremos nuestro Home Assistant listo para añadir todas las funcionalidades y dispositivos que queramos añadir. Habrá que configurar la cuenta y crear un usuario con contraseña. Después se nos pedirá información acerca de la ubicación donde vamos a colocar los dispositivos, tendremos que poner la ubicación exacta en un mapa y la altitud de este punto. Esto es realmente importante ya que se realizarán automatizaciones en función de la elevación del sol. Home Assistant utilizará esta información para saber exactamente las horas de sol que hay en el lugar de implementación o el tiempo meteorológico de forma de que si llueve se puedan crear políticas al haber menos luz, humedad o cualquier cosa que dependa de ello. Después de haber introducido todos estos parámetros ya tendremos nuestro Home Assistant listo para añadir dispositivos.

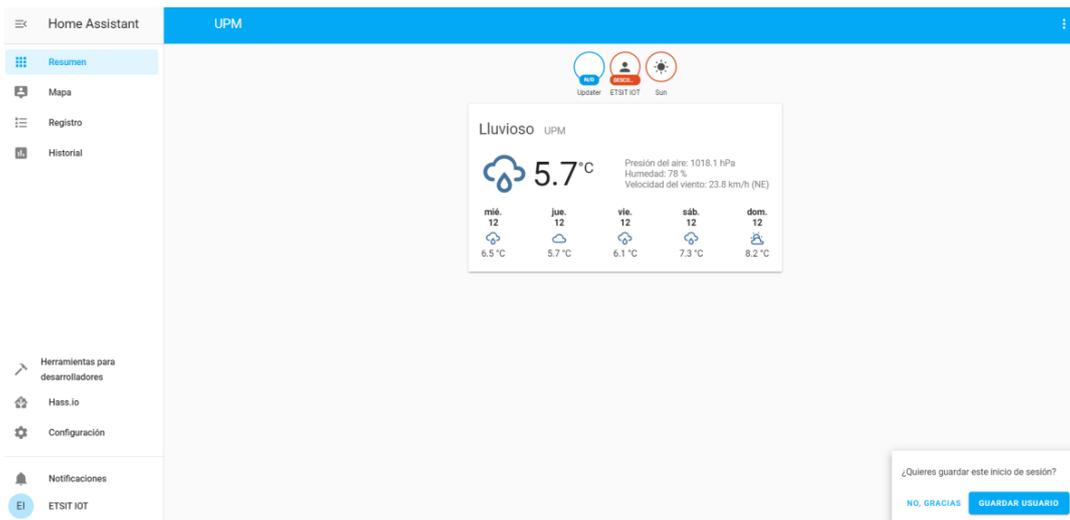


Figura 28 Página de inicio de Home Assistant.

3.5 INSTALACIÓN DE MOSQUITTO BROKER

Una de las grandes ventajas que nos ofrece Home Assistant es que dispone de gran variedad de programas anexados que sirven para complementar las funcionalidades que tiene. Por eso dentro de Home Assistant podemos encontrar una Add-On Store con programas adicionales ofrecidos por hass.io para múltiples funcionalidades. Entre ellos se encuentra el broker de MQTT Mosquitto. En este apartado se verá los aspectos más importantes para su instalación, más adelante con la instalación del primer dispositivo se estudiará su comportamiento y como funciona. Los pasos que seguir son los siguientes:

1) En la pestaña lateral donde aparece “Hass.io” encontraremos todos los add-on oficiales, así como los que ofrece la comunidad de Home Assistant. Es posible que en las versiones más recientes de Hass.io han actualizado el nombre y aparece “Supervisor” en la pestaña lateral. Una vez allí hay que buscar Mosquitto Broker y pulsar sobre Install. Pasados unos minutos ya tendremos instalado Mosquitto.

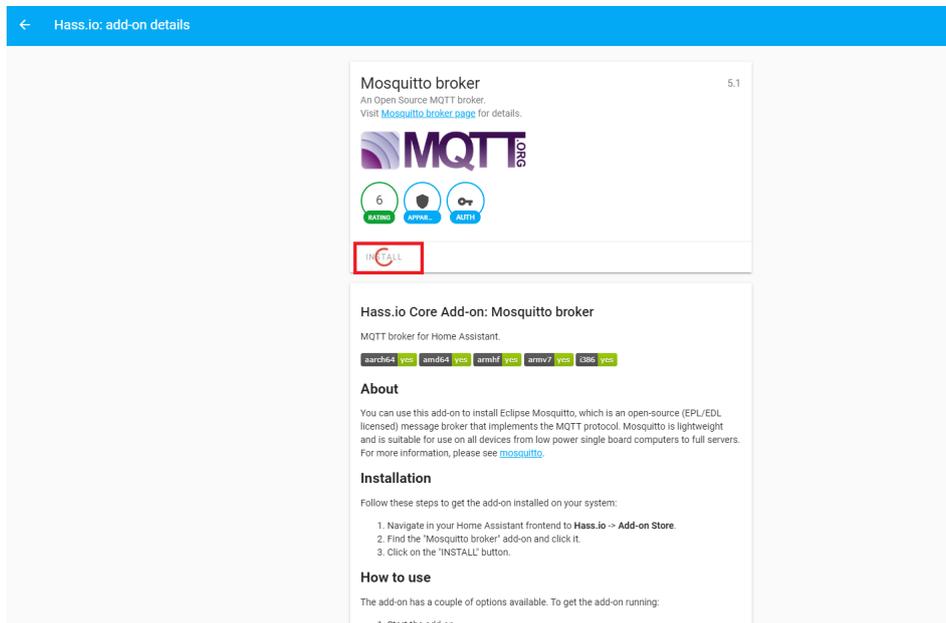


Figura 29 Instalación Mosquitto.

2) Una vez que tenemos instalado Mosquitto Broker hay que configurarlo y dentro de la pestaña de configuración hay que añadir las siguientes líneas:

```
logins:
  - username: your_usernameMQTT
    password: your_passwordMQTT
anonymous: false
customize:
  active: false
  folder: mosquitto
certfile: fullchain.pem
keyfile: privkey.pem
require_certificate: false
```

De esta forma crearemos un usuario y una contraseña para poder conectarnos a nuestro Broker. Otro aspecto importante es la configuración de los puertos por donde se van a mandar

y recibir los mensajes MQTT. Se configura por defecto el puerto 1883 sobre TCP que es el que indicaremos a nuestros dispositivos que utilicen. Por tanto, añadiremos los siguientes parámetros en la configuración:

Network		
Container	Host	Description
1883/tcp	1883	
1884/tcp	1884	
8883/tcp	8883	
8884/tcp	8884	

Figura 30 Configuración de puertos en Mosquitto.

3) Finalizada la configuración podremos darle a START y mosquitto estará funcionando.

3.6 ARCHIVOS DE CONFIGURACIÓN YAML

Como se ha explicado anteriormente Home Assistant utiliza archivos YAML, el archivo que contiene las componentes principales se llama configuración.yaml. Para poder modificar este archivo y otros que se verán más adelante hay que instalar otro add-on de hass.io. En este caso el add-on se llama Configurator. Una vez lo instalemos, le damos a “START” y ya podremos navegar por los archivos de configuración de nuestro sistema. Para ello hay que pulsar sobre “OPEN WEB IU” y nos aparecerá un menú donde veremos las carpetas y todos los archivos YAML de nuestro sistema.

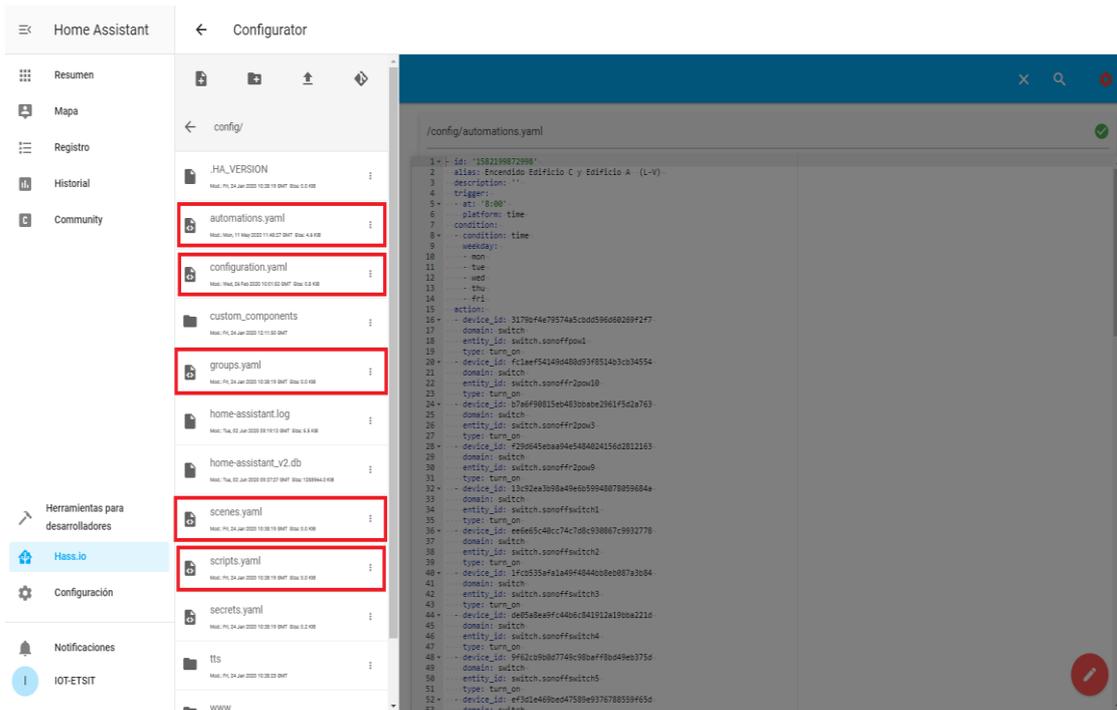


Figura 31 Archivos YAML Home Assistant.

3.7 INTEGRACIÓN DE DISPOSITIVOS

Se ha explicado como flashear los dispositivos Sonoff con Tasmota y a continuación vamos a ver como integrar estos dispositivos dentro de la aplicación de Home Assistant que se ha instalado en la Raspberry pi con un servidor de MQTT Mosquitto Broker. Los pasos que hay que seguir son los que se detallan a continuación.

1) Debemos tener conectado nuestros Sonoff a la red. Para acceder a ellos hay que ir a la dirección IP que se le ha asignado. Desde el terminal de Termite podemos ver cuál ha sido. Accediendo a la dirección <http://X.X.X.X> con esa IP asignada se nos abrirá el menú de configuración Tasmota. Pinchamos en **Configuración/Configuración MQTT** para acceder al siguiente menú.

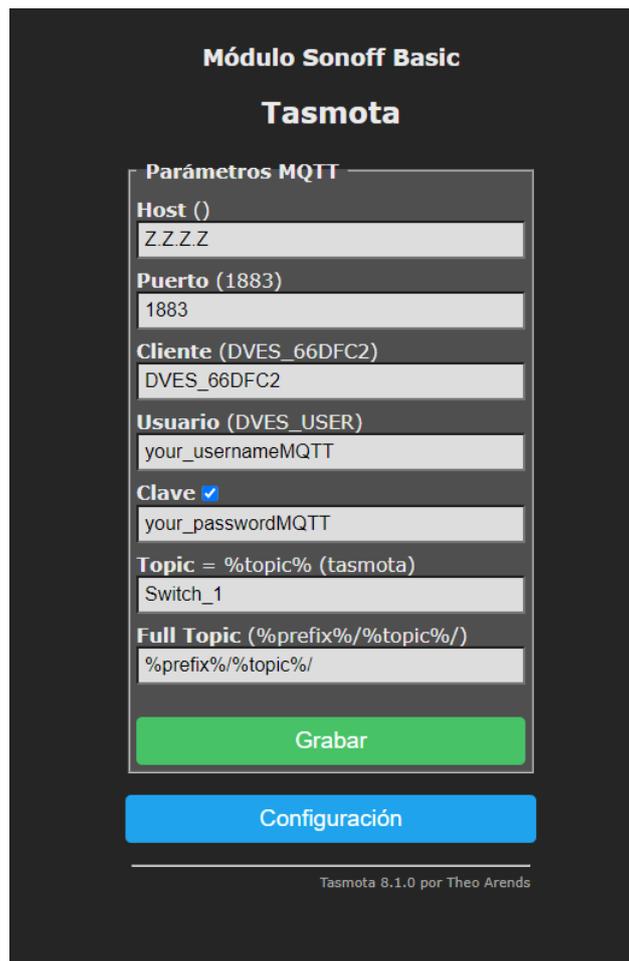


Figura 32 Menú de configuración de MQTT de Tasmota.

2) Introducir los parámetros de configuración MQTT. Los parámetros que se van a configurar son:

- Dirección IP de Home Assistant. Esta es la dirección donde tenemos nuestra Raspberry Pi y servidor MQTT.
- Puerto 1883. Este es el puerto que configuramos en el apartado 3.4 de mosquitto broker.
- Cliente: este debe de ser único es por esta razón por la que pondremos el mismo que nos aparece en la cabecera que está relacionado con su dirección MAC. Para que cuando se identifique sea el único dispositivo con ese nombre.

- Usuario: será el nombre de usuario de MQTT que se introdujo en la configuración del broker.
- Clave: será la password de MQTT que se introdujo en la configuración del broker.
- Topic será distinto para cada dispositivo y nos servirá para mandar mensajes de forma individual a cada uno de ellos. Por tanto, cuando se publique un mensaje con ese Topic lo recibirá el dispositivo que lo tenga asociado.

3) Integrar el dispositivo en la plataforma. Existen dos formas de hacerlo. Una de ellas es bastante sencilla y rápida gracias al Firmware de Tasmota que tiene una opción de descubrir el dispositivo y que Home Assistant automáticamente lo integre en la plataforma. La otra opción y que servirá para todo tipo de dispositivo independientemente del fabricante, es escribir el código en *configuration.yaml* al cuál se explica como acceder a través del add-on de configurator. En primer lugar, se va a explicar en detalle como hacer la segunda opción ya que es la que después puede ser de ayuda para integrar cualquier elemento de diferentes fabricantes.

- Introducir la configuración de nuestro broker en *configuration.yaml*. Hay que añadir las siguientes líneas de código:

```
mqtt:
  broker: Z.Z.Z.Z
  port: 1883
  username: your_usernameMQTT
  password: your_passwordMQTT
  discovery: true
  discovery_prefix: homeassistant
```

De esta forma haremos visible al servidor de MQTT y todos los dispositivos que apunten a la dirección Z.Z.Z.Z serán descubiertos. Recordemos que esta dirección es donde alojamos a la Raspberry Pi. Se ha mantenido la misma nomenclatura durante todo el trabajo de esta forma hay que recordar que la contraseña y usuario que hemos introducido en el servidor mosquitto es la que hay que poner en los Sonoff y en esta parte del elemento de configuración principal de Home Assistant.

- Dependiendo del tipo de dispositivo que se va a utilizar hay que utilizar un código u otro. En este ejemplo detallaremos como añadir el Sonoff Smart Switch Basic y un Sonoff Pow. Las funcionalidades que tienen cada uno son las siguientes:

Switch Smart Basic		Pow	
Switch	Sensor	Switch	Sensor
ON	N/A	ON	Tensión (V)
OFF	-	OFF	Corriente (A)
-	-	-	Potencia consumida durante el día en curso (kWh)
-	-	-	Potencia consumida durante el día de ayer (kWh)
-	-	-	Potencia total consumida desde que se ha puesto en funcionamiento (kWh)

Tabla 5 Tabla de funcionalidades de Sonoff Switch y Pow.

Es por esta razón que para el Sonoff Switch únicamente hay que añadir un código para su funcionalidad de encendido y apagado. Mientras que, para el Pow, además del encendido y apagado están los sensores de corriente y tensión.

Por tanto, el código que hay que añadir para un Sonoff switch es el siguiente:

```
switch:
- platform: mqtt
  name: "Sonoff Switch 1"
  command_topic: "cmdnd/Switch_1/power"
  state_topic: "stat/Switch_1/POWER"
  qos: 1
  payload_on: "ON"
  payload_off: "OFF"
  retain: true
device:
  identifiers: 1
  name: Switch1
```

Es muy importante que el nombre que se ha puesto en el topic del menú de configuración de MQTT de Tasmota coincida con el que aparece en el código aquí puesto para que mosquitto sepa a qué dispositivo le está mandando los mensajes.

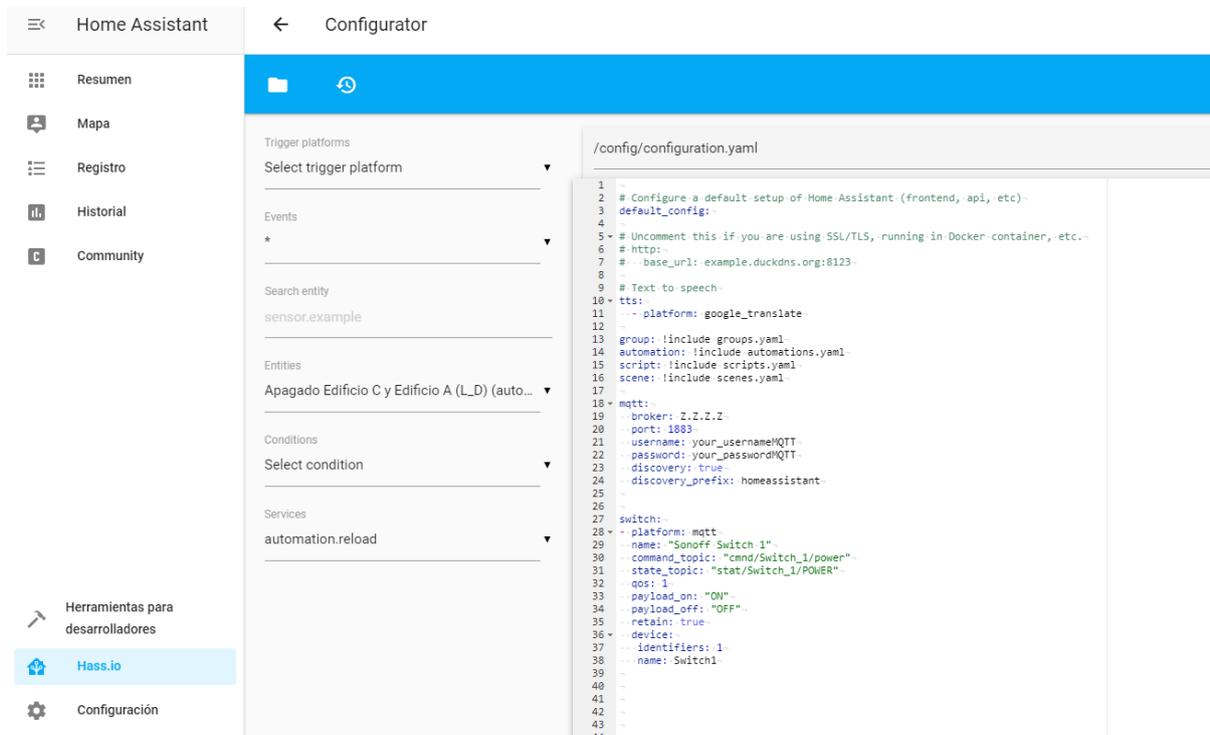


Figura 33 Configuración Sonoff Switch en configuration.yaml.

Por otro lado, para el Sonoff Pow hay que añadir la parte de Switch para su encendido y apagado, y los sensores con las siguientes líneas de código.

```

switch:
- platform: mqtt
  name: "POW"
  state_topic: "stat/pow_1/POWER"
  command_topic: "cmnd/pow_1/POWER"
  qos: 1
  payload_on: "ON"
  payload_off: "OFF"
  retain: false
sensor:
#sensor sonoff_pow_watts:
- platform: mqtt
  name: Current Power Usage
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{value_json['ENERGY'].Power }}"
  qos: 1
  unit_of_measurement : "W"
#sensor sonoff_pow_volts:
- platform: mqtt
  name: Current Voltage
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{ value_json['ENERGY'].Voltage }}"
  qos: 1
  unit_of_measurement : "V"
#sensor sonoff_pow_current:
- platform: mqtt
  name: Current Amps
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{ value_json['ENERGY'].Current }}"
  qos: 1
  unit_of_measurement : "A"
#sensor sonoff_pow_kwh_today:
- platform: mqtt
  name: Todays Power
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{ value_json['ENERGY'].Today }}"
  qos: 1
  unit_of_measurement : "kWh"
#sensor sonoff_pow_kwh_total:
- platform: mqtt
  name: Total Power
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{ value_json['ENERGY'].Total }}"
  qos: 1
  unit_of_measurement : "kWh"

```

```
#sensor sonoff_pow_kwh_yesterday:
- platform: mqtt
  name: Yesterdays Power
  state_topic: "tele/pow_1/SENSOR"
  value_template: "{{ value_json['ENERGY'].Yesterday }}"
  qos: 1
  unit_of_measurement : "kWh"
```

Una vez que se ha introducido el código es muy importante guardar la configuración y reiniciar el sistema. En caso de que haya algún error no dejara guardarlo y tendremos que buscar el posible fallo. Una vez que se ha reiniciado encontraremos los dispositivos en la página principal de Home Assistant.

- La otra forma que hay de añadir los dispositivos es de forma automática. Este proceso es más sencillo y con la que se han integrado los dispositivos en Home Assistant. Para poder hacerlo de forma automática basta con ir al menú de configuración de Tasmota. En el menú hay que ir a la Consola e introducir el siguiente parámetro:

SetOption19 1

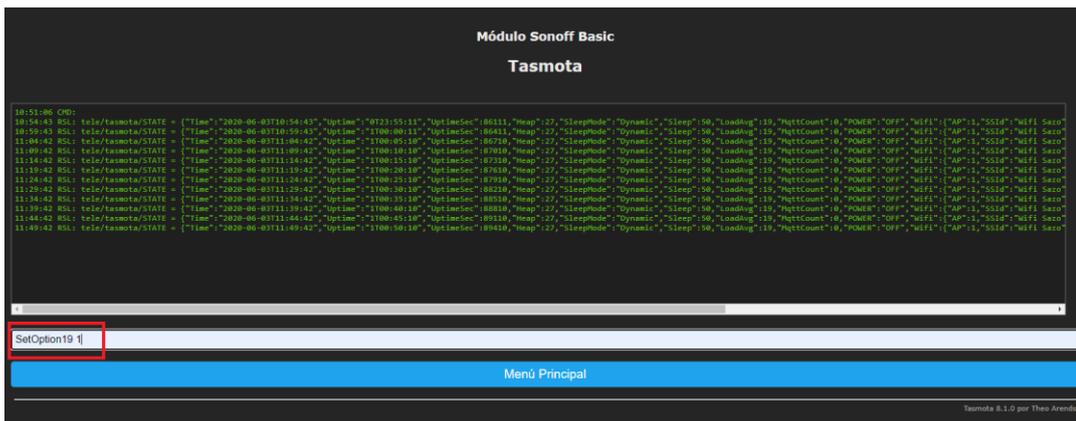


Figura 34 Consola de Tasmota.

Así haremos que el dispositivo aparezca automáticamente en Home assistant y lo podremos encontrar en la ruta **Configuración/Dispositivos**. Donde podremos encontrar todos los dispositivos que tenemos con los datos siguientes:

- Nombre del dispositivo
- Fabricante
- Modelo
- Área
- Batería

Cuando pinchamos sobre alguno de ellos nos podremos encontrar todas las entidades que tiene cada dispositivo. Las *entities* o entidades son variables que proporcionan los datos de un sensor o switch asociado. Para configurar la interfaz trabajaremos con las entidades que se hayan creado de cada dispositivo.

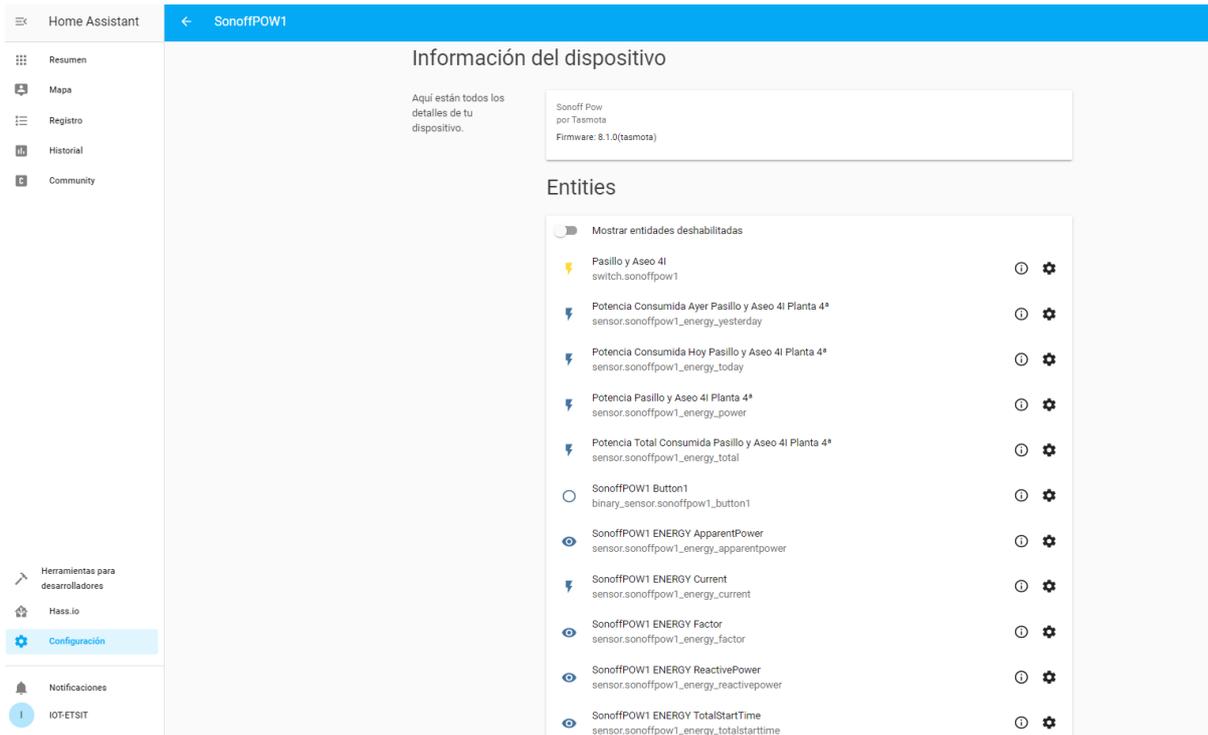


Figura 35 Entidades de un dispositivo.

3.8 INTERFAZ GRÁFICA DE HOME ASSISTANT

Este es uno de los puntos más importantes de home assistant ya que es lo que va a ver el usuario final que utilice la plataforma y es la manera que tenemos de gestionar los datos que nos ofrecen los dispositivos. En este apartado se harán una serie de indicaciones de como hacerlo. En la interfaz se utilizan paneles, grupos y pestañas para organizar los dispositivos que se han integrado. Home Assistant ofrece la posibilidad de hacer la interfaz de usuario de forma gráfica o utilizando el configurador en bruto para añadir el código con las funcionalidades que queremos que tengan o personalizaciones que deseemos.

Disponemos de tarjetas que nos dan la posibilidad de ofrecer diferentes tipos de datos expresado de una manera u otra. Existen 25 tipos de tarjetas personalizables que te permiten configurar como se presentan los valores de los objetos, llamar a funciones y realizar otras tareas. Se pueden distribuir en diferentes pestañas y a continuación ofrecemos un ejemplo de como mostrar la información.



Figura 36 Vistas de Home Assistant.

Se van a hacer cuatro vistas distribuidas de la siguiente forma:

- Página de Home: en esta pestaña encontraremos un listado de todos los dispositivos y los estados en que se encuentran en tiempo real.
- Edificio A: en esta vista se agruparán los dispositivos por bloques correspondientes a cada cuadro eléctrico del edificio.
- Edificio C: en esta vista se agruparán los dispositivos por bloques correspondientes a cada planta del edificio.

- Estudio de potencias y consumo: en esta página se presentarán los datos obtenidos por los sensores que tengamos. Podremos encontrar gráficas y datos sobre las potencias consumidas en el mismo día, día anterior y el total.

Página Home

En ella tendremos una tarjeta de tipo Entidades en las que vamos a añadir todos los dispositivos. Una tarjeta para ver el tiempo que hay en ese momento en el punto que hayamos seleccionado a la hora de configurar Home Assistant. Finalmente, otra tarjeta que tenga las automatizaciones que se hablará más adelante sobre ellas para activarla so desactivarlas. Hay dos formas de hacerlo, añadiendo el siguiente código en el configurador en bruto:

```
views:
  cards:
    - entities:
      - entity: switch.sonoffr2pow1
      - entity: switch.sonoffr2pow2
      - entity: switch.sonoffr2pow3
      - entity: switch.sonoffr2pow4
      - entity: switch.sonoffr2pow5
      - entity: switch.sonoffr2pow6
      - entity: switch.sonoffr2pow7
      - entity: switch.sonoffr2pow8
      - entity: switch.sonoffr2pow9
      - entity: switch.sonoffr2pow10
      - entity: switch.sonoffth161
      - entity: switch.sonoffth162
      - entity: switch.sonoffth163
      - entity: switch.sonoffth164
      - entity: switch.sonoffpow1
      - entity: switch.sonoffswitch1
      - entity: switch.sonoffswitch2
      - entity: switch.sonoffswitch3
      - entity: switch.sonoffswitch4
      - entity: switch.sonoffswitch5
      - entity: switch.sonoffswitch6
      - entity: switch.sonoffswitch7
      - entity: switch.sonoffswitch8
      - entity: switch.sonoffswitch9
      - entity: switch.sonoffswitch10
      - entity: switch.sonoffswitch11
      - entity: switch.sonoffswitch12
      title: Interruptor
      type: entities
    - entity: weather.etsit_upm
      type: weather-forecast
    - entities:
      -
      entity: automation.encendido_edificio_c_y_edificio_a_l_v
      -entity:
      automation.apagado_edificio_c_y_edificio_a_l_d
      title: Automatizaciones
      type: entities
      -entity: sensor.date_time
      type: sensor
    path: 0
    title: Home
```

Y la otra forma es a través de la interfaz de Home Assistant, para ello hay que seleccionar la tarjeta del tipo que queramos.

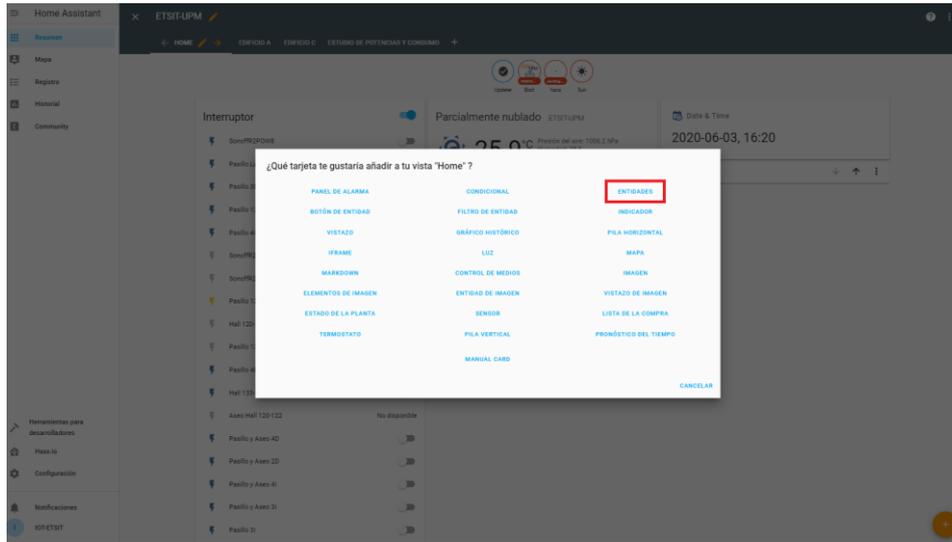


Figura 37 Página de selección de tarjetas de la interfaz de Home Assistant.

Una vez que hayamos seleccionado la tarjeta que se desea, solo hay que añadir las entidades que queramos en ella. En la siguiente imagen vemos como están todas las entidades que aparecen en el configurador en bruto. Así como destacar los siguientes puntos para entender cómo funciona la interfaz gráfica.

- 1) Interruptor de la cabecera. Este interruptor nos permite encender y apagar todos los dispositivos que haya en este grupo a la misma vez. Es opcional y se puede activar o desactivar en cualquier momento pues es editable desde que se crea.
- 2) Estado apagado. Cuando un interruptor está en el estado “OFF” este se encuentra en color azul oscuro
- 3) Estado encendido. Cuando un interruptor está en activo se encuentra la imagen del logo en amarillo y el interruptor en color azul.
- 4) Estado desconocido. Cuando nos aparece en No disponible lo más probable es que el dispositivo no tenga conexión Wifi o que la fuente de alimentación no le este administrando energía. Esto será muy útil ya que nos puede decir si ha habido un corte de luz en función de su estado, es decir, si aparece activo y luego cambia a no disponible puedes saber si ha habido alguna incidencia en el sistema, así como la hora a la que el problema se ha solucionado.

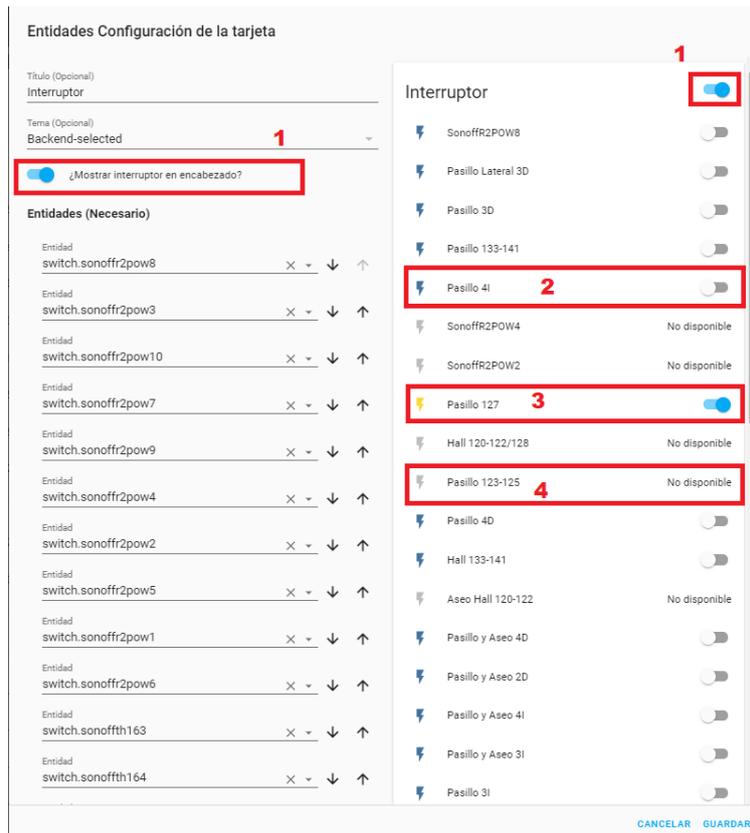


Figura 38 Configuración de tarjeta Entidades.

Otro aspecto muy importante es que en el momento que pinchamos sobre alguna de las entidades podremos ver los estados en los que ha estado en las últimas 24 horas. Así como editar el nombre con el que aparecerá en la página de inicio.



Figura 39 Estado del dispositivo.

Edificio A y Edificio C

En esta página se van a añadir grupos de interruptores ordenador por donde están ubicados en los cuadros eléctricos. Para dar más información al usuario se va a añadir una tarjeta de tipo imagen para añadir una foto con un plano de donde están estos cuadros eléctricos y a que luminarias se corresponden. La tarjeta donde se encuentren los interruptores será de Entidades donde seleccionaremos aquellas que se corresponden con el cuadro eléctrico y les cambiaremos el nombre. Para añadir una imagen, tenemos dos posibilidades, a través de una URL o guardando una imagen dentro de la Raspberry Pi. Para poder tener acceso a los archivos de las Raspberry será necesario instalar un add-on que nos permita acceder a las carpetas. Por ello instalaremos el add-on Samba Share.

Samba es un protocolo de archivos compartidos de Microsoft Windows, de esta forma cuando instalamos este complemento en nuestro Home Assistant podremos acceder a los ficheros de configuración desde cualquier PC que este conectado a la misma red local que la Raspberry Pi y tenga instalado Windows con la opción de archivos compartidos habilitada.

Para instalarlo en la barra lateral hay que buscar Samba Share en la pestaña de ADD-ON STORE dentro de Hass.io. Una vez seleccionado pulsamos sobre “INSTALL” y añadimos la siguiente configuración.

```
workgroup: WORKGROUP
username: nombre_usuario
password: contraseña_usuario
interface: eth0
allow_hosts: []
veto_files:
- .*
- .DS_Store
- Thumbs.db
- icon?
- .Trashes
name: hassio
guest: false
map:
  config: true
  addons: true
  ssl: false
  share: true
  backup: true
```

Debemos de configurar el parámetro de interface como eth0 por tener la Raspberry conectada por cable ethernet a la red local. Una vez configurado pulsamos sobre “START” y reiniciamos el sistema. Pasado unos minutos encontraremos nuestro sistema en Ubicaciones de red.

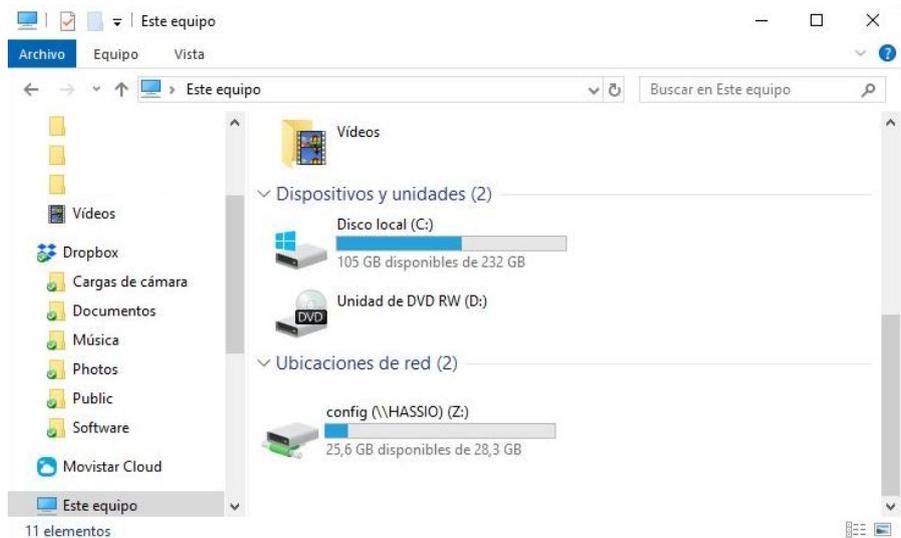


Figura 40 Ubicación de red de Hassio.

Una vez que tenemos instalado Samba y podemos acceder a las carpetas dentro de la carpeta **config/www** podemos guardar todas las imágenes que queramos mostrar después en nuestra tarjeta imagen. Una vez hecho esto podremos añadir el siguiente código para la interfaz gráfica.

```
cards:
  - hold_action:
    action: none
    image: /local/Plano1A.jpg
    tap_action:
    action: none
    type: picture
  - entities:
    - entity: switch.sonoffr2pow7
    - entity: switch.sonoffth164
    title: Cuadro 1.5 Hall aulas 0
    type: entities
  - entities:
    - entity: switch.sonoffr2pow5
    title: Cuadro 1.3 Pasillo 127
    type: entities
  - entities:
    - entity: switch.sonoffr2pow1
    - entity: switch.sonoffr2pow6
    - entity: switch.sonoffth162
    title: Cuadro Hall Pasillo derecha
    type: entities
path: 1
title: Edificio A
```

Al igual que el anterior se puede hacer también a través de la interfaz gráfica. Finalmente, el resultado que tendremos será el siguiente.

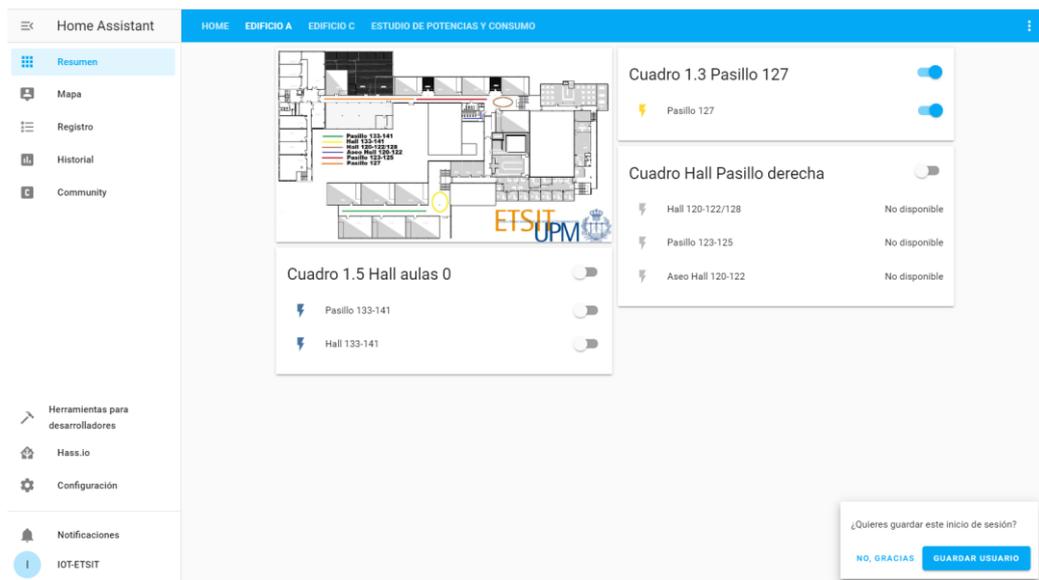


Figura 41 Vista con una tarjeta de imagen y diferentes entidades.

Estudio de potencias y consumo

Los datos que se recogerán en esta vista son los que nos van a permitir tomar decisiones a partir de su análisis para aplicar políticas de eficiencia. Por ello es muy importante la forma de mostrar estos datos. Por un lado, se van a mostrar graficas con la potencia que se está consumiendo en tiempo real ordenada por plantas del edificio C y cuadros del edificio A como la siguiente.

Potencias Planta 4ª Edificio C

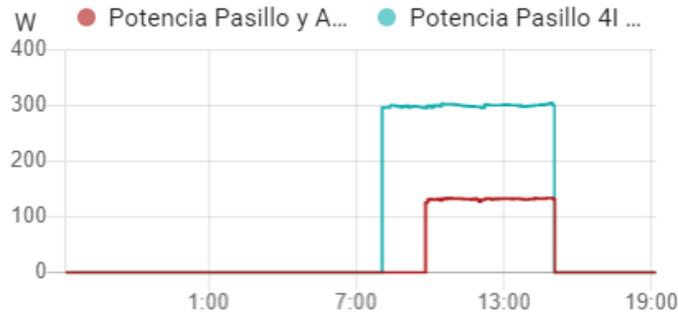


Figura 42 Gráfica de consumo de Home Assistant.

Sabiendo que los dispositivos se han encendido a las 8h y apagado a las 15h, el análisis que podemos extraer en función de los datos de la siguiente gráfica es el siguiente:

- Vemos que entre las 8h y las 15h los dispositivos están desconectados.
- A las 8h hay un consumo de 300 W del pasillo 4I, lo que da a entender que si había una luminaria conectada a ese interruptor cuando el switch paso a ON las luces se encendieron en el mismo estado que se apagaron, es decir, nadie ha ido al interruptor de pared a apagarlas. Se han apagado porque el Switch ha pasado a estar en OFF
- En las luminarias de Pasillo y Aseo vemos que a las 8h que el switch está dejando pasar la corriente todas las luces se encuentran apagadas, es decir, el día anterior se apagaron en el interruptor de pared manual. A las 10 de la mañana hay un consumo de 120 W que se podría corresponder con las luminarias del aseo, es decir, alguien ha entrado a las 10 al aseo y se ha dejado las luces apagadas. A las 15h que el switch ya no deja pasar la corriente y se han apagado. Por tanto, al día siguiente se encenderán a las 8h a no ser que alguien cambie el estado del interruptor del switch.

Es importante entender bien estos datos y las gráficas nos sirven de ayuda para interpretar la información y ver las horas a las que se encienden y se apagan las luces.

Los siguientes datos que hay que priorizar es la toma de potencias. Para ello se propone 3 tablas donde mostrar el consumo total, el consumo del día anterior y el consumo durante el presente día. De esta forma podremos introducir en una tabla diaria los datos y ver su comportamiento. Ver días de la semana que mayor consumo hay, o ver si por algún motivo especial las luces se han pasado más tiempo encendidas.

En la siguiente imagen vemos cómo van a aparecer estos datos en la pestaña de Estudio de potencias y consumo de nuestro Home Assistant.

Potencia Total Consumida	Potencia Total Consumida Ayer	Potencia Total Consumida Hoy
⚡ Potencia Total Consumida Pasillo y As... 2.24 kWh	⚡ Potencia Consumida Ayer Pasillo 4I Pl... 1.324 kWh	⚡ Potencia Consumida Hoy Pasillo y As... 0.687 kWh
⚡ Potencia Total Consumida Pasillo 4... 159.078 kWh	⚡ Potencia Consumida Ayer Pasillo y Ase... 0.24 kWh	⚡ Potencia Consumida Hoy Pasillo 3D Pla... 0.0 kWh
⚡ Potencia Total Consumida Pasillo L... 146.547 kWh	⚡ Potencia Consumida Ayer Pasillo Lateral... 0.0 kWh	⚡ Potencia Consumida Hoy Pasillo Late... 1.267 kWh
⚡ Potencia Total Consumida Pasillo 3... 135.275 kWh	⚡ Potencia Consumida Ayer Pasillo 3D ... 0.135 kWh	⚡ Potencia Consumida Hoy Pasillo 4I Pl... 2.107 kWh
⚡ Potencia Total Consumida Pasillo 1... 202.323 kWh	⚡ Potencia Consumida Ayer Pasillo 133-14... 0.0 kWh	⚡ Potencia Consumida Hoy Pasillo 133-14... 0.0 kWh
⚡ Potencia Total Consumida Pasillo 1... 238.797 kWh	⚡ Potencia Consumida Ayer Pasillo 127 Pr... 0.0 kWh	⚡ Potencia Consumida Hoy Pasillo 127 Pri... 0.0 kWh
⚡ Potencia Total Consumida Hall 12... No disponible	⚡ Potencia Consumida Ayer Hall 120... No disponible	⚡ Potencia Consumida Hoy Hall 120... No disponible
⚡ Potencia Total Consumida Pasillo ... No disponible	⚡ Potencia Consumida Ayer Pasillo 1... No disponible	⚡ Potencia Consumida Hoy Pasillo 1... No disponible

Figura 43 Consumo de los dispositivos Sonoff Pow.

Se pueden añadir de forma manual con las tarjetas de Gráfico Histórico para ver las gráficas de los consumos y añadiendo las entidades que queremos visualizar. Y para ver las potencias nos basta con crear una tarjeta de entidades, pero añadiendo las correspondiente a cada sensor del dispositivo Pow que se han añadido anteriormente. Para poder ver estos datos hay que introducir las siguientes líneas de código en el configurador en bruto.

cards:

- **entities:**
 - **entity:** sensor.sonoffpow1_energy_power
 - **entity:** sensor.sonoffr2pow9_energy_power
- hours_to_show:** 24
- refresh_interval:** 0
- title:** Potencias Planta 4ª Edificio C
- type:** history-graph
- **entities:**
 - **entity:** sensor.sonoffr2pow3_energy_power
 - **entity:** sensor.sonoffr2pow10_energy_power
- hours_to_show:** 24
- refresh_interval:** 0
- title:** Potencias Planta 3ª Edificio C
- type:** history-graph
- **entities:**
 - **entity:** sensor.sonoffr2pow7_energy_power
- hours_to_show:** 24
- refresh_interval:** 0
- title:** Cuadro 1.5 Edificio A Hall aulas 0
- type:** history-graph
- **entities:**
 - **entity:** sensor.sonoffr2pow5_energy_power
- hours_to_show:** 24
- refresh_interval:** 0
- title:** Cuadro 1.3 Edificio A Pasillo 127
- type:** history-graph

```

- entities:
  - entity: sensor.sonoffr2pow1_energy_power
  - entity: sensor.sonoffr2pow6_energy_power
hours_to_show: 24
refresh_interval: 0
title: Cuadro Hall Pasillo Derecha
type: history-graph
- entities:
  - entity: sensor.sonoffpow1_energy_today
  - entity: sensor.sonoffr2pow10_energy_today
  - entity: sensor.sonoffr2pow3_energy_today
  - entity: sensor.sonoffr2pow9_energy_today
  - entity: sensor.sonoffr2pow7_energy_today
  - entity: sensor.sonoffr2pow5_energy_today
  - entity: sensor.sonoffr2pow1_energy_today
  - entity: sensor.sonoffr2pow6_energy_today
show_header_toggle: false
title: Potencia Total Consumida Hoy
type: entities
- entities:
  - entity: sensor.sonoffpow1_energy_total
  - entity: sensor.sonoffr2pow9_energy_total
  - entity: sensor.sonoffr2pow3_energy_total
  - entity: sensor.sonoffr2pow10_energy_total
  - entity: sensor.sonoffr2pow7_energy_total
  - entity: sensor.sonoffr2pow5_energy_total
  - entity: sensor.sonoffr2pow1_energy_total
  - entity: sensor.sonoffr2pow6_energy_total
show_header_toggle: false
title: 'Potencia Total Consumida '
type: entities
- entities:
  - entity: sensor.sonoffr2pow9_energy_yesterday
  - entity: sensor.sonoffpow1_energy_yesterday
  - entity: sensor.sonoffr2pow3_energy_yesterday
  - entity: sensor.sonoffr2pow10_energy_yesterday
  - entity: sensor.sonoffr2pow7_energy_yesterday
  - entity: sensor.sonoffr2pow5_energy_yesterday
  - entity: sensor.sonoffr2pow1_energy_yesterday
  - entity: sensor.sonoffr2pow6_energy_yesterday
show_header_toggle: false
title: Potencia Total Consumida Ayer
type: entities
path: 3

```

3.9 AUTOMATIZACIONES

Debido al análisis de datos obtenidos se pueden crear políticas para mejorar le eficiencia energética. Por ellos Home Assistant te permite hacer cualquier tipo de automatización a través de su lenguaje YAML. Home assistant desde su interfaz te permite hacer automatizaciones siguiendo estos 3 sencillos pasos.

1) Crear un desencadenante. Los desencadenantes son los que inician el funcionamiento de una regla de automatización. Es posible especificar varios desencadenantes para la misma regla. Una vez que se inicia un desencadenante, Home Assistant comprobará las condiciones, si las hubiere, y ejecutará la acción. Los desencadenantes que vienen por defecto en Home Assistant son de los siguientes tipo:

- Dispositivo: Los desencadenantes de dispositivos abarcan un conjunto de eventos definidos por una integración. Esto incluye, por ejemplo, cambios de estado de los sensores, así como eventos de botón desde controles remotos. Los activadores de dispositivos MQTT se configuran mediante la detección automática. A diferencia de los disparadores de estado, los disparadores de dispositivos están vinculados a un dispositivo y no necesariamente a una entidad. Para usar un disparador de dispositivo, configure una automatización a través de la interfaz del navegador. Si desea utilizar un disparador de dispositivo para una automatización que no se administra a través de la interfaz del navegador, puede copiar el YAML del widget de activación en la interfaz y pegarlo en la lista de activadores de su automatización.
- Evento: Se dispara cuando se recibe un evento. Los eventos son los componentes básicos de Home Assistant. Puede hacer coincidir eventos solo con el nombre del evento o también requerir que estén presentes datos de eventos específicos.
- Estado: Se activa cuando cambia el estado de cualquiera de las entidades dadas. Si solo `entity_id` se da, el disparador se disparará para todos los cambios de estado, incluso si solo cambian los atributos de estado.
- Geolocalización: El disparador de geolocalización se dispara cuando una entidad aparece o desaparece de una zona. Las entidades creadas por una plataforma de Geolocalización admiten informes de coordenadas GPS. Debido a que estas plataformas generan y eliminan entidades automáticamente, la identificación de la entidad normalmente no se puede predecir.
- Home Assistant: Se dispara cuando Home Assistant se inicia o se apaga
- MQTT: Se activa cuando se recibe un mensaje específico sobre un tema MQTT dado. Opcionalmente, puede coincidir con la carga útil que se envía sobre el tema. La codificación de carga útil predeterminada es 'utf-8'. Para imágenes y otras cargas de bytes, use `encoding: "`para deshabilitar completamente la decodificación de la carga.
- Estado numérico: Se activa cuando el valor numérico del estado de una entidad cruza un umbral dado. En el cambio de estado de una entidad especificada, intenta analizar el estado como un número y se dispara si el valor cambia de arriba a abajo o de abajo a arriba del umbral dado.
- Sol: Se dispara cuando el sol se está poniendo o saliendo, es decir, cuando la elevación del sol alcanza 0° . Se puede dar una compensación de tiempo opcional para que dispare un tiempo establecido antes o después del evento solar (por ejemplo, 45 minutos antes del atardecer).
- Plantilla: La plantilla desencadena el trabajo mediante la evaluación de una plantilla en cada cambio de estado para todas las entidades reconocidas. El disparador se disparará si el cambio de estado provocó que la plantilla se convirtiera en 'verdadera'.

- Hora: El disparador de tiempo está configurado para dispararse una vez en un momento específico cada día.
- Patrón de tiempo: Con el activador de patrón de tiempo, puede coincidir si la hora, minuto o segundo de la hora actual coincide con un valor específico.
- Zona: El disparador de zona se dispara cuando una entidad está entrando o saliendo de la zona. Para que funcione la automatización de zonas, debe haber configurado una plataforma de seguimiento de dispositivos que admita las coordenadas GPS de informes

2) Condiciones: Las condiciones son una parte opcional de una regla de automatización y se pueden utilizar para evitar que se produzca una acción cuando se desencadena la automatización. Las condiciones se parecen mucho a los desencadenantes, pero son muy diferentes. Un desencadenante analizará los eventos que ocurren en el sistema en cualquier momento, mientras que una condición solo analiza cómo está el sistema en un momento en concreto. Un desencadenante puede detectar que se está activando un interruptor. Una condición solo puede detectar si un interruptor está encendido o apagado. Las condiciones que viene por defecto en Home Assistant son:

- Y Condición: Pasa si todas las condiciones incrustadas son válidas.
- O Condición: Pasa si alguna condición incrustada es válida.
- No Condición: Pasa si todas las condiciones incrustadas no son válidas.
- Condición de estado numérico: Este tipo de condición intenta analizar el estado de la entidad especificada como un número y se dispara si el valor coincide con los umbrales.
- Condición de estado: Comprueba si una entidad es un estado especificado.
- Condición del sol: El estado del sol se puede usar para probar si el sol se ha puesto o ha salido.
- Condición de la plantilla: La condición de plantilla prueba si la plantilla dada representa un valor igual a verdadero.
- Condición de tiempo: La condición de tiempo puede probar si es después de un tiempo específico, antes de un tiempo específico o si es un cierto día de la semana.
- Condición de la zona: Las condiciones de zona prueban si una entidad está en una zona determinada. Para que funcione la automatización de zonas, debe haber configurado una plataforma de seguimiento de dispositivos que admita las coordenadas GPS de informes.

3) Acciones: Las acciones son lo que hará Home Assistant cuando se desencadene la automatización.

Las condiciones también pueden ser parte de una acción. Puede combinar varias llamadas de servicio y condiciones en una sola acción, y se procesarán en el orden en que las puso. Si el resultado de una condición es falso, la acción se detendrá allí, por lo que cualquier llamada de servicio después de esa condición no será ejecutado.

Además de poder hacer este tipo de automatizaciones gracias a la interfaz gráfica de Home Assistant en su menú lateral se podrían hacer otro tipo de reglas a través de *automations.yaml* al que accedemos desde nuestro add-on de Configurator. Como ejemplo ponemos una automatización de encendido de lunes a viernes de un dispositivo a las 8h de la mañana. Y además como segundo ejemplo hemos puesto una regla para el apagado del dispositivo todos los días de la semana a las 15h. Esto es importante hacerlo por si alguien ha encendido el dispositivo de forma manual y asegurarnos que se apague los fines de semana.

Ejemplo 1

```
- id: '1582199872998'
  alias: Encendido Edificio C y Edificio A (L-V)
  description: ''
  trigger:
    - at: '8:00'
      platform: time
  condition:
    - condition: time
      weekday:
        - mon
        - tue
        - wed
        - thu
        - fri
  action:
    - device_id: 3179bf4e79574a5cbdd596d60269f2f7
      domain: switch
      entity_id: switch.sonoffpow1
      type: turn_on
```

Ejemplo 2

```
- id: '1582200030756'
  alias: Apagado Edificio C y Edificio A (L_D)
  description: ''
  trigger:
    - at: '15:00'
      platform: time
  condition: []
  action:
    - device_id: 3179bf4e79574a5cbdd596d60269f2f7
      domain: switch
      entity_id: switch.sonoffpow1
      type: turn_off
```

3.10 VISUALIZACIÓN DEL SISTEMA

Una de las cosas que se ha querido incorporar a la aplicación que se está diseñando es ofrecer la posibilidad de solo mostrar unas vistas determinadas dependiendo del usuario que esté accediendo a Home Assistant. Actualmente, Home Assistant nos ofrece la oportunidad de crear varios usuarios desde su menú de configuración. Estos usuarios pueden tener control total sobre la aplicación, es decir, realizar cualquier tipo de modificación y acceso a toda la información. Y la otra opción es que tenga acceso a todas las vistas que hay creadas, pero sin poder modificar ningún tipo de entidad o parámetro. Lo que se plantea es que si no queremos que un usuario tenga acceso a información como puede ser la toma de datos, cuando inicie la

aplicación con su nombre y usuario solo pueda ver las vistas que el administrador desee. Para ello debemos de instalar la Home Assistant Community Store (HACS) que es la tienda de componentes gratuitos que ofrece la comunidad de Home Assistant. Su funcionamiento es muy sencillo, parecido a los add-on que trae Hass.io por defecto. Para instalarlo hay que seguir estos pasos.

- 1) Ir a la página de GitHub de HACS y descargar la última versión. El archivo es hacs.zip [18]
- 2) Descomprimir hacs.zip
- 3) Gracias a que se ha instalado el add-on de Samba Share desde Windows podemos acceder a las carpetas de nuestro Home Assistant. Dentro de la carpeta que tiene nuestro archivo *configuration.yaml* hay que crear una carpeta llamada *custom_components* y copiar en ella el archivo que se ha descomprimido.
- 4) Reiniciar el sistema y ahora hay que añadir HACS a la configuración.
- 5) Crear un Token en GitHub de acceso personal. Para ello hay que ir a GitHub y pinchar sobre Generate New Token. [19]

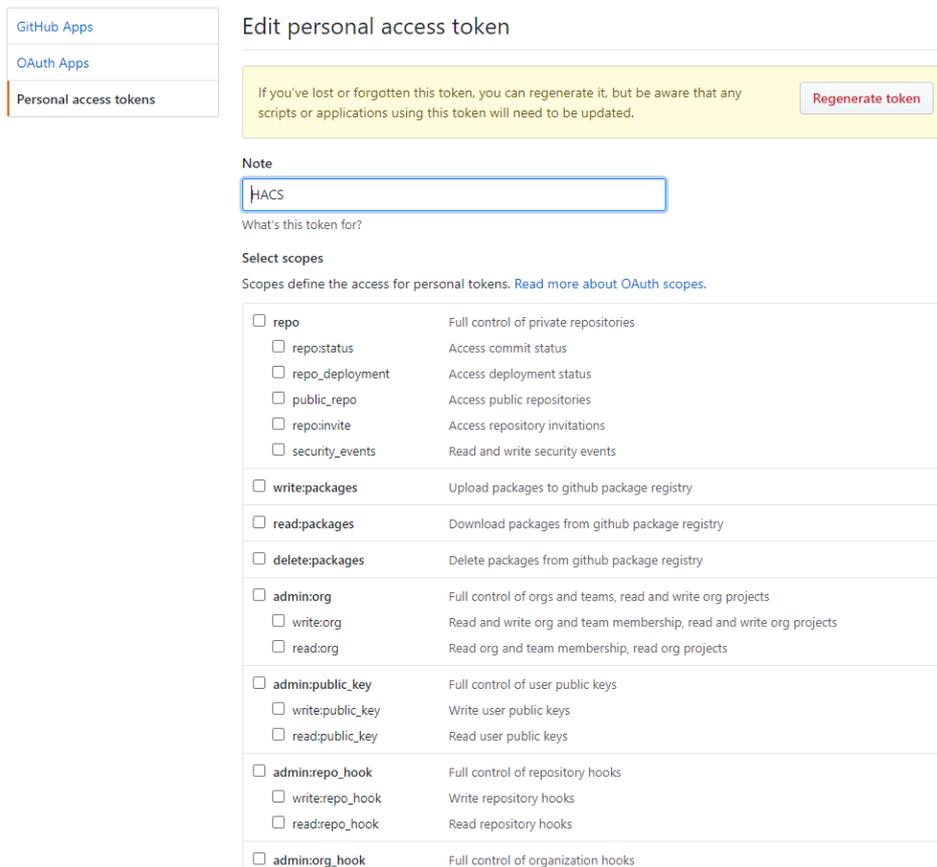


Figura 44 Página para crear un Token de acceso en GitHub.

- 6) Una vez que se genera copia el identificador que nos proporciona el sistema.
- 7) En configuración dentro de integraciones, pulsando el botón de + podremos buscar HACS. Pinchamos sobre ella y añadimos el token que hemos creado en GitHub. Después de unos minutos ya tendremos instalado HACS

Todo este proceso se ha hecho para poder instalar el componente que realmente necesitamos que está dentro de la tienda de HACS que es Compact Custom Header.

Para ello es simplemente ir a la Home Assistant Community y dentro de Plugins instalar Custom Header y cards-tool.

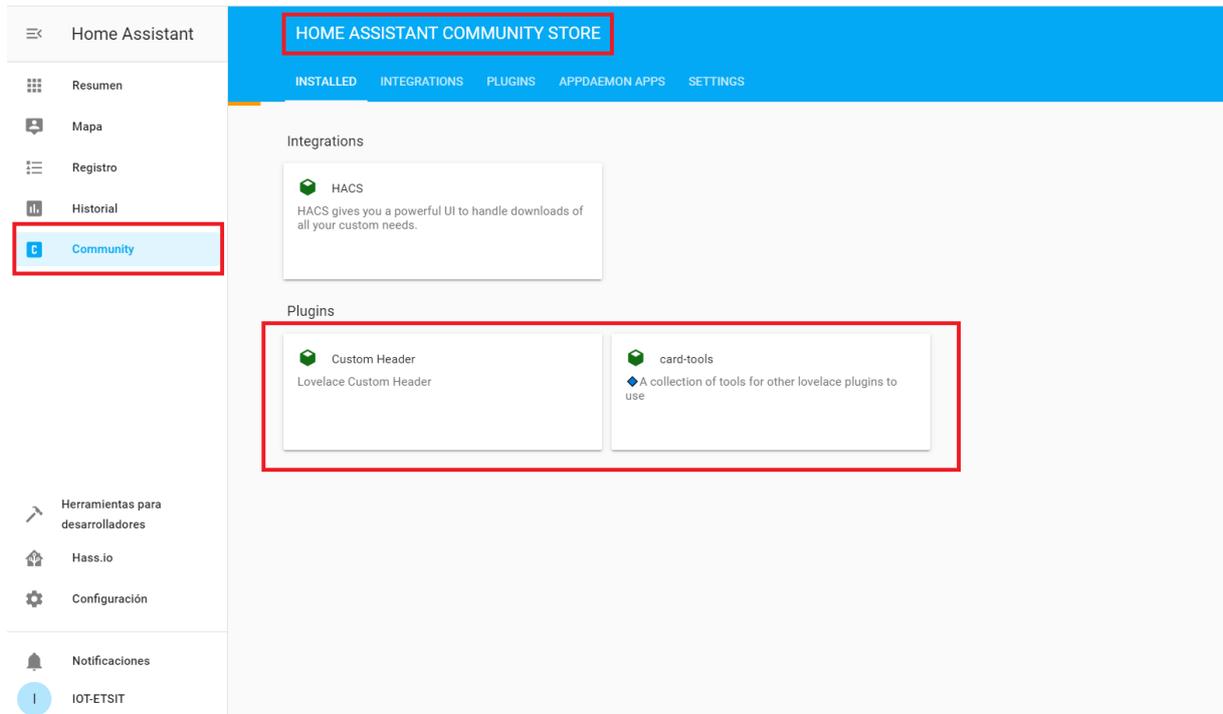


Figura 45 HACS de Home Assistant.

Cuando tengamos instalado Custom Header podremos especificar para cada usuario las vistas que queremos que vea al acceder a la aplicación. Para ello lo único que hay que hacer es añadir las siguientes líneas de código al principio del Configurador en Bruto de la interfaz de Home Assistant.

```

custom_header:
  compact_mode: true
  exceptions:
    - conditions:
      user: Etsit
    config:
      default_tab: '1,2'
      disable_sidebar: true
      hide_config: true
      hide_raw: true
      hide_tabs: '0,3'
      menu_hide: true
      options_hide: true
  
```

Donde vemos que el usuario Etsit solo tendrá acceso a las vistas 1 y 2. Mientras que el usuario administrador tiene acceso a todas las vistas que tiene el sistema.

4. VALIDACIÓN Y EVALUACIÓN

En este capítulo del trabajo se va a presentar la validación del prototipo después de haber estudiado el diseño para un sistema de monitorización. Se trabajará primero en el laboratorio donde se prepararán los equipos para ser instalados inicialmente en un entorno de pruebas, donde se estudiará su comportamiento y se implementarán en la plataforma de Home Assistant. Paralelamente al trabajo en laboratorio se presentará el trabajo a la ETSIT para poder llevarlo a cabo en un entorno de producción. En este capítulo se verán las fases que se han ido siguiendo.

4.1 VALIDACIÓN EN LABORATORIO

Después de estudiar el diseño del sistema se ha realizado una fase inicial para testear los dispositivos en el laboratorio. Para ello se ha creado un primer sistema con el objetivo de monitorizar varios dispositivos Sonoff y ver qué posibilidades se nos ofrecen. Para esta validación en el laboratorio se ha dispuesto de los siguientes Sonoff.

- 4 Sonoff Smart Switch
- 3 Sonoff Pow

Hemos conectado un punto de acceso a la red local de los laboratorios del DIT, no ha sido necesario que tenga conexión a internet. A partir de ese momento se ha creado una aplicación de Home Assistant para pruebas con un servidor mosquito conectado a la red del DIT. Después se han preparado los dispositivos Sonoff con Tasmota y conectados al punto de acceso que se ha instalado previamente. Hemos hecho que tenga conexión a nuestro servidor de mosquito. Durante esta fase de pruebas hemos conectado los dispositivos a calefactores, equipos de aire acondicionado, impresora y enchufes.

Las pruebas que se han hecho han sido las siguientes:

- Comprobar que no ha habido pérdidas de conexión.
- Realizar cambios de estados y ver que se ejecutan sin ningún fallo.
- Apagar el punto de acceso y ver que sucede después de un tiempo se conectó al servidor sin problema.
- Apagar el servidor. Se aprecia que los equipos mantienen su estado.
- Realizar automatizaciones y ver que se ejecutan.
- Realizar medidas de consumo aplicando reglas de automatización y sin aplicar. Se ha comprobado que se puede mejorar la eficiencia.

Una vez que se sabe que el comportamiento es estable se ha querido dar un paso más haya implementado esta solución en las instalaciones de la ETSIT. Para ello después de ver que en el piloto de laboratorio se puede mejorar la eficiencia aplicando reglas de automatización se ha presentado el proyecto a la escuela. En esta presentación que se mantiene con el Subdirector de asuntos económicos, secretaría técnica y equipo de mantenimiento se les va a plantear los siguientes aspectos.

- Presentar los equipos de control eléctrico para despliegue en el piloto y la plataforma de control.
- Identificar y fijar las zonas de control, estableciendo dos posibles fases de implantación.

Después de esta reunión para poder poner en marcha los equipos se necesita poder conectarlos a la red que nos da la universidad ya que sabemos que tiene cobertura en todos los edificios de la escuela. En ese momento hay que ponerse en coordinación con el Rectorado y dar de alta los equipos en la red UPM. Recopilamos las direcciones MAC de los

dispositivos que tenemos para poder conectar a la red y nos coordinamos con el Servicio de Infraestructura e Innovación de la Universidad Politécnica de Madrid. Después de conseguir la autorización, nos dan los equipos de alta en la red “Servicios UPM” con las siguientes indicaciones sobre la estructura del servicio.

- El SSID ServiciosUPM es abierto, oculto y con autenticación por dirección MAC.
- No está permitida la comunicación entre los clientes conectados a la Wi-Fi.
- El direccionamiento IP que se asigna es privado.
- La asignación de direcciones IP se hace usando DHCP estático, para que se asigne a cada dispositivo siempre la misma dirección IP.
- Por defecto, solo estarán abiertos los puertos TCP/UDP y destinos que se especifiquen para el servicio. En este caso, se permite el puerto 1883 tanto en TCP con en UDP.
- Para la comunicación con destinos externos, se hace NAT utilizando una dirección IP pública única para cada servicio, por si hace falta filtrar las direcciones en el extremo que presta el servicio. En este caso, la dirección IP es 138.100.X.X.
- Adicionalmente se permite la conexión con los servidores DNS y NTP de la universidad.
- Los servidores NTP permitidos son:
 - 1-. tic.upm.es
 - 2-. tac.upm.es
- Los servidores DNS se asignan automáticamente por DHCP, siendo dichos servidores los únicos permitidos.

Cabe aclarar que el diseño actual de la infraestructura de red permite únicamente que sean los clientes los que inicien la comunicación, con lo que no es posible que el servidor envíe peticiones a los clientes.

Con estas indicaciones configuramos los equipos con el SSID de “ServiciosUPM” y sin contraseña ya que la red es oculta y los equipos han sido dados de alta.

Se comprueba por el terminal Termite que no se le está asignando la dirección IP que nos han dicho que tiene asignada y falla la conexión.

```
00:00:05 CMD:
00:00:24 WIF: FallÃ³ ConexiÃ³n, timeout de AP
00:00:24 WIF: Conectando a AP1 ServiciosUPM en modo 11N como tasmotaSwitch6-0222...
```

Figura 46 Pantallazo de Termite que muestra el error de conexión de red a ServiciosUPM.

Nos ponemos de nuevo en contacto comunicando lo que está haciendo el sistema al conectarse a la red de ServiciosUPM. Hay un error de conexión y comprueban que es debido a que no se le asignaba correctamente la IP.

Finalmente se consigue conectar con los equipos a través de la Wifi UPM y podemos conectarlos desde cualquier zona de la ETSIT o en cualquier escuela de la UPM que tenga cobertura eduroam.

Después de tener todos los equipos preparados para poder conectar en cualquier zona con cobertura eduroam se procede a identificar esas áreas donde para poder apagar las luces ha de tener que ir un conserje o vigilante de seguridad. Así como las zonas donde creemos que pueden ser sitios donde por su lejanía puedan dejarse olvidadas. Identificamos junto al equipo de mantenimiento aquellos cuadros y zonas para poder integrarlas en nuestra aplicación.

Se identifican varias zonas para poder hacer el diseño en dos fases. Una primera fase que cubra las zonas de menor afluencia de gente, ya que si ocurriera algún problema el corte del

servicio tenga menor impacto y la segunda fase que englobe otras zonas donde el impacto de energía sea mayor.

Las zonas que se identifican son las que se muestran a continuación:

Edificio C (En estas zonas las plantas 2,3,4 son simétricas y nos permiten tener un control de potencia más exhaustivo al ser consumos muy similares)

➤ 4ªPlanta

Cuadro 4D

- Pasillo y Aseos.
- Pasillo.

Cuadro 4I

- Pasillo y Aseos.
- Pasillo.

➤ 3ªPlanta

Cuadro 4D

- Pasillo y Aseos.
- Pasillo.

Cuadro 4I

- Pasillo y Aseos.
- Pasillo.

➤ 2ªPlanta

Cuadro 4D

- Pasillo y Aseos.
- Pasillo.

Cuadro 4I

- Pasillo y Aseos.
- Pasillo.

Edificio A

1ª Planta Edif A

- Hall Aulas 133-141 (Alumbrado Hall y Pasillos)
- Pasillo 127
- Alumbrado Pasillo 127
- Hall Sala Juntas+Sala Grados
- Pasillos Salas 131-133
- Hall Pasillo Aula 120-125
- Servicios Aulas 120-125

Planta Baja Edificio A

- Hall Lab Físicas, Química y GICO.

L-4

- Alumbrado Servicios y Pasillo Izquierda.
- Alumbrado Hall Pasillo Derecho.

L-3

- Alumbrado Hall y Pasillo.

L-2

- Servicios.
- Pasillo Clubs Deportivo.

L-1

- Pasillo Aulas.
- Pasillo Delegación.

4.2 VALIDACIÓN DEL PILOTO

Para la instalación del piloto en la ETSIT se ha procedido a la realización de las siguientes fases:

Fase 1: (27 de enero 2020)

Durante la primera fase se ha procedido a la instalación de 19 sensores en dos emplazamientos:

- Plantas 2,3 y 4 Edificio C, cubriendo luminarias de pasillos y aseos
- Pasillos y Baño 1ª Planta Edificio A.

Se han instalado 11 equipos que actúan como un interruptor (on/off) y 8 equipos que además de actuar como los anteriores nos permiten monitorizar el consumo energético diario. Se han colocado 4 en una planta del edificio C ya que al ser simétrico nos permite tener una aproximación del consumo total de este edificio y el resto en el edificio A.

Inicialmente no se actúa sobre los sensores, monitorizando la actividad habitual diaria, así como el consumo energético. En general se observa que entre las 22:00 h aproximadamente y las 6:45 h los equipos se encuentran apagados, bien mediante interruptores (en pared), bien mediante diferenciales accesibles a través de los cuadros eléctricos.

El mayor impacto está en la planta 2 del edificio C ya que no tiene iluminación y las luces han de estar siempre encendidas.

Se observa que la mayoría de los cuadros eléctricos se encuentran abiertos y por tanto accesibles sin limitación alguna.

Al no disponer de muchos equipos que monitorizan el consumo, no se puede cuantificar cual es consumo total que hay. La aproximación que se puede hacer es puesto que el edificio C es simétrico multiplicar por el número de plantas con el consumo de una de las plantas que se está monitorizando. El principal aspecto que se resuelve es que en zonas como los aseos no siempre se apagan las luces, con la plataforma que se está integrando nos aseguramos de que estas luminarias estén apagadas en las horas donde no hay actividad. Además de observar las horas en las que encienden y apagan los interruptores de las luminarias observamos los siguientes consumos en la primera semana de medidas:

Potencia Consumida (kWh)	10/02/20	11/2/20	12/2/20	13/2/20	14/2/20	15/2/20	16/2/20
4ª Planta Pasillo 4I PowR2(9)	4.253	4.637	4.038	4.581	4.593	0	0
4ª Planta Pasillo y Aseo 4I Pow1	3.354	2.399	3.205	2.863	2.758	0	0
3ª Planta Pasillo lateral 3D PowR2(3)	4.274	4.917	4.430	4.272	4.678	0	0
3ª Planta Pasillo 3D PowR2(10)	4.233	4.326	4.369	4.220	4.302	0	0
Edificio A Cuadro 1.5 Pasillo 133-141 PowR2(7)	8.967	9.095	9.015	8.856	7.486	0	0
Edificio A Cuadro 1.3 Pasillo 127 PowR2(5)	4.346	3.250	3.174	3.266	3.508	4.029	5.282
Edificio A Cuadro Hall Pasillo Derecha Hall 120-122/128 PowR2(1)	7.381	7.760	7.458	7.756	7.648	0	0
Edificio A Cuadro Hall Pasillo Derecha Pasillo 123-125 PowR2(6)	5.069	5.383	5.390	5.324	5.280	0	0
Total(kWh)	41.877	41.767	41.079	41.138	40.253	4.029	5.282

Tabla 6 Medidas de potencia sin aplicar políticas de automatización.

Cabe destacar los siguientes aspectos:

- El mayor consumo viene dado por las luminarias del Pasillo de las aulas 133-141. Estas luces pensamos que en las horas que no hay clases pueden permanecer apagadas ya que no es un pasillo de tránsito si no hay clases.
- Durante el fin de semana en la gran mayoría de los sábados las luces del edificio C no se encienden. Si hemos observado que se han encendido las de la segunda planta ya que no tiene ventana y es necesario la luz. Los domingos permanecen todas apagadas.
- En el pasillo de las aulas 127 hay una cámara de seguridad y es necesario que al menos haya una luminaria encendida. Por esta razón se observa que en los fines de semana también hay consumo en este cuadro.
- El consumo durante los días de diarios se mantiene constante y en los fines de semana observamos que es únicamente el de las luces que han de permanecer encendidas por la cámara de vigilancia.

Una vez que se han recogido estos datos, se ha procedido a configurar Home Assistant para que tengan acceso el personal autorizado por la ETSIT para el control de las luminarias. De esta forma se ha creado un usuario para que puedan identificar los dispositivos en los edificios. En cuanto a las automatizaciones este usuario no puede crearlas desde ese acceso, así como no se puede ver las potencias y los consumos.

Para poder identificar las zonas donde se han instalados estos dispositivos se ha pintado sobre un plano el lugar de los dispositivos instalados. Se han diferenciado en dos vistas, una para el Edificio A y otra para el Edificio C.

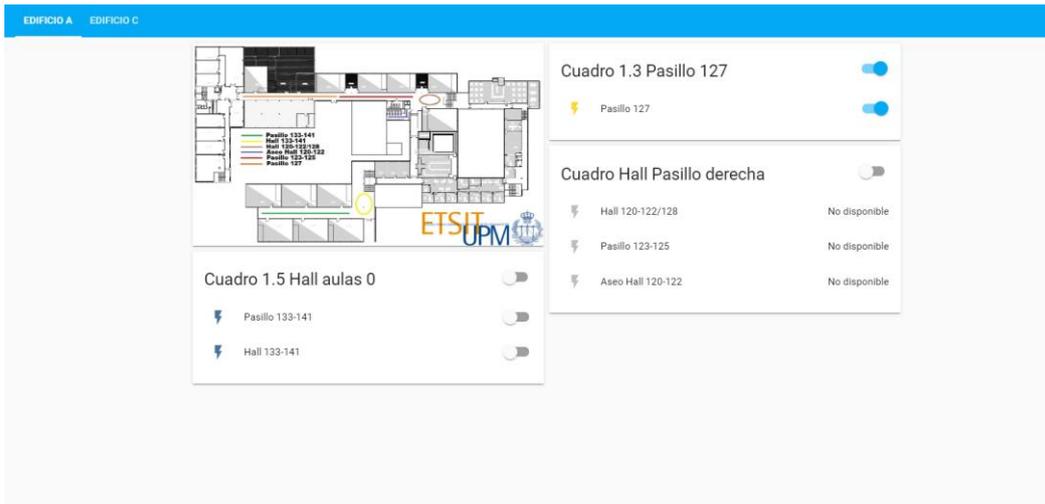


Figura 47 Edificio A en Home Assistant.

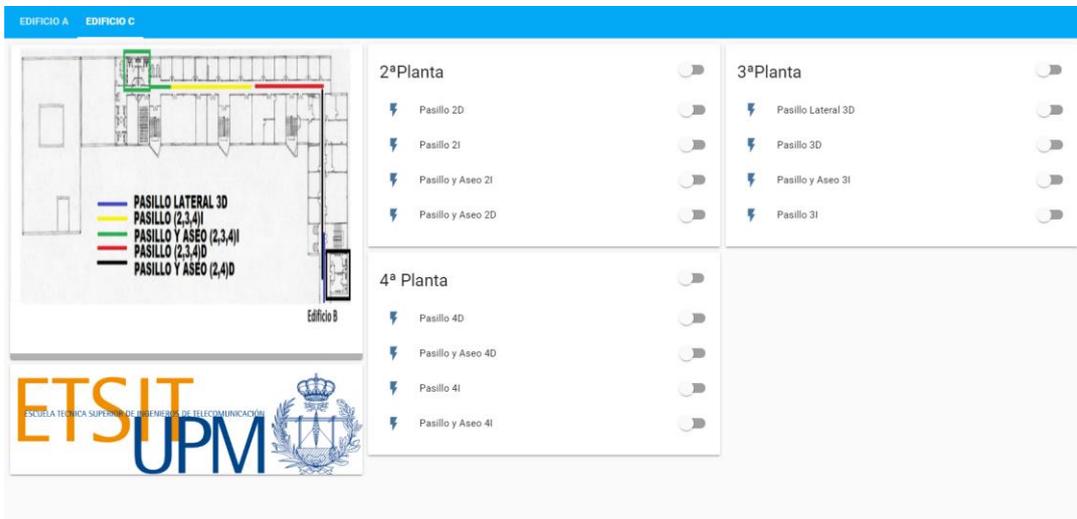


Figura 48 Edificio C en Home Assistant.

Fase 2: (Desde 19 febrero 2020)

Después de crear la plataforma con las vistas correspondientes y los dispositivos instalados se ha creado un manual de usuario de la plataforma Home Assistant que se puede ver en el Anexo E de este trabajo donde se explica paso a paso como acceder a la plataforma y como ver el estado de los dispositivos instalados.

Tras difundir el manual de uso a Conserjes y Vigilantes a través de la Subdirección de Infraestructuras de la ETSIT el miércoles 19/2 y tras hablar con el vigilante Nocturno se inicia una nueva fase en la que los vigilantes y conserjes de forma manual apagan y encienden las luces a través de la plataforma con control de acceso. Se observa que el primer día no se ha procedido a su encendido por lo que se actúa en consecuencia y se prepara un mecanismo automático de encendido que en caso de estar apagadas de L-V proceda a su encendido a las 6:45.

En la siguiente tabla se recogen las medidas de potencia de los dispositivos en esta segunda fase:

Potencia Consumida (kWh)	24/2/20	25/2/20	26/2/20	27/2/20	28/2/20	29/2/20	1/3/20
4ª Planta Pasillo 4I PowR2(9)	4.063	4.835	4.748	4.834	4.732	0	0
4ª Planta Pasillo y Aseo 4I Pow1	1.799	2.510	2.874	2.876	2.348	0	0
3ª Planta Pasillo lateral 3D PowR2(3)	4.281	5.213	5.105	5.196	5.108	0	0
3ª Planta Pasillo 3D PowR2(10)	4.414	5.111	4.768	5.102	5.005	0	0
Edificio A Cuadro 1.5 Pasillo 133-141 PowR2(7)	7.959	8.120	8.396	8.564	8.541	0	0
Edificio A Cuadro 1.3 Pasillo 127 PowR2(5)	4.016	3.871	2.983	4.392	3.245	4.182	4.370
Edificio A Cuadro Hall Pasillo Derecha Hall 120-122/128 PowR2(1)	7.705	8.001	7.516	7.924	7.755	0	0
Edificio A Cuadro Hall Pasillo Derecha Pasillo 123-125 PowR2(6)	5.638	5.641	5.173	5.455	5.638	0	0
Total(kWh)	39.875	43.302	41.563	44.343	42.372	4.182	4.370

Tabla 7 Medidas de potencia después de aplicar políticas de automatización.

Tras estas medidas observamos lo siguiente:

- Ha aumentado el consumo levemente en media. Antes de que se automatizará el vigilante hacía una ronda a eso de las 23:00 y apagaba las luces. Esto no siempre ocurría a la misma hora, algunas veces más tarde y otras más temprano, pero nunca de las 23:00. Ya que el estado en el que se apagan las luces es en el que se van a encender, es decir, si el interruptor de la luz estaba encendido cuando salta la automatización es así como se volverá a encender, de ahí es ese leve aumento de consumo. Pero por otro lado de esta manera garantizas que a la hora que realizas la automatización se van a apagar todas las luces, es posible que luces de aseos se dejaran encendidas ya que hay muchas y no siempre se puede llegar a todas de manera manual. En la siguiente imagen vemos como la automatización apaga todos los dispositivos, excepto en los cuadros del edificio A que el apagado lo realizan los conserjes de manera manual.

En la siguiente imagen se puede observar como es el comportamiento del consumo energético con las automatizaciones aplicadas. Por ejemplo, en el Pasillo 133-141 el consumo es constante de 600 W ya que durante todo el tiempo las luminarias están encendidas, es decir, se observa que cuando se aplica la automatización no hay consumo y a las 6:45 vuelven en el estado anterior. Esto es debido a que nadie ha ido al interruptor de pared a apagar las luces y se han apagado cuando se ha ejecutado la automatización. Ocurre exactamente lo mismo con el resto de los dispositivos. Más adelante se detallara en un apartado de recomendaciones que en las horas donde no hay actividad docente estas luces se pueden apagar, ya que en ese pasillo no hay transito si no hay clase.



Figura 49 Medidas de potencia de Home Assistant.

Otro aspecto importante que se ha observado durante esta segunda fase es el comportamiento de la red con los dispositivos conectados. En este aspecto se ha visto que durante un periodo de 30 minutos el equipo mantenía una conexión TCP estable con el servidor que tenemos en la ETSIT. Pasado este tiempo el dispositivo se desconectaba durante unos 20-30 seg sin tener control sobre el mismo y que en un futuro podría suponer un problema a la hora de lanzar una automatización si el dispositivo en ese periodo estaba desconectado.

Este comportamiento únicamente nos sucede cuando lo conectamos a Servicios UPM, lo que hace pensar que es algo de configuración/disponibilidad de la red. Una vez que detectamos este fallo realizamos las siguientes pruebas:

- Lo primero que hemos hecho es descartar el problema del DHCP poniendo a uno de los sensores dirección IP fija –la asignada al mismo- y conectarlo a la red de Servicios UPM. El comportamiento se mantiene, es decir cada 30 minutos pasa a estar no disponible.
- A continuación, hemos conectado el sensor desde la red Servicios_UPM a una wifi de pruebas que tenemos en el laboratorio (conectada a la red del DIT) comprobando que ese comportamiento nunca sucede, el sensor nunca se desconecta.
- Posteriormente por si era problema de la entrada a la red del DIT (firewall,..) hemos conectado el sensor desde casa accediendo por internet al servidor. Tampoco ha generado dicho comportamiento. El sensor ha estado siempre disponible.
- Finalmente, otro aspecto que repasamos es los servidores NTP a los que se conecta el sensor por defecto. Fijamos manualmente los servidores de la UPM, pero el problema sigue ocurriendo.

En las siguientes imágenes vemos cómo se comportan los dispositivos según lo que se ha descrito.

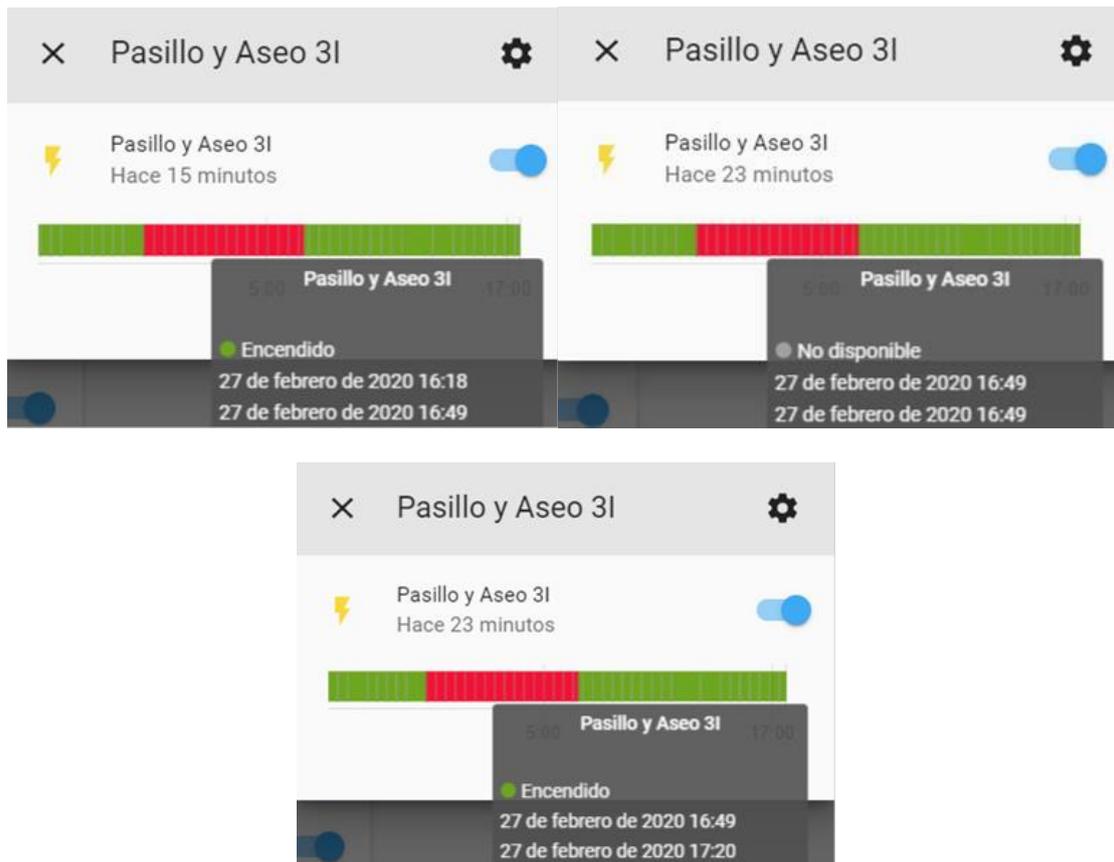


Figura 50 Problema de conectividad en los dispositivos.

Una vez realizadas todas las pruebas posibles por nuestra parte ponemos en conocimiento del rectorado el problema. Pasado unos días vemos que el dispositivo se comporta como debería y el problema ha sido resuelto. El rectorado actualiza uno de los parámetros de los routers y el comportamiento es el adecuado.

Durante esta nueva fase se analiza el despliegue realizado y la posibilidad de cubrir la 2ª, 3ª y 4ª planta del edificio B. Estas plantas mantienen una estructura simétrica y sus luminarias son controladas mediante 4 diferenciales, dos para las luces de Hall y 2 para luces de pasillo. En todos los casos, una de las fases de las luces de Hall se encuentra apagada. Además, las luces de pasillo están distribuidas entre los dos diferenciales de forma alternada, de modo que en caso de apagar una de las fases, se mantendría iluminado todo el pasillo, pero con menor intensidad. De este modo, con la instalación de 9 sensores podría controlarse y actuarse sobre dicha infraestructura eléctrica.

En este sentido se recomienda la instalación de estos 9 sensores, los cuadros eléctricos permiten su integración sin tener problemas de capacidad, y la aplicación de políticas de eficiencia energética como las siguientes:

- Control Manual (Encendido/Apagado) por parte de personal autorizado (Conserjes/Vigilantes).
- Automatización de encendido y apagado automático en horas/días habituales (L-V: OFF: 23:00-6:45, S: ON:8-14, D: OFF 0-24), coincidiendo con resto de instalación
- Racionalización de consumo en horas valle (Ej a partir de las 19:30) eliminando una de las fases de luminarias de Pasillo.

4.3 EVALUACIÓN DE LA PLATAFORMA

Después de realizar el despliegue en un entorno de producción, hemos observado que la plataforma creada ha logrado el objetivo que se planteaba al principio del trabajo. Se ha conseguido gracias a la monitorización de los equipos que se han instalados ver el comportamiento que hay sobre ellos sin aplicar ningún tipo de política, gracias a esto hemos observado en que aspectos se puede mejorar para ser más eficientes.

Las ventajas que ofrece la plataforma son las siguientes

- Integrar cualquier dispositivo sin necesidad de depender de software de terceros. Este aspecto es muy importante, ya que uno de los grandes problemas que se tuvo al inicio del trabajo es que cuando se compra un dispositivo de IoT trae un firmware y una aplicación instalada por defecto. Debido a esto, cuando realizamos la conexión con los dispositivos enviamos los datos a un servidor de la empresa a la que hemos comprado los mismo. Gracias a la plataforma que hemos creado todos estos datos se almacenan en nuestras Raspberry Pi y es más complejo que pueda sufrir cualquier vulnerabilidad.
- Se han aplicado políticas para mejorar la eficiencia energética de la escuela. Gracias a los datos que se han recogido en la plataforma, se han analizado los consumo durante las horas de actividad en la escuela. Una vez que se han analizado los datos, se ha visto que aplicando reglas de automatización se garantiza el apagado de las luces.
- Monitorizar la potencia en tiempo real nos permite ver cualquier circunstancia que ocurra en las zonas donde hay dispositivos instalados, es decir, si ocurriera alguna incidencia de corte de suministro se puede detectar en tiempo real.
- Debido a que se han integrado los dispositivos de IoT a la red de Servicios UPM es posible conectarlos en cualquier campus de la UPM que tenga conexión Wifi UPM.
- La plataforma que se ha diseñado permite un control de los dispositivos muy intuitivo y que permite a cualquier persona autorizada tener controlado todos los dispositivos sin la necesidad de desplazarse para apagarlos o encenderlos.

Durante el desarrollo de la plataforma se han tenido algunos problemas que se detallan a continuación:

- Por motivos del Covid-19 no se ha podido seguir realizando un mayor despliegue de dispositivos por la escuela. El día 11 de marzo de 2020 se declara la suspensión de toda actividad en las universidades de Madrid, los dispositivos se habían preparado para realizar otro despliegue en las zonas que se plantearon al inicio de este capítulo y no se pudieron llegar a instalar. Sin embargo, gracias a la plataforma durante este periodo de inactividad se ha garantizado el apagado de las luminarias. Se ha procedido a desactivar la automatización de encendido y comprobado durante el periodo el apagado de las mismas.
- No se puede cuantificar el consumo energético debido a que se disponía de un número menor de dispositivos que midan potencia. La gran ventaja de la plataforma es que, gracias a la monitorización de potencia, se puede actuar en consecuencia. La propuesta es que estos dispositivos son muy fáciles de cambiar de un sitio a otro. Por tanto, a pesar de no disponer de todos los dispositivos para medir potencia, realizar un periodo de medida y después instalar el Sonoff Basic que es más económico sabiendo como actuar tras haber analizado el consumo.
- Si hay un corte de luz se han de volver a flashear los dispositivos. Durante el periodo en que la escuela ha permanecido cerrada debido al Covid-19, hubo un corte de luz y cuando vuelve a reestablecerse el servicio se produce una fuerte subida de tensión lo

que hizo que los dispositivos perdieran la información y se tuvieron que volver a realizar el flasheo.

4.4 RECOMENDACIONES

En general y tras los primeros días de uso posteriores a las pruebas exhaustivas de laboratorio se considera que la plataforma puede prestar un importante beneficio a la Escuela en los siguientes aspectos principales:

- Como herramienta de control y establecimiento de políticas energéticas desde la dirección afectando a zonas comunes (luminarias, baños) que hasta ahora no son cubiertas por soluciones alternativas (ej: sensores de presencia). Si bien la plataforma de control y monitorización admite múltiples mecanismos de actuación, se considera que la automatización de horarios habituales junto con la actuación manual por parte de personal autorizado (vigilantes y conserjes) forma un buen compromiso entre control y automatización.
- Se considera que la plataforma puede contribuir de forma directa a evitar consumos innecesarios en el caso de equipos HVCA (Heating, Ventilation, and Air-Conditioning). En la escuela existen equipos controlados mediante relojes, que cortan su uso entre las 22:30h y las 8h de forma centralizada. Sin embargo, existen muchos equipos ubicados en despachos que no disponen de esta funcionalidad. Mediante la instalación sencilla de estos equipos con un coste de unos 15€ podría aplicarse una política similar y evitar el consumo energético innecesario aplicando las mismas políticas que en el caso actual de los equipos controlados.
- Durante el periodo de evaluación de la plataforma debido al confinamiento tuvimos un problema con la Raspberry Pi que dejó de funcionar. Después de evaluarlo se detecta que no podía arrancar debido a que la tarjeta de memoria estaba dañada como consecuencia de las altas temperaturas. En consecuencia, lo que se recomienda es tener un Backup de la imagen del sistema de la Raspberry Pi en otra tarjeta de memoria y en cualquier momento realizar el cambio si fuera necesario.
- En caso de no disponer de los equipos suficientes para la monitorización de la energía, se recomienda instalar un equipo medidor de potencia (Sonoff Pow) durante un determinado periodo de tiempo para ver el comportamiento y después reemplazar por el interruptor básico (Sonoff Switch Basic). De esta forma se estudia el consumo y comportamiento para después aplicar medidas para mejorar la eficiencia en esos sitios.
- Dado que los dispositivos se han integrado en la red de Servicios UPM tiene la gran ventaja de que la infraestructura es muy estable, pero requiere de un tiempo para darlos de alta previo a su utilización. Por ello se recomienda tener dispositivos dados de alta para su sustitución en caso de que algún equipo fallara como, por ejemplo, por una subida de tensión y así el reemplazo sería inmediato. Durante el desarrollo de este trabajo se ha observado, que una vez que se ha ido la luz de la escuela, al volver el servicio algunos de los equipos tuvieron que ser desconectados y reflasheados.

Finalmente debe considerarse la necesidad de limitar el acceso a los cuadros eléctricos por motivos de seguridad. En este sentido el uso de estos equipos puede eliminar la necesidad del acceso en el funcionamiento habitual de la Escuela por parte de (Conserjes y bedeles) restringiendo su uso a personal especializado (Mantenimiento).

5. CONCLUSIONES

5.1 CONCLUSIONES

La necesidad de frenar el cambio climático y mejorar la eficiencia energética de los sistemas actuales nos hace afirmar que gracias a las tecnologías actuales de IoT, se puede realizar un sistema domótico para aplicar políticas que nos ayuden a ser más eficientes.

La gestión de determinados elementos del día a día, sin necesidad de tener que realizar operaciones sobre ellos de forma manual, nos hacen ser más eficientes. Así como el poder obtener datos del comportamiento de los dispositivos serán las claves para aplicar automatizaciones que nos ayuden en el día a día.

Durante el desarrollo del trabajo, se ha cerrado la escuela por el Covid-19. Gracias al uso de esta aplicación se ha podido mantener apagada todas las luminarias implementadas sin necesidad de que alguien haya tenido que ir revisando todos los lugares.

Gracias a la comunidad de usuarios de Home Assistant y a su amplio abanico de posibilidades para la personalización del sistema podemos crear una plataforma para implementar los dispositivos de IoT sin depender de software de terceros.

En la Universidad Politécnica de Madrid se trabaja para fomentar este tipo de trabajos para mejorar las infraestructuras y poder lograr los ODS. Gracias al proyecto RES2+U este trabajo ha sido posible y servirá de ayuda para poder implementar dichos sistemas en el resto de Escuelas de la UPM.

5.2 TRABAJOS FUTUROS

Como trabajo futuro relacionado con este proyecto, se proponen las siguientes líneas de acción:

- Despliegue de dispositivos que no se han podido instalar como consecuencia del Covid-19. En el desarrollo de este trabajo se han instalado un total de 19 dispositivos y se tenía pensado la instalación de otros dispositivos en otras zonas del edificio A.
- Ampliar los dispositivos al edificio B de la ETSIT. En este sentido se pueden establecer políticas para determinar tener encendidas un número de luminarias en una franja horaria y otro más reducido en las últimas horas del horario de la escuela. La diferencia entre el edificio C es que cuando apagas una luminaria están colocadas en serie y se apagaría los 20 o 25 metros de pasillo.
- Implementar políticas en función de otros factores como pueden ser las horas de Sol, climatología, temperatura, etc.
- Extender las funcionalidades de la plataforma para funcionalidades de otros equipos en la plataforma, como pueden ser cámaras de videovigilancia, sensores de movimientos, etc. Home Assistant te permite integrar otros equipos, aunque sean diferentes, sería añadir los equipos que le permitan al conserje realizar todas sus funciones desde una misma plataforma.
- Divulgar los resultados obtenidos para el resto de las Escuelas de la UPM. Este trabajo ha obtenido una beca y uno de los compromisos que se ha adquirido es la presentación del trabajo en octubre en una reunión donde se presentarán todos los TFM becados en un entorno abierto donde participan todas las escuelas. El año pasado se realizó una exposición para presentar la idea del trabajo en este entorno.

6. BIBLIOGRAFÍA

- [1] «Portal web de la ONU,» [En línea]. Available: <https://www.unenvironment.org/es/node/1411>. [Último acceso: 25 04 2020].
- [2] «Hacia un campus sostenible: El Proyecto RES2+U de la UPM,» [En línea]. Available: <https://www.iit.comillas.edu/docs/IIT-18-162A.pdf>. [Último acceso: 25 04 2020].
- [3] «Informe Digital 2019,» [En línea]. Available: <https://www.orientamartamouliaa.es/wp-content/uploads/2019/03/Informe-Digital-2019-Espa%C3%B1a-Hootsuite-2019.pdf>. [Último acceso: 25 04 2020].
- [4] «La importancia del IoT para las Smart Cities,» [En línea]. Available: <https://iurban.es/la-importancia-del-iot-para-las-smart-cities/>. [Último acceso: 09 06 2020].
- [5] «Principales estándares de comunicación en IoT,» [En línea]. Available: <https://www.ubuntupit.com/top-15-standard-iot-protocols-that-you-must-know-about/>. [Último acceso: 24 04 2020].
- [6] «Tecnología Sigfox,» [En línea]. Available: <http://www.dset-energy.com/2019/06/05/tecnologia-sigfox/>. [Último acceso: 25 04 2020].
- [7] «Internet de las cosas MQTT Broker,» [En línea]. Available: <https://www.internetdelascosas.cl/2019/09/25/raspberry-pi-mqtt-broker/>. [Último acceso: 27 04 2020].
- [8] «Principales broker MQTT Open Source,» [En línea]. Available: <https://www.luisllamas.es/principales-broker-mqtt-open-source-para-proyectos-iot/>. [Último acceso: 26 04 2020].
- [9] «Protocolos de comunicación de IoT,» [En línea]. Available: <https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones>. [Último acceso: 24 04 2020].
- [10] «Página de Sonoff,» [En línea]. Available: <https://sonoff.tech/>. [Último acceso: 02 06 2020].
- [11] «Comparativa para dispositivos ESP8266,» [En línea]. Available: <https://www.alferez.es/iot/espurna-tasmota-o-espeasy-para-dispositivos-esp8266/>. [Último acceso: 28 05 2020].
- [12] «Página de GitHub de Tasmota,» [En línea]. Available: <https://github.com/arendst/Tasmota/>. [Último acceso: 31 05 2020].
- [13] «Página de GitHub de la herramienta Pyflasher,» [En línea]. Available: <https://github.com/tasmota/tasmota-pyflasher>. [Último acceso: 05 06 2020].
- [14] «Página de GitHub de Esptool,» [En línea]. Available: <https://github.com/espressif/esptool>. [Último acceso: 05 06 2020].
- [15] «Página de descarga de Termite,» [En línea]. Available: https://www.compuphase.com/software_termite.htm. [Último acceso: 05 06 2020].
- [16] «Página de Home Assistant para descargar la Imagen de Hassio,» [En línea]. Available: <https://www.home-assistant.io/hassio/installation/>. [Último acceso: 08 06 2020].
- [17] «Página de BalenaEtcher para formatear tarjetas SD,» [En línea]. Available: <https://www.balena.io/etcher/>. [Último acceso: 08 06 2020].

- [18] «Página de GitHub de HACS,» [En línea]. Available:
<https://github.com/hacs/integration/releases/tag/1.0.3>. [Último acceso: 08 06 2020].
- [19] «Página de GitHub para generar un Token,» [En línea]. Available:
<https://github.com/settings/tokens> . [Último acceso: 02 06 2020].

ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

A.1 INTRODUCCIÓN

Este proyecto nace por la necesidad de recopilar información sobre el uso del consumo energético en la Universidad y a través del análisis de estos, intentar aplicar políticas para mejorar la eficiencia energética de las infraestructuras. Por ello, el objetivo de este estudio es diseñar, desarrollar e implementar una plataforma que sirva para el control y monitorización de infraestructuras energéticas dentro de los campus universitarios de la UPM, utilizando sensores de la marca Sonoff, de bajo coste e indicados para medir potencia, y actuadores que se utilizarán como un switch. Para ello, se ha realizado un despliegue inicial en laboratorio configurando estos dispositivos y preparándolos de cara a una su posterior instalación y análisis de datos en las infraestructuras.

A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO

El trabajo se ha desarrollado inicialmente dentro de la ETSIT para el control y monitorización de los alumbrados en los edificios A y C. Se han tomado datos sobre el consumo energético para después tomar decisiones sobre ellos en forma de políticas que permitan una mayor eficiencia. Estas políticas se basan en aplicar automatizaciones en los dispositivos instalados y evitar que en horas donde no haya actividad dentro de la universidad, se produzca un mal uso de la energía eléctrica.

A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS

El principal impacto que tiene este trabajo es sobre el consumo energético. Gracias a las medidas que se han tomado aseguramos que en horas donde no hay actividad los alumbrados están apagados. Además, el trabajo sirve de ayuda al personal de la ETSIT para realizar su actividad. El objetivo de este proyecto es que sirva a otras escuelas para poder implementar medidas que ayuden al campus universitario para mejorar sus instalaciones. Debido a que muchas escuelas no disponen de sensores de movimiento y haya lugares dentro de estas que sean de difícil acceso para comprobar si el alumbrado está encendido o apagado, con un bajo coste podremos monitorizar estos sitios y en horas donde no haya actividad hacer que se apague el alumbrado.

A.4 CONCLUSIONES

Finalmente, se ha conseguido con este trabajo hacer que la universidad sea un espacio más responsable y eficiente. Se plantean una serie de trabajos futuros para poder seguir trabajando en estos aspectos tan importante que determinará nuestro futuro y así contribuir en frenar las consecuencias del cambio climático.

ANEXO B: PRESUPUESTO ECONÓMICO

COSTE DE MANO DE OBRA (coste directo)

Horas	Precio/hora	Total
810	18 €	14.580 €

COSTE DE RECURSOS MATERIALES (coste directo)

	Unidades	Precio de compra	Uso en meses	Amortización (en años)	Total
Ordenador portatil	1	874,90 €	6	5	145,82 €

COSTE TOTAL DE RECURSOS MATERIALES

145,82 €

GASTOS GENERALES (costes indirectos)	15%	sobre CD	2.208,87 €
BENEFICIO INDUSTRIAL	6%	sobre CD+CI	1.016,08 €

MATERIAL FUNGIBLE

Raspberry Pi 4 Model B + Carcasa+ Disipadores	2	80,74 €	161,48 €
Tarjeta de Memoria 32 GB Categoría 10	4	12,95 €	51,80 €
FTDI FT232RL USB	1	5,99 €	5,99 €
Jumper Wire Cables de Puente	1	9,99 €	9,99 €
Tiras 40 Pin Macho	1	6,99 €	6,99 €
Soldador JBC + Estaño	1	46,05 €	46,05 €
Switch Smart Sonoff	10	9,52 €	95,20 €
Sonoff Pow	5	10,90 €	54,50 €
Sonoff Pow R2	10	12,41 €	124,10 €
Sonoff TH16 + Sensor	5	9,48 €	47,40 €
Sonoff 4CH	4	16,16 €	64,64 €
Sonoff S20	5	13,09 €	65,45 €

SUBTOTAL PRESUPUESTO		18.684,36 €
IVA APLICABLE	21%	3.923,72 €
TOTAL PRESUPUESTO		22.608,08 €

ANEXO C: GUÍA PARA CAMBIAR EL FIRMWARE A TASMOTA EN DISPOSITIVOS SONOFF

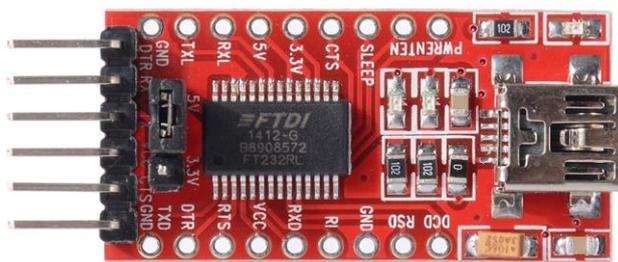
C.1 INTRODUCCIÓN Y OBJETIVOS

En la siguiente guía se va a especificar como cambiar el firmware de un dispositivo Sonoff a Tasmota. La ventaja de Tasmota es que nos ofrece todos los dispositivos integrados en una misma imagen y es seleccionable una vez accedes el dispositivo sobre el que está instalado. El objetivo principal es poder configurar nuestros dispositivos sin depender de software de terceros y disponer de más control sobre el dispositivo.

C.2 HARDWARE NECESARIO

Los elementos físicos que se necesitan para poder flashear los dispositivos son:

- **FTDI FT232RL USB:** se trata de una controladora que nos permite la transferencia de datos en serie de un solo chip. De esta forma nos podemos comunicar con los dispositivos Sonoff que no poseen una interfaz USB-Serial integrada en el dispositivo. Puede funcionar con potencia de 3.3V o 5V, en nuestro caso debemos colocar el jumper sobre el FTDI para trabajar en 3.3V.



- **Soldador + Estaño + Pin:** se necesitarán soldar a la placa base de los Sonoff los pines para poder conectarlos a la controladora y mandarle la información. Se necesitan 4 pines macho-macho para cada placa base que modifiquemos.



- Cables para conexiones tipo Dupont hembra-hembra, para la conexión entre el FTDI y el dispositivo Sonoff que queremos modificar.



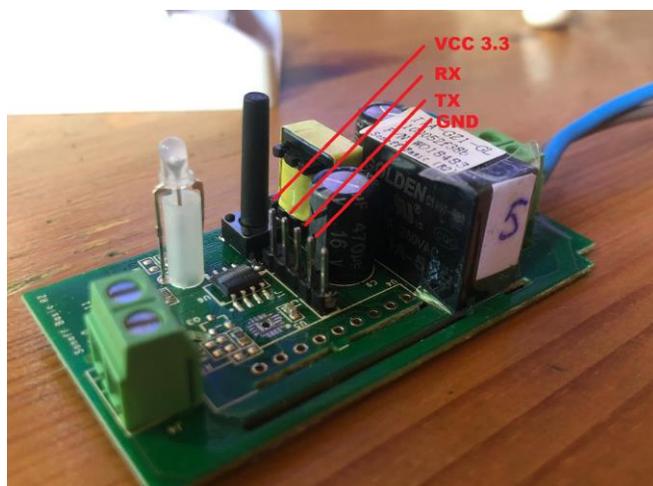
- Cable USB tipo B mini para conectar el FTDI con el ordenador desde donde le mandaremos los datos a los dispositivos.



C.3 CONSIDERACIONES PREVIAS

Una vez que tenemos preparados todos los elementos físicos para poder modificar el firmware comenzamos con el proceso para poder realizar este cambio. De todos los Sonoff que se han descrito anteriormente se pueden modificar más fácilmente al traer los agujeros para los pines en su placa base los siguientes dispositivos, se ha adjuntado una imagen en la que se puede ver en los dos primeros la placa base con los pines ya soldados, así como con que se corresponde cada puerto de esos pines y en las dos siguientes las placas base sin realizar ningún tipo de soldadura:

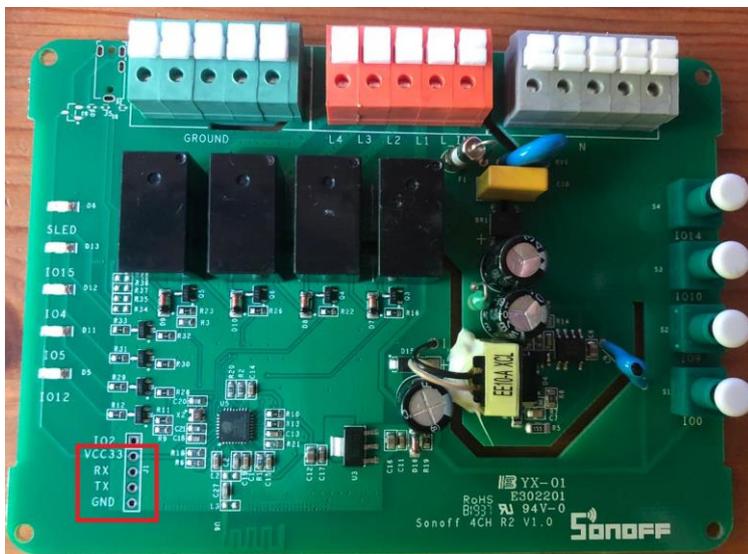
- **Sonoff Smart Switch**



- Sonoff POW R2



- Sonoff 4CH



- Sonoff S20

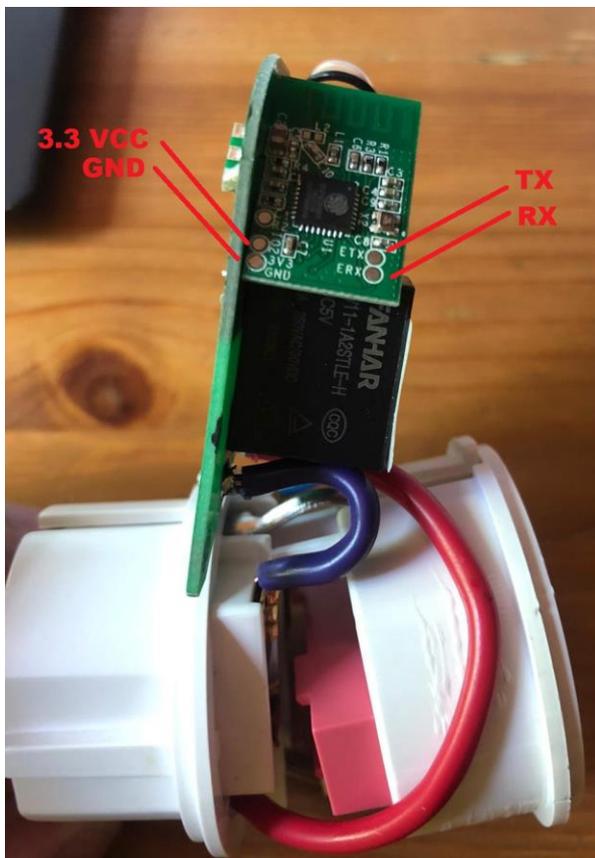


El resto de dispositivos también se pueden modificar pero a la hora de realizar la soldadura hay que tener más experiencia realizando este tipo de soldaduras.

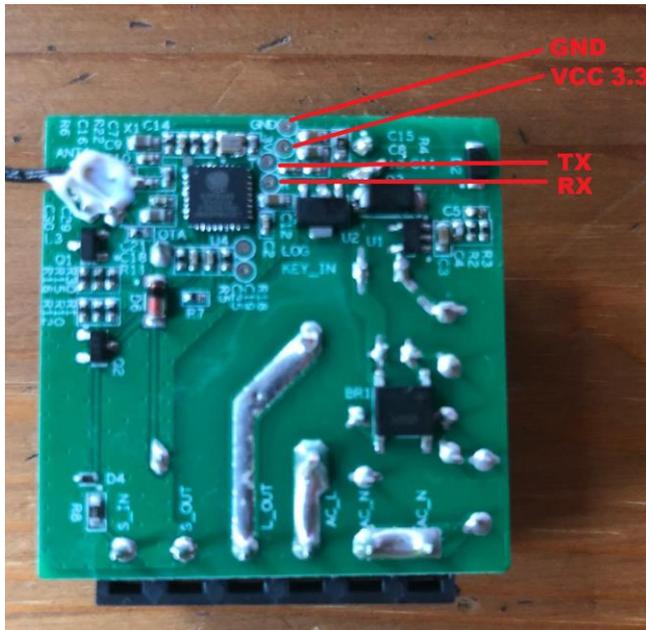
- **DIY Smart Switch:** es el más “fácil” de los otros tres dispositivos ya que trae los agujeros correspondientes para realizar la soldadura, pero el espacio para realizarla es muy pequeño y al tener una placa debajo hay que tener mejor manejo realizando las soldaduras en espacios tan reducidos.



- **Sonoff S26:** al tratarse de un enchufe más pequeño que el S20 no trae incorporados los agujeros para realizar las soldaduras. Por tanto hay que soldar un cable que se conecte directamente a la placa en los siguientes puntos:



- **DIY Mini:** se trata del dispositivo más pequeño de todos, por tanto es el más complejo a la hora de realizar las soldaduras en la placa base. Se deben de conectar cables en los siguientes puntos:



C.4 INSTALACIÓN TASMOTA

Después de haber explicado donde están los punto de soldadura en cada placa base de los Sonoff, los pasos que hay que seguir son los siguientes:

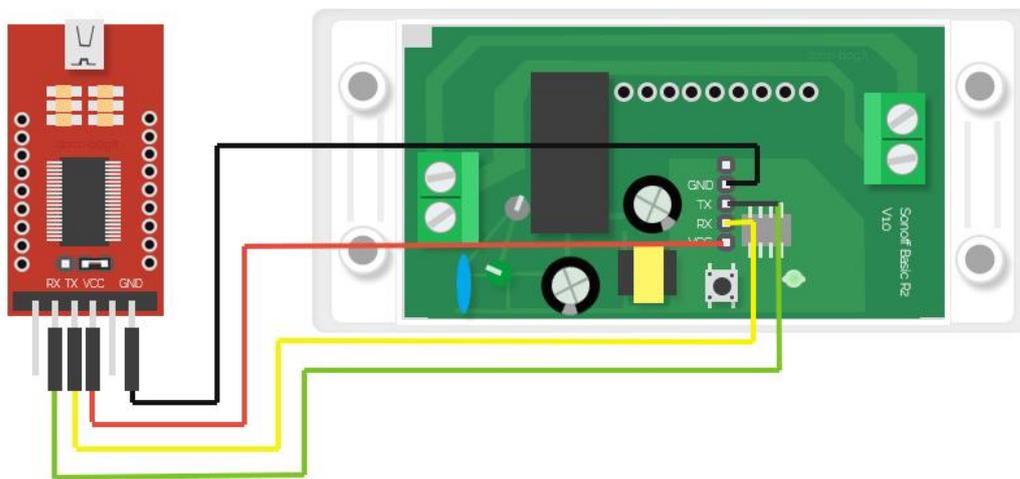
1) Soldar los pines a la placa base del dispositivo. Después de haber identificado los puntos de 3.3V, RX, TX y GND hay que soldar un PIN si hay un orificio en la placa para ello o un cable para los que traigan el punto de soldadura. Dependiendo del dispositivo puede aparecer indicado o no a que se corresponde cada punto u orificio. También destacar que es diferente para cada tipo de dispositivo la posición donde se encuentran. En el FTDI los puntos a considerar son los siguientes:



Lo más importante es que la conexión entre el FTDI y el Sonoff debe de quedar de la siguiente forma:

Pin FTDI	Pin ESP8266
3,3v	3,3 v
Tx	Rx
Rx	Tx
Gnd	Gnd

En la siguiente imagen se puede ver como quedaría la interconexión entre la placa de un Sonoff Smart Switch y el FTDI. Lo más importante es conectar el TX del FTDI con el RX del Sonoff y el RX del FTDI con el TX del Sonoff. Además se puede apreciar como hay que colocar el jumper en el FTDI para trabajar a 3.3V.



2) Conectar el dispositivo con el ordenador con el cable USB. Es muy importante que antes de conectar el USB al ordenador hay que mantener presionado el GPIO0 que es el switch que trae el sonoff para realizar el cambio de estado ON/OFF de forma manual en caso de que ocurra algún problema de conexión con él.



En el caso del Sonoff 4CH el GPIO se corresponde con el botón L1, ya que al tener 4 canales podría llegar a cierta confusión. En el Sonoff Mini DIY está situado sobre el siguiente punto:



3) Realizar el flasheo. Para realizar el cambio de firmware necesitaremos irnos a la página de Github de Tamota [12] y descargarnos la última versión del archivo binario de Tasmota.bin que queremos mandar a nuestro Sonoff. Para realizar el Flasheo necesitaremos utilizar una herramienta software para proceder de la forma más sencilla. Existen varias y son muy intuitivas, yo he utilizado las dos siguientes y el resultado ha sido satisfactorio en ambos casos, solo es necesario instalar una de las dos.

- Tasmota PyFlasher: herramienta de flasheo diseñada para tasmota. Es muy sencilla de utilizar e instalar a través del ejecutable .exe que encontraras en la página de GitHub. Es la que recomiendo para realizar el flasheo por primera vez [13]
- Esptool.py: esta herramienta al igual que la anterior es muy sencilla de utilizar, pero su instalación es algo más compleja ya que requiere tener instalado previamente una versión de Python 2.7 o 3.4 [14]

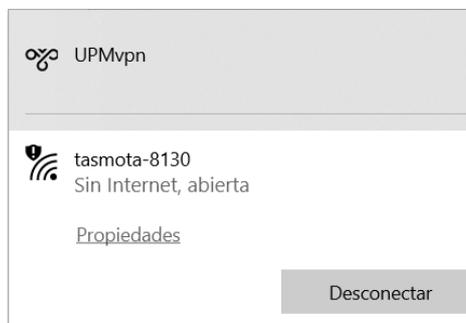
Una vez que hemos instalado alguna de estas herramientas, hay que realizar el flasheo. Para ello hay que seleccionar el puerto serie donde hemos conectado previamente el Sonoff presionando el botón de Switch, seleccionar la imagen que hemos descargado de la última versión de Tasmota y realizar el Flash hasta que se completa el 100%. Durante este proceso el software lo que hace es eliminar el firmware antiguo y guardar la versión de Tasmota que hemos descargado.

A screenshot of the Tasmota PyFlasher application window. The 'Serial port' dropdown is set to 'COM3'. The 'Tasmota firmware' field contains the path 'D:\TFM\ESPEasy_mega-20191208\bin\tasmota-ES.bin'. The 'Erase flash' option is set to 'yes, wipe all data!!!'. The console output shows the command 'esptool.py --port COM3 --baud 115200 --after_no_reset write...' and the progress of the flashing process, including 'Writing at 0x00014000... (24 %)'.

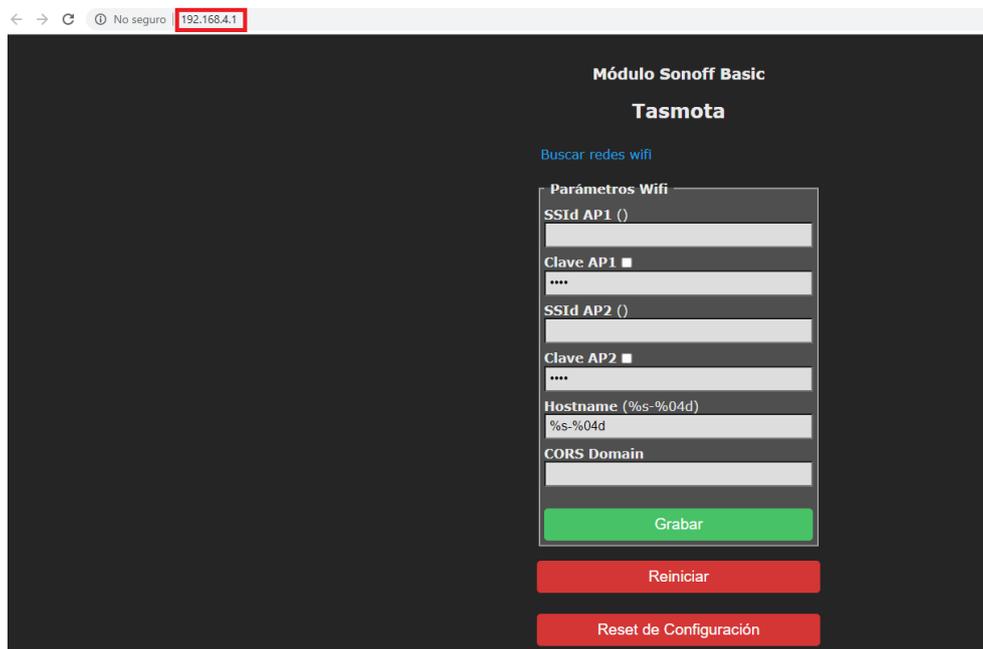
Una vez que se ha completado hay que desconectar el dispositivo y volverlo a conectar sin presionar nada.

4) Introducir los parámetros de la red Wifi que queremos conectar al dispositivo. Para ello hay dos formas de hacerlo.

- Tasmota ofrece un punto de acceso inalámbrico de configuración de Wifi una vez que se ha flasheado y conectado al ordenador o fuente de alimentación. Por tanto, una vez conectado a la alimentación hay que buscar un AP llamado tasmota-XXXX y conectarse.



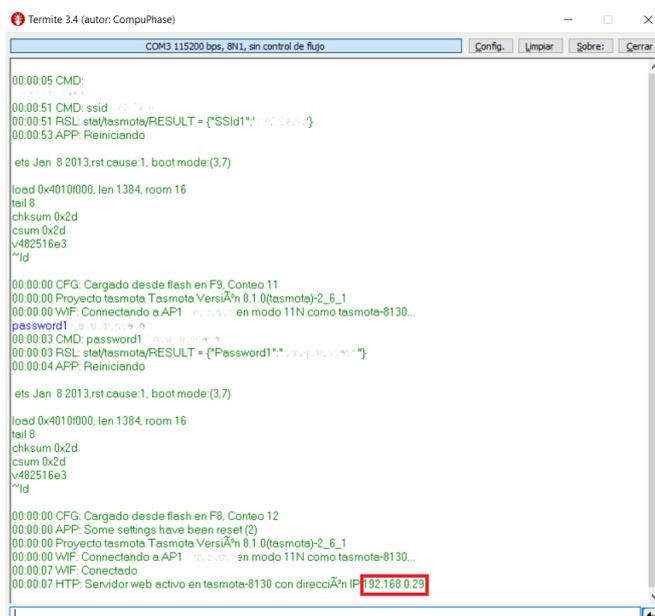
Después de que se haya conectado al AP lo normal es que nos redirija a una página para poder configurar los parámetros que deseamos. En caso de no se haya redirigido podemos abrir en el navegador esta página accediendo a la dirección <http://192.168.4.1>. En esta página se puede hacer que Tasmota busque redes disponibles o ingresar manualmente los parámetros de SSID y Password de la red Wifi. También es recomendable introducir otra red Wifi alternativa para que se conecte en caso de problemas de conexión con la que se ha asignado en primer lugar.



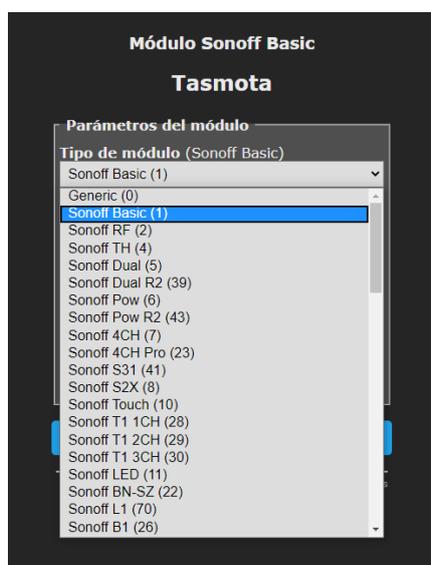
- La otra opción y la más rápida es la de utilizar un Terminal Serie. En mi caso he utilizado Termite [15], desde su página Web te puedes descargar el ejecutable y es muy fácil de instalar. Con esta herramienta software podremos conectarnos al dispositivo a través del USB que utilizamos para realizar el Flasheo utilizando comandos para enviarle los parámetros de la red wifi de la siguiente forma:

```
ssid1 <yourssid>; password1 <your_password>; ssid2
<your_ssid2>; password2 <your_password>
```

El dispositivo se reiniciará y se conectará a la red Wifi que le hemos asignado. Aunque el otro método es mucho más intuitivo por la interfaz grafica que nos ofrece, una de las ventajas que tenemos al utilizar Termite es que en su consola podremos ver la dirección IP que se le ha asignado al módulo Sonoff. Con el otro método habría que meterse en el Router y ver la dirección IP que se le ha asignado al dispositivo. Es importante saber esta dirección IP que se le ha asignado ya que para poder modificar cualquier parámetro de Tasmota y tener más control sobre el dispositivo hay que acceder de la forma <http://X.X.X.X> donde tendremos que introducir la dirección IP asignada.



5) Por último, hay que seleccionar el tipo de módulo Sonoff que tenemos. Para ello en la página de inicio de Tasmota hay que acceder a **Configuración- Configuración del Módulo** y seleccionar el dispositivo que estamos configurando.



ANEXO D. GUÍA DE INSTALACIÓN DE HOME ASSISTANT Y SERVIDOR MQTT MOSQUITTO

D.1 HARDWARE NECESARIO

El hardware necesario para la instalación de Home Assistant con un servidor de MQTT es el siguiente:

-Raspberry Pi (Se recomienda Model 3 o Model 4)



-Tarjeta Micro SD de al menos 32 GB y categoría 10



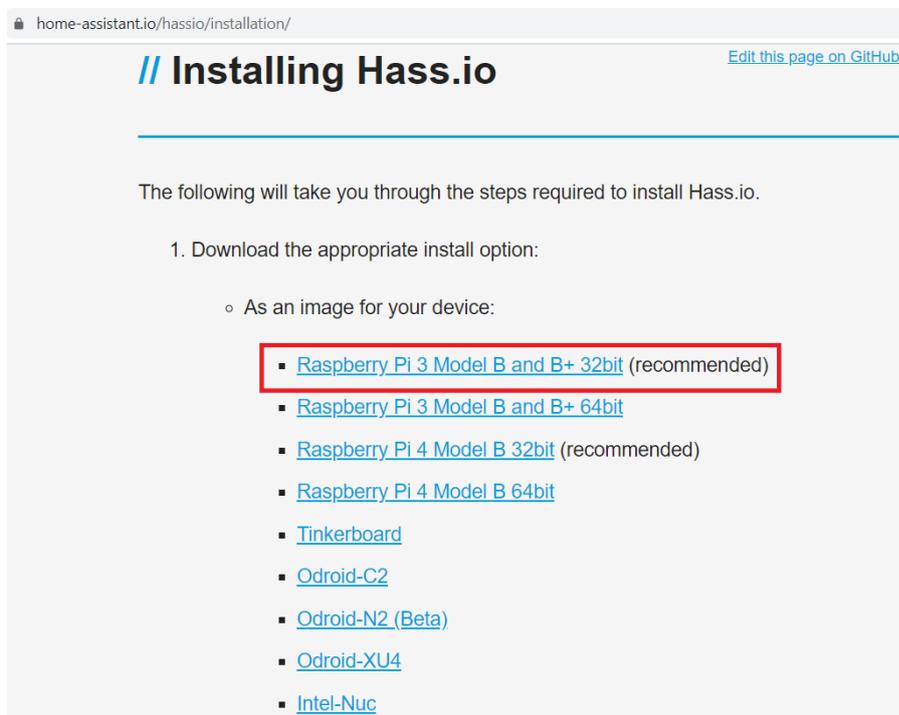
D.2 SOFTWARE NECESARIO

Programa para formatear Micro SD.

Se recomienda Balenaetcher para escribir la imagen en la SD. [17]

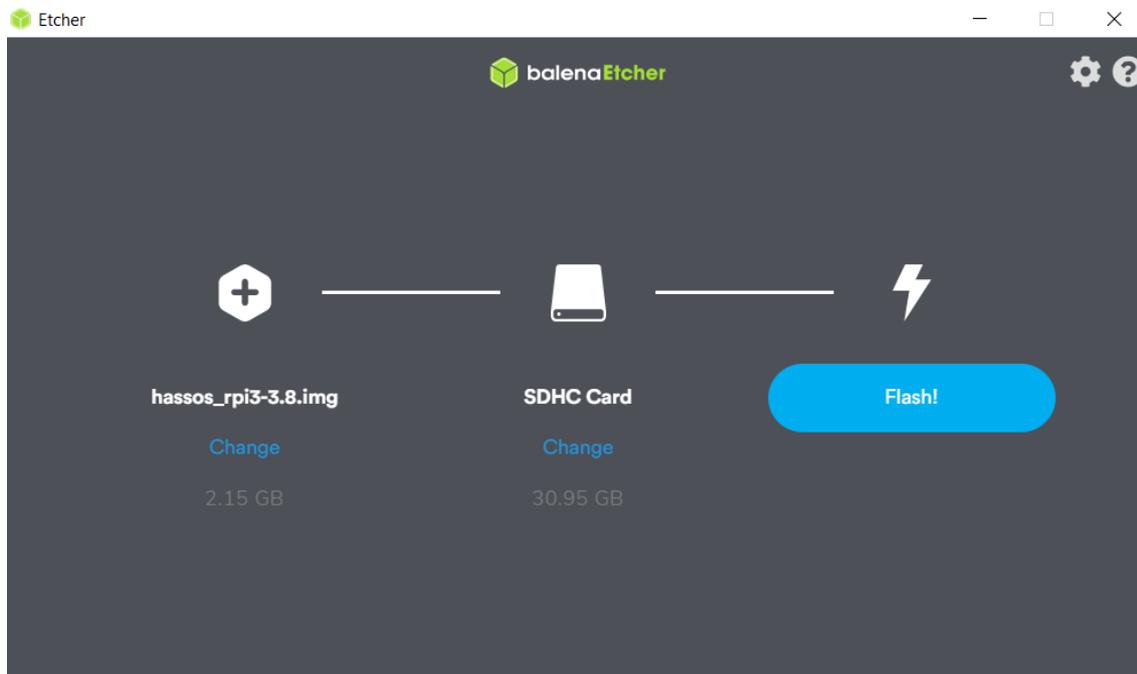
D.3 INSTALACIÓN DE HOME ASSISTANT

1º Descargar la imagen dependiendo del tipo de dispositivo [16]:



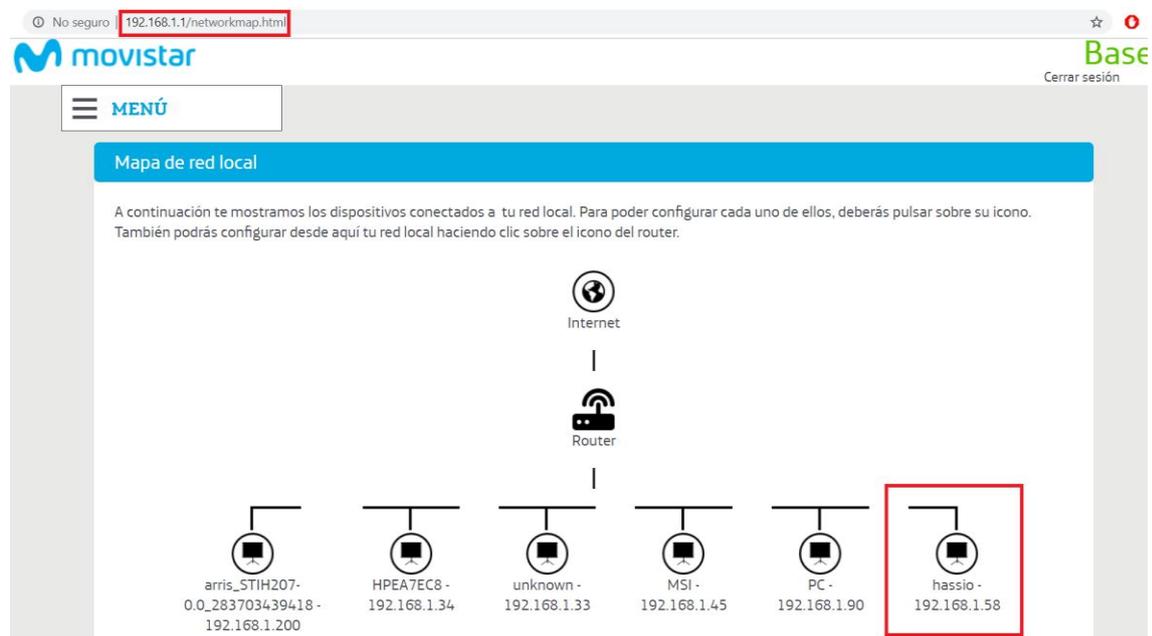
The screenshot shows the 'Installing Hass.io' page on home-assistant.io. The page title is '// Installing Hass.io' and it includes a link to 'Edit this page on GitHub'. The main content states: 'The following will take you through the steps required to install Hass.io.' The first step is '1. Download the appropriate install option:', followed by a sub-step 'o As an image for your device:'. A list of options is provided, with 'Raspberry Pi 3 Model B and B+ 32bit (recommended)' highlighted by a red box. Other options include Raspberry Pi 3 Model B and B+ 64bit, Raspberry Pi 4 Model B 32bit (recommended), Raspberry Pi 4 Model B 64bit, Tinkerboard, Odroid-C2, Odroid-N2 (Beta), Odroid-XU4, and Intel-Nuc.

2º Guardar la imagen en la SD de la Raspberry Pi con el programa que se ha instalado previamente para ello:

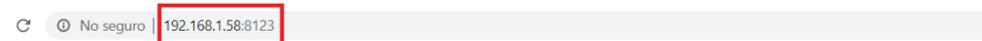


The screenshot shows the Etcher application interface. The title bar reads 'Etcher'. The main window features the 'balenaEtcher' logo and a progress indicator with three stages: a plus sign, a SD card icon, and a lightning bolt icon. Below the progress indicator, the source image is 'hassos_rpi3-3.8.img' (2.15 GB) and the target is 'SDHC Card' (30.95 GB). A blue 'Flash!' button is visible on the right side of the interface.

3º Introducimos la SD dentro del Raspberry Pi. Procedemos a conectar el cable de alimentación y un cable Ethernet. Una vez iniciada buscamos la IP que se le asigna por DHCP automáticamente en la dirección del router.



4º Una vez que sabemos la dirección IP del dispositivo podremos acceder a él de la siguiente forma <http://192.168.1.58:8123> . Habría que sustituir la IP: 192.168.1.58 por la que tengas asignada en tu red. Tardará 20 minutos aproximadamente en descargar la última versión de Hassio



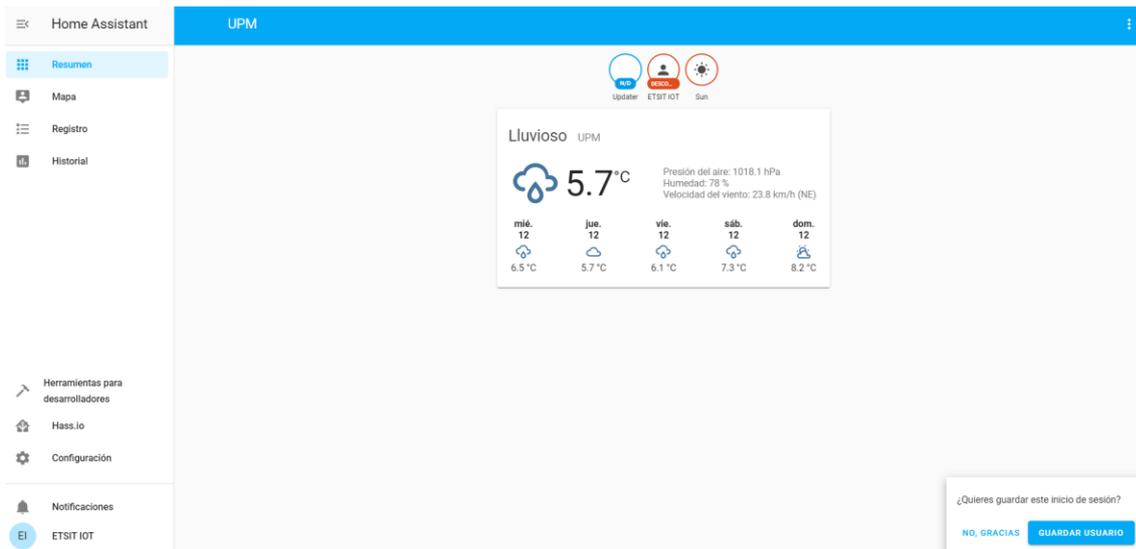
Preparing Hass.io

(this can take up to 20 minutes)

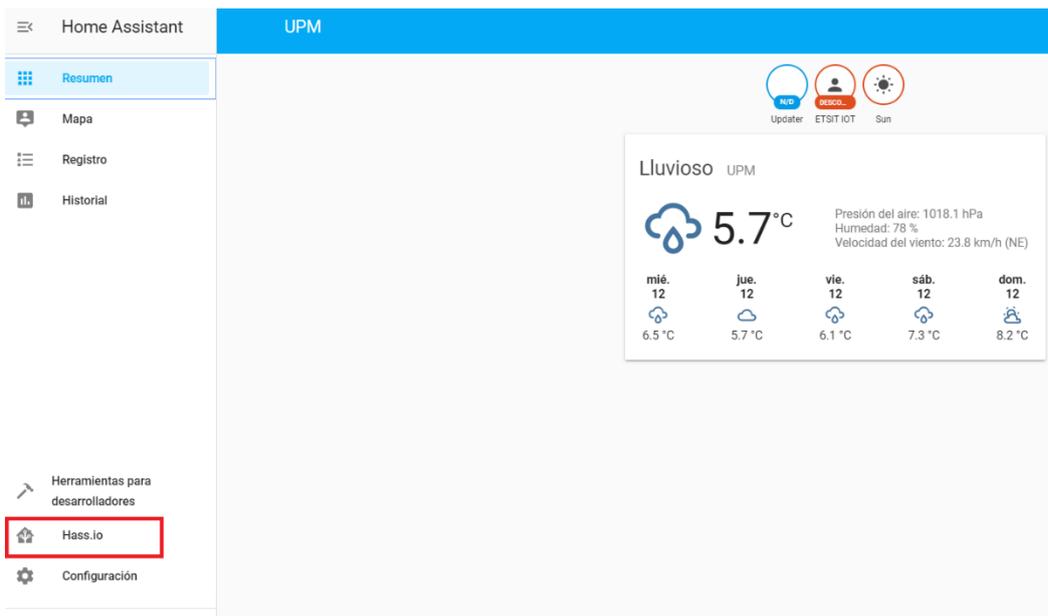
5º Crearemos una cuenta de usuario:

6º Rellenar la información de la localización. Será muy importante para después programar automatizaciones en función de las horas del Sol y la elevación. Así como la hora correcta

7º Una vez hecho esto ya tendremos creado nuestro Home assistant donde integraremos todos nuestros dispositivos



8º Instalar Mosquitto Broker para MQTT. Para ello hay que ir a la pestaña lateral donde aparece HASS.IO



9º Dentro de HASS.IO buscar el repositorio de Mosquitto Broker e instalarlo

The screenshot shows the Home Assistant interface with the Hass.io Add-on Store selected. The 'ADD-ON STORE' tab is highlighted in red. Below the navigation bar, there are sections for 'Repositories' and 'Official add-ons'. In the 'Official add-ons' section, the 'Mosquitto broker' add-on is highlighted with a red box. The sidebar on the left shows 'Herramientas para desarrolladores' with 'Hass.io' highlighted in red.

The screenshot shows the details page for the 'Mosquitto broker' add-on. The page title is 'Mosquitto broker' with a version of 5.1. It includes a description, the MQTT.ORG logo, and a grid of icons for '6 RATING', 'APPROVED', and 'AUTH'. The 'INSTALL' button is highlighted in red. Below the button, there is a section for 'Hass.io Core Add-on: Mosquitto broker' with supported architectures listed: aarch64 yes, amd64 yes, armhf yes, armv7 yes, i386 yes. The 'About' section describes it as an open-source message broker. The 'Installation' section provides a 3-step guide: 1. Navigate to Hass.io -> Add-on Store, 2. Find the 'Mosquitto broker' add-on and click it, 3. Click on the 'INSTALL' button. The 'How to use' section starts with '1. Start the add-on'.

10º Una vez instalado haremos una configuración con un login y password que llevaran nuestros dispositivos para conectarse con mosquitto. También se podrán configurar el puerto por el que deseas que se manden los mensajes.

Config

```
{
  "logins": [
    {
      "username": "mosquitto",
      "password": "1234"
    }
  ],
  "anonymous": false,
  "customize": {
    "active": false,
    "folder": "mosquitto"
  },
  "certfile": "fullchain.pem",
  "keyfile": "privkey.pem",
  "require_certificate": false
}
```

RESET TO DEFAULTS

SAVE

Network

Container	Host	Description
1883/tcp	1883	
1884/tcp	1884	
8883/tcp	8883	
8884/tcp	8884	

RESET TO DEFAULTS

SAVE

11º Una vez hecho todo esto le daremos a START para arrancar el servicio de mosquitto bróker.

Mosquitto broker 5.1

An Open Source MQTT broker.
Visit [Mosquitto broker page](#) for details.

MQTT.ORG

6 **RATING** APPAR... AUTH

Start on boot

Auto update

Protection mode

UNINSTALL **START**

Hass.io Core Add-on: Mosquitto broker
MQTT broker for Home Assistant.

ANEXO E. GUÍA DE ACCESO A LA PLATAFORMA HOME ASSISTANT PARA PERSONAL DE LA ETSIT

E.1 ACCESO A LA PLATAFORMA

Para acceder a la plataforma debemos de ingresar la siguiente dirección en el navegador.

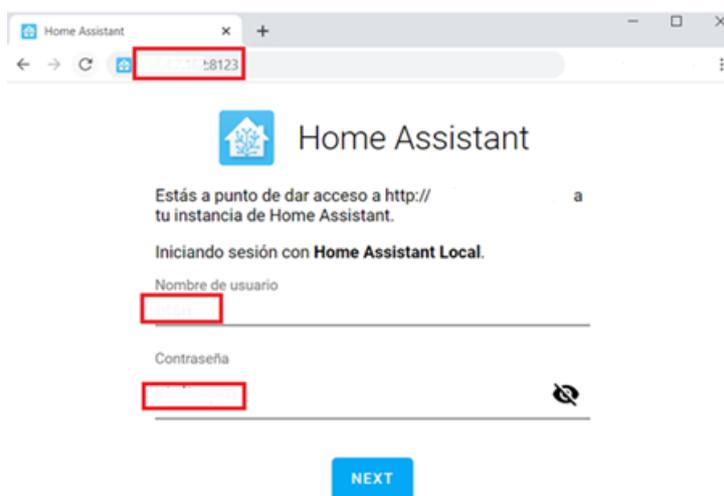
<http://Z.Z.Z.Z:8123>

El acceso debe de hacerse a través de cualquier navegador (Google Chrome, Internet Explorer, Firefox, Safari, etc). Desde la red de la UPM (Eduroam) está restringido el acceso. Por eso se debe de realizar desde conexión cableada o datos móviles.

Una vez que accedamos, debemos de introducir las siguientes credenciales en la página de inicio:

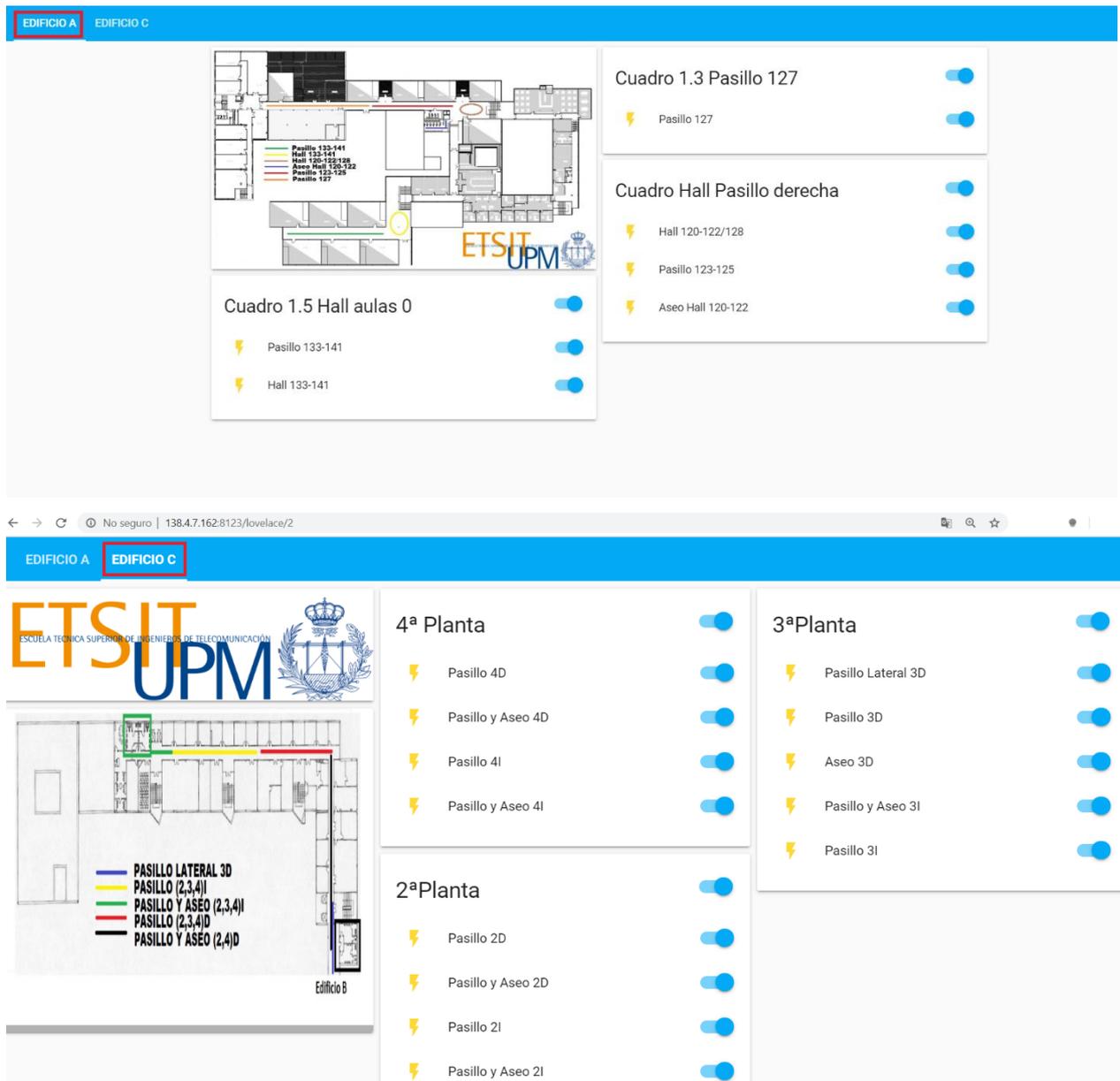
Nombre de usuario: xxxx

Contraseña: xxxx



E.2 NAVEGACIÓN POR LA PLATAFORMA

Una vez dentro de la plataforma tendremos acceso a el control de las luces del Edificio A y Edificio C. Estos están diferenciados en dos pestañas diferentes que se encuentran en la parte superior. En cada una de estas pestañas veremos los cuadros de cada edificio. En el caso del edificio A están ordenador por pasillos y aulas. En el caso del edificio C están colocados por plantas. En este caso además tendremos un mapa donde se verán a que se corresponden cada una de las etiquetas de la plataforma. En las siguientes imágenes se observa lo que nos encontraremos en cada una de las pestañas:



E.3 FUNCIONALIDADES DE LA PLATAFORMA

En la plataforma tendremos dos opciones principales:

- Apagar/Encender una luz determinada
- Apagar/Encender un grupo de luces

Empezamos por el primer caso. Si queremos apagar una luz determinada habría que ir al interruptor que se encuentra a la derecha de esa luz en concreto y pulsarlo, como se muestra en la siguiente imagen:

EDIFICIO A EDIFICIO C

4ª Planta

- Pasillo 4D
- Pasillo y Aseo 4D
- Pasillo 4I
- Pasillo y Aseo 4I

3ªPlanta

- Pasillo Lateral 3D
- Pasillo 3D
- Aseo 3D
- Pasillo y Aseo 3I
- Pasillo 3I

2ªPlanta

- Pasillo 2D
- Pasillo y Aseo 2D

Si por el contrario queremos apagar un grupo determinado, hay que pulsar en la parte derecha de las luces que están agrupadas por plantas en el Edificio C o por cuadros eléctricos en el Edificio A.

EDIFICIO A EDIFICIO C

4ª Planta

- Pasillo 4D
- Pasillo y Aseo 4D
- Pasillo 4I
- Pasillo y Aseo 4I

3ªPlanta

- Pasillo Lateral 3D
- Pasillo 3D
- Aseo 3D
- Pasillo y Aseo 3I
- Pasillo 3I

2ªPlanta

- Pasillo 2D
- Pasillo y Aseo 2D

EDIFICIO A EDIFICIO C

Cuadro 1.3 Pasillo 127

- Pasillo 127

Cuadro Hall Pasillo derecha

- Hall 120-122/128
- Pasillo 123-125
- Aseo Hall 120-122

Cuadro 1.5 Hall aulas 0

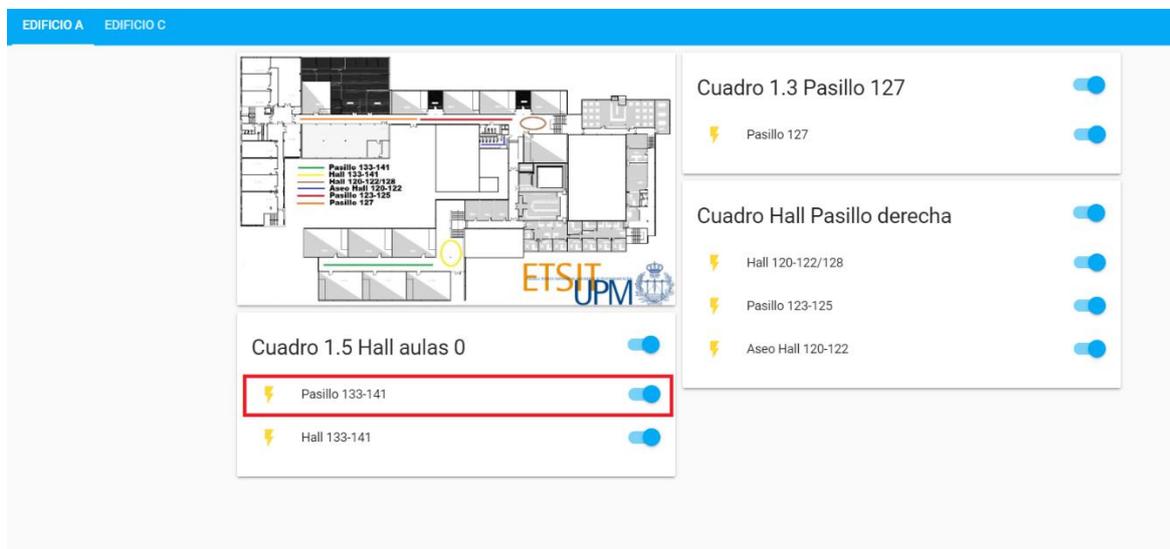
- Pasillo 133-141
- Hall 133-141

E.4 ESTADO DE LOS DISPOSITIVOS

Los dispositivos hacen como un interruptor y hay dos posibles estados:

- Encendido
- Apagado

Si el interruptor está en encendido, se mostrará en amarillo.



Si está apagado pasará a estar en azul oscuro, y el interruptor cambiará de posición.

