

Instalación y configuración de un sistema de hogar inteligente integral basado en plataformas de software libre.

MEMÓRIA PRESENTADA POR:

Antonio Camarena Ivars

GRADO DE INGENIERÍA INFORMÁTICA

Autor del TFG: Antonio Camarena Ivars

Fecha: 15/02/19

Tutor Académico: Pedro José Ramiro Zafra

Agradecimientos

A Pedro José Ramiro Zafra, tutor de este trabajo, por su tiempo y dedicación.

Y por supuesto a mis padres que han sido, como siempre, un apoyo constante e incondicional

Resumen

Este trabajo, enmarcado dentro del ámbito de la domótica, aborda una perspectiva de software libre, multidispositivo y multifabricante, el diseño y construcción de un sistema domótico completo para una vivienda unifamiliar tipo. Para ello, integra diversos fabricantes y utiliza herramientas de software libre para abordar áreas tan diversas que van desde la seguridad y el confort hasta el riego inteligente, con el fin de convertir la vivienda en un hogar inteligente y sostenible.

Palabras clave: *OpenHAB, Zigbee, protocolo MQTT, domótica, ESP8266.*

Resum

Aquest treball, emmarcat dins de l'àmbit de la domòtica, aborda una perspectiva de software lliure, multidispositiu i multifabricant, el disseny i construcció d'un sistema domòtic complet per a una vivenda unifamiliar estàndard. Per aquesta raó integra diversos fabricants i utilitza ferramentes de software lliure, per a tractar àrees tan diferents que van des de la seguretat i el confort fins el reg intel·ligent, amb la finalitat de convertir la vivenda en una llar intel·ligent i sostenible.

Paraules clau: *OpenHAB, Zigbee, protocol MQTT, domòtica, ESP8266.*

Abstract

This study is framed within the area of automotion, which addresses a free software perspective, multi-devices and multi-producers, the design and construction of a complet automotion system for a single family home. In order to this, different producers are integrated and free software resources are used so that different areas can be adressed not only security and comfort but also intelligent irrigation, achieving as a result an intelligent and sustainable housing.

Keywords: *OpenHAB, Zigbee, MQTT protocol, home automation, ESP8266.*



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Contenido

Introducción	1
1.1. Motivación.....	1
1.2. Objetivos	4
1.3. Estructura de la memoria	4
1.4. Principales referencias utilizadas.....	5
1.5. Relación con las asignaturas cursadas	6
Estado del arte.....	7
2.1. Jeedom.....	7
2.2. Home Assistant.....	8
2.3. OpenHAB.....	8
2.4. Otras alternativas.....	9
Especificación de requisitos.....	10
3.1. Alcance.....	10
3.2. Perspectiva del sistema domótico	10
3.3. Tipos de dispositivos.....	10
3.3.1. <i>Funcionalidad del sistema domótico</i>	11
3.3.2. <i>Restricciones</i>	11
Planificación y costes.....	12
4.1. Fases de desarrollo.....	12
4.2. Recursos.....	15
4.2.1. <i>Hardware</i>	15
4.2.2. <i>Software</i>	15
4.3. Proyección de costes.....	16
Herramientas y tecnologías utilizadas.....	17
5.1. HPE ProLiant MicroServer Gen10.....	17
5.2. Raspberry Pi.....	18
5.2.1. <i>Raspberry Pi 3B</i>	18
5.2.2. <i>Puesta en marcha</i>	18
5.2.3. <i>Consumo energético</i>	19
5.3. Protocolo MQTT.....	19
5.4. Mosquitto (Eclipse).....	22
5.5. Protocolo ZigBee	23
5.5.1. <i>Bandas de operación</i>	23
5.5.2. <i>Nodos y topología de red</i>	23

5.5.3.	<i>Seguridad</i>	25
5.5.4.	<i>Debilidades en ZigBee</i>	25
5.6.	Proyecto openHAB.....	25
5.7.	Sonoff (ITEAD).....	27
5.8.	Chip ESP8266.....	28
5.9.	Firmware Tasmota.....	28
5.10.	FTDI	29
5.11.	Xiaomi Router 3.....	29
5.12.	Dispositivos ZigBee en Mi Home.....	30
	Instalación y configuración.....	32
6.1.	Configuración del servidor HP	32
6.2.	Configuración Raspberry Pi 3.....	33
6.3.	Instalación y configuración de Mosquitto	34
6.4.	Modificación y configuración del firmware Tasmota	35
6.5.	Instalación y configuración de openHAB	37
6.5.1.	<i>Descarga e instalación</i>	37
6.5.2.	<i>Interfaz gráfica de usuario</i>	39
6.5.3.	<i>Configuración y funcionamiento inicial</i>	40
	Configuración automática.....	41
	Configuración manual.....	43
6.5.4.	<i>Sitemaps</i>	44
6.5.5.	<i>Bindings utilizados</i>	45
6.5.6.	<i>Automatizaciones (Rules)</i>	46
6.5.7.	<i>Integración myOpenHAB</i>	47
6.5.8.	<i>Configuración de MQTT</i>	47
6.6.	Otras Instalaciones y configuraciones.....	49
6.6.1.	<i>Instalación y configuración del router</i>	49
6.6.2.	<i>Instalación y configuración de la aplicación Mi Home</i>	49
6.6.3.	<i>Instalación y configuración de Visual Studio Code</i>	50
	Instalación SMB.....	51
	Acceso remoto	52
	Descripción gráfica del entorno real, automatizaciones y pruebas de funcionamiento	53
7.1.	Descripción.....	53
7.2.	Plano 2D	53
7.3.	Ejemplos de automatizaciones.....	54
7.4.	Pruebas en automatizaciones	55

7.5. Pruebas relacionadas con el sitio web myopenHAB	56
Conclusiones y líneas futuras	57
8.1. Líneas futuras	58
8.2. Opinión personal.....	58
Referencias y bibliografía	59
Ficheros de configuración openHAB.....	64

Índice de ilustraciones

Ilustración 1. Microsoft, 2017. IoT para empresas.....	2
Ilustración 2. CEDOM.¿Qué es la domotica?	3
Ilustración 3. Icono Jeedom.....	7
Ilustración 4. Icono Home Assistant.	8
Ilustración 5. Icono openHAB.	8
Ilustración 6. Diagrama de Gantt.....	14
Ilustración 7. HP Gen10.	17
Ilustración 8. Raspberry Pi 3.	18
Ilustración 9. Raspi.TV. How much power does Pi Zero W use?	19
Ilustración 10. MQTT (Message Queue Telemetry Transport).....	19
Ilustración 11. Garcia Soria, A. Funcionamiento elementos MQTT.....	20
Ilustración 12. Garcia Soria, A. Comunicación establecida entre cliente y bróker.....	21
Ilustración 13. Garcia Soria, A. Calidad de nivel QoS0.....	21
Ilustración 14. Garcia Soria, A. Calidad de nivel QoS1.....	22
Ilustración 15. Garcia Soria, A. Calidad de nivel QoS2	22
Ilustración 16. Marín ,J y Ruiz,D. Informe técnico protocolo ZigBee	24
Ilustración 17. Marín ,J y Ruiz,D. Informe técnico protocolo ZigBee.....	24
Ilustración 18. Funcionamiento bindings openHAB. Imagen propia.....	26
Ilustración 19. FTDI.	29
Ilustración 20. Gateway Xiaomi. Disponible en: https://	30
Ilustración 21. Programa Win32Diskimager. Imagen Propia.....	32
Ilustración 22. Instalación Ubuntu Server 18.04. Imagen Propia	33
Ilustración 23. Programa PuTTY. Imagen Propia	34
Ilustración 24. Instalación MQTT bróker. Imagen Propia	34
Ilustración 25. Comprobación netstat Bróker MQTT. Imagen Propia	35
Ilustración 26. Soldado pines Sonoff. Imagen Propia	35
Ilustración 27. Conexión FTDI con el Sonoff. Imagen Propia.....	35
Ilustración 28. Programa ESPEasy. Imagen Propia	36
Ilustración 29. interfaz web de Tasmota. Imagen Propia	36
Ilustración 30. Interfaz de bienvenida de openHAB. Imagen Propia.....	39
Ilustración 31. Interfaz Paper UI. Imagen Propia.....	39
Ilustración 32. Interfaz Basic UI. Imagen Propia	40
Ilustración 33. Interfaz HABpanell. Imagen Propia.....	40
Ilustración 34. Instalación binding NTP. Imagen Propia	41
Ilustración 35. Configuración binding NTP. Imagen Propia	41
Ilustración 36. Configuración del canal. Imagen Propia	42
Ilustración 37. Enlazar canal al ítem. Imagen Propia	42
Ilustración 38. Prueba correcto funcionamiento NTP. Imagen Propia	43
Ilustración 39. Configuración NTP en sitemap. Imagen Propia	44
Ilustración 41. Conector myOpenHAB. Imagen Propia.....	47
Ilustración 42. Interfaz myOpenHAB. Imagen Propia	47
Ilustración 43. Configuración MQTT de Sonoff. Imagen Propia	48



Ilustración 44. Prueba relé Sonoff. Imagen Propia	49
Ilustración 45. Vinculación Gateway Xiaomi. Imagen Propia	50
Ilustración 46. Vinculación Gateway con openHAB. Imagen Propia	50
Ilustración 47. VS Code Plugin openHAB. Imagen Propia	51
Ilustración 48. Acceso a la carta compartida openHAB. Imagen Propia	52
Ilustración 49. Diseño inmueble floorplanner. Imagen Propia.....	53

CAPÍTULO 1

Introducción

En este apartado de introducción se recogerán los conceptos básicos tratados en esta memoria, como la relevancia que está adquiriendo en el presente el “Internet de las Cosas”, especialmente en la vida de las personas y empresas.

Este proyecto surge por un interés propio respecto al ámbito de “Internet de las cosas”, en adelante IoT, aplicada al hogar, conocido como domótica. Además, este trabajo busca una solución alcanzable para muchas personas, ya que uno de los requisitos indispensables siempre ha sido el tener “un bajo coste” y eso se consigue gracias a la utilización de software de código abierto (*open-source*) y hardware a precio razonable.

1.1. Motivación

La principal motivación de este proyecto es la expansión de las soluciones y tecnología que han ido apareciendo, relacionadas con el IoT. Económicamente hablando, la gran mayoría de estas tecnologías son muy elevadas en el mercado, lo que supone que se lo pueda permitir un reducido campo de clientes.

Se dividen en diferentes ramificaciones, como puede ser el uso del IoT en el hogar, en ciudades, en la industria, o incluso en los diferentes dispositivos que las personas utilizan de uso cotidiano. Un buen ejemplo de eso puede ser: una lavadora conectada a Internet que te avisa cuando ha acabado, un frigorífico que te indique que alimentos faltan en él, y que decir de la automoción con la conducción autónoma de algunos modelos.

La expansión de internet y su acceso han crecido de forma exponencial en los últimos años. Vivimos en una sociedad hiperconectada ya que podemos comunicarnos con cualquier persona del mundo en tiempo real, por lo que no sabremos convivir sin esta tecnología.

El IoT está vigente en una gran diversidad de entornos, muy diferentes entre sí, como pueden ser:

- **La industria**

La tecnología IoT en industrias ayuda a la cooperación conjunta entre los seres humanos, las máquinas y los sistemas empresariales, mejorando en gran parte la eficiencia de producción y en la reducción de costes.

- **Los edificios inteligentes**

Con las nuevas implementaciones de IoT se está consiguiendo una seguridad, eficiencia y confort de las personas que lo utilizan, gracias a la recopilación y analizado de datos diariamente.

- **La automoción**

Los vehículos del futuro estarán altamente conectados y dispondrán de sensores que, unidos con otros vehículos, formarán una gran red que recopilarán y enviarán datos a los servidores, los cuales manipularán para recopilar información sobre tráfico, aparcamientos libres, etc.

Por otra parte, IoT amplía otros campos de trabajo. Por ejemplo, según (*BlinkEdTech, 2015*) en [1], el uso en la educación permitirá que los estudiantes aprendan más y mejor en tiempo real. Lo más importante será el aprendizaje en tiempo real y adaptado, es decir, que la universidad aprenderá de sus estudiantes y podrá mejorar simultáneamente sus procesos al aplicar esos cambios. Otro ejemplo, tal y como comenta (*Marta B, 2017*) en [2], sería referente al uso de IoT en la salud mediante la conectividad constante y segura entre profesionales y pacientes. Otro sector imprescindible en la implementación del IoT según (*Microsoft, 2016*) en [3], será proporcionar un transporte conectado para mejorar el tráfico en ciudades con mayor movilidad y eficiencia buscando una mejora del tráfico incrementando una mayor calidad respecto a los pasajeros y reducir la congestión.

En resumidas cuentas, podemos observar que las empresas más importantes a nivel mundial ven en el IoT una gran oportunidad de negocio. Por ejemplo, (*Microsoft, 2017*) en [4], informa sobre las pautas necesarias para transformar una empresa en una inteligente. Dichas pautas serían: desarrollar productos que efectúen tecnologías basadas en IoT, como dispositivos o sensores inteligentes y mantener un control de los datos capturados en tiempo real para posteriormente realizar un análisis de los mismos con el objetivo de mejorar la eficiencia y rendimiento de la empresa.



Ilustración 1. Microsoft, 2017. IoT para empresas. [Consulta: 11 de noviembre de 2017].
Disponible en: https://azure.microsoft.com/es-es/overview/iot/?site=mscom_io

Para (*Intel, 2016*) en [5], la evolución de IoT viene fundamentada por 3 pasos:

1. Conectar los dispositivos que aún no lo estén mediante sensores con los que recopilar información y que el centro de datos los analice.
2. Crear sólo dispositivos con inteligencia artificial directamente conectados a los servidores de datos.

3. Creación de un mundo autónomo donde los datos recogidos y analizados por los centros de datos puedan actuar libremente en la toma de decisiones.

La domótica (según CEDOM, 2017) en [6], es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda que permite una gestión eficiente del uso de la energía, aportando seguridad y confort.

Un sistema domótico es capaz de recoger información proveniente de sensores o entradas, procesarlas y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información.

La domótica aporta mejoras en la calidad de vida del usuario, como, por ejemplo:

- **Accesibilidad:** facilitar el manejo de los elementos del hogar a las personas con discapacidad de la forma que más se ajuste a sus necesidades.
- **Ahorro de energía:** gestionar inteligentemente la iluminación, climatización, agua caliente, el riego, los electrodomésticos, etcétera, aprovechando mejor los recursos naturales y utilizando las tarifas horarias de menor coste para una reducción en la factura energética.
- **Seguridad:** se incrementa el nivel de seguridad del hogar mediante vigilancia 24/7, actuando de manera autónoma ante posibles averías o incidencias. Para ello se emplean elementos de control de intrusiones, cámaras de vigilancia, alarmas para detección de incendios, fugas de agua, etc.
- **Confort:** el uso de la domótica permite tener un hogar más confortable a través de la gestión de dispositivos y actividades domésticas. La domótica permite el control a distancia de electrodomésticos, climatización, ventilación, luces, persianas, etc.
- **Comunicaciones:** garantiza la conexión entre el sistema domótico y los usuarios del hogar mediante un control y supervisión remota a través de aplicaciones en el teléfono, ordenador, servidores web..., que permitirá avisos sobre anomalías o averías en tiempo real.

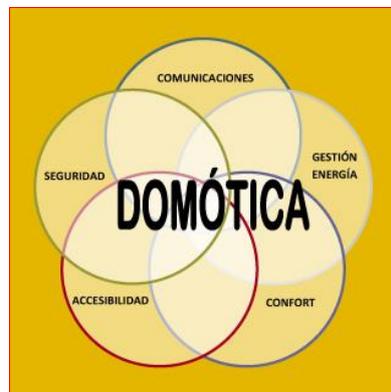


Ilustración 2. CEDOM. ¿Qué es la domótica? [Consulta: 13 de noviembre de 2017]. Disponible en: <http://www.cedom.es/sobre-domotica/que-es-domotica>

1.2. Objetivos

En este apartado serán expuestos los objetivos principales de este trabajo. Hay dos objetivos generales bien estructurados y diferenciados uno del otro que a su vez están compuestos por otros específicos.

El primer objetivo es implementar un servidor domótico económico. Para las comunicaciones se usarán protocolos ZigBee y MQTT (*bróker Mosquitto*) con licencia de software libre.

El segundo de los objetivos está relacionado con el diseño e implementación de la interfaz web para el sistema domótico. Debe permitir su monitorización y control de forma remota y segura a través de cualquier dispositivo conectado a Internet. Dicha interfaz tiene que ser entendible, dinámica e intuitiva.

1.3. Estructura de la memoria

En este apartado se llevará a cabo un resumen de la estructura de la presente memoria y se realizará una breve descripción de cada uno de ellos.

La estructura de este proyecto está basada en 11 apartados:

- **Capítulo 1. Introducción.**
En este apartado se realizará una breve introducción de los conceptos de IoT y domótica, además de la situación actual y el crecimiento que se está experimentando. Por otra parte, se explicarán los objetivos perseguidos en este documento, la estructura, algunas referencias a modo de información bibliográfica y una relación con las asignaturas cursadas en la carrera universitaria.
- **Capítulo 2. Estado del arte.**
En esta sección se realizará un estudio de las diferentes soluciones existentes en el mercado y se hará una comparación entre ellas. Finalmente, se procederá a la elección de una de las plataformas domóticas en la que se basará este proyecto.
- **Capítulo 3. Especificación de requisitos.**
En este punto se definirán las características y funcionalidades del sistema. Se detallará el alcance, la perspectiva, los tipos de dispositivos, su funcionalidad y sus correspondientes restricciones.
- **Capítulo 4. Planificación y coste.**
Aquí se presentarán las diferentes fases de desarrollo del proyecto junto con el tiempo invertido en cada una de ellas. Además, se detallarán los recursos hardware y softwares utilizados.
Por último, se realizará un cálculo del coste total aproximado del proyecto.

- **Capítulo 5. Herramientas y tecnologías a utilizar.**
En esta sección se realizará un estudio detallado de las diferentes tecnologías y protocolos utilizados. Además, se explicarán las herramientas usadas para llegar a la solución del problema.
- **Capítulo 6. Instalación y configuración.**
En este capítulo se expondrán todos los detalles para la implementación del servidor domótico openHAB. Por otro lado, se definirá un caso práctico para la inicialización y aprendizaje básico del sistema.
- **Capítulo 7. Descripción gráfica del entorno real, automatizaciones y pruebas de funcionamiento.**
En el apartado de diseño se realizará un plano con la distribución de todos los dispositivos localizados entre las diferentes estancias de la vivienda. Además, se definirán las automatizaciones que integrará el sistema domótico.
Además, se recogerán las pruebas realizadas al sistema. Se comprobará el buen funcionamiento de cada uno de los elementos que lo componen, tanto las automatizaciones como el acceso remoto al sistema.
- **Capítulo 8. Conclusiones y líneas futuras.**
Como en todos los proyectos, es una de las partes más importantes. Se definirán las conclusiones a las que se ha llegado durante la realización del proyecto. Posteriormente se ampliará con unas posibles mejoras futuras que se podría implementar en el futuro.
- **Capítulo 9. Referencias y bibliografía.**
Se recogerán las referencias de documentos y/o enlaces bibliográficos consultados durante la investigación y la realización del proyecto.
- **Capítulo 10. Anexos.**
En este apartado se incluirán algunos de los ficheros de configuración de los elementos del servidor openHAB.

1.4. Principales referencias utilizadas

En este apartado se explicarán las principales referencias utilizadas para el estudio y realización de este proyecto:

- Documentación de la página web oficial de openHAB [7]. Este sitio dispone de toda la documentación necesaria para la instalación y configuración del sistema domótico. Además, dispone de una amplia comunidad de usuarios para generar dudas y solucionar problemas.
- Documentación del sitio web oficial del protocolo MQTT [8]. Esta web dispone de toda la documentación necesaria para la instalación y configuración de MQTT.

1.5. Relación con las asignaturas cursadas

Tal y como se ha comentado anteriormente, la mayor parte del desarrollo de este proyecto se basa en los conocimientos adquiridos a lo largo de estos cuatro años de carrera. Las asignaturas con una vinculación directa han sido las referentes a la programación, gestión de proyectos, ingeniería del Software, etc.

Aunque no todas las asignaturas se han visto vinculadas directamente con el proyecto, en su conjunto han sido muy importantes para proporcionarme habilidades, conocimientos y competencias.

A continuación, se muestra la relación directa de las asignaturas cursadas con las partes de este proyecto:

Apartado del TFG: 3. Especificación de requisitos

<i>Asignatura</i>	Ingeniería del software Gestión de Proyectos Gestión de servicios de SI-TI
<i>Justificación</i>	En estas asignaturas se ha aprendido a realizar una especificación de requisitos completa.

Apartado del TFG: 4. Planificación y costes

Subapartados: 4.1 Fases de desarrollo, 4.3. Proyección de coste.

<i>Asignaturas</i>	Gestión de proyectos Modelos de negocio y áreas funcionales de la organización
<i>Justificación</i>	Estas asignaturas han ayudado a la comprensión y planificación del proyecto.

Apartado del TFG: 5. Herramientas y tecnologías utilizadas

Subapartados: 5.6. Sonoff (ITEAD), 5.7. ESP8266, 5.8. FTDI.

<i>Asignatura</i>	Tecnología de computadores
<i>Justificación</i>	Esta asignatura ha sido de gran ayuda para conocer y programar placas de desarrollo, diferenciar métricas, etc.

Apartado del TFG: 6. Instalación y configuración

<i>Asignaturas</i>	Tecnología de computadores Fundamentos de sistemas operativos Interfaces persona computador Programación Concurrencia y sistemas distribuidos
<i>Justificación</i>	Estas asignaturas han sido de gran ayuda. Comenzando por las instalaciones de Linux en los servidores, pasando por la programación de las automatizaciones y acabando con la realización de la interfaz gráfica de usuario.

CAPÍTULO 2

Estado del arte

En esta sección se realizará un estudio del estado del arte, donde se expondrán las características más relevantes de las diferentes alternativas software que existen actualmente en el mercado. Además, se explicará la elección del sistema domótico que se va a decidir utilizar en este proyecto.

En el avance tecnológico en el que vivimos se necesitará saber de la necesidad que se tiene y el uso que se le va a dar a un sistema domótico de estas características.

A continuación, se definirán algunas de las alternativas de sistemas domóticos de código abierto que se presentan en el mercado actualmente:

2.1. Jeedom

Jeedom [9], es un software de código abierto. Una persona con conocimientos básicos de Linux puede hacer uso de él sin mayores problemas. La instalación y configuración se puede realizar sobre una Raspberry Pi, aunque el desarrollador ha puesto en el mercado sus propios tipos de controladores domóticos.



*Ilustración 3. Icono Jeedom. Disponible en:
<https://www.jeedom.com/site/fr/>*

La marca dispone de dos modelos, uno de características similares a la Raspberry Pi 3 y otro a nivel profesional. El precio del dispositivo más económico es de aproximadamente 220 € mientras que el modelo profesional de aproximadamente 900€.

Jeedom es una alternativa interesante, proporciona seguridad, automatización y ahorro de energía, pudiendo monitorizar el consumo de energía, la comunicación por voz, SMS o aplicaciones móviles junto con los automatismos del hogar como sensores, luces, cámara, etc. Además, el desarrollador proporciona una aplicación móvil con compatibilidad para dispositivos Android e iOS.

Ventajas	Inconvenientes
<input checked="" type="checkbox"/> Código abierto	<input checked="" type="checkbox"/> Plugins de pago (2-6€)
<input checked="" type="checkbox"/> Multi-protocolo	
<input checked="" type="checkbox"/> Autónomo	
<input checked="" type="checkbox"/> Personalizable	

2.2. Home Assistant

Home Assistant [10] también es de software libre. Esta plataforma de automatización del hogar funciona sobre Python3. Es una alternativa muy interesante para la instalación en una Raspberry Pi. Además, permite manejar los dispositivos inteligentes sin necesidad de un servidor externo, aumentando la privacidad.



Ilustración 4. Icono Home Assistant. Disponible en: <https://www.home-assistant.io/>

Home Assistant funciona con reglas programables ante posibles cambios en el sistema del hogar. Además, dispone de una comunidad muy grande de usuarios y un sitio web para realizar cualquier consulta.

Ventajas	Inconvenientes
<input checked="" type="checkbox"/> Código abierto <input checked="" type="checkbox"/> Multi-protocolo <input checked="" type="checkbox"/> Autónomo <input checked="" type="checkbox"/> Personalizable <input checked="" type="checkbox"/> Comunidad de usuarios	<input checked="" type="checkbox"/> Actualizaciones inestables

2.3. OpenHAB

OpenHAB [11] es una tecnología cuyo software es de código abierto, diseñado para integrar diferentes sistemas de automatización de viviendas, dispositivos y tecnologías dentro de una misma solución.



Ilustración 5. Icono openHAB. Disponible en: <https://www.openhab.org/>

OpenHAB es una solución basada en Java y proporciona una interfaz de usuario uniforme, centralizado y con un enfoque común para las reglas de automatización de todo el sistema.

Ventajas	Inconvenientes
<input checked="" type="checkbox"/> Código abierto <input checked="" type="checkbox"/> Multi-protocolo <input checked="" type="checkbox"/> Autónomo <input checked="" type="checkbox"/> Personalizable <input checked="" type="checkbox"/> Actualizaciones estables	

2.4. Otras alternativas

Las tres alternativas software comentadas anteriormente son ahora mismo las más utilizadas por los usuarios. No obstante, existen muchas más alternativas existentes, como, por ejemplo:

- **Eedomus.**
Consta de un controlador, aplicaciones y dispone de un servicio en la nube opcional. Este controlador dispone de un puerto Ethernet y dos puertos USB. Soporta la tecnología Z-Wave, de serie, y EnOcean de manera opcional. Además dispone de un sinfín de dispositivos para la automatización del hogar.
El precio del controlador oscila entre 250-300 euros.
- **Zipato Zipatile.**
Utiliza los protocolos Z-Wave y ZigBee. Es un dispositivo muy útil para situarlo junto a la pared del hogar. Cuenta con un gran número de sensores y módulos, además de termostato, cámara IP, controlador de automatización y alarma.
El precio es de unos 350 euros y está disponible en el sitio web oficial.
- **Xiaomi Smart Home**
Es un kit domótico desarrollado por la empresa china Xiaomi. Este puede comprarse en forma de kit o cada dispositivo individualmente. La gestión la determina el controlador domótico y los dispositivos irán vinculados a dicha centralita.
Este sistema tiene una peculiaridad de ser integrable en los diferentes sistemas domóticos open-source que existen hoy en día, por lo que lo convierte en un gran candidato a la hora de comprar este tipo de dispositivos.
El precio es relativamente económico, desde los 25 euros que puede costar la centralita a unos 3-25 euros para los demás dispositivos.

En conclusión, tras haber recopilado información de las alternativas disponibles en el mercado del sector domótico, se tenía la duda entre Home Assistant y openHAB, ya que las dos disponen de un gran potencial. Sin embargo, se ha elegido la opción de openHAB, ya que las actualizaciones permanecen más estables. Las actualizaciones de Home Assistant crean muchos problemas por ser muy constantes, provocando la modificación y configuración de algunos archivos, afectando al sistema.

CAPÍTULO 3

Especificación de requisitos

La especificación de requisitos tiene como propósito definir las características y funcionalidades para complacer las necesidades de los usuarios del sistema. El siguiente documento sirve para precisar las funcionalidades que adoptan los dispositivos que componen el sistema domótico.

3.1. Alcance

Este proyecto nace de la necesidad de diseñar e integrar un sistema domótico inteligente que actúe de manera lo más autónomo posible para la mejora de calidad de vida de los usuarios. El sistema tiene que ser capaz de comunicarse de manera fiable, rápida y segura.

Otro aspecto a tener en cuenta será el alcance de las señal inalámbrica del sistema, que tendrá que ser capaz de alcanzar todas y cada una de las estancias de la vivienda.

3.2. Perspectiva del sistema domótico

El sistema domótico a desarrollar debe tener compatibilidad con los dispositivos móviles Android e iOS. Además, deben ser de bajo consumo y se alimentarán mediante red eléctrica o pila.

Por otra parte, los dispositivos que forman el sistema domótico se comunicarán de forma inalámbrica mediante WI-FI, Bluetooth, ZigBee o MQTT.

3.3. Tipos de dispositivos

El sistema domótico estará compuesto por los siguientes dispositivos:

- **Centralita domótica.** Se comunicará con el resto de dispositivos mediante el protocolo ZigBee y con el servidor, a través del WI-FI. La alimentación de este dispositivo es mediante red eléctrica.
- **Sensores.** Se dispondrán de varios tipos de dispositivos. Sensores de presencia, agua, humo, puertas y ventanas, temperatura y humedad, etc. La comunicación de todos los sensores serán a través del protocolo ZigBee que se conectarán a la centralita domótica. Estos dispositivos utilizarán una pila de botón para su alimentación.

- **Actuadores.** Este tipo de dispositivos se utilizarán para controlar el apagado y encendido de luces, riego, calefacción, subir y bajar persianas, etc. Depende de la funcionalidad se dispondrá de un tipo de actuador, como por ejemplo: enchufes inteligentes, interruptores de pared, pulsadores inalámbricos, etc. La comunicación se realizará a través de los protocolos ZigBee, MQTT o WI-FI. La alimentación de estos dispositivos será a través de pilas de botón o directamente a la red eléctrica.

3.3.1. Funcionalidad del sistema domótico

Con el objetivo de automatizar una vivienda serán indispensables una serie de dispositivos que proporcionen confort, ahorro de energía, comunicación y seguridad.

En la siguiente tabla se puede observar un breve resumen de las principales funcionalidades del sistema domótico:

Funcionalidad	Descripción
<i>Temperatura</i>	Lectura de la temperatura en grados centígrados.
<i>Humedad</i>	Lectura de la humedad relativa.
<i>Gestión de luces</i>	Controlar mediante un interruptor o relé el apagado / encendido de luces.
<i>Gestión de puertas</i>	Controlar el estado de puertas o ventanas.
<i>Gestión de enchufes</i>	Controlar mediante un relé o enchufe inteligente un determinado punto de corriente.
<i>Gestión de riego</i>	Actuar sobre una electroválvula para encendido / apagado del riego.
<i>Gestión termostato</i>	Actuar sobre calefactores en función de la temperatura de cada estancia.
<i>Autonomía de baterías</i>	Conocer el estado de las de batería de los dispositivos.
<i>Temporización</i>	Controlar el tiempo de apagado o encendido de algunos de los dispositivos.
<i>Sistema inteligente</i>	El sistema gestionará persianas, modos, luces, temperaturas, etc.

3.3.2. Restricciones

Las restricciones existentes más relevantes de este proyecto son las siguientes:

- La función de cada dispositivo tiene que estar disponible para cada usuario en todo momento.
- El sistema tiene que ser escalable para poder añadir más dispositivos en un futuro.
- La comunicación entre los diferentes dispositivos tiene que ser rápida, segura y fiable.
- La distancia entre dos dispositivos tiene que ser la correcta para una buena comunicación entre ellos.

CAPÍTULO 4

Planificación y costes

En este apartado se definirá la planificación que se ha seguido para la consecución de este proyecto. Se detallarán las fases de desarrollo, el tiempo que se ha destinado a cada una de ellas, los recursos empleados y una aproximación de los costes totales.

4.1. Fases de desarrollo

En las fases de desarrollo se recogerán los aspectos más importantes en este proyecto y se realizará una estimación del tiempo de dedicación. Por último, se incluirá un diagrama de Gantt con todas las tareas realizadas en periodos de tiempo.

Presentación de las fases:

- **Fase 1. Definición del proyecto.**
En esta fase se plantearán los objetivos del proyecto y se realizará un estudio sobre las herramientas y protocolos utilizados. La plataforma elegida ha sido la de openHAB y se utilizarán los protocolos MQTT, ZigBee, Bluetooth y Wi-Fi.
- **Fase 2. Especificación de requisitos.**
Una vez definidos los objetivos de la primera fase, se definirán las características y funcionalidades de sistema. Por otra parte, se realizarán pruebas para familiarizarse con las herramientas elegidas.
- **Fase 3. Estudio de otros sistemas alternativos.**
Fase en la que se realizará un estudio de las diferentes plataformas que existen en el mercado y se hará una comparación entre ellas buscando el que más de ajusta a la solución del problema.
- **Fase 4. Toma de contacto con las herramientas utilizadas.**
En esta fase se define un estudio completo de las diferentes herramientas y protocolos utilizados. Para una primera toma de contacto, dicho estudio se realizará sobre la documentación oficial de cada uno de los sistemas domóticos y se buscarán ejemplos con casos prácticos.
- **Fase 5. Diseño Hardware y software.**
Fase encargada de definir todo lo elementos hardware y software que se utilizarán en el presente proyecto.

- Fase 6. Instalación y configuración.**
Se realizará el montaje, instalación y configuración del servidor domótico. Esta fase es la más laboriosa de todas ya que alberga una gran parte de acciones que deben ser realizadas correctamente para un funcionamiento óptimo del sistema.
- Fase 7. Descripción gráfica del entorno real, automatizaciones y pruebas de funcionamiento.**
En esta fase se diseñará un plano de la vivienda con los dispositivos localizados en cada estancia. Además, se realizará una descripción de las automatizaciones que dispondrá el sistema domótico para el hogar. Además, se realizarán baterías de pruebas para comprobar el correcto funcionamiento.
- Fase 8. Transcripción de la memoria.**
Esta etapa abarca la redacción completa de la memoria, aunque esta se realiza en todo el transcurso del proyecto. Se hará una revisión exhaustiva para que sea lo más completa y entendible.
- Fase 9. Revisión de referencias y bibliográfica**
Es la última fase antes de la finalización del proyecto. Así pues, se realizará un repaso general a todas las referencias bibliográficas y se comprobará que toda la información que se ha extraído para este proyecto esté bien citada y en su formato correspondiente.

Una vez definidas todas las fases del proyecto, se realizará una estimación del tiempo invertido en cada una de ellas en el estudio de tecnologías, en la implementación y en la documentación. Esta planificación se puede estudiar siguiendo el diagrama de Gantt.

Tabla resumen con el tiempo de dedicación que se le ha asignado a cada una de las fases:

Cantidad en horas de las etapas del proyecto		
<i>Fases</i>		<i>Tiempo</i>
Fase 1	Definición del proyecto	12 horas
Fase 2	Especificación de requisitos	10 horas
Fase 3	Estudio de otras plataformas	35 horas
Fase 4	Toma de contacto con las herramientas utilizadas	60 horas
Fase 5	Diseño hardware y software	20 horas
Fase 6	Instalación y configuración	160 horas
Fase 7	Descripción gráfica del entorno real	10 horas
Fase 8	Transcripción de la memoria	80 horas
Fase 9	Revisión de referencias y bibliográfica	10 horas
<i>Estimación temporal de la planificación del proyecto</i>		397 horas

La totalidad de horas invertidas en este proyecto asciende a 397. Como se puede observar, la inversión de mayor número de horas ha sido en la implantación y en la instalación y configuración del sistema domótico.

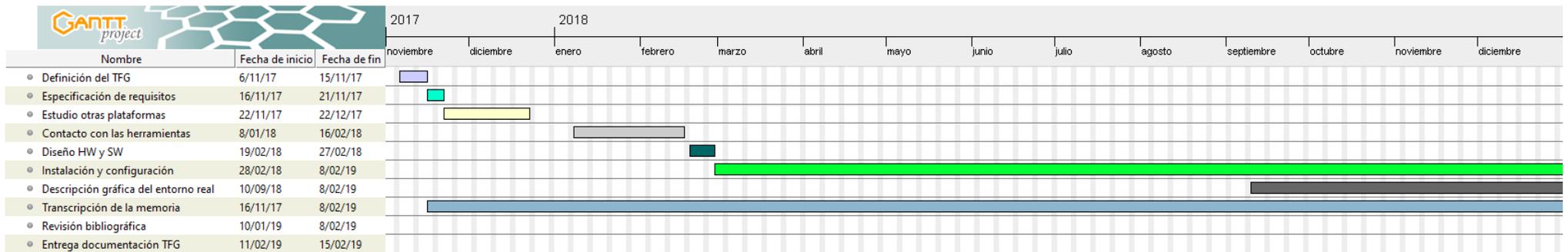


Ilustración 6. Diagrama de Gantt. Imagen propia

4.2. Recursos

En esta apartado se definirán los recursos hardware y software utilizados en este proyecto. Los recursos humanos se sobreentiende que es el creador de este proyecto.

4.2.1. Hardware.

- Ordenador de sobremesa, con un procesador AMD FX-8350 de 8 núcleos a 4,34 GHz y 16 GB de RAM.
- Móvil Xiaomi Pocophone F1, con un procesador Snapdragon 845, 6 GB de RAM y 64 de ROM.
- Raspberry Pi 3B, con un procesador de 64 bits quad-core ARM Cortex-A53 a 1,2GHz y 1GB de RAM.
- HP Gen10, con un procesador AMD 6X86MX a 1,6 GHz, 8GB de RAM y 16TB de ROM.
- Orange pi PC, con un procesador 1.6GHz QUAD Cortex-A7, 1GB de RAM y 16 de ROM.
- Dispositivos Sonoff Basic, Sonoff Dual R2 y Sonoff S20.
- Pulsador Dash Button de Amazon.
- FTDI USB
- Bombillas inteligentes Yeelight.
- Tiras de LED 12v.
- Dispositivos Xiaomi. Esta marca nos proporcionará la centralita domótica y sensores de presencia, puertas y ventanas, temperatura y humedad, detección de agua y humo, vibración y de plantas. Además, también proporcionará dispositivos como interruptores inalámbricos, enchufes inteligentes y pulsadores.

4.2.2. Software

- Sistema operativo Windows 10, utilizado en el ordenador de sobremesa para dar apoyo a los diferentes elementos y servicios.
- Sistema operativo openHABian (basado en Raspbian) instalado en la Raspberry Pi 3, en el cual estará alojado el servidor Mosquitto de MQTT.
- Sistema operativo Ubuntu Server 18.04 instalado en el servidor HP gen10, con el cual se dará soporte a todo el sistema domótico y openHAB.
- Sistema operativo openELEC instalado en una OrangePi PC para la reproducción multimedia mediante KODI.
- Sistema operativo MIUI Global 10.2 instalado en el dispositivo móvil.
- El programa Visual Studio Code que se utilizará como editor de código mediante el plugin de openHAB.
- El programa PuTTY para conexiones remotas SSH desde cualquier otra máquina conectada a la red.
- Los programas ESPEasy y Ternite para la modificación del firmware Tasmota a los diferentes Sonoff.

- Aplicación Mi Home de Xiaomi para la vinculación de las centralitas y dispositivos domóticos.
- El programa Floorplanner para diseñar un plano de la vivienda con sus respectivos sensores y actuadores.

4.3. Proyección de costes

En este apartado se realizará un análisis de los costes que suponen los recursos hardware y software. Como se ha visto en puntos anteriores, la gran mayoría del software utilizado es de código abierto, por esta razón no conllevará ningún incremento en la cuantía total al proyecto.

Por otro lado, los recursos hardware tendrán los siguientes costes:

Nombre	Precio	Udes	Total
<i>HP gen10</i>	240€	1	240€
<i>Discos duros 4TB Red</i>	120€	1	120€
<i>Raspberry Pi 3 B</i>	35€	1	35€
<i>Orange Pi PC</i>	15€	1	15€
<i>Centralita domótica</i>	20€	3	60€
<i>Sensor de temperatura</i>	7€	7	49€
<i>Sensor de presencia</i>	8€	3	24€
<i>Sensores de puerta</i>	5€	5	25€
<i>Sensor de detección de humo</i>	20€	1	20€
<i>Sensor de detección de agua</i>	10€	1	10€
<i>Sensor de plantas</i>	7€	2	14€
<i>Pulsadores</i>	5€	5	25€
<i>Pulsador Dash Button</i>	2€	4	8€
<i>Enchufes inteligentes</i>	10€	3	30€
<i>Relé SonOff basic</i>	4€	8	32€
<i>Relé SonOff Dual R2</i>	9€	4	36€
<i>Interruptores inalámbricos</i>	10€	8	80€
<i>Bombilla Yeelight RGB</i>	9€	1	9€
<i>Bombilla Yeelight Blanca</i>	10€	3	30€
<i>Bombilla Philips Blanca</i>	8€	1	8€
<i>Motor persiana</i>	35€	4	140€
<i>Electroválvula (N/C)</i>	12€	1	12€
<i>Tira LED RGB 5metros</i>	5€	2	10€
<i>Cableado y otros</i>	50€	1	50€
Total			1082€

Precio de los dispositivos redondeados al alza

CAPÍTULO 5

Herramientas y tecnologías utilizadas

En este apartado se llevará a cabo un estudio de las diferentes herramientas, equipamiento y tecnologías utilizadas para poner en marcha el sistema domótico objeto de este TFG respecto a las fases nombradas anteriormente. Además, también se citarán los diferentes protocolos y tecnologías implicadas en cada una de las fases para el correcto funcionamiento del sistema. Se explicarán los conceptos siguientes:

- Características del Servidor HP Gen10.
- Características de la Raspberry Pi 3
- Definición y comunicación del protocolo MQTT.
- Definición del bróker Mosquitto para protocolos MQTT.
- Definición y comunicación del protocolo ZigBee.
- Definición y características del proyecto openHAB.
- Definición de los productos de la empresa Sonoff (ITEAD).
- Definición del chip ESP8266.
- Definición del firmware Tasmota.
- Definición y para que se utiliza el dispositivo FTDI.
- Especificaciones Xiaomi router 3.
- Dispositivos y centralita ZigBee utilizados en el sistema domótico.

5.1. HPE ProLiant MicroServer Gen10

HPE ProLiant MicroServer Gen10 [12], ofrece un servidor de nivel básico, compacto y asequible, diseñado específicamente para entornos de pequeñas oficinas o pequeñas empresas. Este servidor compacto y fácil de usar se puede colocar casi en cualquier lugar. El fácil acceso a las unidades de disco duro, memoria y ranuras PCIe permiten una gestión, instalación y actualización sencilla.

Las especificaciones técnicas son las siguientes: procesador AMD Opteron™ Serie X3000, 2 núcleos a una velocidad de 3,4 GHz, memoria RAM de 8GB DDR4, 2 PCIe 3.0 y 2 controladoras de red Broadcom 5720.



*Ilustración 7. HP Gen10. Disponible en:
<https://www.hpe.com/es/es/product-catalog/servers/proliant-servers.html>*

5.2. Raspberry Pi

Raspberry Pi [13] nace del trabajo de una organización benéfica con sede en el Reino Unido, con el objetivo de poner en manos de la gente de todo el mundo un enfoque digital.

La misión que persigue esta organización es la de crear minicomputadoras de bajo coste y proporcionar un alto rendimiento. Además, incitar a las personas de todo el mundo el arte de aprender, crear, innovar, etc.

5.2.1. Raspberry Pi 3B

En la actualidad se pueden encontrar varios modelos de minicomputadoras de este tipo, pero para la realización de este proyecto nos centraremos en la Raspberry Pi 3.

La Raspberry Pi 3 [14], en adelante RPi3, es actualmente la versión más potente que proporciona la fundación. Se observan cambios considerables en componentes hardware respecto a su antecesora. Estos cambios incrementan sus prestaciones y consiguen un aumento del rendimiento.



Ilustración 8. Raspberry Pi 3. Disponible en:
<https://www.raspberrypi.org/>

Entre las características más importantes de esta tercera generación es la integración de un SoC Broadcom BCM2837, Quad-Core de 64 bits con estructura ARMv8-A, con procesador Cortex-A53 a 1,2 GHz y una memoria RAM de 1GB.

La RPi3 contiene un procesador gráfico Broadcom VideoCore IV 400MHz, que soporta OpenGL ES 2.0. Este procesador gráfico de baja potencia posee una arquitectura DSP bidimensional que hace posible decodificar eficientemente una serie de códecs.

El resto de características se pueden estudiar en la web oficial de la fundación en [15], donde destaca una conectividad Ethernet 10/100 Mbps, chip para Wi-Fi 802.11n y Bluetooth v4.1 (BLE). Además, integra 4 puertos USB, 40 pines GPIO, 1 HDMI, conector Jack de 3.5 mm y una ranura para tarjeta de memoria micro SD.

La RPi3 tiene un precio aproximado de unos 35-38 €.

5.2.2. Puesta en marcha

Para una primera puesta en marcha de la RPi3 se elegirá para su instalación algunos de los sistemas operativos disponibles. El sistema principal recomendado por los desarrolladores es Raspbian. Además, también se pueden instalar otras distribuciones, como Ubuntu MATE, Windows 10 IoT Core, LibreELEC, etc. [16]

En segundo lugar, con la imagen del sistema operativo dentro de la SD, se introducirá en la ranura de la RPI3. A continuación, para una primera configuración, es conveniente conectar la placa a una pantalla mediante HDMI, teclado y ratón.

Respecto a la conexión a internet, se puede conectar mediante cable Ethernet o Wi-Fi, mientras que para su alimentación se utilizará un cable micro USB.

Para finalizar, se esperará que inicie el sistema operativo

5.2.3. Consumo energético

Estas minicomputadoras poseen un bajo consumo energético. El estudio (según Raspi.TV, 2017) [17] respecto al consumo energético se realiza mediante 4 comprobaciones: su estado en reposo, cuando carga la interfaz gráfica, con la visualización de un vídeo a resolución 1080p y con la realización de una captura de vídeo. En la siguiente tabla se muestran los resultados del consumo de los diferentes modelos de Raspberry Pi:

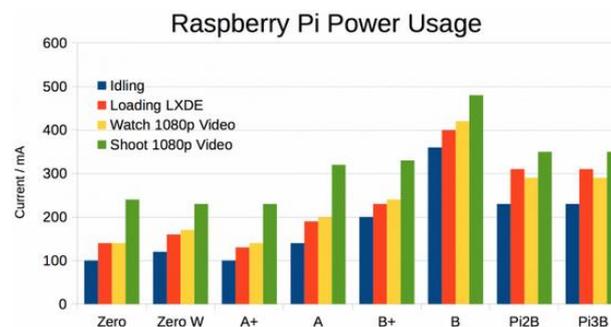


Ilustración 9. Raspi.TV. How much power does Pi Zero W use? [Consulta: 11 de enero de 2018]. Disponible en: <http://raspi.tv/2017/how-much-power-does-pi-zero-w-use>

5.3. Protocolo MQTT

MQTT (Message Queue Telemetry Transport), según su sitio web oficial [18] y (según Geeky Theory, 2018) en [19], es un protocolo usado para la comunicación machine-to-machine (M2M) en el "Internet de las cosas". Este protocolo está orientado a la comunicación de sensores, debido al poco consumo de ancho de banda y puede ser utilizado en la mayoría de los dispositivos empotrados con pocos recursos.



Ilustración 10. MQTT (Message Queue Telemetry Transport). Disponible en: <https://mqtt.org/>

El protocolo MQTT permite a diferentes dispositivos publicar información sobre un *tema* en un servidor que actúa de intermediario entre los publicadores de datos y los consumidores. Una vez que el servidor haya recibido los datos, este los envía a aquellos clientes que se han suscrito a ellos. Los *topic* o *temas* en MQTT, tienen una estructura jerarquizada, donde los clientes deciden en qué nivel se suscriben. Con esta jerarquía, este protocolo es una buena alternativa para redes inalámbricas donde hay restricciones de ancho de banda o conexiones no confiables. Por ejemplo, si se interrumpe la comunicación con un cliente, el servidor mantendrá la información y se la volverá a enviar cuando la comunicación haya sido restablecida.

MQTT, según su estudio en [20], [21], [22] fue desarrollado por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom en el año 1999, para conectar diferentes dispositivos de monitorización utilizados en la industria petrolífera cuyos servidores estaban en ubicaciones distintas. El desafío consistía en enviar datos desde los sensores estando en lugares alejados. Por ello, diseñaron una topología de comunicación publicación/suscripción basado en el modelo TCP/IP.

A sus inicios, era un protocolo específicamente para IBM, pero desde 2013, es un protocolo abierto.

El protocolo MQTT posee puertos TCP/IP estandarizados. El 1883 está reservado para su uso básico, y, por otro lado, tiene registrado el puerto 8883 para conexiones seguras mediante TLS/SSL.

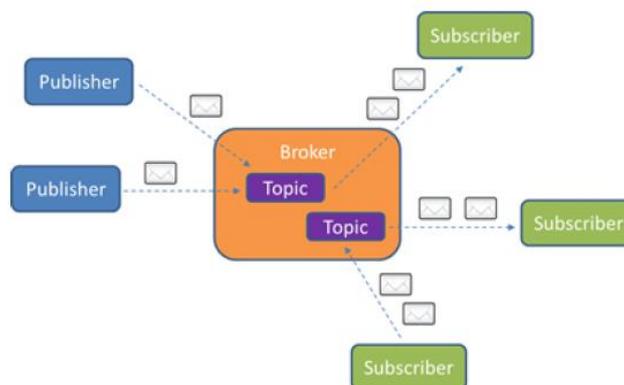


Ilustración 11. Garcia Soria, A. *Funcionamiento elementos MQTT* [Consulta: 27 de diciembre de 2018]. Disponible en: http://wpd.uqr.es/~jorgenavarro/thesis/2017_TFG_AlejandraGarciaSoria.pdf

El servidor puede gestionar una o varias comunicaciones con uno o múltiples clientes simultáneamente. Las comunicaciones MQTT se dividen en diferentes etapas: conexión, autenticación, suscripción, anulación de suscripción, publicación y desconexión.

En la etapa de conexión, el cliente intenta establecer una unión con el intermediario o servidor del protocolo MQTT. Seguidamente, se realiza la autenticación con la secuencia de paquetes CONNECT/CONACK.

Durante la comunicación entre clientes y servidor (bróker), el primero, se puede suscribir a un *topic*, para ello utilizará la pareja de mensajes SUSCRIBE/SUBACK. La estructura de los "*topics* es jerárquica", separados con el carácter "/" para los diferentes niveles.

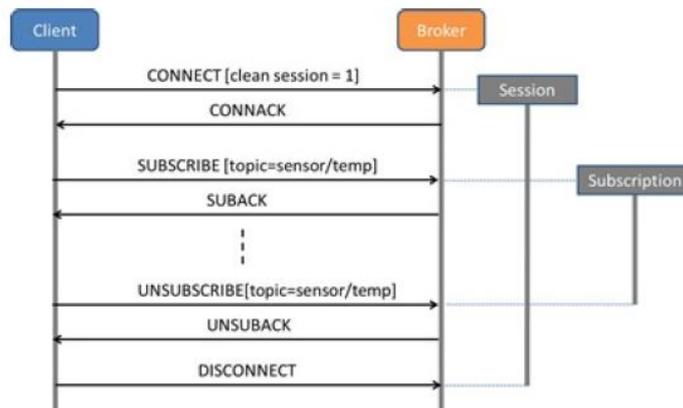


Ilustración 12. Garcia Soria, A. Comunicación establecida entre cliente y bróker [Consulta: 27 de diciembre de 2018]. Disponible en: http://wpd.ugr.es/~jorgenavarro/thesis/2017_TFG_AlejandroGarciaSoria.pdf

Para realizar una anulación de suscripción, también se deben utilizar estas cadenas de caracteres, pero en este caso se realizaría un intercambio con el servidor mediante el mensaje UNSUBSCRIBE/UNSUBACK.

Otra de las alternativas que posee el cliente en la comunicación con el bróker consta del cierre de la conexión, el envío del mensaje DISCONNECT.

MQTT reconoce tres niveles diferentes de calidad del servicio. Los niveles superiores permiten una mayor calidad del servicio, pero esto puede llevar mayor latencia y ancho de banda.

En primer lugar, está el nivel más básico, recibe el nombre de “como máximo una vez” (QoS0). Este nivel envía una secuencia de paquetes PUBLISH al bróker para un *topic* determinado y este lo reenvía una sola vez hacia los clientes suscritos a dicho *topic*. Por lo tanto, no se hace uso de ningún mecanismo que compruebe su recepción y el servidor no mantiene los mensajes en su memoria.

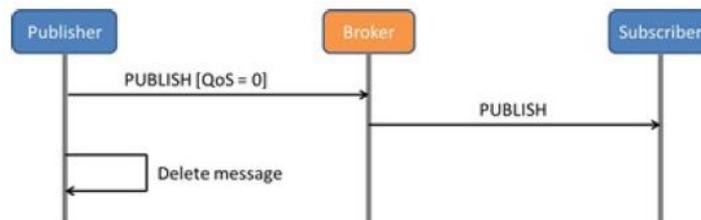
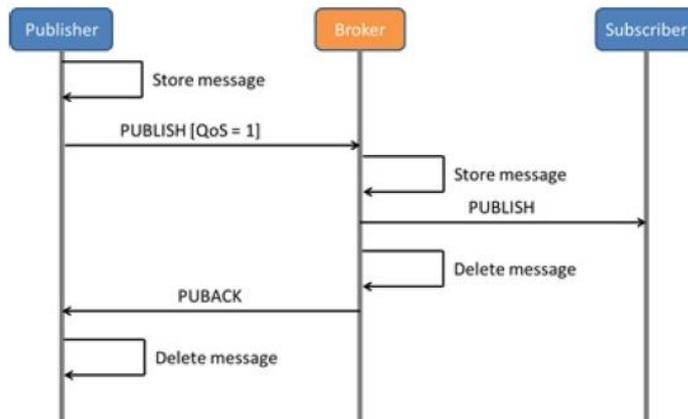


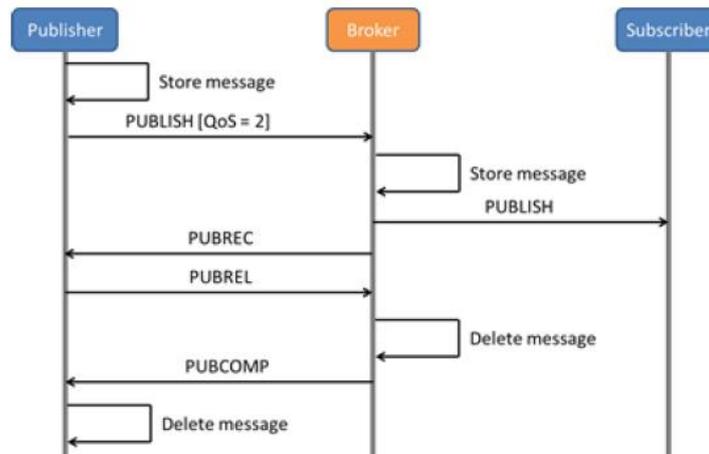
Ilustración 13. Garcia Soria, A. Calidad de nivel QoS0 [Consulta: 27 de diciembre de 2018]. Disponible en: http://wpd.ugr.es/~jorgenavarro/thesis/2017_TFG_AlejandroGarciaSoria.pdf

El segundo nivel de calidad de servicio es llamado “al menos una vez” (QoS1). Utiliza una secuencia de mensajes PUBLISH/PUBACK entre el publicador y el bróker. Además, se utilizará un acuse de recibo para indicar la correcta recepción de la información. De este modo, si no se recibe la información, el bróker la reenviará de nuevo.



Il·lustració 14. Garcia Soria, A. *Calidad de nivel QoS1* [Consulta: 27 de diciembre de 2018]. Disponible en: http://wpd.uqr.es/~jorgenavarro/thesis/2017_TFG_AlejandroGarciaSoria.pdf

Para finalizar, se tiene el tercer nivel de calidad, denominado como “exactamente una vez” (QoS2) o servicio asegurado. La entrega se realiza en dos mensajes. Un primer mensaje compuesto por paquetes PUBLISH/PUBREC y un segundo por PUBREL/PUBCOM. Este mecanismo asegura que la información sea enviada solamente una vez, independientemente del número de intentos realizados.



Il·lustració 15. Garcia Soria, A. *Calidad de nivel QoS2* [Consulta: 27 de diciembre de 2018]. Disponible en: http://wpd.uqr.es/~jorgenavarro/thesis/2017_TFG_AlejandroGarciaSoria.pdf

5.4. Mosquitto (Eclipse)

Según la documentación oficial [23], Mosquitto es un bróker open-source que implementa la versión 3.1.1 del protocolo MQTT. Esto lo transfigura en un servidor de un gran potencial en lo

referente a IoT en un sistema con recursos limitados. Se utiliza para mensajería ligera, tanto en máquinas potentes como en sistemas de baja potencia.

Habitualmente, el bróker Mosquitto consume 3MB de RAM con una conexión máxima de 1000 clientes conectados simultáneamente. Además, es capaz de establecer conexiones con otros servidores MQTT a modo de puente, consiguiendo una conectividad total entre diferentes puntos de este tipo de redes.

El proyecto Mosquitto es miembro de la fundación de eclipse y consta de dos partes bien diferenciadas:

- Un servidor Mosquitto.
- El cliente “mosquitto_pub”, utilizado para realizar publicaciones en el servidor y el cliente “mosquitto_sub”, utilizado para poder suscribirse a un tema.

5.5. Protocolo ZigBee

ZigBee (según Marín Moreno, J y Ruiz Fernández, D. 2007) en [24], es una tecnología inalámbrica de corto alcance y bajo consumo que se define como una solución de baja capacidad para aplicaciones del hogar como la seguridad y la automatización.

El objetivo de esta tecnología no es obtener velocidades muy altas, ya que solo puede alcanzar una tasa de 20 a 250kbps en un rango de 10 a 75 metros, si no conseguir que los sensores tengan un bajo consumo energético. De hecho, algunos dispositivos alimentados con pilas pueden aguantar hasta 2 años. Por tanto, estos dispositivos pasan su mayor parte “durmiendo” para consumir menos.

5.5.1. Bandas de operación

Se basa en un estándar IEEE 802.15.4 y emplea una de las bandas de frecuencia de uso no regulado:

- 868-868,8 MHz: Europa, permite un canal de comunicación, extendido a tres en la revisión de 2006.
- 902.928 MHz: Norte América, hasta cinco canales, extendidos a treinta en 2006.
- 2400-2483,5 MHz: Se puede usar en todo el mundo, hasta 16 canales.

La banda ISM de 2,4 GHz es la más popular ya que no limita geográficamente el ámbito de uso de los dispositivos.

5.5.2. Nodos y topología de red

En una red ZigBee (según efectoLED, 2018) en [25], puede haber hasta 254 nodos, pero según la agrupación que se haga, se pueden crear hasta 255 conjuntos/clusters llegando a

poder tener 64770 nodos. Existe la posibilidad de utilizar varias tipologías de red: en estrella, en malla o en grupos de árboles, como pueden ver a continuación:

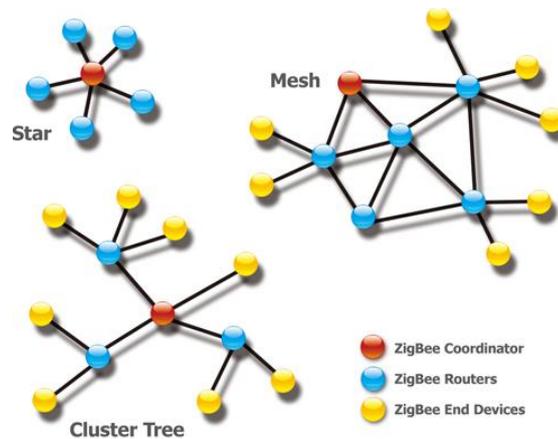


Ilustración 16. Marín ,J y Ruiz,D. Informe técnico protocolo ZigBee [Consulta: 15 de enero de 2018]. Disponible en: https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf

Si hablamos de los tipos de dispositivos, nos encontramos con tres categorías de nodos:

- El primero y más importante ya que ha de existir obligatoriamente en una red. El Coordinador ZigBee se encarga de controlar toda la red y los caminos para su comunicación.
- Por debajo tenemos el Router ZigBee que interconecta los nodos para poder ejecutar código del usuario, es decir, ofrece un nivel de aplicación dentro de la torre de protocolos.
- Por último, el Dispositivo final ZigBee sólo recibe información y se comunica únicamente con el nodo padre. La ventaja de este dispositivo es que puede permanecer dormido y “despertarse” en ciertos momentos para alargar la duración de batería.

ZigBee es un protocolo inalámbrico, desarrollado por la *ZigBee Alliance* que adopta el estándar IEEE 802.15.4 para las capas bajas del modelo OSI. De este modo, se asienta sobre la capa física y de enlace (subcapa de acceso al medio MAC), operando directamente por encima de estos niveles.



Ilustración 17. Marín ,J y Ruiz,D. Informe técnico protocolo ZigBee [Consulta: 15 de enero de 2018]. Disponible en: https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf

5.5.3. Seguridad

La seguridad en la comunicación es uno de los puntos fuertes de ZigBee. El protocolo proporciona mecanismos de control de acceso de los dispositivos a la red (autenticación) y de cifrado (utilizando criptografía de clave simétrica) así como de integridad, asegurando que las tramas transmitidas sufran manipulación con comprobaciones de integridad de mensaje (MIC).

Los aspectos relacionados con las claves criptográficas en ZigBee son fundamentales en la seguridad. ZigBee fundamenta su arquitectura de seguridad en la utilización de criptografía de clave simétrica contando con un elaborado protocolo para la gestión de claves. ZigBee utiliza 3 tipos distintos de claves según se asocien a una red, a un grupo de dispositivos o bien a un enlace entre dos elementos:

- **Clave maestra (master key):** Clave a partir de la cual se generan las diferentes claves de enlace. Dada su importancia, la clave maestra inicial ha de obtenerse por medios seguros.
- **Clave de enlace (link key):** Cifra las comunicaciones punto a punto a nivel de aplicación, y sólo es conocida por los elementos que participan en dicho enlace. Esta clave sólo se comparte entre dos elementos de la red y varía para cada pareja de elementos.
- **Clave de red (network key):** Clave utilizada a nivel de red y conocida por todos los elementos pertenecientes a ésta. La clave de red también se utiliza en agrupaciones de más de dos elementos dentro de una red.

5.5.4. Debilidades en ZigBee

La principal debilidad que existe en la implementación de mecanismos de seguridad en ZigBee, deriva directamente de la limitación de recursos en los nodos, debido a que la mayoría de ellos se alimenta mediante baterías, tienen poco poder de cálculo y poca memoria.

Las claves utilizadas en los dispositivos ZigBee se guardan en memoria, por lo que un intruso puede simplemente leer la clave directamente de la memoria si tiene acceso físico al mismo (con software específico) y no se encuentran implementados mecanismos de seguridad. [26]

5.6. Proyecto openHAB

OpenHAB es un proyecto de software libre creado por Kai Kreuzer. Es un software desarrollado en Java y basado en Eclipse SmartHome. Está diseñado para ser absolutamente independiente del proveedor, así como del hardware y del protocolo de comunicación, haciendo de esto su principal característica.

La interoperabilidad es una de las principales barreras a las que se enfrenta el IoT. Existen muchas soluciones en el mercado pero resulta casi imposible combinarlas entre sí, ya que es el

fabricante el que define los casos de uso y protocolos. OpenHAB surge para salvar esta barrera y reivindicar las necesidades de los usuarios, pudiendo conectar dispositivos de proveedores diferentes a un mismo sistema.

Este software es personalizable y flexible lo que conlleva un coste en términos de tiempo de usuario, el cual se tiene que aprender los conceptos básicos de funcionamiento y configuración para aplicarlo a sus necesidades.

OpenHAB es una apuesta muy acertada, ya que la privacidad de los datos de los usuarios se los gestionan ellos mismos, dicho de otra forma, el usuario decide qué información sale al exterior de la red. Esta solución puede ser utilizada sin acceso remoto, por lo que todos los datos recogidos por los diferentes elementos que compongan el sistema domótico se mantendrán seguros dentro de la intranet del hogar.

El sistema openHAB no nace como una alternativa del sector del “Internet de las cosas” que pretende desbancar a otras soluciones, su pretensión es mejorar el funcionamiento y la integración entre los diferentes sistemas utilizados.

En la documentación oficial de openHAB se recogen los conceptos más relevantes. Las denominadas “cosas”, en adelante *things*, son entidades que pueden ser añadidas al sistema físicamente y pueden desarrollar diferentes funcionalidades a la vez. Cabe resaltar que las *things* no tienen por qué ser dispositivos físicos, también puede ser servicios web o cualquier otra fuente que proporcione información. Cada *thing* dispone de un canal, en adelante *channel*, que añaden alguna funcionalidad al sistema. En una configuración individual, los *channels* suelen ser utilizados a través de “objetos”, en adelante los llamaremos *items*.

Cada *Item* representa una funcionalidad que puede ser utilizada por una aplicación, una interfaz de usuario o para reglas de automatización. Los *Items* pueden tener estados y son capaces de recibir comandos. La unión entre las *things* y los *items* son los “enlaces”, en adelante *bindings*.

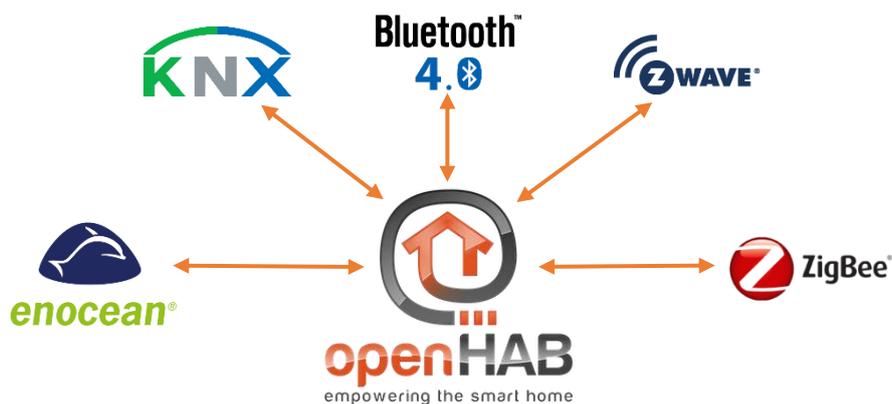


Ilustración 18. Funcionamiento bindings openHAB. Imagen propia

5.7. Sonoff (ITEAD)

Sonoff (según Luis, Llamas, 2018) en [27] y [28], es una empresa productos referentes a la domótica muy interesantes. Es una alternativa atractiva y económica para convertir tu casa en domótica. La gama Sonoff de ITEAD se presenta como una alternativa de bajo coste y muy bien implementada, motivo por el cual está cogiendo una gran popularidad.

De momento, en este proyecto, solo se va a tener esta marca para dispositivos de tipo relé. Los dispositivos de la marca Sonoff utilizados están basados en el chip *ESP8266* y su uso es muy sencillo. Dispone de una entrada (input) y una salida (output). Además, el relé incorpora un botón para accionarlo de forma manual.

A continuación se presentan las características de los Sonoff escogidos:

Sonoff Basic

Especificaciones

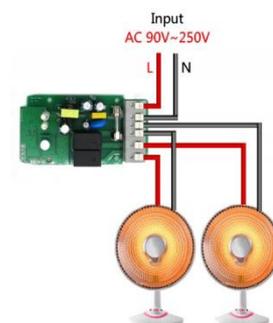
- Rango de voltaje: 90-250v AC (50/60Hz)
- Corriente máxima: 10^a
- Potencia máxima: 2200W
- Dimensiones: 88 * 38 * 23 mm
- Material de la caja: ABS ignífugo
- Humedad: 5% -90% HR, anti condensación.
- Estándar inalámbrico: 802.11 b / g / n
- Mecanismo de seguridad: WPA-PSK / WPA2-PSK
- Temp. de funcionamiento: 0°C-40°C
- 1 Interruptor por dispositivo.



Sonoff Dual R2

Especificaciones

- Rango de voltaje: 90-250v AC (50/60Hz)
- Corriente máxima: 16A
- Potencia máxima: 3500W
- Dimensiones: 114 * 52 * 32 mm
- Material de la caja: ABS ignífugo
- Humedad: 5% -90% HR, anti condensación.
- Estándar inalámbrico: 802.11 b / g / n
- Mecanismo de seguridad: WPA-PSK / WPA2-PSK
- Temperatura de funcionamiento: 0°C-40°C
- 2 Interruptores por dispositivo.



La empresa Sonoff [29] dispone de aplicación propia para sus dispositivos, disponible tanto para iOS como para Android. De momento, la empresa no ha dado el salto a la integración de sus servicios en el sistema domótico openHAB, aunque existe un firmware alternativo para hacer posible dicha unificación mediante MQTT, denominado "*Tasmota*".

5.8. Chip ESP8266

El chip ESP8266E-X según su documentación oficial en [30], presenta una solución completa de Wi-Fi en un SoC integrado desarrollado para satisfacer las necesidades de los usuarios en implementaciones de eficiencia energética y diseños compactos relacionados directamente con el IoT.

Una característica muy importante presente en todas las versiones de este módulo es la posibilidad de conectar sensores u otros dispositivos mediante los pines GPIOs.

En la siguiente tabla se recogen algunas de sus características técnicas más importantes referentes a las conexiones, hardware y software, del módulo ESP8266E-X.

Características	Objetos	Parámetros
<i>Wi-Fi</i>	Estándares	FCC/CE/TELEC/SRRC
	Protocolos	802.11 b/g/n
	Rango de frecuencias	2.4G-2.5G
<i>hardware</i>	CPU	Microcontrolador Tensilica L106 32-bit
	Voltaje de operación	3.0~3.6V
	Corriente de operación	Valor medio 80mA
	Rango de temperatura de operación	-40°~125°
<i>Software</i>	Actualización del firmware	Soporta Cloud Server / Firmware y SDK para la programación rápida en el chip
	Protocolos de red	IPv4, TCP /UDP / HTTP / FTP
	Configuración de usuario	Conjunto de instrucciones AT, Cloud Server, aplicación Android / iOS

Este tipo de módulos son utilizados principalmente para proyectos relacionados con IoT. Algunas de sus aplicaciones más frecuentes son referentes a la automatización del hogar, control de la iluminación, redes de sensores, cámaras IP, etc.

Hay diferentes módulos del chip ESP8266 (según Community Wiki, 2018) en [31], los cuales se diferencian en la distribución y número de pines GPIO, el valor de resistencias, limitadores de corriente, etc.

5.9. Firmware Tasmota

Tasmota (según Michael Ingraham, 2018) en [32], es un firmware alternativo para dispositivos basados en el chip ESP8266, permitiendo interactuar y configurar el dispositivo por vía web. Además, permite configurar temporizaciones, actualizaciones de firmware y es compatible con

sensores de todo tipo. Con la modificación de este firmware en placas con chip ESP8266, se hace posible la integración en la plataforma openHAB mediante el protocolo MQTT.

Existen varias opciones para la modificación del chip ESP8266; mediante el IDE de Arduino, con el programa SonOTA o mediante ESPEasy.

5.10. FTDI

FTDI (Future Technology Devices International) [33], es una empresa privada escocesa de dispositivos semiconductores, especializada en tecnología Universal Serial Bus (USB).

Desarrolla, fabrica y da apoyo a dispositivos y a sus correspondientes controladores de software (drivers) para la conversión de transmisiones serie RS-232 o TTL a señales USB, con el fin de permitir la compatibilidad de dispositivos heredados con los ordenadores modernos.

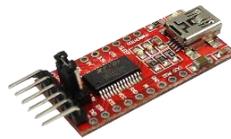


Ilustración 19. FTDI. Disponible en: <https://iotquider.in/ftdi-usb-to-ttl/programming-using-ftdi-usb-to-ttl-serial/>

5.11. Xiaomi Router 3

Para una mejor gestión y configuración de los dispositivos se ha decidido presidir del router de la compañía. Los motivos han sido los siguientes: mayor rendimiento, mejor rango de señal, portabilidad y seguridad.

Después de un estudio y comparación entre los routers existentes en el mercado, se ha elegido el “Xiaomi Router 3”.

Se trata (según Rubén G. G. 2018) en [34], de un dispositivo de gama media. Dispone de un procesador MediaTek con 128MB de RAM y 128MB de ROM, seguridad WPA-PSK, WPA2-PSK, 4 antenas de amplificación de señal, 1 puerto USB 2.0 y 3 puertos Ethernet (1 WAN, 2 LAN). Además, cuenta con dos bandas simultáneas de Wi-Fi AC, 2,4GHz con una tasa de transferencia de hasta 300Mbps con mayor rango de alcance de señal y una segunda banda de 5GHz enfocada para dar conectividad a los dispositivos más modernos, los cuales podrán trabajar con unas tasas de transferencias de hasta 867Mbps, sacrificando el rango de alcance de la señal.

Además, se ha sustituido su firmware de fábrica por el conocido firmware de *Padavan*, de código abierto y similar al *OpenWRT*.

Con el cambio de firmware (según U/Dr_Schmoctor, 2015) en [35], este dispositivo se convierte en un producto más potente, incorporando una interfaz mucho más configurable e intuitiva. El firmware de Padavan dispondrá de soporte de descargas, Servidor y cliente VPN, FTP, cliente Torrent, servidor SMB, UPnP/DLNA y muchas más funcionalidades.

5.12. Dispositivos ZigBee en Mi Home

Mi Home (según Juan Jesús, 2016) en [36], es la aplicación Oficial de Xiaomi que permitirá vincular y gestionar todos los productos de la marca. Así, la aplicación puede comportarse perfectamente como un sistema domótico. Se podrán automatizar luces, sensores, aspiradora, ventilador, cámaras, etc.

Para la utilización de la aplicación Mi Home se necesitará de una cuenta de usuario. Esto permitirá la sincronización de todo tipo de dispositivos a la aplicación.

Los dispositivos referentes a la domótica se vincularán a la centralita domótica o "Gateway". Este dispositivo funciona como controlador domótico, es el corazón de todo. El Gateway controla muchas de las funciones y es un elemento indispensable para controlar dispositivos que cuenten con tecnología ZigBee.



Ilustración 20. Gateway Xiaomi. Disponible en:
<https://npirtube.com/gateways-para-el-entorno-mijia-que-modelos-tenemos-disponibles/>

Características técnicas.

- Dispone de Radio Online.
- Luz nocturna integrada con soporte de 16 millones de colores.
- Sensor de luminosidad.
- Timbres predefinidos y personalizables
- Capacidad de vincularse con otros sensores de la marca como sensores de presencia, de puertas y ventanas, de temperatura, de humo y alguno más.

A la centralita domótica se vincularán todos los dispositivos de la marca que funcionen mediante el protocolo ZigBee. En la siguiente tabla se pueden ver los dispositivos vinculados a la centralita domótica:

Dispositivos ZigBee	
Sensor de presencia Xiaomi Mijia	
Sensor de puertas/ventanas Xiaomi Mijia	
Sensor de temperatura Xiaomi Aqara	

Sensor de temperatura Xiaomi Mijia	
Sensor de detección de agua Xiaomi Mijia	
Sensor de detección de humo Xioami Mijia Honeywell	
Sensor de plantas Mi Flora	
Pulsador Xiaomi Mijia (3 pulsaciones)	
Pulsador Xiaomi Aqara (2 pulsaciones)	
Enchufe inteligente Xiaomi	
Xiaomi Aqara (3 pulsaciones)	
Xiaomi Aqara (2 pulsaciones)	

CAPÍTULO 6

Instalación y configuración

En este apartado se recopilarán los pasos necesarios para la instalación y configuración de los diferentes servicios y/o herramientas utilizadas en este proyecto.

En primer lugar, se realizará la descarga e instalación de los sistemas operativos para los dos servidores. Una vez instalados los sistemas operativos, se realizará la instalación de openHAB y el bróker MQTT, denominado Mosquitto.

A continuación, se explicará con un ejemplo práctico el funcionamiento del sistema domótico openHAB. Además, se procederá al cambio de firmware de los dispositivos de Sonoff y se integrarán al sistema mediante el protocolo MQTT.

Finalmente, Habrá un último apartado de otras configuraciones necesarias para el funcionamiento del sistema.

6.1. Configuración del servidor HP

Como se ha dicho en apartados anteriores, este modelo de servidor será el que contendrá, además del servidor domótico, albergará un gestor para las cámaras y un NAS de datos en red.

El sistema elegido para el servidor será un Ubuntu Server 18.04. La descarga de la imagen se realizará desde la página oficial de Ubuntu, en [37]. Una vez descargada, se montará con la herramienta “win32DiskImager” [38]. Su utilización es sencilla, basta con buscar la imagen del sistema operativo, seleccionar le letra del USB correcta donde se quiera cargar y pulsar sobre “Write”. En pocos minutos estará disponible la imagen para instalar en el servidor.

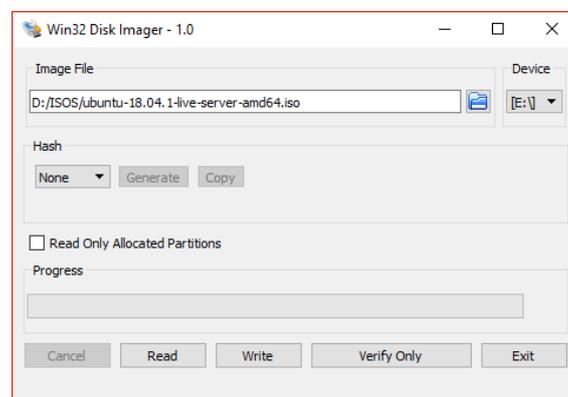


Ilustración 21. Programa Win32Diskimager. Imagen Propia

A continuació, se accedirà a la BIOS del servidor HP y se elegirà la opció de arranque mediante USB. Después del reinicio del sistema aparecerá por pantalla la configuración de instalación.

El primer paso será seleccionar el idioma y la distribución del teclado que se desee.

En el menú de bienvenida, se seleccionará la opción de “Instalar Ubuntu”. Por otro lado, la conexión de red se dejará por defecto, en DHCP ya que de la gestión de direcciones IP estáticas se encargará el router.

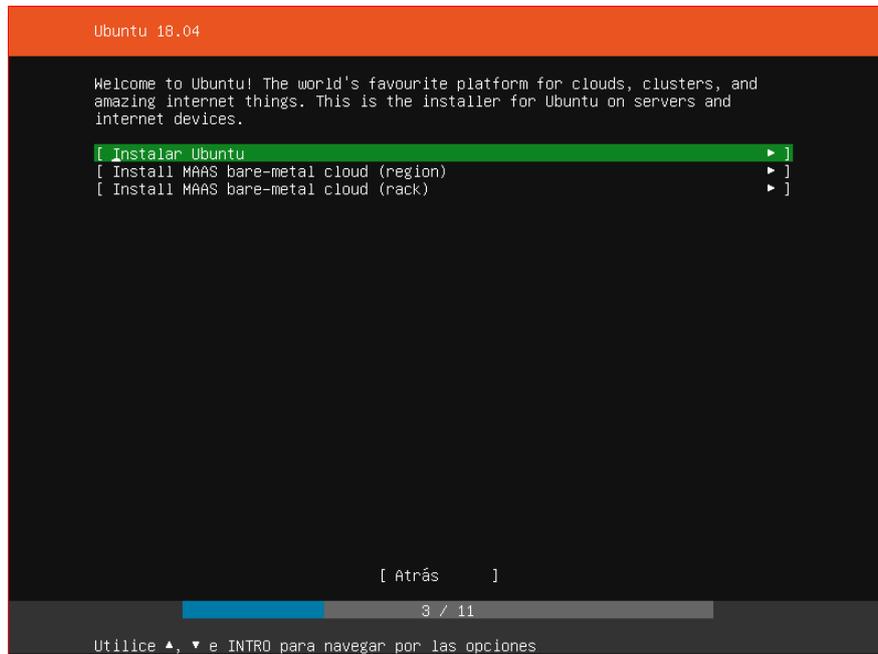


Ilustración 22. Instalación Ubuntu Server 18.04. Imagen Propia

El siguiente paso será la gestión de particiones del disco duro. En este caso, se utilizará toda la memoria del disco para la instalación del sistema operativo.

Ahora, será el momento de la configuración del perfil de usuario. Se añadirá el nombre de usuario y la contraseña de acceso. Finalmente, se procederá a la instalación del sistema operativo en el servidor.

6.2. Configuración Raspberry Pi 3

En esta sección se llevará a cabo una explicación de los pasos necesarios para la descarga y configuración de openHABian en la RPi3. Es una distribución basada en Raspbian [39] y contiene en su interior la última versión de openHAB. La RPi3 contendrá tanto el servidor Mosquitto como una versión a modo de prueba de openHAB, por lo que antes de realizar cualquier prueba de riesgo directamente en el servidor HP, se realiza en el servidor de la RPi3 para prevenir posibles errores y no entorpecer el funcionamiento correcto del servidor HP.

El procedimiento de instalación es semejante a la de Ubuntu server del punto anterior. Para montar la imagen del sistema operativo en la tarjeta SD, utilizaremos la herramienta “win32DiskImager”. Se seleccionará la imagen y la tarjeta SD y se procederá a la espera hasta su

finalización. El siguiente paso será inserta la tarjeta SD en la Rpi3 y esperar a que el sistema (aproximadamente 15-45 minutos).

OpenHABian asigna un usuario y contraseña por defecto, por lo que se deberá cambiar para mayor seguridad. Se accederá a la configuración del sistema en “*sudo raspi-config*” para realizar los siguientes cambios:

```
$ sudo raspi-config
```

1. Cambiar contraseña.
2. Cambiar usuario.
3. Cambiar el nombre del host.
4. Habilitar el SSH para conexiones remotas.

Con la habilitación del protocolo SSH se podrá acceder de forma remota mediante el programa “PuTTY”. Este programa, según Wikipedia [40] es un cliente SSH con licencia libre.

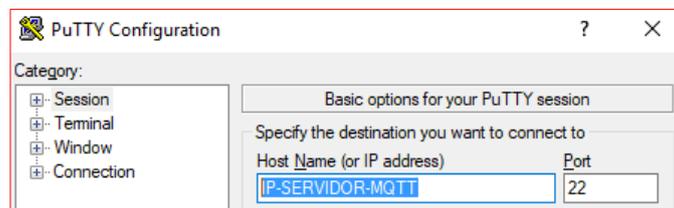


Ilustración 23. Programa PuTTY. Imagen Propia

6.3. Instalación y configuración de Mosquitto

En este apartado se recogen los pasos necesarios para la instalación y configuración de Mosquitto como bróker del protocolo MQTT. Como ya se ha comentado anteriormente, la distribución openHABian dispone de todos los paquetes necesarios para la instalación de Mosquitto. Se accederá al archivo de las herramientas de configuración con el siguiente comando:

```
$ sudo openhabian-config
```

Se buscará entre todas las opciones el servicio Mosquitto y ejecutaremos la instalación del servicio.

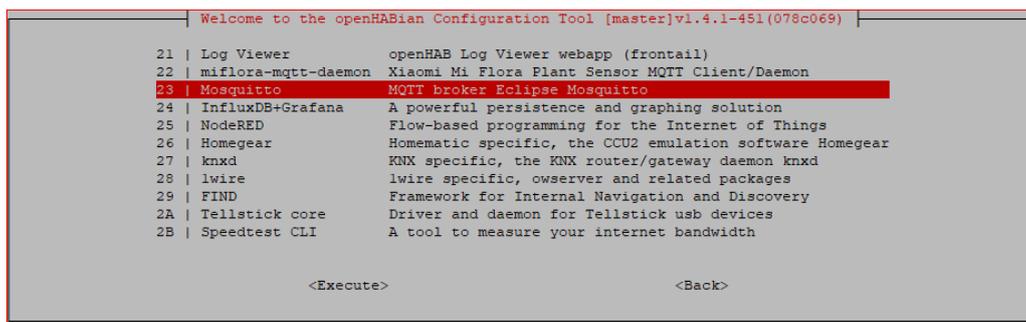


Ilustración 24. Instalación MQTT bróker. Imagen Propia

Si todo el proceso ha finalizado satisfactoriamente, el servidor Mosquitto deberá estar funcionando. Su puerto por defecto es el 1883. Para la comprobación de funcionamiento del servicio, basta con ejecutar el comando “netstat” en consola y observar si está escuchando por su puerto. Además, para más seguridad, con el comando “sudo systemctl status mosquitto.service” podemos comprobar su estado de ejecución.

```
netstat
sudo systemctl status mosquitto.service
```

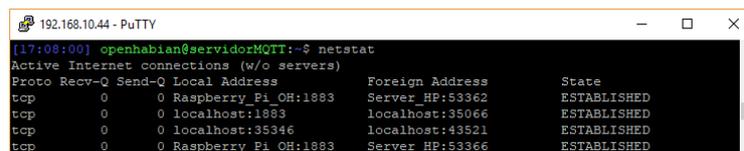


Ilustración 25. Comprobación netstat Bróker MQTT. Imagen Propia

6.4. Modificación y configuración del firmware Tasmota

Para la integración de alguno de los productos de la marca Sonoff se tiene la necesidad de un cambio de firmware para la vinculación con openHAB. Este firmware es denominado Tasmota.

Los accesorios necesarios para la modificación de este tipo de dispositivos son los siguientes:

- FTDI (3,3v) y jumpers macho/hembra.
- Soldador de estaño para soldar los pines a la placa.

Depende de los modelos disponibles en el mercado, los pines a soldar estarán situados en un lugar u otro. En internet se puede encontrar mucha información al respecto.

La soldadura se debe hacer de manera cuidadosa para no dañar ningún componente.



Ilustración 26. Soldado pines Sonoff. Imagen Propia

Una vez realizada la soldadura, se procederá a la conexión la placa con el FTDI, tal y observa en la siguiente ilustración:

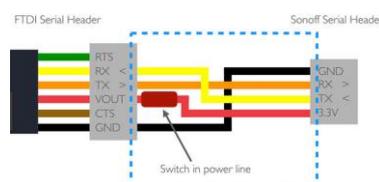
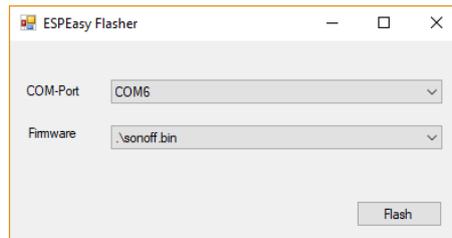


Ilustración 27. Conexión FTDI con el Sonoff. Imagen Propia

A continuació, se procedirà a la descàrrega del firmware Tasmota [41], el programa ESPEasy [42] y el programa Ternite [43] para la configuració de red del dispositiu. El firmware *sonoff.bin* descàrregat se col·locarà en la raïz del programa ESPEasy.

Para el canvi de firmware, el relé Sonoff té que estar en mode flasheo. Depende del model de relé se realitzarà de una manera u otra.

Estanto el Sonoff en mode flasheo, se connectarà el FTDI al ordinador mitjançant la connexió USB y se executarà el programa ESPEasy. Apareixerà una finestra on se afegirà el port COM y se seleccionarà el firmware prèviament posat en la raïz del programa. Finalment, se esperarà a que finalitzi el procés.

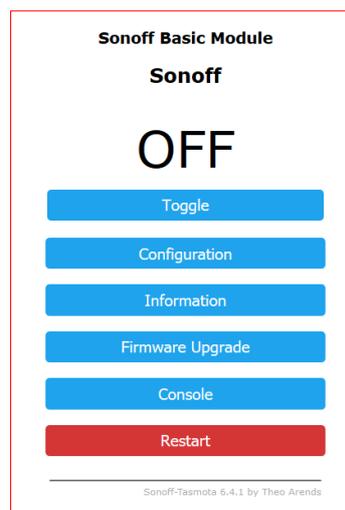


Il·lustració 28. Programa ESPEasy. Imagen Propia

Una vegada finalitzat el canvi de firmware, se haurà d'utilitzar el programa *Ternite*, que contactarà via chat (mitjançant el port COM corresponent) al relé. Con aquest programa se realitzarà la connexió de red entre el dispositiu y el router, assignant-li al dispositiu una direcció IP per poder accedir via web. Els comandos a utilitzar per a la connexió de red en la consola del programa Ternite són:

Ssid [ssid_router]
Password [contraseña_router]

Si el procés de modificació del firmware se ha completat correctament, *Ternite* mostrarà la direcció IP que se li ha assignat.



Il·lustració 29. Interfície web de Tasmota. Imagen Propia

6.5. Instalación y configuración de openHAB

En este punto se recogerá todo lo referente a la instalación y configuración de openHAB. Se definirá la descarga e instalación del sistema domótico y el proceso para un funcionamiento inicial. Además, se explicará la creación de la interfaz de usuario y todos los enlaces que han sido utilizados para el desarrollo de este proyecto. Además, se explicará la configuración para crear grupos y automatizaciones.

Se definirá un apartado con la configuración del protocolo MQTT y su vinculación con el sistema domótico openHAB.

Para finalizar, se definirán otras configuraciones no presentes en el sistema domótico pero que son de una gran trascendencia para su buen funcionamiento.

6.5.1. Descarga e instalación

Tal y como se recoge en la documentación de la página oficial de openHAB, en [44], la instalación se puede realizar de diferentes maneras. El primer paso será ver que versión de Java disponible tiene el servidor. Para ello, desde el terminal, se comprobará la versión:

```
java -version
```

En caso de no disponer de Java instalado, se deberá de instalar con los siguientes comandos:

```
sudo add-apt-repository ppa:webupd8team/java  
sudo apt update  
sudo apt-get install oracle-java8-set-default
```

A continuación, se procederá a la instalación de openHAB. Tal y como nos muestra la el sitio web oficial, la instalación se puede hacer mediante repositorios de paquetes o manualmente. En este caso, el procedimiento de instalación se va a realizar mediante la instalación de paquetes:

1. Para la utilización del protocolo HTTPS, se agregará la clave del repositorio de openHAB Bintray, utilizando las siguientes sentencias:

```
jqget-q0-  
'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' |  
sudo apt-key add -  
sudo apt-get install apt-transport-https
```

2. A continuación, se puede elegir entre tres compilaciones diferentes: la estable, beta o instantánea. La diferencia entre las tres son las siguientes:
 - **Versión estable:** compilación de la última versión oficial con sus características comprobadas y revisadas.

- **Versión beta:** compilación que posee alguna característica en fase de pruebas, por lo tanto, puede que no estén revisadas y comprobadas completamente.
 - **Versión de instantáneas:** los desarrolladores lanzan instantáneas diariamente que incluyen cambios en diferentes complementos a modo de prueba. Su utilización solo debe estar destinado a pruebas y desarrollo.
3. En definitiva, se procederá a utilizar la versión estable, puesto que será la que menos errores contenga. El repositorio se agregará con el siguiente comando:

```
echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee /etc/apt/sources.list.d/openhab2.list
```

4. El siguiente paso será actualizar los paquetes instalados y proceder a la instalación de openHAB.

```
sudo apt-get update  
sudo apt-get install openhab2  
sudo apt-get install openhab2-addons
```

5. La siguiente opción a instalar se utilizará para mantener los enlaces o *bindings* configurados de versiones anteriores en caso de una nueva actualización.

```
sudo apt-get install openhab2-addons-legacy
```

6. Hay una serie de comandos que serán de gran ayuda a lo largo de todo el proceso de aprendizaje de este sistema domótico. Los comandos para poder iniciar y controlar el servicio openHAB:

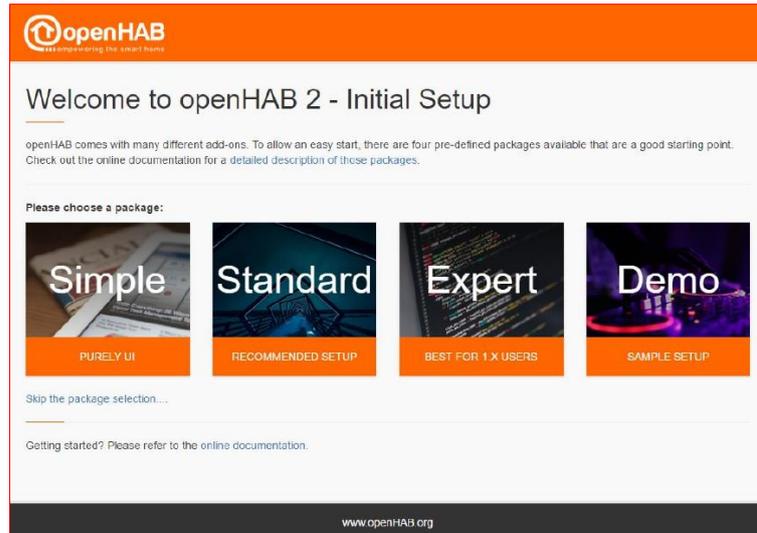
```
sudo systemctl start openhab2.service  
sudo systemctl status openhab2.service
```

7. Finalmente, para que se pueda ejecutar el servicio cuando inicie el servidor, se deberán ejecutar las siguientes sentencias:

```
sudo systemctl daemon-reload  
sudo systemctl enable openhab2.service
```

Con los pasos anteriores realizados correctamente se procederá a la comprobación del funcionamiento de openHAB. Se accederá mediante la URL:

```
http://<IP\_servidorHP\_gen10>:8080.
```

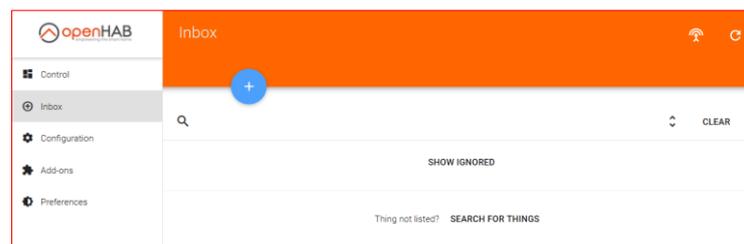


Il·lustració 30. Interfaz de bienvenida de openHAB. Imagen Propia

6.5.2. Interfaz gràfica de usuario.

La primera vez que se inicia el servidor web, se dispone de tres configuraciones diferentes a elegir. Una configuración simple donde los recursos están limitados, una versión estándar para usuarios principiantes y una configuración para expertos. La recomendada por la organización es la “Standard”. Dentro de la configuración estándar podemos encontrar tres modos:

- **Paper UI:** Esta interfaz se utiliza para la instalación de complementos al servidor de manera automática y con interfaz gráfica. Permite descubrir las cosas o *things*, asociarlos a los ítems y conectarlos a los diferentes canales o *channels* de cada ítem. Este método es ideal para usuarios principiantes.



Il·lustració 31. Interfaz Paper UI. Imagen Propia

- **Basic UI:** Es un panel de control para los diferentes interfaces de usuario o *sitemaps* que se tengan creados. Es la parte visual del sistema y dispone de un diseño adaptable a cualquier tipo de pantalla y actualización en tiempo real.

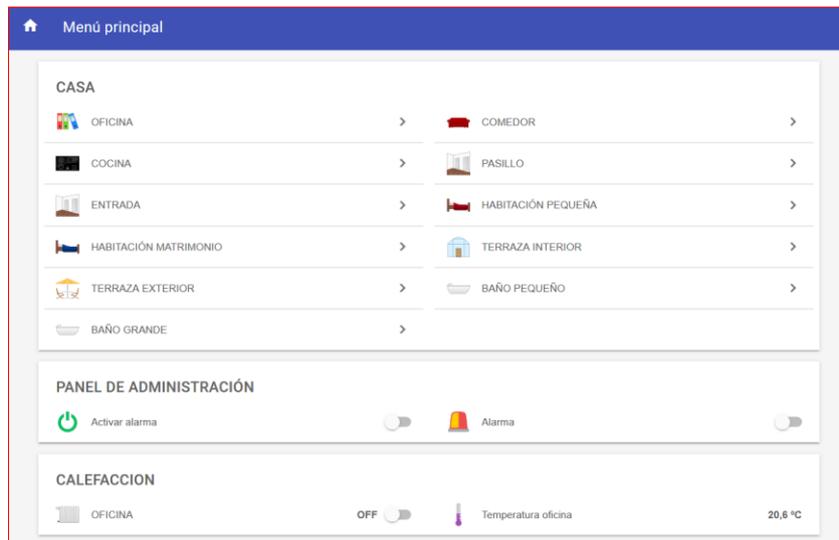


Ilustración 32. Interfaz Basic UI. Imagen Propia

- **HABPanel:** Es una interfaz de usuario enfocada a tabletas electrónicas. Funciona mediante creación de paneles donde se van añadiendo los dispositivos y servicios.

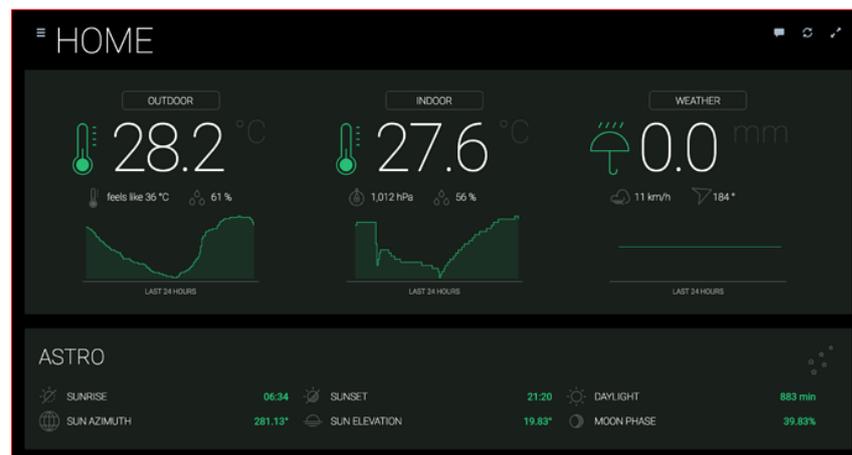


Ilustración 33. Interfaz HABpanell. Imagen Propia

6.5.3. Configuración y funcionamiento inicial.

El desarrollo de esta sección puede ser definida en dos partes. La primera, la vinculación de dispositivos de forma automática desde la interfaz amigable que dispone openHAB.

La segunda, para usuarios más experimentados, la vinculación de dispositivos mediante los archivos de configuración que dispone openHAB.

En los siguientes puntos a tratar se definirá un ejemplo de configuración automático y otro mediante archivos de configuración.

Configuración automática

Para una primera configuración de los dispositivos o servicios reconocidos por el sistema domótico openHAB, se debe utilizar la interfaz “Paper UI”.

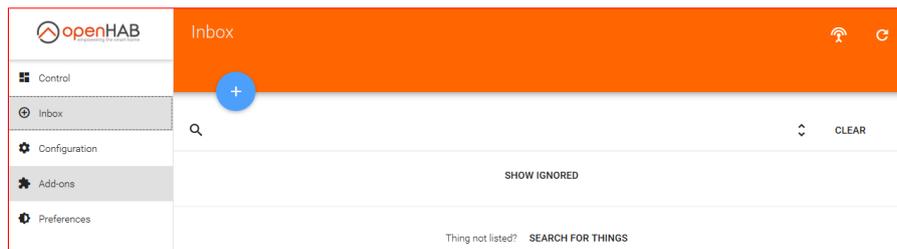
Cabe destacar, tal y como se ha comentado anteriormente, que las *things* pueden ser tanto dispositivos físicos como no físicos. Por ejemplo, un sensor es un dispositivo físico, mientras que la predicción del tiempo se considera un servicio de openHAB no físico.

Iniciada la interfaz de “Paper UI” se procederá a integrar un servicio mediante un binding. El primer paso será acceder a la opción “add-ons > bindings” e instalar el binding “NTP Binding”, que se utilizará para mostrar la fecha y hora completa basándose en el servidor NTP configurado.



Ilustración 34. Instalación binding NTP. Imagen Propia

A continuación, dentro de la opción de “Inbox”, mediante el botón azul se buscará el *binding* previamente instalado.



En la siguiente ventana de configuración, se añadirá el nombre del *thing*, la localización y el servidor que proporcionará la fecha y hora. Los otros parámetros se pueden configurar en función de la necesidad respecto al tiempo en los intervalos de refresco.

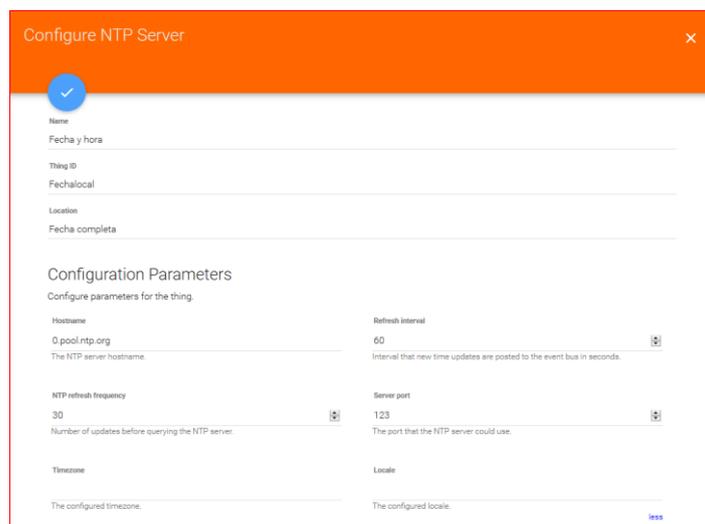
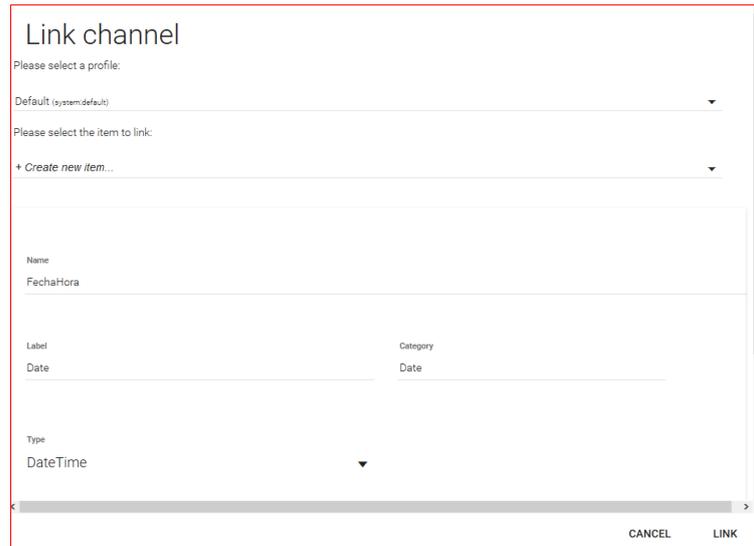


Ilustración 35. Configuración binding NTP. Imagen Propia

Si la configuración se ha realizado correctamente, en “*Configuration > Things*” aparecerá el servicio enlazado con el estado “en línea”.

En este momento entran en juego los canales o channels. Se deberán enlazar los things a los ítems y cada ítem con sus correspondientes canales. Por ejemplo, el binding NTP dispone de dos canales. El primer canal devuelve un dato de tipo fecha (*datetime*) mientras que el segundo devuelve un dato de tipo texto (*string*).

En el interior de cada canal se presenta su configuración. Para la configuración añadiremos el nombre que presentará el ítem, el texto que aparecerá en la interfaz gráfica junto con la fecha completa y el tipo de datos de retorno, en este caso de tipo fecha.

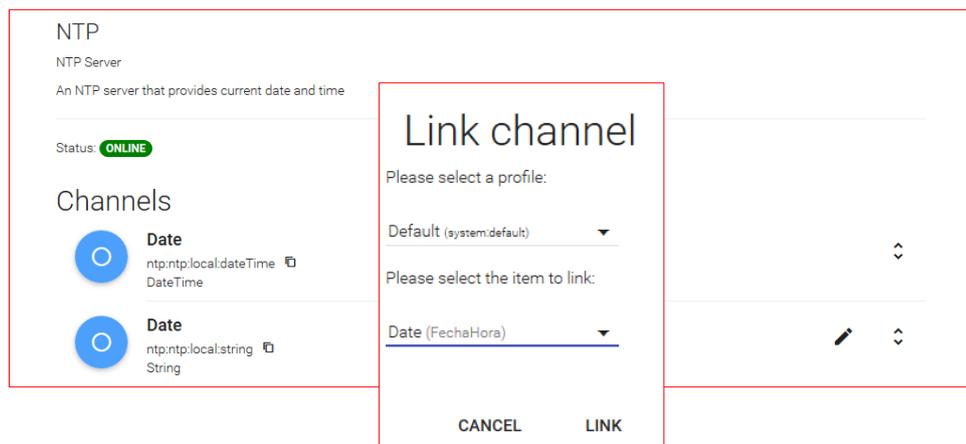


The screenshot shows a dialog box titled "Link channel". It contains the following elements:

- A dropdown menu for "Please select a profile:" with "Default (system:default)" selected.
- A dropdown menu for "Please select the item to link:" with "+ Create new item..." selected.
- A "Name" field containing "FechaHora".
- A "Label" field containing "Date" and a "Category" field containing "Date".
- A "Type" dropdown menu containing "DateTime".
- "CANCEL" and "LINK" buttons at the bottom right.

Ilustración 36. Configuración del canal. Imagen Propia

A continuación, haciendo clic sobre el punto azul, aparecerá el procedimiento para enlazar el canal al ítem. Para ello, se deberá de seleccionar el ítem creado anteriormente, tal y como se muestra en la siguiente ilustración:



The screenshot shows the NTP server configuration page with a "Link channel" dialog overlaid. The background page shows:

- NTP Server status: ONLINE.
- Channels list:
 - Date (ntp:ntp.local.dateTime, DateTime)
 - Date (ntp:ntp.local.string, String)

The "Link channel" dialog is the same as in the previous image, but with "Date (FechaHora)" selected in the "Please select the item to link:" dropdown.

Ilustración 37. Enlazar canal al ítem. Imagen Propia

Finalmente, la pestaña de “Control” mostrará por pantalla la configuración realizada previamente.

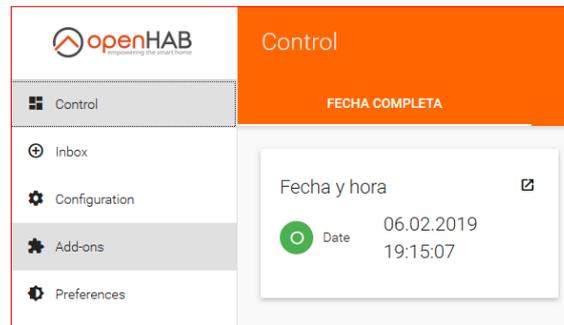


Ilustración 38. Prueba correcto funcionamiento NTP. Imagen Propia

Configuración manual

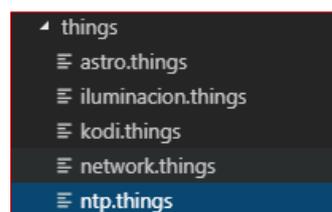
La integración de dispositivos y servicios de forma manual tiene una mayor dificultad, pero con la experiencia, se obtienen unos resultados mucho más óptimos.

Los archivos de configuración se encuentra en la ruta “/etc/openhab2/”. Hay distintos archivos de configuración, aunque los más utilizados serán los archivos *items*, *rules*, *sitemaps* y *things*. Para una simulación, se va a proceder a realizar el mismo ejemplo que en el caso anterior, donde se realizaba la configuración del enlace de NTP.

Otra forma de acceder a dichos archivos de configuración es usando el programa Visual Studio Code, para acceder de forma remota desde otro ordenador.

En la documentación oficial de openHAB, en [45] se obtiene toda la información necesaria para la integración de los dispositivos y /o servicios.

El primer paso será crear un nuevo archivo dentro de la carpeta *things*, como por ejemplo “*ntp.things*”.



A continuación, y siguiendo la documentación oficial, se definirá la nomenclatura del *thing*:

```
Thing <binding_id>:<type_id>:<thing_id> "Label" @ "Location" [<parameters>]
```

Respecto a la sintaxis anterior, se definirá el *thing* aplicado al binding NTP.

```
Thing ntp:ntp:local "NTP" [ hostname="n1.pool.ntp.org",
refreshInterval=60, refreshNtp=30 ]
```

Para la creación de los ítems, se utilizará como ejemplo la sintaxis siguiente:

```
itemtype itemname "labeltext [stateformat]" <iconname> (group1) ["tag1",  
"tag2", ...] {bindingconfig}
```

Finalmente, se ajustará la nomenclatura del ítem descrito arriba al binding NTP, así:

```
DateTime Datos_NTP "Fecha completa [%1$tA, %1$td.%1$tm.%1$tY %1$tH:%1$tM]"  
<time> { channel="ntp:ntp:local:dateTime"}
```

Esta es la filosofía de la configuración tanto de forma manual como automática de los diferentes dispositivos y/o que proporciona la plataforma openHAB.

6.5.4. Sitemaps

La implementación de los archivos de configuración *sitemaps* es fundamental para un aspecto visual detallado y ordenado de la interfaz de usuario “Basic UI”.

Para la creación de un *sitemap* se necesita de los archivos de configuración. Como se ha visto anteriormente, la definición de un *sitemap* será tal como así: <nombre>.sitemaps.

Para la visualización en la interfaz gráfica de usuario “Basic UI” se creará un *sitemap*. La sintaxis respecto a la documentación oficial de openHAB es la siguiente:

```
sitemap <sitemapname> label="<title of the main screen>"  
{  
    ItemType = ItemName label = "Description on the item"  
}
```

Es obligatorio que el archivo comience con la palabra *sitemap*, seguido del nombre y una etiqueta con el título. Los ítems *ItemType* e *ItemName* deberá ser el mismo que el fichero de definición, tal que así:

```
sitemap Home label= "Menú principal"  
{  
    Default item=Datos_NTP label="Prueba Fecha y Hora"  
}
```

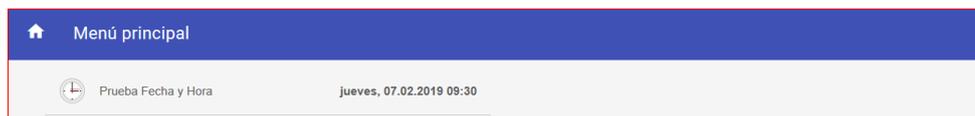


Ilustración 39. Configuración NTP en sitemap. Imagen Propia

6.5.5. Bindings utilizados

En este apartado se definirán todos los *bindings* que se han ido instalando a lo largo del presente proyecto. Como se puede estudiar en [46], existen muchos más.

- **Astro Binding**
Este binding hace referencia a información sobre la luna y el sol. Por ejemplo, puede dar datos de cuando amanece, atardece, la estación del año, la radiación solar, el día que se producirá un eclipse, etc.
- **GPSTracker Binding**
GPSTracker proporciona información sobre la situación en tiempo real de las personas inquilinas en la vivienda. Es muy interesante para, por ejemplo, la desconexión de la alarma automáticamente cuando el usuario esté relativamente cerca de su hogar.
- **HTTP Binding**
Útil para enviar comandos a los elementos con la pretensión que actualice su estado mediante sondeos en una URL. Por ejemplo, se utilizará para acceder al servicio de las cámaras de vigilancia.
- **KODI Binding**
Este binding enlazará cualquier dispositivo que se tenga KODI con openHAB para la reproducción de películas, videos, imágenes, etc.
- **Network Binding**
Útil para saber si los dispositivos de nuestra red están en conexión, además de indicar la latencia y la hora a la que se ha actualizado dicho estado.
- **NTP Binding**
Es utilizado para mostrar la fecha y hora basándose en el servidor NTP que tengamos configurado.
- **SystemInfo Binding**
Proporciona información del dispositivo donde esté instalado openHAB. Es muy útil para saber la memoria RAM que está utilizando el sistema, la memoria ROM disponible, la temperatura del servidor, etc.
- **Xiaomi Mi IO Binding**
Útil para la integración de diferentes dispositivos fabricados por la marca china Xiaomi. Por ejemplo, se puede integrar el robot aspirados, lámparas de escritorio, purificadores de aire, etc.
- **Xiaomi Mi Smart Home Binding**
Este binbing permite la integración de dispositivos de la marca Xiaomi para el sistema domótico del hogar.

- **Yeelight Binding**
Útil para la integración de productos de la marca Yeelight, empresa dedicada a la iluminación.
- **MQTT Binding**
Permite que openHAB actúe como servidor MQTT. Puede ser utilizado para accionar relés desde la mota MQTT y encender / apagar cualquier dispositivo conectado.
- **Weather Binding**
Proporciona información sobre el tiempo actual, como la temperatura exterior, humedad, condición meteorológica, velocidad del viento, etc.

6.5.6. Automatizaciones (Rules)

Las automatizaciones se realizan mediante la programación de reglas (*rules*). Las automatizaciones son el comportamiento que adoptará el sistema domótico, por lo tanto, es de las partes más importantes de la domótica. La carpeta para su desarrollo se encuentra en `/etc/openhab2/rules` y en el sitio web oficial se dispone de toda la información necesaria para su aprendizaje. La sintaxis es la siguiente:

```
rule "<RULE_NAME>"
when
  <TRIGGER_CONDITION> [or <TRIGGER_CONDITION2> [or ...]]
then
  <SCRIPT_BLOCK>
end
```

Un buen ejemplo de uso de automatizaciones sería con la utilización del binding de Astro, mediante la ejecución de un interruptor ON/OFF, con estado ON cuando se atardezca y OFF cuando amanezca. De este modo se podría programar automatizaciones como luces, subir y bajar persianas, etc.

```
rule "Amanecer"
when
  Channel 'astro:sun:home:rise#event' triggered START
then
  modoNoche.postUpdate(OFF)
end

rule "Atardecer"
when
  Channel 'astro:sun:home:set#event' triggered START
then
  modoNoche.postUpdate(ON)
end
```

6.5.7. Integración myOpenHAB

MyopenHab [47] es un servicio de nube creado por los desarrolladores de openHAB que permite el acceso remoto a las instancias sin la necesidad de una configuración VPN. Además, permite el envío de notificaciones “Push” al teléfono.

El primer paso será el de la instalación el conector desde la plataforma openHAB, en “*add-ons > Misc > openHAB Cloud Connector*”. Este conector realizará una conexión cliente/servidor.



Ilustración 40. Conector myOpenHAB. Imagen Propia

En la ruta “*/usr/share/openhab2/addons*” y con el comando *wget* se instalará el *binding openhabCloud*:

```
wget  
https://dl.bintray.com/openhab/bin/org.openhab.io.openhabcloud_1.9.0.  
201612192331.jar
```

La instalación del *binding* generará dos ficheros, *UUID* y *Secret*. Estos se pueden encontrar en las rutas */var/lib/openhab2* y */var/lib/openhab2/openhabcloud* respectivamente. El contenido de estos ficheros se añadirá junto al registro en la web oficial <https://myopenhab.org/>.

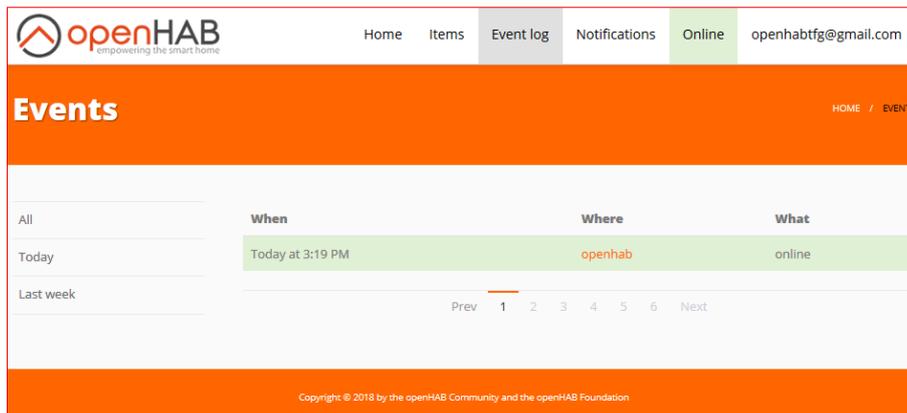


Ilustración 41. Interfaz myOpenHAB. Imagen Propia

6.5.8. Configuración de MQTT

Con la instalación del binding de MQTT, se procederá a la configuración del mismo para que el sistema openHAB actúe como cliente del protocolo MQTT. El fichero de configuración se encuentra en “*/etc/openhab2/services/mqtt.cfg*”.

En el fichero *mqtt.cfg* se modificarán las siguientes líneas:

```

# URL to the MQTT broker, e.g. tcp://localhost:1883 or
ssl://localhost:8883
ejemplobroker.url=tcp://192.168.X.X:1883

# Optional. Client id (max 23 chars) to use when connecting to the
broker.
# If not provided a random default is generated.
ejemploMqtt.clientId=openhab2

#broker.user=mqttopenhab# Optional. User id to authenticate with the
broker.
ejemploUsuario.user=sonoffmqtt

# Optional. Password to authenticate with the broker.
ejemploPass.pwd=passUsuario
  
```

Una vez configurado el protocolo MQTT, se procederá a la definición de elementos que utilizarán este protocolo. Según la documentación oficial, la sintaxis utilizada para esta configuración es la siguiente:

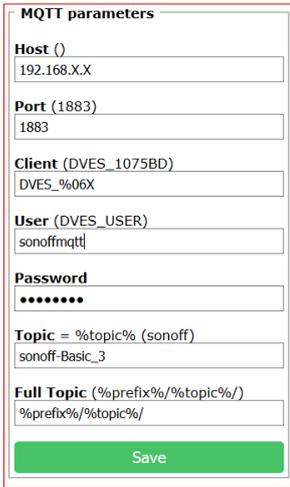
```

Item myItem {mqtt="<direction>" [<broker>:<topic>:<transformer>],
<direction> [<broker> : <topic> : <type>: <transformation>], ...}
  
```

Este proyecto, solo dispone de los relés de Sonoff y del sensor de plantas funcionando con el protocolo MQTT. En un futuro, se pretende añadir muchos más dispositivos mediante este protocolo.

El siguiente ejemplo hace referencia al extractor de aire de la cocina, donde la acción que implementa es la de apagado/encendido. Para ello, se utilizará el relé Sonoff Basic y se accederá con su dirección IP a su configuración.

Una vez accedido a la dirección IP del dispositivo, en "*configuration > Configure MQTT*", se configurarán los parámetros para la conexión con el servidor Mosquitto, tal y como muestra la siguiente imagen:



MQTT parameters
Host ()
 192.168.X.X
Port (1883)
 1883
Client (DVES_1075BD)
 DVES_%06X
User (DVES_USER)
 sonoffmqtt
Password
 ••••••••
Topic = %topic% (sonoff)
 sonoff-Basic_3
Full Topic (%prefix%/ %topic%/)
 %prefix%/ %topic%/

Save

*Ilustración 42. Configuración MQTT de Sonoff.
 Imagen Propia*

Finalmente, siguiendo la sintaxis proporcionada por la documentación oficial y teniendo configurado el servidor y cliente MQTT, se procederá a la creación del ítem.

```
Switch Extractor_Cocina "Extractor cocina" <pump> {mqtt="> [broker:cmdnd/sonoff-Basic_3/POWER:command:*.default],<[broker:stat/sonoff-Basic_3/POWER:state:default]"}
```

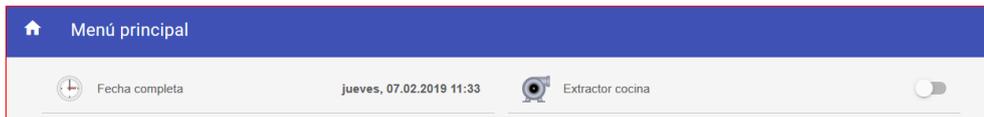


Ilustración 43. Prueba relé Sonoff. Imagen Propia

6.6. Otras Instalaciones y configuraciones

En este apartado se definirán otras instalaciones y configuraciones. Por una parte, la instalación y configuración del router que aportará portabilidad, escalabilidad y seguridad. Por otra parte, para mejorar la accesibilidad a los archivos de configuración, se instalará el editor de código *VS Code*. Finalmente se explicará la integración de la centralita domótica mediante la aplicación de Mi Home de Xiaomi a la plataforma openHAB.

6.6.1. Instalación y configuración del router

A veces, se hace necesario que las direcciones IP de nuestros dispositivos no cambien, ya que es necesario tener acceso remoto a los servidores. Otro motivo es la organización, ya que interesará tener los dispositivos etiquetados en orden de rangos de direcciones IP para cada tipo de ellos.

En resumidas cuentas, se etiquetarán los dispositivos en el DHCP server del router y se asignará una dirección IP del rango del tipo de dispositivo y se le asociará con la MAC. De este modo, el dispositivo siempre se conectará a la dirección IP estática.

El router estará en una LAN diferente respecto al de la compañía. Dicha configuración se realizará en la “*configuración avanzada >> LAN*”, donde se añadirá la dirección IP y la máscara de subred para las nuevas direcciones de IP.

6.6.2. Instalación y configuración de la aplicación Mi Home

Para la sincronización en la plataforma openHAB de los dispositivos de la marca Xiaomi, se necesitará previamente de su vinculación en la aplicación oficial Mi Home.

Como ya se ha explicado anteriormente, la centralita domótica o *gateway* será la encargada de controlar los dispositivos como sensores, pulsadores, interruptores, etcétera, mediante el protocolo ZigBee.

La vinculación de la centralita a la aplicación Mi Home se realiza de la manera siguiente:

1. Una vez descargada la aplicación Mi Home en el dispositivo móvil, bastará con pulsar unos segundos el botón incrustado en la centralita domótica y la aplicación la detectará automáticamente.

2. A continuación se elegirá la red Wi-Fi a la que se vinculará con su respectiva contraseña.
3. Una vez realizada la conexión con éxito, se tendrá el dispositivo vinculado.



Ilustración 44. Vinculación Gateway Xiaomi. Imagen Propia

4. Para la detección automática de la centralita con la plataforma openHAB se necesitará activar el modo desarrollador en ella. Dentro de la centralita domótica, "Acerca de", se pulsará en repetidas veces el icono "Plug-in version" para la habilitación de las opciones ocultas del sistema. Por último en la pestaña "Wireless communication protocol", se activará el modo desarrollador.
5. Con todos los pasos realizados correctamente y mediante el *binding* de *Xiaomi Mi Smart Home* instalado, la centralita aparecerá preparada para ser vinculada a la plataforma openHAB.

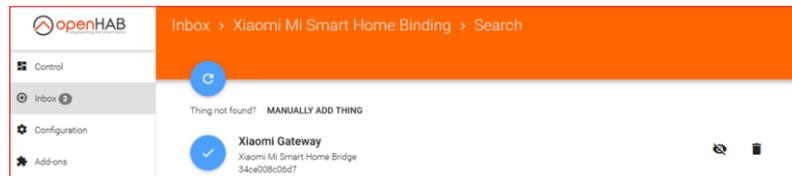
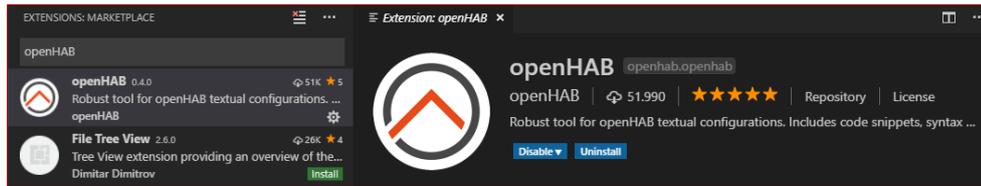


Ilustración 45. Vinculación Gateway con openHAB. Imagen Propia

6.6.3. Instalación y configuración de Visual Studio Code

VS Code (según Wikipwdia, 2019) en [48], es un editor de código fuente multiplataforma portable, ligero y rápido. Nos permitirá trabajar con diversos lenguajes de programación. Este editor ha sido desarrollado por Microsoft y se puede utilizar tanto para Windows, Linux y macOS.

Permite, mediante el plugin de openHAB, trabajar y desplegar proyectos sin necesidad de utilizar los archivos de configuración del servidor. Para ello, necesita de la instalación y configuración del protocolo SMB (*Samba*).



Il·lustració 46. VS Code Plugin openHAB. Imagen Propia

Instalación SMB

SMB es un protocolo de red que permitirá compartir archivos, impresoras, etc. La instalación se realizará con este comando:

```
Sudo apt-get install samba samba-common-bin
```

El archivo de configuración de Samba estará en “/etc/samba/smb.conf” y su primera configuración será habilitar el soporte para Windows, “descomentando” la siguiente línea:

```
wins support = yes
```

A continuación, se procederá a la instalación de los paquetes del repositorio.

```
[openHAB2-userdata]
comment=openHAB2 userdata
path=/var/lib/openhab2
browseable=Yes
writeable=Yes
only guest=no
public=no
create mask=0777
directory mask=0777

[openHAB2-conf]
comment=openHAB2 site configuration
path=/etc/openhab2
browseable=Yes
writeable=Yes
only guest=no
public=no
create mask=0777
directory mask=0777
```

Se creará un usuario con su contraseña para mantener la privacidad de los archivos de configuración.

```
sudo smbpasswd -a openhab
```

El usuario creado tendrá permisos de escritura a los archivos de configuración.

```
sudo chown -hR openhab:openhab /etc/openhab2
```

Finalmente, se reiniciarà el servici SMB.

```
sudo systemctl restart smbd.service
```

Acceso remoto

El acceso remoto a las carpetas compartidas del servidor openHAB, se accederá desde el PC local mediante la dirección IP. Se deberá de poner el usuario y contraseña para autenticarse.

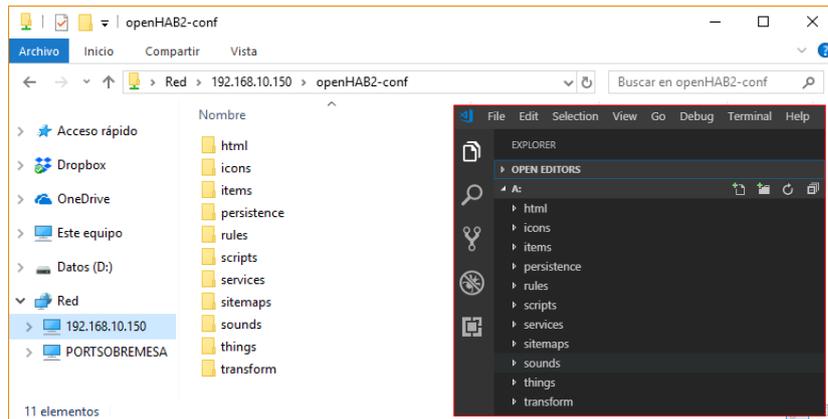


Ilustración 47. Acceso a la carta compartida openHAB. Imagen Propia

	Sensor de presencia		Cámara de vigilancia
	Sensor de temperatura		Servidor openHAB y MQTT
	Luz con interruptor		Extractor de humo
	Red		Sensor de puertas / ventanas
	Enchufe inteligente		Electroválvula
	Medidor de consumo		Reproductor multimedia KODI
	Detector de humo		Router WI-FI
	Pulsador		

7.3. Ejemplos de automatizaciones

Se denomina automatización (Según Julián Pérez Porto y María Merino. 2016) en [50], a hacer que determinadas acciones se vuelvan automáticas, es decir, que se desarrollen por sí solas y sin la participación directa de un individuo.

Las automatizaciones en la domótica tienen como objetivo principal hacer la vida más fácil, mejorar la seguridad y el confort. Las automatizaciones que dispone la vivienda unifamiliar a día de hoy son las siguientes:

- Automatización total de la iluminación en gran parte de la vivienda. Las luces del pasillo y entrada tienen inteligencia propia para encenderse y apagarse automáticamente dependiendo de la luminosidad o de la salida y puesta del sol. Por ejemplo, cuando amanece, se dispara el “trigger” y cambia el disparador del estado ON a OFF, por lo tanto, las luces pasan a ser manuales, mientras que si se atardece y el estado cambia a ON, las luces del pasillo son automáticas siempre y cuando detecte movimiento (sensor de presencia).
- Automatización de persianas, permitiendo controlar estos elementos por voz, manualmente, con la plataforma openHAB o programadas. Esta automatización viene relacionada con la anterior, se subirán las persianas si el modo noche está en el estado OFF y viceversa. Además, si el sensor de detección de agua cambia de estado OFF a ON, o lo que es lo mismo, que está lloviendo, las persianas se bajarán.
- Sistema de control de la calefacción para todas las estancias de la casa, dependiendo de la temperatura ambiente de cada una de ellas. Dependiendo de la estancia de la casa se necesitará una temperatura. El cambio de estado de OFF a ON de los calefactores lo dictamina el sensor de temperatura. Si el sensor está en el rango que marca la automatización, el estado del calefactor cambia su estado a ON. Si se sobrepasa la temperatura permitida, el relé del calefactor cambia su estado a OFF. Además, si el calefactor está encendido y algún usuario abre alguna ventana o puerta de esa misma estancia, automáticamente el calefactor pasa al estado OFF hasta que dicha puerta o ventana se cierre.

- El control del sistema de riego lo gestionará el sensor de plantas. Si la tierra está por debajo de la humedad que necesita, activará el relé Sonoff Basic a ON y él mismo accionará la electroválvula de agua. Cuando la humedad llegue al porcentaje que se necesita, automáticamente el sensor de plantas dará la señal al relé, cambiando el estado a OFF de la electroválvula.
- Control de algunos de los electrodomésticos, funcionando solamente en horas donde la tarifa de la luz es más baja. Electrodomésticos como la lavadora, secadora, lavaplatos, etcétera, solo cambiarán de estado a ON en horas donde la tarifa de luz es más baja (siempre y cuando no se necesite urgentemente su uso).
- La alarma está configurada con las tres centralitas domóticas disponibles en la vivienda. Cuando se sale de la vivienda, se activa la alarma mediante un pulsador y se dispone de 60 segundos para abandonar el hogar antes de que se active la alarma. Pasados ese tiempo la alarma será activada. Por lo tanto, cuando la puerta vuelva a ser abierta, se accionará el estado del sensor de puerta de CERRADA a ABIERTA y se dispondrán de 10 segundos de cortesía para desactivar la alarma. Si trascurridos esos 10 segundos no se ha desactivado la alarma, los tres altavoces integrados de las controladoras harán de sirena.
- Otra automatización hace referencia al sensor de humo situado en la cocina. Si el sensor de humo pasa al estado ON, automáticamente se accionará el estado ON del relé del extractor de humo.
- Se disponen de automatizaciones referentes a escenas. Las escenas se han definido para el modo televisión. Por ejemplo, si se está viendo un partido de tenis, se puede activar el modo deportes para que las luces creen un ambiente personalizado.
- Otra automatización hace referencia al sensor de vibración situado en el telefonillo y timbre. En el momento que alguien llame al timbre, el sensor de vibración cambia de estado a ON y avisa a todos los usuarios de la casa vía chat.
- Además, este sistema domótico dispone de automatizaciones de alerta. Se crean alertas si a la salida de casa hay alguna luz encendida o ventana abierta. La controladora domótica puede alertar de ello mediante sus sonidos personalizados.

Cabe resaltar, que todas las automatizaciones definidas anteriormente tienen configurado aviso de mensajería al usuario mediante Telegram. Por lo tanto, nos avisará de todo lo que está pasando en la vivienda vía chat.

7.4. Pruebas en automatizaciones

Es una parte complicada del proyecto. La programación de automatizaciones provoca condiciones de carrera entre dispositivos, por lo que se tienen que controlar todas las situaciones posibles antes de verificar un buen funcionamiento.

Una vez programada una automatización, hay que ver si afecta a otra indirectamente, por lo que es una tarea que puede llevar muchas horas o incluso días en la detección de bugs del sistema. Además, se deben contemplar todos los casos posibles que puedan afectar a esa automatización.

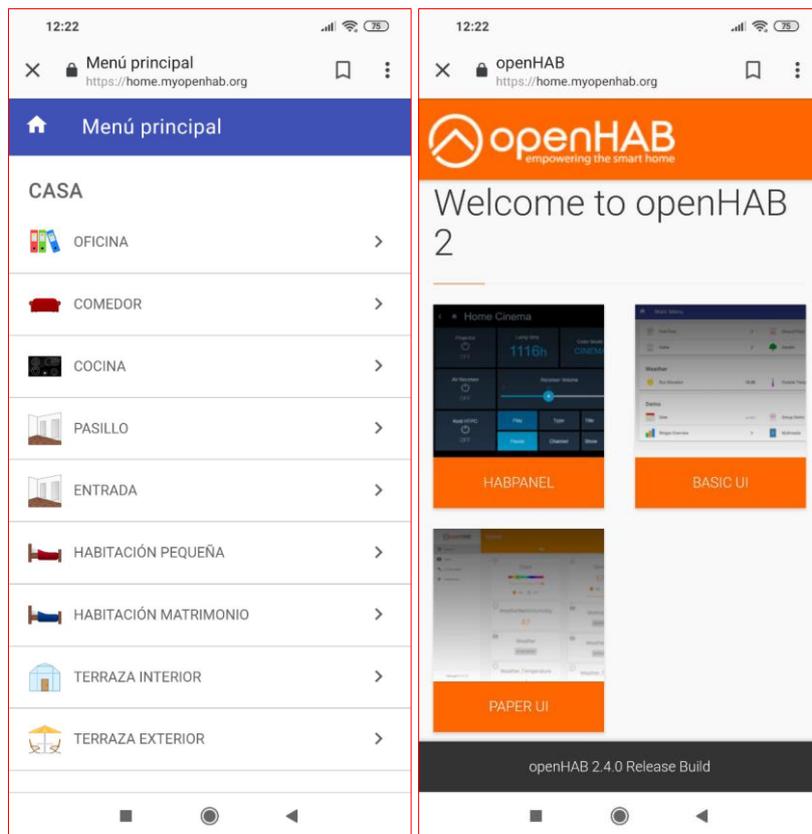
En las diferentes pruebas que se han realizado en este proyecto, se ha ido depurando el sistema hasta llegar a un punto donde se comporta de la manera que tiene que comportarse, sin errores no fallos.

7.5. Pruebas relacionadas con el sitio web myopenHAB

Otra prueba a realizar la conexión remota desde un dispositivo móvil al sistema. Esta conexión se realiza mediante el sitio web <https://myopenhab.org/>. Se accederá desde el móvil a la URL indicada y se deberá poner el usuario y contraseña para acceder al sitio.

Una vez realizada la conexión correctamente se seleccionará una de las tres opciones, en este caso, "Basic UI". De este modo se tendrá acceso remotamente a la plataforma openHAB para interactuar con los dispositivos de cualquier lugar.

Esta prueba ha sido verificada para la comprobación correcta del funcionamiento del sistema ya que es muy importante saber el estado de los dispositivos a distancia.



CAPÍTULO 8

Conclusiones y líneas futuras

Al inicio de este proyecto se presentaban una serie de objetivos bien estructurados y diferenciados. El primero consistía en implementar un servidor domótico de bajo coste y asequible para cualquier tipo de usuarios. Por ello, se optó por herramientas Open-Source y dispositivos físicos económicos.

Un detalle muy importante era la integración de diferentes protocolos y fabricantes en una misma plataforma, convirtiéndola en multi marca. Por ejemplo, poder utilizar un dispositivo mediante el protocolo ZigBee de la marca Xiaomi y al mismo tiempo la utilización de otro de diferente marca. Además, como se ha dicho anteriormente, la plataforma openHAB no viene a desbancar a otras, sino a ayudar a que los sistemas domóticos funcionen de la mejor manera posible.

El segundo objetivo consistía en crear una interfaz de usuario básica e intuitiva. Además, dicha interfaz debía permitir la monitorización y control en cualquier dispositivo móvil o tableta, desde cualquier lugar y distancia. Por otra parte, la vivienda debía de tener inteligencia con la programación de automatizaciones de todo tipo para mejorar la calidad de vida y confort de los usuarios

Tras la finalización de este proyecto se puede considerar que los objetivos descritos anteriormente han sido alcanzados.

Los beneficios que aporta este tipo de sistemas son muy importantes tanto para la utilización empresarial como a nivel de usuario. El futuro está preparando para el uso de este tipo de sistemas. Por otro lado, cada vez hay más empresas que ofrecen este tipo de servicios a los clientes. Este tipo de servicios serán fundamentales para la mejora de la calidad de vida y el confort de los usuarios, convirtiéndose en imprescindibles.

Algunas de las limitaciones que se han experimentado en este trabajo es la poca información en sus inicios. Además, el desarrollo de este tipo de sistemas es complejo si no se tienen nociones de programación. Otra de las limitaciones podría el precio de los dispositivos, ya que a nivel regional es bastante elevado.

La motivación para realizar este proyecto fue por interés del autor, ya que es aficionado en este ámbito desde hace años. Además, gracias al estudio para la realización de este trabajo se ha descubierto el verdadero potencial de este tipo de sistemas, con la motivación de continuarlo en un futuro.

8.1. Líneas futuras

En este proyecto se ha conseguido la gran mayoría de los objetivos propuestos, pero como se ha comentado anteriormente, el término domótica es tan amplio que la palabra “final” no existe.

A continuación, se definirán las líneas futuras que se van a alcanzar en un futuro próximo que:

- Souliss (según Pedro Santamaría. 2015.) en [51], es un proyecto totalmente abierto basado en nodos donde siempre tiene que haber un principal que haga de controlador. A partir de esa configuración se pueden configurar dispositivos como sensores, leds, relés, motores de persianas, etc. Souliss tiene binding propio en openHAB, por lo que se puede asociar a los demás dispositivos sin ningún problema. Dispone de un framework para Arduino basado en C/C++. Además, dispone de un foro en español en Google. [52]
- Otra línea futura será la de aplicar más seguridad al sistema domótico. Investigadores han comprobado los peligros que presenta la domótica inalámbrica, es decir, el control automatizado del hogar. Sus comunicaciones pueden ser accesibles por terceros, que pueden conocer los patrones de comportamientos de los habitantes, e incluso planear un robo. Por estos motivos, se intentará aplicar seguridad lo más rápido posible para afianzar el sistema ante intrusos.
- Se pretende integrar un módulo CC2531 para los dispositivos ZigBee para que funcionen independientemente de la centralita domótica. Por lo tanto, ya no se requerirá de la aplicación Mi Home.

8.2. Opinión personal

Este Trabajo Final de Grado es el final de un largo trayecto donde he aprendido y adquirido capacidades tanto a nivel personal como profesional. Los cuatro años universitarios me han servido para afrontar los problemas encontrados y darles solución.

Respecto al proyecto, he conseguido cada uno de los objetivos que tenía en mente presentar en él. Para ello, han sido necesarios varios meses de estudio y aprendizaje.

En un principio pensé que el proyecto no seguiría adelante, pero poco a poco las cosas iban saliendo y eso me creó una motivación extra.

A día de hoy creo que tengo un sistema bastante completo y de fácil utilización para todos los usuarios de la casa, pero esto solo es el principio. En un futuro, en mi propia vivienda, me gustaría implementar un hogar domótico al 100%, a grande escala.

Para concluir, se puede decir que no hay mayor satisfacción que haber logrado finalizar este proyecto, ya que en un futuro me gustaría dedicarme profesionalmente a implementar este tipo de sistemas

Referencias y bibliografía

Referencias

[1]

BlinkEdTech. *Impacto de IoT en la educación*. [WWW Document] URL
<<https://www.realinfluencers.es/2016/12/29/como-impactara-internet-cosas-educacion/>>
[Consulta: 10 de noviembre de 2017]

[2]

Marta Benedet 2017. *Impacto de IoT en la salud*. [WWW Document].URL
<<https://blog.mdcloud.es/sector-salud-por-que-las-plataformas-iot-son-imprescindibles/>>
[Consulta: 11 de noviembre de 2017]

[3]

Microsoft, 2016. *Impacto de IoT en el transporte*. [WWW Document].URL
<<https://www.microsoft.com/es-es/internet-of-things/transportation>>
[Consulta: 11 de noviembre de 2017]

[4]

Microsoft, 2017. *Internet de las cosas (IoT) para cada empresa*. [WWW Document].URL
<https://azure.microsoft.com/es-es/overview/iot/?site=mscom_iot>
[Consulta: 11 de noviembre de 2017]

[5]

Intel, 2016. *Datos potentes. Negocio inteligente*. [WWW Document].URL
<<https://www.intel.es/content/www/es/es/internet-of-things/overview.html>>
[Consulta: 13 de noviembre de 2017]

[6]

CEDOM. *¿Qué es la domótica?* [WWW Document].URL
<<http://www.cedom.es/sobre-domotica/que-es-domotica>>
[Consulta: 13 de noviembre de 2017]

[14]

Brian Benchoff, 2016. *Introducing the Raspberry Pi 3*. [WWW Document].URL
<<https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>>
[Consulta: 8 de enero de 2018]

[17]

Raspi.TV, 2017. *How much power does Pi Zero W use?* [WWW Document].URL
<<http://raspi.tv/2017/how-much-power-does-pi-zero-w-use>>
[Consulta: 11 de enero de 2018]

[19]

Geeky Theory. *¿Qué es MQTT?* [WWW Document].URL
<<https://geekytheory.com/que-es-mqtt/>> [Consulta: 19 de febrero de 2018]

[20]

Margaret Rouse. *MQTT (MQ Telemetry Transport)*. [WWW Document].URL
<<https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport>>
[Consulta: 22 de febrero de 2018]

[21]

OASIS, 2014. *MQTT Version 3.1.1 OASIS Standard* / PDF. [WWW Document].URL
<<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>>
[Consulta: 23 de febrero de 2018]

[24]

Marín Moreno, J y Ruiz Fernández, D. 2007. *Informe Técnico: Protocolo ZigBee (IEEE 802.15.4)*
[WWW Document].URL
<https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf>
[Consulta: 15 de enero de 2019]

[25]

EfectoLED. 2018. *¿Qué es ZigBee?* [WWW Document].URL
<<https://www.efectoled.com/blog/que-es-zigbee/>> [Consulta: 15 de enero de 2019]

[26]

INCIBE. 2016. *Seguridad en comunicaciones ZigBee*. [WWW Document].URL
<<https://www.incibe-cert.es/blog/seguridad-comunicaciones-zigbee>>
[Consulta: 15 de enero de 2019]

[30]

Jecrespom, 2017. *MQTT* [WWW Document].URL
<<https://aprendiendoarduino.wordpress.com/tag/protocolos/>>
[Consulta: 25 de febrero de 2018]

[27]

Luis Llamas, 2018. *Tú casa domótica con los dispositivos Sonoff de Itead* [WWW Document].URL
<<https://www.luisllamas.es/tu-casa-domotica-con-los-dispositivos-sonoff-de-itead/>> [Consulta: 1 de noviembre de 2018]

[30]

Espressif Systems, 2015. *ESP8266EX Datasheet Version 4.3* [WWW Document].URL
<https://www.esp8266.com/wiki/lib/exe/fetch.php?media=0a-esp8266_datasheet_en_v4.3.pdf>
[Consulta: 24 de marzo de 2018]

[31]

Community Wiki, 2018. *Módulos ESP8266*. / PDF [WWW Document].URL
<<https://www.esp8266.com/wiki/doku.php?id=esp8266-module-family>>
[Consulta: 28 de marzo de 2018]

[32]

Michael Ingraham, 2018. *Sonoff-Tasmota*. [WWW Document].URL
<<https://github.com/arendst/Sonoff-Tasmota/wiki>> [Consulta: 28 de marzo de 2018]

[33]

Wikipedia, 2017. *FTDI (Future Technology Devices International)*. [WWW Document].URL
<<https://es.wikipedia.org/wiki/FTDI>> [Consulta: 29 de marzo de 2018]

[34]

Rubén G. G. 2018. *Xiaomi Router 3, toda la gama al completo*. [WWW Document].URL
<<https://npirtube.com/xiaomi-router-3/>> [Consulta: 10 de Abril de 2018]

[35]

U/Dr_Schmoctor, 2015. *How to install Padavan custom firmware on the Xiaomi Mi Router 3*.
[WWW Document].URL
<https://www.reddit.com/r/Xiaomi/comments/5d0wmx/guide_how_to_install_padavan_custom_firmware_on/> [Consulta: 25 de Abril de 2018]

[36]

Juan Jesús, 2016. *Gestiona tus dispositivos de Xiaomi desde el móvil*. [WWW Document].URL
<<https://mi-home.uptodown.com/android>> [Consulta: 10 de enero de 2019]

[40]

Wikipedia, 2019. *PuTTY*. [WWW Document].URL <https://es.wikipedia.org/wiki/PuTTY>
[Consulta: 11 de enero de 2019]

[48]

Wikipedia, 2019, *Visual Studio Code*. [WWW Document].URL
<https://es.wikipedia.org/wiki/Visual_Studio_Code> [Consulta: 10 de septiembre de 2019]

[50]

Julián Pérez Porto y María Merino. 2016. Definición de automatización. [WWW Document].URL
<<https://definicion.de/automatizacion/>>
[Consulta: 7 de enero de 2019]

[51]

Pedro Santamaría. 2015. *Souliss, el proyecto que gustará a los makers fans de Arduino y la internet de las cosas*. [WWW Document].URL
<<https://www.xatakahome.com/domotica/souliss-el-proyecto-que-gustara-a-los-makers-fans-de-arduino-y-la-internet-de-las-cosas>>

Bibliografía

[7]

Web oficial del software openHAB: <https://www.openhab.org/>

[8]

Web oficial del software MQTT: <https://mqtt.org/>

[9]

Web oficial del software Jeedom: <https://www.jeedom.com/site/fr/>

[10]

Web oficial del software Home Assistant: <https://www.home-assistant.io/>

[11]

Web oficial del software openHAB: <https://www.openhab.org/>

[12]

Web oficial de HPE: <https://www.hpe.com/es/es/product-catalog/servers/proliant-servers/pip.specifications.hpe-proliant-microserver-gen10.1009955118.html>

[13]

Web oficial de la organización Raspberry: <https://www.raspberrypi.org/about/>

[15]

Características del modelo Raspberry Pi 3: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

[16]

Descarga: Sistemas operativos Raspberry Pi: <https://www.raspberrypi.org/downloads/>

[18]

Web oficial del software MQTT: <https://mqtt.org/>

[23]

Web oficial del protocolo Mosquitto: <https://mosquitto.org/>

[28]

Empresa Sonoff: <https://sonoff.itead.cc/en/>

[29]

Empresa Ewelink para productos Sonoff: <https://sonoff.itead.cc/en/ewelink>

[37]

Descarga de Ubuntu Server: <https://www.ubuntu.com/#download>

[38]

Descarga del programa Win32diskimager: <https://sourceforge.net/projects/win32diskimager/>

[39]

Guía de instalación openHABian:
<https://www.openhab.org/docs/installation/openhabian.html#raspberry-pi>

[41]

Firmware Tasmota: <https://github.com/arendst/Sonoff-Tasmota/releases>

[42]

Programa ESPEasy: <https://github.com/letscontrolit/ESPEasy/releases>

[43]

Programa Ternite: <https://github.com/letscontrolit/ESPEasy/releases>

[44]

Guía instalación openHAB: <https://www.openhab.org/docs/installation/linux.html>

[45]

Configuración binding NTP: <https://www.openhab.org/addons/bindings/ntp/>

[46]

Addons disponibles openHAB: <https://www.openhab.org/addons/>

[47]

Configuración MyopenHAB: <https://www.openhab.org/addons/integrations/openhabcloud/>

[49]

Programa diseño de planos: <https://floorplanner.com/>

[52]

Grupo de Google de Souliss: https://groups.google.com/forum/#!topic/souliss-es/pb_fBadA6IE

Ficheros de configuración openHAB

En este anexo se recogerán los ficheros de configuración de openHAB. Algunos de esos ficheros han sido simplificados por redundantes.

Los ficheros de configuración serán 3: things, items y rules.

Things

xiaomi.things

```
// Dispositivos XIAOMI
// Gateway OFICINA
Bridge mihome:bridge:34ce00... "Bridge Xiaomi Gateway Oficina"
[ serialNumber="34ce00...", ipAddress="192.168.X.X" , port=9898, key="30io...",
pollingInterval=6000]
{
  // GATEWAY
  Thing mihome:gateway:34ce00... "Gateway Oficina" [itemId="34ce00..."]
  //SENSORES
  // Sensores de puertas / ventanas
  Thing mihome:sensor_magnet:158d0... "Sensor ventana OficinaD"
    [itemId="158d0..."]
  Thing mihome:sensor_magnet_aq2:158d0..."Sensor ventana OficinaI"
    [itemId="158d0..."]

  // Sensor de temperatura y humedad
  Thing mihome:sensor_ht:158d0... "Sensor tempertaura Oficina"
    [itemId="158d0..."]

  // Sensor de presencia
  Thing mihome:sensor_motion:158d0... "Sensor de presencia Oficina"
    [itemId="158d0..."]

  //INTERRUPTORES
  Thing mihome:86sw2:158d0... "Interruptor 1 doble Oficina"
    [itemId="158d0..."]
  Thing mihome:86sw2:158d0... "Interruptor 2 doble Oficina"
    [itemId="158d0..."]
}
```

```
//PULSADORES
Thing mihome:sensor_switch:158d0... "Pulsador 1: Flexo oficina"
  [itemId="158d0..."]
//ENCHUFES INTELIGENTES
Thing mihome:sensor_plug:158d0... "Enchufe inteligente 1"
  [itemId="158d0..."]
}
// Gateway COMEDOR
Bridge mihome:bridge:34ce0... "Bridge Xiaomi Gateway Comedor"
[ serialNumber="34ce0...", ipAddress="192.168.X.X" , port=9898, key="nzlcr...",
pollingInterval=6000]
{
  // GATEWAY
  Thing mihome:gateway:34ce0... "Gateway Comedor" [itemId="34ce0..."]

  //SENSORES
  // Sensores de puertas / ventanas
  Thing mihome:sensor_magnet_aq2:158d0... "Sensor Puerta Comedor"
    [itemId="158d0..."]

  // Sensor de temperatura y humedad
  Thing mihome:sensor_ht:158d0... "Sensor tempertaura Comedor"
    [itemId="158d0..."]

  // Sensor de inundación
  Thing mihome:sensor_wleak_aq1:158d0... "Detector de inundación Terraza
interior" [itemId="158d0..."]

  // Alarma detector de humo
  Thing mihome:smoke:158d0... "Detector de humo Cocina " [itemId="158d0..."]

  //PULSADORES
  Thing mihome:sensor_switch_aq2:158d0... "PulsadorC 3: Extractor Cocina"
    [itemId="158d0..."]
}

// Gateway ENTRADA
Bridge mihome:bridge:7811d... "Bridge Xiaomi Gateway Entrada"
[ serialNumber="7811d...", ipAddress="192.168.X.X" , port=9898, key="rkole...",
pollingInterval=6000]
{
  // GATEWAY
  Thing mihome:gateway:7811d... "Gateway Entrada" [itemId="7811d..."]

  // Interruptores Wireless
  Thing mihome:86sw1:158d0... "Interruptor pasillo"
```

```
[itemId="158d0..."]
Thing mihome:86sw1:158d0... "Interruptor pasillo2" [itemId="158d0..."]

//SENSORES
// Sensores de puertas / ventanas
Thing mihome:sensor_magnet_aq2:158d0... "Sensor Puerta Entrada"
[itemId="158d0..."]

// Sensor de temperatura y humedad
Thing mihome:sensor_weather_v1:158d0... "Sensor tempertaura Cocina"
[itemId="158d0..."]

// Sensor de presencia
Thing mihome:sensor_motion_aq2:158d0... "Sensor de presencia Entrada"
[itemId="158d0..."]
Thing mihome:sensor_motion:158d0... "Sensor de presencia Pasillo"
[itemId="158d0..."]

// Sensor de vibración
Thing mihome:sensor_vibration:158d0...Sensor de vibracion Entrada"
[itemId="158d0..."]

//PULSADORES
Thing mihome:sensor_switch:158d0... "Pulsador 2: Alarma oficina"
[itemId="158d0..."]
}
```

astro.things

```
// Astro binding
astro:sun:home "Astro: Datos sobre el sol " [ geolocation="30.0,-0.2662300",
interval=300]

astro:moon:home "Astro: Datos sobre la luna " [ geolocation="39.0719500,-0.2662300",
interval=300]
```

iluminacion.things

```
// Bombillas Yeelight
// Bola blanca
Thing yeelight:Dolphin:0x00000... "Yeelight Blanca Oficina" [itemId="0x00000..."]

// Bola RGB
Thing yeelight:Wonder:0x00000... "Yeelight RGB Oficina1" [itemId="0x00000..."]
Thing yeelight:Wonder:0x00000... "Yeelight RGB Oficina2" [itemId="0x00000..."]
Thing yeelight:Wonder:0x00000... "Yeelight RGB Oficina3" [itemId="0x00000..."]
```

```
// Bombillas Xiaomi Philips
Thing mio:basic:039... "Philips Xiaomi " [ host="192.168.X.X", token="f9b..." ]
```

kodi.things

```
// Kodi OrangPi comedor
Thing kodi:kodi:c1e862c0... "Kodi OrangePi Comedor" [ ipAddress="192.168.X.X",
port=9090, httpPort=8080]
```

network.things

```
// Network binding
// DISPOSITIVOS
// Gateway
Thing network:pingdevice:gateway_1 "Dispositivo de red: Gateway 1" [
hostname="192.168.X.X" , retry=1, timeout=5000, refreshInterval=60000 ]
Thing network:pingdevice:gateway_2 "Dispositivo de red: Gateway 2" [
hostname="192.168.X.X" , retry=1, timeout=5000, refreshInterval=60000 ]
Thing network:pingdevice:gateway_3 "Dispositivo de red: Gateway 3" [
hostname="192.168.X.X" , retry=1, timeout=5000, refreshInterval=60000 ]

// Robot aspirador
Thing network:pingdevice:robot "Dispositivo de red: Robot aspirador Xiaomi" [
hostname="192.168.X.X" , retry=1, timeout=5000, refreshInterval=60000 ]
```

ntp.things

```
// Ntp Binding
Thing ntp:ntp:local "NTP" [ hostname="nl.pool.ntp.org", refreshInterval=60,
refreshNtp=30 ]
```

robotAspirador.things

```
// Robot aspirador
Thing mio:vacuum:0... "Robot aspirador Xiaomi" [ host="192.168.X.X",
token="6a6e..."]
```

servidorHP.things

```
// System info binding
// Información del servidor HP
```

```
systeminfo:computer:acamarena "Información del servidor HP"[ interval_high=3,
interval_medium=60 ]
```

Items

grupos.items

```
// Grupos de la casa
Group gComedor      "Comedor"          <sofa>
Group gCocina       "Cocina"          <kitchen>
Group gPasillo     "Pasillo"         <corridor>
Group gEntrada     "Entrada"         <corridor>
Group gHab1        "Habitación pequeña" <bedroom_red>
Group gHab2        "Habitación matrimonio" <bedroom_blue>
Group gTerrazaInt  "Terraza interior"   <greenhouse>
Group gTerrazaExt  "Terraza exterior"   <terrace>
Group gBanoP       "Baño principal"   <bath>
Group gBanoS       "Baño secundario"  <bath>
Group gOficina     "Oficina"          <office>

// Subgrupo Oficina
Group gOficinaGateway "Gateway Oficina" (gOficina)
Group gOficinaSensores "Sensores" (gOficina)
Group gOficinaIluminacion "Iluminación" (gOficina)
Group:Switch:OR(ON,OFF) gAlloficina "Luces techo [%d]" [ "Lighting" ]
    Group gLumOficina "Brillo [%d %%]"
    Group gColorOficina "Color"
    Group gTempOficina "Color temperatura [%d]"

Group gOficinaInterruptores "Interruptores" (gOficina)
Group gOficinaServerHP "Servidor HP" (gOficina)
    Group gParamRed "Parámetros de red" (gOficinaServerHP)
    Group gMemRAM "Memoria RAM" (gOficinaServerHP)
    Group gMemROM "Almacenamiento" (gOficinaServerHP)
    Group gParamCPU "Parámetros de la CPU" (gOficinaServerHP)
    Group gInfoUnidadProceso "Unidad de procesos" (gOficinaServerHP)
    Group gMemSWAP "Memoria SWAP" (gOficinaServerHP)
    Group gBateria "Batería" (gOficinaServerHP)
    Group gSensores "Sensores" (gOficinaServerHP)
    Group gDisplay "Display" (gOficinaServerHP)
    Group gUnidad "Unidad" (gOficinaServerHP)
Group gOficinaPulsadores "Pulsadores" (gOficina)
Group gOficinaCalefaccion "Calefacción" (gOficina)
Group gOficinaEnchufes "Enchufes" (gOficina)

// Subgrupo Comedor
Group gComedorGateway "Gateway Comedor" (gComedor)
```

```

Group gComedorSensores      "Sensores"      (gComedor)
Group gComedorCalefaccion   "Calefacción"   (gComedor)

// Subgrupo Entrada
Group gEntradaGateway       "Gateway Entrada" (gEntrada)
Group gEntradaSensores      "Sensores"      (gEntrada)
Group gEntradaPulsadores    "Pulsadores"    (gEntrada)
Group gEntradaCalefaccion   "Calefacción"   (gEntrada)

// Subgrupo Pasillo
Group gPasilloSensores      "Sensores"      (gPasillo)
Group gPasilloIluminacion   "Iluminación"   (gPasillo)
Group gPasilloInterruptores "Interruptores" (gPasillo)
Group gPasilloCalefaccion   "Calefacción"   (gPasillo)

// Subgrupo Cocina
Group gCocinaSensores       "Sensores"      (gCocina)
Group gCocinaExtractor       "Extractor"      (gCocina)
Group gCocinaCalefaccion    "Calefacción"   (gCocina)

// Subgrupo Habitacion Matrimonio
Group gHab2Calefaccion      "Calefacción"   (gHab2)

// Subgrupo Habitacion pequeña
Group gHab1Calefaccion      "Calefacción"   (gHab1)

// Subgrupo terraza interior
Group gTerrazaIntSensores   "Sensores"      (gTerrazaInt)
Group gTerrazaIntRiego      "Riego"          (gTerrazaInt)

// Subgrupo baño pequeño
Group gBanoPInterruptores   "Interruptores" (gBanoP)
Group gBanoCalefaccion      "Calefacción"   (gBanoP)

// Subgrupo baño grande
Group gBanoSCalefaccion     "Calefacción"   (gBanoS)

// GRUPOS complejos mixtos

// Apagado general de todas las luces (Oficina x4,Pasillo x1)
Group:Switch:OR(ON,OFF) gAllLuces "Todas las luces [(%d)]"

// Apagado general de todas los relés
Group:Switch:OR(ON, OFF) gAllReles "Todos los relés [(%d)]"

// Apagado general de todo
Group:Switch:OR(ON, OFF) gAll "Todo [(%d)]"

```

```
// Temperatura min,max,AVG de toda la casa (Comedor,Cocina,Oficina)
Group:Number:AVG gTempAVG "Temp. Media interior [%.1f °C]" <temperature>
Group:Number:MIN gTempMIN "Temp. Mínima interior [%.1f °C]" <temperature>
Group:Number:MAX gTempMAX "Temp. Máxima interior [%.1f °C]" <temperature>

// Grupos de todos los sensores de ventanas y puertas
Group:Contact:OR(OPEN,CLOSED) gSensoresPVall "Todos los sensores P/V menos
    Entrada" (gSensores)

Group:Contact:OR(OPEN,CLOSED) gAllSensoresPV "Todos los sensores P y V [(%d)]"

// Apagado general de la calefacción
Group:Switch:OR(ON, OFF) gAllCalefaccion "Toda la calefacción [(%d)]"

// Group gSensores
Group:Contact:OR(OPEN,CLOSED) gSensorV_Oficina // Sensores de ventana
// MODOS
Switch modoNoche "Modo noche [%s]" // Modo noche (Astro)
```

tiempo.items

```
// Tiempo
// Grupos
Group gTiempo "Más información" <zoom>
Group gAtmosfera "Atmósfera" (gTiempo)
Group gNubes "Nubes" (gTiempo)
Group gCondicion "Condición" (gTiempo)
Group gPrecipitacion "Precipitaciones" (gTiempo)
Group gTemperaturaExt "Temperatura exterior" (gTiempo)
Group gViento "Viento" (gTiempo)
Group gEstacionMete "Estación meteorológica"(gTiempo)

// Atmósfera
Number Atmosfera_Hum "Humedad [%d %%]" <humidity> (gAtmosfera)
    {weather="locationId=f, type=atmosphere, property=humidity"}
Number Atmosfera_Pres "Presión atmosférica [%.2f mb]" <pressure> (gAtmosfera)
    {weather="locationId=f, type=atmosphere, property=pressure"}

// Nubes
Number Nubes_Por "Nubes [%.0f %%]" (gNubes)
    {weather="locationId=f, type=clouds, property=percent"}

// Condición
String Condicion "Condición [%s]" <sun_clouds> (gCondicion)
    {weather="locationId=f, type=condition, property=text"}
```

```
// Precipitaciones
Number Precipitacion_LL "Lluvia [%.2f mm/h]" (gPrecipitacion)
    {weather="locationId=casaW, type=precipitation, property=rain"}
Number Precipitacion_Nieve "Nieve [%.2f mm/h]" (gPrecipitacion)
    {weather="locationId=casaW, type=precipitation, property=snow"}
Number Precipitacion_Prob "Probabilidad [%d %%]" (gPrecipitacion)
    {weather="locationId=casaW, type=precipitation, property=probability"}

// Temperatura
Number Temperatura "Temperatura [%.1f °C]" <temperature> (gTemperaturaExt)
    {weather="locationId=f, type=temperature, property=current"}

// Viento
Number Viento_Vel "Velocidad del viento [%.2f km/h]" <wind> (gViento)
    {weather="locationId=f, type=wind, property=speed"}
```

servidorHP.items

```
// Servidor HP
// Parámetros de red
String Server_IP "Dirección IP [%d]" (gParamRed)
    { channel="systeminfo:computer:acamarena:network#ip" }
String Server_Nom "Red" (gParamRed)
    { channel="systeminfo:computer:acamarena:network#networkName" }

// Memoria RAM
Number Server_MemD "Memoria RAM disponible" (gMemRAM)
    { channel="systeminfo:computer:acamarena:memory#available" }

// Almacenamiento ROM
Number Server_ADisp "Almacenamiento disponible"(gMemROM)
    { channel="systeminfo:computer:acamarena:storage#available" }

// Informaciones sensores
Number Server_SensorCPU "Temperatura de la CPU" (gSensores)
    { channel="systeminfo:computer:acamarena:sensors#cpuTemp" }
Number Server_SensorCPUV "Voltaje de la CPU" (gSensores)
    { channel="systeminfo:computer:acamarena:sensors#cpuVoltage" }
```

sensores.items

```
// OFICINA
// Sensores de ventana Oficina
Contact SensorV_Izda_Oficina "Estado ventana izquierda [MAP(es.map):%s]"
<window> (gOficinaSensores,gSensoresPVall,gSensorV_Oficina,gAllSensoresPV)
{ channel="mihome:sensor_magnet_aq2:158d0...:isOpen" }

DateTime SensorF_Izda_Oficina "Hora: [%1$tY-%1$tm-%1$td %1$tH:%1$tM]"
<time> (gOficinaSensores)
{ channel="mihome:sensor_magnet_aq2:158d0...:lastOpened" }

Number SensorA_Izda_Oficina "Tiempo abierta I [%d seg]"
<line> (gOficinaSensores)
{ channel="mihome:sensor_magnet_aq2:158d0...:isOpenAlarmTimer" }

Number SensorB_Izda_Oficina "Nivel de batería I "
<battery> (gOficinaSensores)
{ channel="mihome:sensor_magnet_aq2:158d0...:batteryLevel" }

Switch SensorB_BajaIzda_Oficina "Batería baja I"
<energy> (gOficinaSensores)
{ channel="mihome:sensor_magnet_aq2:158d0...:lowBattery" }

// Sensor de temperatura y humedad
Number SensorTempT_Oficina "Temperatura oficina [%1.1f °C]"
<temperature> (gOficinaSensores,gTempAVG,gTempMIN, gTempMAX)
{ channel="mihome:sensor_ht:158d0...:temperature" }

Number SensorTempH_Oficina "Humedad [%1.1f %]" <humidity> gOficinaSensores)
{ channel="mihome:sensor_ht:158d0...:humidity" }

// Sensor de presencia
Switch SensorPresE_Oficina "Estado [%s]" <motion> (gOficinaSensores)
{ channel="mihome:sensor_motion:158d0...:motion" }

Number SensorPresT_Oficina "Motion Off Timer" <clock> (gOficinaSensores)
{ channel="mihome:sensor_motion:158d0...:motionOffTimer" }

DateTime SensorPresU_Oficina "Última actividad [%1$tY-%1$tm-%1$td %1$tH:%1$tM]"
<clock-on> (gOficinaSensores)
{ channel="mihome:sensor_motion:158d0...:lastMotion" }

// ENTRADA
// Sensor de vibracion
DateTime SensorVibU_Entrada " Última actividad [%1$tY-%1$tm-%1$td %1$tH:%1$tM]"
<time> (gEntradaSensores)
{ channel="mihome:sensor_vibration:158d0...:lastAction" }
```

```

Number SensorVibA_Entrada "Ángulo" <line> (gEntradaSensores)
  { channel="mihome:sensor_vibration:158d0...:tiltAngle" }

Number SensorVibX_Entrada "Corrdenada X" <line> (gEntradaSensores)
  { channel="mihome:sensor_vibration:158d0...:orientationX" }

Number SensorVibY_Entrada "Corrdenada Y" <line> (gEntradaSensores)
  { channel="mihome:sensor_vibration:158d0...:orientationY" }

Number SensorVibZ_Entrada "Coordenada Z" <line> (gEntradaSensores)
  { channel="mihome:sensor_vibration:158d0...:orientationZ" }

Number SensorVibBed_Entrada "Modo dormir" <bedroom>(gEntradaSensores)
  { channel="mihome:sensor_vibration:158d0...:bedActivity" }

// COCINA
// Sensor de detección de humos
Number SensorHumoD_Cocina "Densidad de las partículas [%d]" <fire>
  (gCocinaSensores) { channel="mihome:smoke:158d0...:density" }

Switch SensorHumoDet_Cocina "Humo detectado [%s]" <smoke> (gCocinaSensores)
  { channel="mihome:smoke:158d0...:alarm" }

String SensorHumoE_Cocina "Estado [%s]" <status> (gCocinaSensores)
  { channel="mihome:smoke:158d0...:status" }

// TERRAZA INTERIOR
// Sensor detector de inundación de aguas
Switch SensorAguaD_TeInt "Fuga de agua detectada [%s]" <water>
  (gTerrazaIntSensores) {channel="mihome:sensor_wleak_aq1:158d0...:leak"}

// Sensor de planta Mi flora
Number Sensor_PlantaI_TeIntB "Intensidad de luz del sol B [%d lux]" <sun>
  (gTerrazaIntRiego) {mqtt="<[broker:miflora/AloeVera:state:JSONPATH($.light)]"}

Number Sensor_PlantaT_TeIntB "Temperatura del aire B[%d.%1f °C]" <temperature>
  (gTerrazaIntRiego)
  {mqtt="<[broker:miflora/AloeVera:state:JSONPATH($.temperature)]"}

Number Sensor_PlantaH_TeIntB "Humedad B[%d %%]" <humidity> (gTerrazaIntRiego)
  {mqtt="<[broker:miflora/AloeVera:state:JSONPATH($.moisture)]"}

Number Sensor_PlantaF_TeIntB "Fertilidad del terreno B[%d µS/cm]" <smoke>
  (gTerrazaIntRiego)
  {mqtt="<[broker:miflora/AloeVera:state:JSONPATH($.conductivity)]"}

Number Sensor_PlantaB_TeIntB "Nivel de batería B[%d %%]" <battery>
  (gTerrazaIntRiego)
  {mqtt="<[broker:miflora/AloeVera:state:JSONPATH($.battery)]"}

```

reles.items

```
// Relés
// COCINA
Switch Extractor_Cocina "Extractor cocina" <power> (gCocinaExtractor)
{mqtt=">[broker:cmd/sonoff-Basic_3/POWER:command:*:default],<[broker:
stat/sonoff-Basic_3/POWER:state:default]"}
```

pulsadores.items

```
// OFICINA
// Pulsadores redondos
// Flexo escritorio
Number PulsadorB_Oficina "Nivel de batería" <battery> (gOficinaPulsadores)
{ channel="mihome:sensor_switch:158d0...:batteryLevel" }

Switch PulsadorB_Baja_Oficina "Batería baja" <energy> (gOficinaPulsadores)
{ channel="mihome:sensor_switch:158d0...:lowBattery" }

// OFICINA
// Pulsador cuadrado
// Alarma
Number PulsadorB_Cocina "Nivel de batería" <battery> (gEntradaPulsadores)
{ channel="mihome:sensor_switch_aq2:158d0...:batteryLevel" }

Switch PulsadorB_Baja_Cocina "Batería baja" <energy> (gEntradaPulsadores)
{ channel="mihome:sensor_switch_aq2:158d0...:lowBattery" }
```

ntp.items

```
// NTP binding
DateTime Datos_NTP "Fecha completa [%1$tA, %1$td.%1$tm.%1$tY %1$tH:%1$tM]"
<time> { channel="ntp:ntp:local:dateTime"}
```

network.items

```
// Dispositivos Online
// OFICINA
// Gateway
Switch DispGat1_Oficina "Gateway oficina" <network> (dispositivos)
{ channel="network:pingdevice:gateway_1:online" }
Number DispGat1L_Oficina "Latencia" (dispositivos)
{ channel="network:pingdevice:gateway_1:latency" }
DateTime DispGat1U_Oficina "Última vez vista [%1$tH:%1$tM]"(dispositivos)
{ channel="network:pingdevice:gateway_1:lastseen" }
```

kodi.items

```
// Kodi
//COMEDOR
Group gKodiComedor "Kodi Comedor" (gComedor)

Switch Kodi_Sil_Comedor "Silenciar [%s]" (gKodiComedor)
{ channel="kodi:kodi:c1e862c0...:mute" }

Dimmer Kodi_Vol_Comedor "Volumen [%d %%]" (gKodiComedor)
{ channel="kodi:kodi:c1e862c0...:volume" }

Player Kodi_Cont_Comedor "Control" (gKodiComedor)
{ channel="kodi:kodi:c1e862c0...:control" }

Switch Kodi_Stop_Comedor "Stop" (gKodiComedor)
{ channel="kodi:kodi:c1e862c0...:stop" }

String Kodi_Play_Comedor "PlayerURI" (gKodiComedor)
{ channel="kodi:kodi:c1e862c0...:playuri" }
```

interruptores.items

```
// OFICINA
// Interruptores inalámbricos
Number Inte2B_Oficina "Nivel de batería" <battery> (gOficinaInterruptores)
{ channel="mihome:86sw2:158d0...:batteryLevel" }

Switch Inte2B_Baja_Oficina "Batería baja" <energy> (gOficinaInterruptores)
{ channel="mihome:86sw2:158d0...:lowBattery" }
```

iluminacion.items

```
// Bombillas Yeelight
// Bombilla blanca Xioami Yeelight
Switch Luz_Flexo_Oficina "Flexo" (gOficinaIluminacion,gAllLuces)[ "Lighting" ]
{ channel="yeelight:Dolphin:0x00000...:Brightness" }

Dimmer Luz_FlexoB_Oficina "Brillo [%d %%]" (gOficinaIluminacion)
{ channel="yeelight:Dolphin:0x00000...:Brightness " }
// Bombilla RGB Xioami Yeelight
Switch Luz_Techo_Oficina1 "Lampara techo uno"
(gOficinaIluminacion,gAlloficina,gAllLuces) [ "Lighting" ]
{ channel="yeelight:Wonder:0x00000...:Brightness" }
```

```
Dimmer Luz_TechoB_Oficina1 "Brillo [%d %%]"  
  (gOficinaIluminacion,gLumOficina )  
  { channel="yeelight:Wonder:0x00000...:Brightness " }  
  
Color Luz_TechoC_Oficina1 "Color"  
  (gOficinaIluminacion,gColorOficina )  
  { channel="yeelight:Wonder:0x00000...:Color" }  
  
Dimmer Luz_TechoT_Oficina1 "Color temperatura [%d]"  
  (gOficinaIluminacion,gTempOficina )  
  { channel="yeelight:Wonder:0x00000...:Color_Temperature" }
```

gateway.items

```
//Centralita domòtica Oficina  
  
Switch Gateway0_AddDevice "Añadir dispositivo" <smiley>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:joinPermission" }  
  
Switch Gateway0_LuzSwitch "Luz []" <light> (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:brightness" }  
  
Dimmer Gateway0_Brillo "Brillo [%d %%]" <movecontrol>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:brightness" }  
  
Color Gateway0_Color "Color RGB" <rgb>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:color" }  
  
Dimmer Gateway0_ColorTemp "Color temperatura" <heating>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:colorTemperature" }  
  
Number Gateway0_AmbiLight "Ambilight" <sun> (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:illumination" }  
  
Number Gateway0_Sonido "Sonido actual" <soundvolume-0>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:sound" }  
  
Switch Gateway0_Mute "Silenciar []" <soundvolume_mute>  
  (gOficinaGateway,gAjustesExtra)  
  { channel="mihome:gateway:34ce0...:enableSound" }
```

```
Dimmer Gateway0_SonidoVol "Volumen" <soundvolume>
  (gOficinaGateway,gAjustesExtra)
  { channel="mihome:gateway:34ce0...:volume" }

// Grupos
Group gAjustesOficina "Más ajustes" <settings>
```

enchufes.items

```
// OFICINA
// Enchufes inteligentes Xiaomi Plug
Switch Plug_Switch "Calefacción [%s]" <poweroutlet> (gOficinaEnchufes)
  { channel="mihome:sensor_plug:158d0...:power" }

Switch Plug_Active "Estado [%s]" <status> (gOficinaEnchufes)
  { channel="mihome:sensor_plug:158d0...:inUse" }

Number Plug_Power "Energía suministrada" <energy> (gOficinaEnchufes)
  { channel="mihome:sensor_plug:158d0...:loadPower" }

Number Plug_Consumption "Energía consumida" <line-incline> (gOficinaEnchufes)
  { channel="mihome:sensor_plug:158d0...:powerConsumed" }
```

Rules

alarma.rules

```
var Timer timer = null
var Timer timer2= null

rule "Activar y desactivar la alarma de entrada"
  when
    Channel "mihome:sensor_switch:158d0...:button" triggered
  then
    var actionName = receivedEvent.getEvent()
    switch(actionName) {
      case "SHORT_PRESSED":
        {
          if (Alarma_Entrada.state == OFF)
            {
              timer = createTimer(now.plusSeconds(15))
              [sendCommand(Alarma_Entrada,ON) timer = null ]
              sendTelegram("bot1","Alarma activada")
            }
        }
    }
}
```

```

    }
  }
  case "DOUBLE_PRESSED":
  {
    if (Alarma_Entrada.state== ON)
    {
      Alarma_Entrada.postUpdate(OFF)
      sendCommand(GatewayO_Mute,OFF)
      sendTelegram("bot1","Alarma desactivada")
    }
  }
  case "LONG_PRESSED":
  {

  }
  case "LONG_RELEASED":
  {

  }
}
end

rule "Deteccion de intruso con alarma activada"
  when
    Item SensorP_Entrada changed from CLOSED to OPEN
  then
    if(Alarma_Entrada.state == ON)
    {
      timer = createTimer(now.plusSeconds(15))
      [sendCommand(GatewayO_Sonido,1) timer = null ]
    }
  end
end

```

calefacción.rules

```

// OFICINA ////////////////////////////////////////
rule "OFICINA Encender calefacción"
  when
    Item SensorTempT_Oficina changed or
    Item CalefaccionA_Oficina changed or
    Item gSensorV_Oficina changed
  then
    if (CalefaccionA_Oficina.state == ON &&
        SensorTempT_Oficina.state <= 22 && gSensorV_Oficina.state == CLOSED)
    {
      Plug_Switch.sendCommand(ON)
      CalefaccionA_Oficina.sendCommand(ON)
      sendTelegram("bot1","Calefacción encendida")
    }
  end
end

```

```
    }  
    else if ((CalefaccionA_Oficina.state == ON &&  
             SensorTempT_Oficina.state >= 22.5) ||  
            gSensorV_Oficina.state == OPEN)  
    {  
        Plug_Switch.sendCommand(OFF)  
        sendTelegram("bot1","Calefacción apagada")  
    }  
end  
  
rule "OFICINA Apagar calefaccion"  
when  
    Item CalefaccionA_Oficina changed from ON to OFF  
then  
    Plug_Switch.sendCommand(OFF)  
    CalefaccionA_Oficina.sendCommand(OFF)  
    sendTelegram("bot1","Calefacción apagado forzada")  
end
```

iluminación.rules

```
var Timer timer = null  
// OFICINA ///////////////////////////////////////  
// Flexo escritorio  
rule "Pulsador redondo Xiaomi"  
when  
    Channel "mihome:sensor_switch:158d0...:button" triggered  
then  
    var actionName = receivedEvent.getEvent()  
    switch(actionName) {  
        case "SHORT_PRESSED":  
            {  
                if (Luz_Flexo_Oficina.state == OFF)  
                {  
                    Luz_Flexo_Oficina.sendCommand(ON)  
                }else if (Luz_Flexo_Oficina.state == ON)  
                {  
                    Luz_Flexo_Oficina.sendCommand(OFF)  
                }  
            }  
        case "DOUBLE_PRESSED":  
            {  
                sendCommand(GatewayE_Sonido,10004)  
            }  
        case "LONG_PRESSED":  
            {
```

```
    }
    case "LONG_RELEASED":
    {
    }
}
end
// Lámpara techo
rule "Interruptor Aqara Wirelss (2 Botones)"
when
    Channel "mihome:86sw2:158d0...:ch1" triggered SHORT_PRESSED
then
    if (gAlloficina.state == OFF)
    {
        gAlloficina.sendCommand(ON)
    }else if (gAlloficina.state == ON){
        gAlloficina.sendCommand(OFF)
    }
end

rule "Interruptor Aqara Wirelss (2 Botones)"
when
    Channel "mihome:86sw2:158d0...:ch1" triggered SHORT_PRESSED
then
    if (gAlloficina.state == OFF)
    {
        gAlloficina.sendCommand(ON)
    }else if (gAlloficina.state == ON)
    {
        gAlloficina.sendCommand(OFF)
    }
end

// Rule Sensor de presencia + luz
rule "Reset de los dispositivos de presencia a OFF en el inicio"
when
    System started
then
    SensorPresE_Oficina.sendCommand(OFF)
end

rule "Sensor presencia oficina update"
when
    Item SensorPresE_Oficina changed
then
    if (SensorPresE_Oficina.state == ON && modoNoche.state==ON)
    {
        gAlloficina.sendCommand(ON)
    }
end
```

```
        timer = createTimer(now.plusSeconds(5))
        [ gAlloficina.sendCommand(OFF) timer = null ]

    }
    else if (SensorPresE_Oficina.state == OFF)
    {
    }
end

// PASILLO ////////////////////////////////////////////////////////////////////////
// Lámpara techo
rule "Interruptor Aqara Wirelss (1 Boton)"
when
    Channel "mihome:86sw1:158d0...:ch1" triggered SHORT_PRESSED
then
    if (Luz_Techo_Pasillo.state == OFF)
    {
        Luz_Techo_Pasillo.sendCommand(ON)
    }else if (Luz_Techo_Pasillo.state == ON){
        Luz_Techo_Pasillo.sendCommand(OFF)
    }
end

rule "Interruptor Aqara Wirelss (1 Boton)"
when
    Channel "mihome:86sw1:158d0...:ch1" triggered SHORT_PRESSED
then
    if (Luz_Techo_Pasillo.state == OFF)
    {
        Luz_Techo_Pasillo.sendCommand(ON)
    }else if (Luz_Techo_Pasillo.state == ON){
        Luz_Techo_Pasillo.sendCommand(OFF)
    }
end

// Rule Sensor de presencia + luz
rule "Sensor presencia pasillo update"
when
    Item SensorPresE_Pasillo changed
then
    if (SensorPresE_Pasillo.state == ON && modoNoche.state==ON)
    {
        Luz_Techo_Pasillo.sendCommand(ON)
        timer = createTimer(now.plusSeconds(5))
        [ Luz_Techo_Pasillo.sendCommand(OFF) timer = null ]
    }
    else if (SensorPresE_Pasillo.state == OFF)
    {
```

```
    }  
  end  
  
// ENTRADA ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
// Rule Sensor de presencia + luz  
rule "Sensor presencia pasillo update"  
  when  
    Item SensorPresE_Entrada changed  
  then  
    if (SensorPresE_Entrada.state == ON && modoNoche.state==ON)  
    {  
      GatewayE_LuzSwitch.sendCommand(ON)  
      timer = createTimer(now.plusSeconds(5))  
      [ GatewayE_LuzSwitch.sendCommand(OFF) timer = null ]  
    }  
    else if (SensorPresE_Entrada.state == OFF)  
    {  
    }  
  }  
end
```

modo.rules

```
// Rules de todos los modos del sistema domótico  
// MODO NOCHE ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
rule "Amanecer"  
  when  
    Channel 'astro:sun:home:rise#event' triggered START  
  then  
    modoNoche.postUpdate(OFF)  
    sendTelegram("bot1","Modo Noche OFF")  
end  
rule "Atardecer"  
  when  
    Channel 'astro:sun:home:set#event' triggered START  
  then  
    modoNoche.postUpdate(ON)  
    sendTelegram("bot1","Modo Noche ON")  
end
```

sensores.rules

```
rule "Aviso si hay alguna puerta o ventana abierta al salir"
  when
    Item SensorP_Entrada changed from CLOSED to OPEN
  then
    if (gSensoresPVall.state == OPEN && SensorPresE_Entrada.state == ON)
    {
      sendTelegram("bot1","Hay alguna puerta o ventana abierta")
      // Sonido como que se ha dejado algo abierto
    }else{
      sendTelegram("bot1","Todas las puertas y ventanas cerradas")
    }
  end

rule "Timbre + encendido cuando sensor vibracion se active"
  when
    Item SensorVibU_Entrada changed
  then
    sendTelegram("bot1","Alguien esta llamando")
    sendCommand( GatewayO_Sonido, 10)
    sendCommand(GatewayO_SonidoVol,25)
  end
```