This document presents the work carried out for the installation and configuration of a home automation system based on Openhab in a single-family house, whose objective is the installation of home automation devices that help to improve the quality of life of the family and can monitor each of the elements.

Throughout this work, different phases of the software development are presented: listing of requirements, analysis and planning, design, implementation, testing and evaluation of the software and hardware to achieve the objective of the project.

Francisco Javier Jiménez Legaza is the student in charge of the implementation and configuration of the project, and with this work he finishes his degree in Computer Engineering with a specialisation in Software Engineering.

Andrés María Roldán Aranda is the academic head of the present project, and the student's tutor. He is professor in the Departament of Electronics and Computers Technologies

BACHELOR THESIS

Single-family home automation portal based on OpenHAB

Francisco Javier Jiménez Legaza

COMPUTER ENGINEERING

22/23

UNIVERSITY OF GRANADA

Degree in Computer Engineering

Bachelor Thesis

Single-family home automation portal based on OpenHAB

Francisco Javier Jiménez Legaza

2022/2023

Tutor: Andrés María Roldán Aranda

Credits for the cover: **NASA**.
Printed in Granada, November 2022.

“Single-family home automation portal based on OpenHAB”

DEGREE IN
COMPUTER ENGINEERING

Bachelor Thesis

“Single-family home automation portal based on OpenHAB”

ACADEMIC COURSE: 2022-2023

Francisco Javier Jiménez Legaza

DEGREE IN COMPUTER ENGINEERING

“Single-family home automation portal based on OpenHAB”

AUTHOR:

Francisco Javier Jiménez Legaza

SUPERVISED BY:

Prof. Andrés Roldán Aranda

DEPARTMENT:

Electronics and Computers Technologies

D. Andrés María Roldán Aranda, Profesor del departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como director del Trabajo Fin de Grado de D. Francisco Javier Jiménez Legaza,

Informa:

Que el presente trabajo, titulado:

Single-family home automation portal based on OpenHAB

ha sido realizado y redactado por el mencionado alumno bajo mi dirección, y con esta fecha autorizo a su presentación.

Granada, a 16 de noviembre de 2022

Andrés Roldan

Fdo. Prof. Andrés María Roldán Aranda

Los abajo firmantes autorizan a que la presente copia de Trabajo Fin de Grado se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a 16 de noviembre de 2022

[illegible] [illegible]

Fdo. Francisco Javier Jiménez Legaza

Fdo. Prof. Andrés María Roldán Aranda

”Single-family home automation portal based on OpenHAB”

Francisco Javier Jiménez Legaza

KEYWORDS:

OpenHAB, Domotic, Binding, Amazon Alexa, Docker, Automatization, Centralization, Rules, OpenHAB Cloud, OpenHAB Mobile App, Sitemap, Pages, Graphics.

ABSTRACT:

The main objective of this project is based on initialising a home automation portal in a single-family house with OpenHAB as the core. In addition, it is intended to install and configure electronic devices to monitor and control them from the same site (OpenHAB).

This project consists of 3 phases present in any work related to home automation. But as a summary, we can divide it into three main blocks: market analysis, design and implementation and configuration of the elements. During the analysis phase, we will compare different smart devices to see which device is the best suited to meet the client’s needs. Once chosen, we move on to the design phase, which shows how the device is installed in the house and how it communicates with the OpenHAB system. Finally, the implementation phase will explain how the devices have been implemented and configured to add them to the OpenHAB home automation system, as well as the design of the user interfaces to control them.

This final degree project should be seen as the basis of a much larger project, as home automation is constantly evolving and new devices are coming out all the time. Thanks to OpenHAB I have been able to acquire knowledge and skills not only in the field of computer engineering but also in electronics and telecommunications. I have also been able to gather experience in the field of home automation, a sector that is currently booming.

Therefore, the result of all the above culminates in obtaining a basic and functional home automation environment, thus fulfilling the requirements defined in the initial stages of the project, and which closes my university stage in the Degree in computer engineering.

”Single-family home automation portal based on OpenHAB”

Francisco Javier Jiménez Legaza

PALABRAS CLAVE:

OpenHAB, Domotic, Binding, Amazon Alexa, Docker, Automatizacion, Centralizacion, Rules, OpenHAB Cloud, OpenHAB Mobile App, Sitemap, Pages, Gráficas.

RESUMEN:

El objetivo principal del presente proyecto se basa en inicializar un portal domótico en una casa unifamiliar teniendo como nucleo OpenHAB. Y además, se pretende instalar y configurar dispositivos electrónicos para monitorizarlos y controlarlos desde el mismo sitio (OpenHAB).

Este proyecto consta de 3 fases presentes en cualquier trabajo relacionado con la domótica. Pero a modo de resumen, podemos dividirlo en tres grandes bloques: análisis de mercado, diseño e implementación y configuracion de los elementos. Durante la fase de analisis compararemos diferentes dispositivos inteligentes para ver que dispositivo es el que mejor nos conviene para cubrir las necesidad del cliente. Una vez elegidos, pasamos al diseño, fase en la que se muestra como esta instalado el sispositivo en la casa y su comunicacion con el sistema OpenHAB. Por ultimo, la fase de implementación en la que se explicaran como se han implementado y configurado los dispositivos para añadirlos al sistema domotico de OpenHAB, asi como el diseño de las interfaces de usuario para controlarlos.

Este trabajo de fin de grado debe verse como la base de un proyecto mucho mas grande, ya que la domótica esta en constante evolucion y cada vez salen nuevos dispositivos. Gracias a OpenHAB he podido adquirir conocimientos y habilidades no solo en el ambito del Grado de ingeniería informática sino tambien en electrónica y telecomunicaciones. Asimismo he podido reunir experiencia en el sector de la domótica, un sector que ahora mismo esta en auge.

Por lo tanto, el resultado de todo lo expuesto culmina la obtención de un entorno domotico básico y funcional, cumpliendo asi con los requisitos definidos en las etapas inciales del proyecto, y con el cual se cierra mi etapa universitaria en el Grado de ingeniería informática.

'It is fine to celebrate success, but it is more important to pay attention to the lessons of failure.'

Bill Gates, Microsoft co-founder

Acknowledgments:

This work has been possible thanks to a very small number of people, but to whom I owe a great deal of gratitude, not only for the support and understanding they have given me along the way, but also for being the fundamental pieces that have made it possible for me to be where I am today.

For this reason, I would like to dedicate this project first of all to my family, my parents Francisco Javier and Marta. For supporting me and putting up with the bad times and my anger that I have had during the development of this work, thank you very much for supporting me unconditionally.

To my girlfriend, Bea, for trusting me even when I didn't trust myself, for always being by my side supporting me and making me see that I was capable of achieving and doing many more things than I thought I was capable of.

I would also like to thank my tutor Andrés Roldán Aranda and my lab mates Pedro and Antonio for their advice on certain parts of the project. I would also like to mention Luis, a friend of Professor Andres and the Openhab Telegram group who helped me with the problems that arose during the development of the project and who were able to solve my doubts.

And last but not least, I would like to thank all the teachers who have taught me in our School of Computer Engineering and Telecommunications, and the colleagues I have met and have accompanied me during this beautiful stage of my life. All of them have made this long journey more enjoyable and fun thanks to their company both in the good times and in the not so good times, thus enjoying one of the best experiences of my life accompanied by fantastic people.

Agradecimientos:

Este trabajo ha sido posible gracias a un número muy reducido de personas, pero a las cuales debo un gran agradecimiento, no solo por el apoyo y la compresión que me han otorgado a lo largo de este camino, sino por ser las piezas fundamentales, que han hecho que hoy en día me encuentre donde estoy.

Por ello, quisiera dedicar este proyecto en primer lugar a mis familiares, mis padres Francisco Javier y Marta. Por apoyarme y aguantarme en los malos momentos y mis enfados que he tenido durante el desarrollo de este trabajo, muchas gracias por apoyarme incondicionalmente.

A mi pareja, Bea, por confiar en mi incluso cuando ni yo mismo confiaba, por siempre estar a mi lado apoyandome y hacerme ver que era capaz de lograrlo y hacer muchas más cosas de las que no me creía capaz.

Quisiera agradecer también a mi tutor Andrés Roldán Aranda y a los compañeros de laboratorio, Pedro y Antonio por su asesoramiento en ciertas partes del proyecto. Ademas de mencionar a Luis, amigo del profesor Andres y al grupo de Telegram de Openhab los cuales me ayudaron con los problemas que me iban surgiendo durante el desarrollo del proyecto y pudiendo asi resolver mis dudas.

Y por último, pero no por ello menos importante, quisiera agradecer a todos los profesores que me han dado clase en nuestra Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones, y a los compañeros que he conocido y me han acompañado durante esta bonita etapa de mi vida. Todos ellos han hecho que esta larga travesia fuera mas amena y divertida gracias a su compañia tanto en los buenos momentos como en los no tan buenos, disfrutando asi de una de las mejores experiencias de mi vida acompañado de gente fantastica.

Contents

List of Figures

List of Tables

Glossary

Application Programming Interface is a set of subroutines, functions and procedures (or methods, in object-oriented programming) that provides a certain library to be used by other software as an abstraction layer [54].

Binding can be considered as software adapters, which makes the Things available to your home automation system, in other words it is the communication link between the Thing and your system [5].

Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances and building personal area networks (PANs) [55].

C is a high-level and general-purpose programming language that is ideal for developing firmware or portable applications. Originally intended for writing system software, C was developed at Bell Labs by Dennis Ritchie for the Unix Operating System in the early 1970s [50].

C++ Programming language developed in 1980, with the characteristics of being a compiled language, object-oriented and imperative (they tell the computer how to perform a specific task) [53].

Docker is an open source software platform to create, deploy and manage virtualized application containers on a common operating system (OS), with an ecosystem of allied tools[**?**].

Docker Compose is a tool for defining and running multi-container Docker applications. A YAML file is used to configure the Docker application services. Then, with a single command, all the services in the configuration are created and started [12].

Dockerfile is a simple text file or document that includes a series of instructions that need to be executed consecutively to accomplish the processes necessary for the creation of a new image [23].

Domotic a set of techniques aimed at automating a home, integrating technology in the security, energy management, welfare or communications systems.

Extensible Markup Language is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable[63].

HTTPS is an application protocol based on the HTTP protocol, intended for the secure transfer of hypertext data, i.e. it is the secure version of HTTP [57].

HyperText Markup Language is the most basic component of the Web. It defines the meaning and structure of web content[26].

Hypertext Transfer Protocol is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems [56].

IFTTT is an online platform that makes it possible to integrate and make compatible with each other services and connected objects that are not natively connected [14].

Internet of Things describes the network of physical objects "things" that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet [40].

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc [19].

JavaScript Object Notation is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other serializable values). [59].

Modbus is a data communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs)[60].

MPPT Maximum Power Point Tracker (MPPT) is a charge controller used to regulate the amount of electrical energy supplied to a battery [46].

MQ Telemetry Transport is a lightweight, publish-subscribe, machine to machine network protocol for Message queue/Message queuing service [61].

NAS is a storage device connected to the network. Its function is to make backup copies of the files that you indicate in the configuration, both those of your personal computer and those of any other mobile device, although it also has many other functionalities [69].

OpenHAB is an open source home automation platform or system that is the controller or center of your Smart Home [43].

OpenHAB Cloud is a companion cloud service and backend for the OpenHAB open-source home automation software. The OpenHAB Cloud backend provides secure remote access and enables OpenHAB users to remotely monitor, control and steer their homes through the internet, collect device statistics of their OpenHABs, receive notifications on their mobile devices or collect and visualize data etc [39].

Pages is the new user interface released in OpenHAB version 3, to interact with the items. It is not very explored yet but you can create custom widgets or use the default ones to represent the items [35].

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python [52].

Raspberry Pi is a series of small-board, low-cost single-board computers developed in the UK by the Raspberry Pi Foundation, with the aim of putting the power of computing and digital creation into the hands of people around the world [62].

REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding [42].

Rules are rules that are used for process automation. A rule can be activated in many ways, for example when an item changes state (plug goes from ON to OFF) or at a specific time expressed by a cron expression [5].

Sitemap are a way to select and compose items in a user-oriented representation through user interfaces (UI), including the OpenHAB application for Android. That is, it is a UI for the user to interact with the items, being able to modify the status of the items or view data that has been set, such as weather, humidity, etc [35].

SMA Energy Meter also known as the Home Manager energy meter enables precise electrical metering for each phase conductor and in the form of heat balances, i.e. as a feed-in meter or current consumption meter [44].

SSH or Secure Shell, is a remote administration protocol that allows users to control and modify their remote servers over the Internet through an authentication mechanism [22].

Universal Plug and Play consists of a series of standardized communication protocols to facilitate connectivity between different devices in your private network [6].

Unix time is defined as the number of seconds elapsed since midnight UTC on January 1, 1970 [27].

Virtual Private Network an arrangement whereby a secure, apparently private network is achieved using encryption over a public network, typically the internet.

WiFi is a family of wireless network protocols, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access, allowing nearby digital devices to exchange data by radio waves [64].

Z-Wave is a wireless communications protocol used primarily for residential and commercial building automation. It is a mesh network using low-energy radio waves to communicate from device to device, allowing for wireless control of smart home devices [65].

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection [66].

Acronyms

API	Application Programming Interface.
DIY	Do It Yourself.
HTML	HyperText Markup Language.
HTTP	Hypertext Transfer Protocol.
IP	Internet Protocol.
IR	Infra Red.
JSON	JavaScript Object Notation.
LED	Light-Emitting Diode.
MQTT	MQ Telemetry Transport.
PWS	Personal Weather Station.
TCP/IP	Transmission Control Protocol / Internet Protocol.
UGR	University of Granada.
UPnP	Universal Plug and Play.
UV index	Ultraviolet index.
VPN	Virtual Private Network.
XML	Extensible Markup Language.

Chapter 1

Introduction

The Bachelor Thesis presented here, shows the Final Degree Project of the University Degree in Computer Engineering, with specialising in Software Engineering, taken by the student Francisco Javier Jiménez Legaza at the Engineering School of Technology and Telecommunications in Granada. The main objective of this work is to demonstrate the knowledge and skills acquired by the student during the degree and to obtain new knowledge in a completely new area for him. For this objective, the following project has been carried out, which proposes the installation and configuration of a centralised domotics system based on Openhab.

This Final Degree Project is carried out in collaboration with the academic project GranaSAT. This is an aerospace development group of the University of Granada (UGR), formed only by students from different fields of Engineering, such as Aerospace Engineering, Electronic Engineering, Computer Engineering or Telecommunications Engineering among others, under the supervision of Professor Dr. Andrés María Roldán Aranda.

Figure 1.1 – *Granasat logo*

The GranaSAT laboratory gave me an office in the laboratory, and provided me with the necessary equipment and materials to carry out this project. GranaSAT is located in the I+D Josefa Castro Visozo building, which is next to the Aulario of the International Graduate School of Granada and the Clinical Hospital of Granada (Spain). It should also be noted that this project has been carried out both on-site in the laboratory and remotely.

1.1 Motivation

The reason for choosing this project was my interest in home automation and the boom in home automation in recent years. The current rise in popularity of Internet of Things devices and the continued interest in controlling environments has led to an increasing number of people turning to this technology to make their daily lives easier. Many companies are taking advantage of this to develop their own technology, manufacturing equipment and integration software to make it easier to control and help people.

This work is also the base to start a bigger and more ambicious project. This work consisted of designing, installing and configuring the automation of a house by a central Domotic system called OpenHAB, in which its main mission is to unify all the information and functionalities of different ecosystems from different domotic companies under a single system.

1.2 Project objectives

Before I continue with the development and documentation of the project, I would like to explain what I intend to achieve after completing this dissertation.

1. To demonstrate the knowledge and skills acquired in the Bachelor's Degree in Computer Engineering
2. To get out of my comfort zone and acquire knowledge in a new and expanding field such as domotics.
3. Understand systems such as OpenHAB. Learn about hardware of Domotic devices and how they communicate.
4. Learn how to compare devices and how to search for information about them.
5. To build a functional low-cost home automation system, which allows it to be monitored and controlled remotely and securely through any device connected to the Internet with an understandable and intuitive interface (mobile and computer).
6. Unify different Domotic ecosystems under the application of OpenHAB according to the needs that the user may have.

1.3 Project structure

Following the objectives listed above, the project was developed after planning the next structure:

- Chapter 1: Introduction

  This is the current chapter, in which a short introduction, the objectives to be achieved and the structure of the project have been shown.
- Chapter 2: Requirement list

  Section presenting the different requirements necessary to achieve the project's purpose.
- Chapter 3: Market analysis

  Section comparing different hardware and software devices to see which is most suitable for the customer's requirements.
- Chapter 4: House design

  Section showing the design of the different devices chosen in the in-house analysis, as well as their installation in the home.
- Chapter 5: Implementation and configuration

  Section showing what OpenHAB is, its installation and configuration along with the chosen devices.
- Chapter 6: System verification and testing

  Section to show the validation tests done during the implementation to ensure the correct functioning of the OpenHAB system, as well as those of all the installed devices.

- Chapter 7: Conclusions, future work and lessons learned

  Section expressing the opinion on the achievements of the project and discussing possible future improvements of the project.

Chapter 2

Requirement list

The aim of this section is to present in its entirety the list of requirements that has been elaborated during the numerous interviews with the client. The main purpose of this list is to define what our system will be able to do.

The client explained his ideas in mind that he wanted to incorporate in the system to be set up in the different interviews we had. Also, the client described graphically what he wanted in the home and his network architecture for a better understanding.

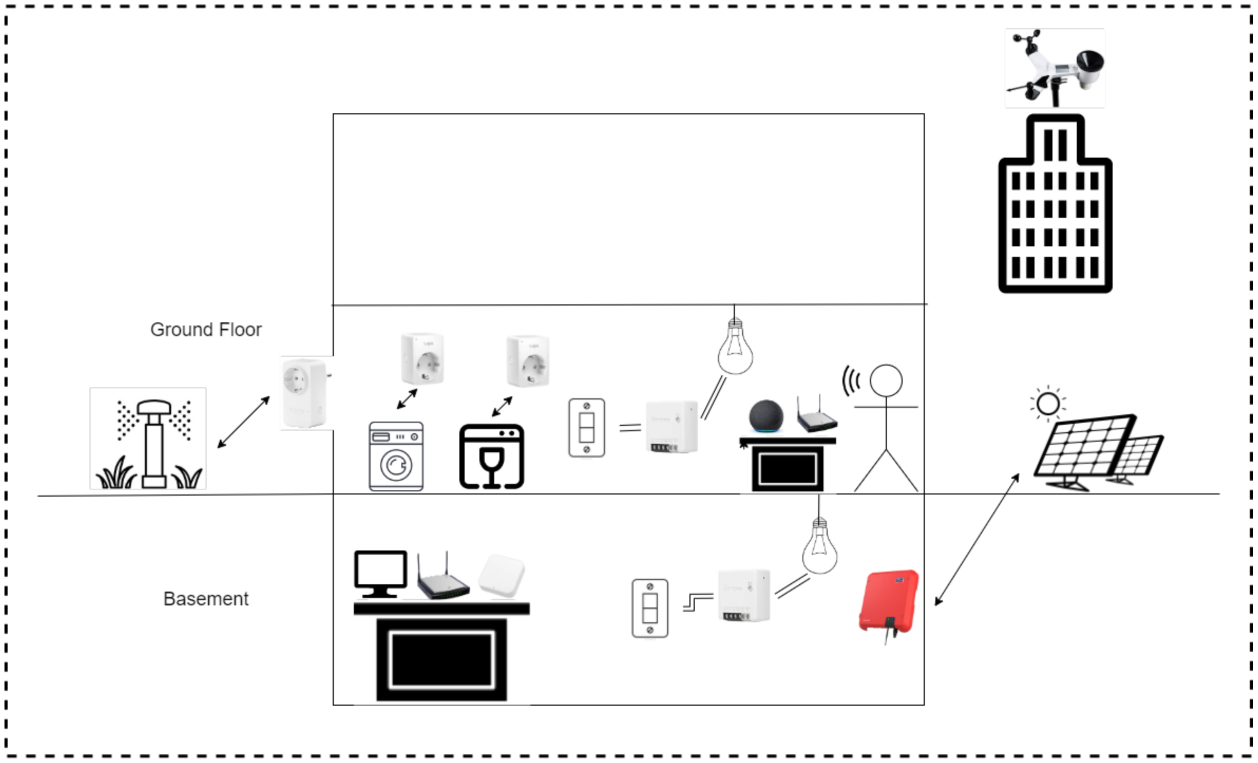


Figure 2.1 – *House with devices*

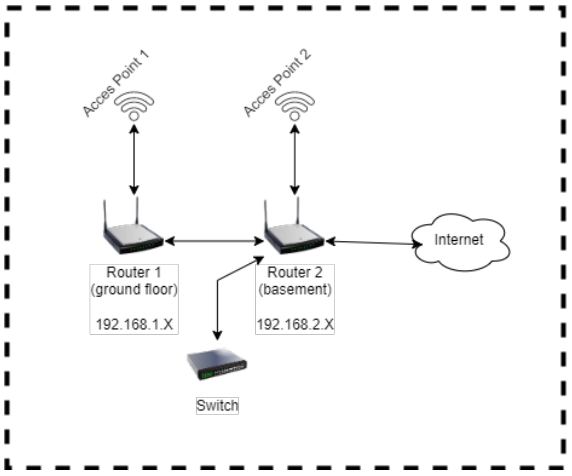


Figure 2.2 – *Network architecture in the home*

The main requirements requested by the customer are as follows:

Ref.	**Main Objetives**
MObj.1	Home automation software analysis and selection
MObj.2	Respect existing network architecture in the house.
MObj.3	Control the system via a mobile app
MObj.4	Being able to access the system remotely via a computer/mobile phone outside the home
MObj.5	Control different devices to be able to manage the lights in the rooms of the house, activating and deactivating them at will.
MObj.6	Control the plugs placed in the system, activating and deactivating them at will.
MObj.7	Voice control of devices, using a virtual assistant and all devices to be compatible with this.
MObj.8	Get information about solar energy devices, Sunny Boy and Home Manager
MObj.9	Obtain information from a personal weather station
MObj.10	Get information from a weather API
MObj.11	Possibility to have future weather forecasts
MObj.12	Group devices according to their functions
MObj.13	Represent the obtained data in graphs for a better visibility of the data.
MObj.14	Possibility of modifying time intervals in the graphs of the data obtained.
MObj.15	Persistence of data across a database
MObj.16	Create user interfaces for the user to control system devices.
MObj.17	Documenting step-by-step installations and configurations of each device
MObj.18	Create rules to control actions of certain devices or to obtain specific values.
MObj.19	Control and play music from the system
MObj.20	Mount the Domotics system in a Docker container
MObj.21	Version control in GitLab

Table 2.1 – *List of main objetives of the project*

Once we are clear about the main objectives of the project given by the customer, we will proceed to list the functional and non-functional requirements covering the whole system from the main objectives:

Ref.	Description
FR.1	The system shall be able to allow only correctly identified and verified administrator users to modify system configurations.
FR.2	The system shall store the data obtained from the different devices in a database to guarantee their persistence.
FR.3	The system shall represent the data obtained by means of graphs for a better visibility of the data.
FR.4	The user shall be able to operate the system by means of a device (computer/mobile) outside the local network.
FR.5	The user will be able to make use of the different devices installed, activating and deactivating them at will.
FR.6	The user will be able to control by voice the devices by means of a virtual assistant and that everything is compatible with this one.
FR.7	The system shall obtain information from the solar energy devices, Sunny Boy solar inverter and Home Manager energy meter, for display in the system.
FR.8	The system shall obtain information from a weather station for display in the system.
FR.9	The system shall obtain information from a meteorological API for display on the system.
FR.10	The system shall be able to obtain metrics of the weather forecast of the future weather for its visualisation in the system.
FR.11	The system shall group the devices according to their functions.
FR.12	The user shall have the possibility to modify the time intervals in the graphs of the data obtained from the different devices.
FR.13	The system shall include interfaces for the user to control the system devices.
FR.14	The user shall be able to control and play music from the system.

Table 2.2 – *List of functional requirements*

Ref.	Description
NFR.1	The system shall be scalable and exportable.
NFR.2	The system shall be easily maintainable.
NFR.3	Step-by-step documentation of each installation and configuration of any device or service installed on the system.
NFR.4	The home automation system must be mounted in a Docker container.
NFR.5	The distance between the host and each device shall not exceed the threshold to ensure proper communication.
NFR.6	The system shall always be available except when there is an Internet outage or power failure.
NFR.7	The physical devices shall always be available except when there is an internet connection outage or power failure.
NFR.8	The system shall display a message for any device or service error.
NFR.9	The system will have version control in GitLab.

Table 2.3 – *List of non-functional requirements*

Chapter 3

Market analysis

Once we have the customer's main requirements, we analyse the market to find the best products that meet the customer's requirements.

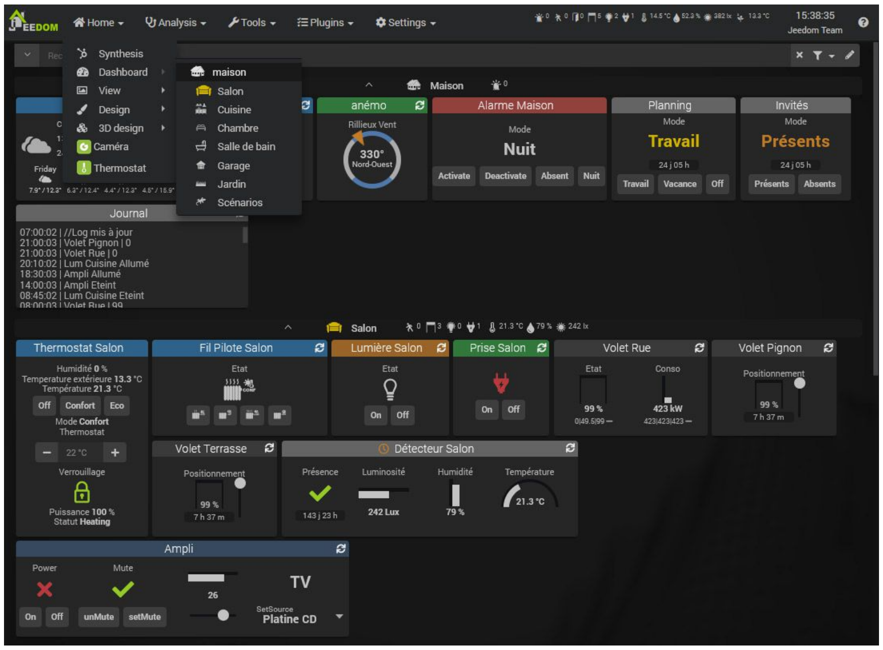
3.1 Open source domotic systems

In this section, a study of the different open source home automation systems will be carried out, in which the most relevant characteristics of the different software alternatives currently available on the market will be presented. In addition, we will explain the choice of the home automation system that we have decided to use in this project.

The **home automation** software is a program that brings together all the **devices of different ecosystems** in one place, thanks to its different **communication protocols** that facilitate the incorporation of these devices and without the need to have an infinite number of applications to manage the different devices that we can have at home. In addition, **routines** can be programmed and their appearance can be **customised** to suit each individual's preferences. You can also create your **own DIY device** and through certain communication protocols connect it to this software and manage it together with the other devices.

There are many open source softwares dedicated to home automation, but we will explain the most famous and the most used ones:

- Jeedom is free open-source software that can be installed on any Linux system, on a Raspberry Pi 2, 3 or 4 or on NAS. It is based on a kernel with multiple functionalities : scenario management, textual and sound interaction with the home automation installation, visualization of histories and generation of curves and graphics, linking of all equipment and connected objects, customization of the interface. Its clear interface and intuitive allows you to set up a complete solution without development knowledge.



```
07:00:02 | //Log mis à jour
21:00:03 | Volet Pignon | 0
21:00:03 | Volet Rue | 0
20:10:02 | Lum Cuisine Allumé
18:30:03 | Ampli Allumé
14:00:03 | Ampli Eteint
08:45:02 | Lum Cuisine Eteint
08:00:03 | Volet Rue | 99
```

Figure 3.1 – *Jeedom interface*

Jeedom does not require access to external servers to operate. Your entire installation is managed locally and you are therefore the only ones who have access to it to guarantee complete confidentiality.

Thanks to its flexibility and numerous customization parameters, each user can create their own Jeedom home automation. Using widgets, views and designs, you have complete freedom to imagine your own interface if you wish [3].

- Domoticz is a very lightweight home automation system developed in C/C++ designed to run on a variety of operating systems, including Linux, Windows, Mac OS and other embedded systems such as Raspberry Pi, network storage equipment (NAS).

  The interaction with the user is done by HTML through an integrated web server, and it is adaptive to the type of screen, being able to use it from tablets or mobiles, although it is somewhat archaic and unattractive.

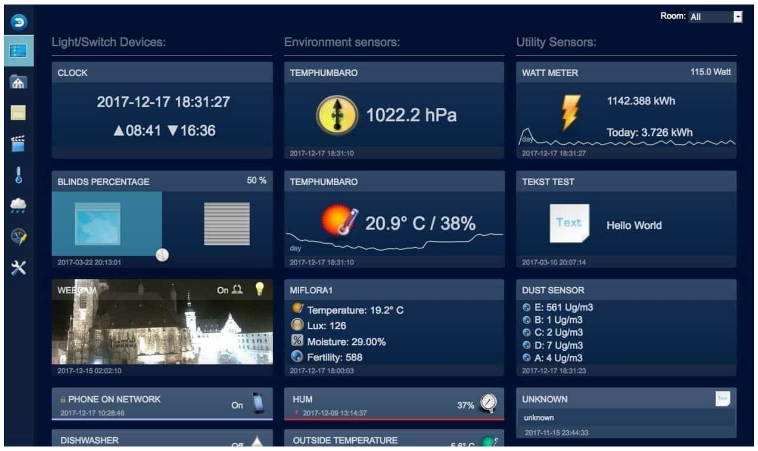


Figure 3.2 – *Domoticz interface*

Domoticz allows monitoring and configuration of various devices including lights, switches, various sensors/meters such as temperature, rain, wind, ultraviolet radiation, electricity usage/production, gas consumption, water consumption and many more. Notifications/alerts can be sent to any mobile device [1].

- Home Assistant, one of the most popular, developed in Python, also compatible with virtually any server technology for local, Docker or Raspberry Pi installation. This free software has a large community behind it and it is very easy to make new integrations into the system.

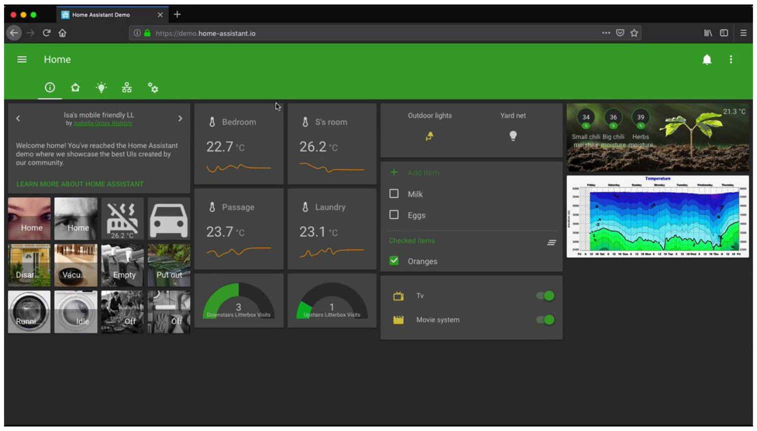


Figure 3.3 – *Home Assistant interface*

Its design is also attractive and responsive. Furthermore, Home Assistant is a standalone system and does not rely on third-party cloud services, focusing on local control of information. Support for third-party devices and services is handled in its community forum.

It is always up to date with the latest in the sector, but this has an impact on the stability of the updates, because the new protocol or gadget integrations have not yet been well studied [2].

- OpenHAB, one of the most famous, programmed in Java, is characterised by the large number of technologies and brands supported thanks to its architecture based on "bindings" (plugins), small code modules developed to communicate with specific devices, which makes possible the infinity of devices compatible with it.

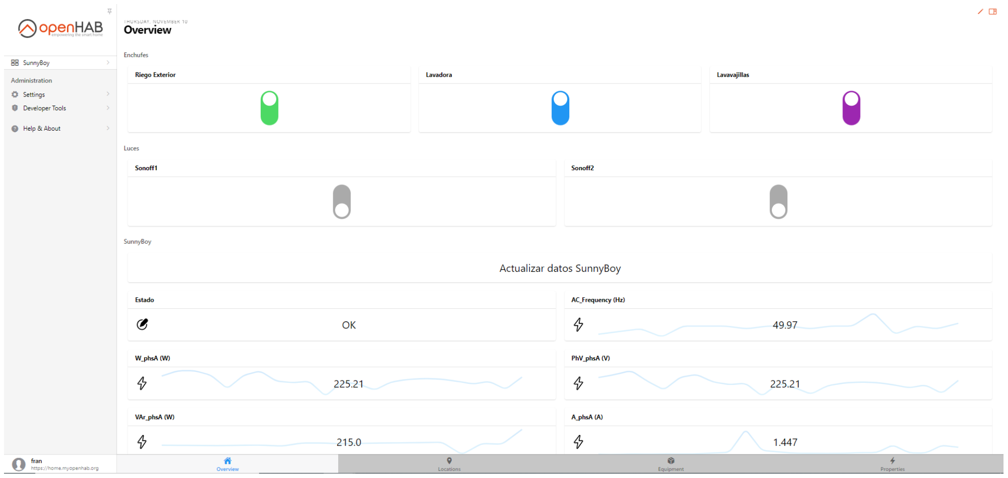


Figure 3.4 – *OpenHAB interface*

OpenHAB is also compatible to be installed on various platforms, locally, Docker or Raspberry Pi, as well as the flexibility for its data control by users.

The pace of development sometimes feels too slow, but this is done to make the updates as stable as possible, for the safety of its users. But you can use the less stable versions that are in development [5].

A comparative table will be made between all of them, to see which one is better:

Features	Jeedom	Domoticz	Home Assistant	OpenHAB
Open Source	Yes	Yes	Yes	Yes
Multi-protocol	Yes	Yes	Yes	Yes
Autonomous	Yes	Yes	Yes	Yes
Customizable	Yes	Yes	Yes	Yes
Community	Medium sized community (2º)	Small community (3º)	Big community (1º)	Big community (1º)
Mobile App	Yes	Yes	Yes	Yes
Graphical interface	2º	3º	1º	1º
Paid plugins	Yes	No	No	No
Weight	67.5 MB	14.86 MB	150 MB	96 MB
Documentation	3º	4º	2º	1º
Device compatibility	3º	4º	1º	2º
Updates	stable	outdated	unstable	stable
Election	✗ (3º)	✗ (4º)	✗ (2º)	✓ (1º)

Table 3.1 – *Comparative table of Home Automation systems [47, 67]*

After evaluating the different options, the decision was between two: OpenHAB or Home Assistant, as the others either had payments or were older and were not adapting to the new changes, such as protocols, privacy or security. It **was decided to install** OpenHAB, on the one hand because the client wanted that system, for its large community and compatibility of devices with different bindings and on the other hand, for its good documentation and customization.

Figure 3.5 – *OpenHAB logo*

3.2 Virtual Assist

Before we start, we will explain what a virtual assistant is and how it works inside.

Virtual assistants are artificial intelligences that respond to **voice commands**, which are becoming increasingly sophisticated and intelligent. These systems **learn** through interactions with the consumer, and continuously improve their repertoire. The more they interact with people, the more they learn and are moving beyond the mobile screen to become business solutions and functional devices for the home. [21].

These functional devices for the home are normally loudspeakers and work through a **keyword** that wakes them up to attend to you, for this **they are listening all the time in standby mode**, each brand has its own keyword, for example Amazon is "Alexa", Google with "Ok Google" and Apple with "Hey Siri".

When the speaker **detects the keyword** it will activate and start recording, then send it over the Internet to the voice recognition and processing system, which decodes the message and performs the reverse process by sending a human-comprehensible response back or executing the command given, e.g. play music or activate/deactivate some connected device.

The **voice recognition** system is 'intelligent' and through an algorithm is able to learn how we use our words in order to understand us better and better, to give us an adequate service and to do it in a more and more natural way [28].

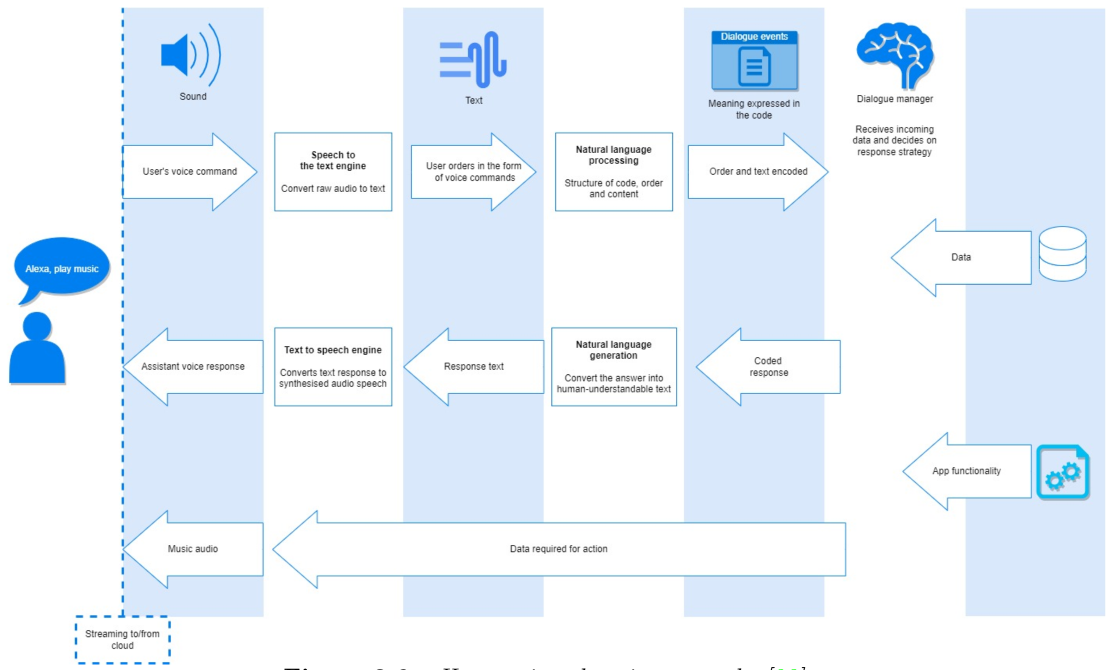

Sound
Text
Dialogue events
Meaning expressed in the code
Dialogue manager
Receives incoming data and decides on response strategy
User's voice command
Speech to the text engine
Convert raw audio to text
User orders in the form of voice commands
Natural language processing
Structure of code, order and content
Order and text encoded
Data
Alexa, play music
Assistant voice response
Text to speech engine
Converts text response to synthesised audio speech
Response text
Natural language generation
Convert the answer into human-understandable text
Coded response
App functionality
Music audio
Data required for action
Streaming to/from cloud

Figure 3.6 – *How a virtual assistant works [28]*

Having explained what a virtual assistant is and how it works, we continued with the analysis. The client wanted to use a virtual assistant to control most of the elements connected to OpenHAB, so a comparative was made between the main assistants on the market: **Amazon Alexa** (echo dot 4), **Google Home** (google nest mini) and **Siri** HomeKit (homepod mini).

Features	Alexa	Google Home	Siri
Company	Amazon	Google	Apple
General questions	2º	3º	1º
Sound quality	1º	2º	2º
Response Speed	1º	2º	3º
Meals and recipes	2º	1º	3º
Online shopping	1º	2º	2º
Calls and messages	2º	3º	1º
Addresses and navigation	3º	1º	2º
Device compatibility	1º	2º	3º
Routines	1º	1º	2º
Price	39.99 €	59 €	99 €
Extras	Skills	Google services (google maps, google calendar...)	Apple services
Election	✓ (1º)	✗ (2º)	✗ (3º)

Table 3.2 – *Comparative table of virtual assist [28, 15]*

There are also other assistants such as **Bixby (Samsung)** or **Cortana (Microsoft)**, but these do not have devices to talk to and interact with them without the phone or the computer, such as the echo dot from Alexa or the Google Nest mini, they are only integrated in their devices (such as mobiles, samsung televisions by bixby) or software (such as the windows operating system in which cortana is integrated) and they could not even connect to OpenHAB.

Another important aspect is the **security and privacy** of our data, since their speakers, as has been mentioned, are always listening and recording the messages we have with them. This has caused users to be concerned about their privacy.

In general, all companies explain that the recordings are made to improve the user experience by learning from user interaction, but we will never know 100%. These recordings are perfectly erasable at the user's request.

In conclusion Alexa and Google are now above Siri, because Apple does not have as much compatibility with all devices on the market because it has more stringent security protocols, but lately many companies are adapting their devices to meet these protocols, in order to integrate them with the 3 assistants. Our decision was to stay with **Alexa (echo dot) from Amazon** because of the speed of response that is superior to the other two, the customisation with the different skills (exclusive to Amazon Alexa) and the infinity of routines that can be done, Amazon Alexa allows you more customisation of routines for devices, such as listening to a dog barking or glass breaking to be able to execute a routine.

Figure 3.7 – *Amazon Alexa*

3.3 Plugs

Smart plugs are very useful devices to turn on and off any appliance or use them as a switch, for example to turn on the radiator, heater or coffee maker, to save energy and make it more convenient for the user, and not having to be plugging and unplugging the cable of the device.

The customer wanted switches to turn on and off the programmed kitchen appliances, such as the washing machine and dishwasher.

These **smart plugs** are electronic plugs that connect to a normal power socket, but allow us to control any other device that we connect to it from our mobile device. These plugs are mainly connected via WiFi, although they can also be connected via Zigbee (a compatible Zigbee hub is required), and their main function is to switch any device on or off and obtain its consumption information instantly, if the plug allows it. In this way, we can adapt the electrical consumption of our homes to the most optimal conditions [68].

The reason for using smart plugs is not only to control the switching on and off of the connected device, but also to save on electricity bills and optimise time.

They are also compatible with virtual assistants and can be controlled by voice commands. There are many types of plugs:

- **Basic smart plugs**, which are plugged into the normal socket.
- **Smart embedded plugs**, which replace normal plugs and are, as the name suggests, embedded in the wall.
- **Smart outdoor plugs**, like normal plugs, but with materials and to resist the weather conditions.
- **Smart power strips**, which can be controlled both plug by plug and all at the same time.

Most smart plugs are **protected** against excessive currents, short circuits, overheating and overloading, making them safer to use than even traditional plugs. [4].

The comparison will be about basic smart plugs, as we have chosen **Amazon Alexa** as our virtual assistant we will make sure that the chosen plug(s) are **compatible** with it.

Features	Smart Plug	Tapo P100	Tapo P110	Hue Smart Plug	Smart Plug
Company	Amazon	TP-Link	TP-Link	Philips	Aqara
Voice control with alexa	Yes	Yes	Yes	Yes	Yes
Dimensions	10,11x5,6x7,94 cm	5,1x7,7x7,2 cm	10,6x8,5x7,5 cm	5,1x5,1x8,4 cm	8,5x6,3x6,3 cm
Weight	166 g	100 g	230 g	100 g	110 g
Power supported	2300 W	2300 W	2300 W	2300 W	2300 W
Energy Meter	No	No	Yes	No	Yes
Communication protocol	WiFi	WiFi	WiFi	Zigbee Bluetooth	Zigbee
Timer	No	Si	Si	No	Si
Price	24,99 €	10,99 €	14,59 €	31,99 € (more Hub connected by Zigbee)	28,99 € (more Hub)
Election	✓ (2º)	✓ (1º)	✗ (3º)	✗ (5º)	✗ (4º)

Table 3.3 – *Comparative table of smart plugs [68, 4]*

In terms of price-quality comparison, the winner is the Tapo P110, as it offers everything that its Zigbee competitors offer, but based on the WiFi protocol and at a much lower price.

Unfortunately the customer made a mistake when buying the plugs and bought the Tapo P100 plugs, which are the same as the Tapo P110 but lighter and without a energy meter. The Amazon Smart Plug was also chosen, as it was on offer in a pack with the Alexa.

(a) *Amazon Smart Plug*

(b) *Tapo P100*

Figure 3.8 – *Smart plugs*

3.4 Relays

Another option for not filling the house with intelligent plugs, are the relays or intelligent switches that allow us to **control the passage of current** simply and allow us to continue using our **traditional switch or plug**, with the advantage of making them intelligent to be able to integrate them with our home automation system, just place the relay in the junction box, in the junction box or in the box of the device, whether light switch, plug, etc, for example to control the light switches, sockets, garage motor, blinds, doorbell, etc.

These relays have several connection options mainly: WiFi or Zigbee, there is also Z-Wave but there are not so many relays with this technology, so it has been discarded.

Features	WiFi	Zigbee
Product variety	Most products are WiFi	There will be more and more products as Zigbee is a very optimal technology for home automation and all the big brands are working and opting for it.
Data transfer	Higher data transfer	Lower data transfer
Price	Cheaper, due to the large production that exists.	A little more expensive than WiFi, although little by little they are increasing the production, so soon they will lower the prices.
Individual vs centralised connection	WiFi routers allow a maximum of 32 connected devices to guarantee the stability of the connection.	Allows hundreds of Zigbee products to be connected simultaneously and with high stability to the network via WiFi as a single device via the gateway (hub), without affecting the stability of the router connection.
Energy efficiency	They cost more than Zigbee-connected products.	They use very low energy.

Table 3.4 – *Comparative table of communication protocol*

Having explained what a relay is and what it is used for, we are going to compare different alternatives of these relays, both WiFi and Zigbee. The best known brands are Shelly and Sonoff, although there are already companies such as Tuya selling this technology.

Features	Shelly 1	Sonoff mini	Tuya relay mini
Voice control with Alexa	Yes	Yes	Yes
Dimensions	41x36x17 mm	42.6x42,6x20 mm	40x40x21 mm
Weight	29g	36 g	36 g
Amperage max	16 A	10 A	16 A
Power max	3500 W	2200 W	3450 W
Communication protocol available	WiFi	WiFi Zigbee	WiFi Zigbee
Timer	Yes	Yes	Yes
Routines	Yes	Yes	Yes
Price	WiFi: 12,71 €	WiFi: 19,99 € Zigbee: 19,30 €	WiFi: 8,65 € Zigbee: 11,90 €
Election	✗ (3º)	✓ (1º)	✗ (2º)

Table 3.5 – *Comparative table of relay*

Sonoff devices have been chosen in their Zigbee version, as they are the most famous brand on the market. With them the lights of a room will be controlled, which is what the client wanted, it is not necessary that they have so much maximum power and therefore they spend less energy, although it is not that they spend a lot.

As we have opted for Zigbee we need a Zigbee **gateway/hub**, we chose a multipurpose hub such as the Tuya hub and its beautiful Smart Life application interface, although not all Zigbee devices connect to all hubs on the market, you have to look at compatibility with the hubs, in this case the hub is compatible with the relays..

We have **discarded** the WiFi versions, because in the long term, if the customer wants to control more things, the WiFi network could be damaged by installing more devices, causing a loss of stability in the communication between the devices.

(a) *Sonoff mini Zigbee* **(b)** *Tuya Hub Zigbee*

Figure 3.9 – *Relay and hub Zigbee*

3.5 Weather station

A weather station is a set of measuring tools that help to predict the weather both at the present time and to predict it over a short period of time. They can be installed either inside or outside the house, depending on the type of weather station.

Weather stations could be divided into two specific types, **domestic** and **professional**, although the main difference is **the quality of the sensors** which affects both the reliability of the data and the price of the station. However, when it comes down to it, increasingly those designed for home use have better measuring tools and a much wider range of instruments.

The main purpose of a weather station is to **collect enough information** to be able to predict the weather [17].

Depending on the number of sensors in the station, the variables they are able to measure can be:

- **Wind speed and direction**
- **Barometric pressure**
- **Precipitation and cloudiness**
- **UV index and solar radiation**
- **Air pollution density**
- **Temperature and humidity**

Now we are going to make a comparison between different stations to see which one we choose for the project.

Features	Froggit HP1000SE PRO Single Sensor Edition	Froggit WH3000 SE	BRESSER 4 CAST Pro WLAN 7003210
Display	Colour and dynamism Dynamic tables and charts	Colour and dynamism Information without charts	Colour and dynamism Information without charts Multiday forecast
Possibility of extra sensors	Yes	Yes	Yes
Connection to web-based weather platforms	Weather Wunderground WeatherCloud Ecowitt	Weather Wunderground WeatherCloud Ecowitt	Weather Wunderground WeatherCloud ProWeatherLive AWEKAS
Optional measurements with extra sensors	Lightning and storm detection Ground humidity Air quality	Lightning and storm detection Ground humidity Air quality	Lightning and storm detection Ground humidity
Extras	Measurement of ultraviolet rays Measurement of solar radiation Solar panel with rechargeable batteries Weather forecast	Measurement of ultraviolet rays Measurement of solar radiation Weather forecast	Measurement of ultraviolet rays Measurement of solar radiation Measurement of cloudiness and air quality Solar panel with rechargeable batteries Weather forecast
Future forecasts	Yes (API)	Yes (API)	No
Price	295 €	150 €	259 €
Election	✗ (3º)	✓ (1º)	✗ (2º)

Table 3.6 – *Comparative table of weather station*

After the comparison, the **Froggit WH3000 SE** weather station was chosen, as it was the best seller in 2021 due to its great **quality-price ratio** and the other two were out of budget, and covers the customer's needs, which were to obtain basic measurements such as temperature, humidity, precipitation, wind speed and solar radiation, have a screen showing the current measurements and the possibility of connecting to the most important meteorological platforms. This station can have several names depending on the supplier

Froggit WH3000 SE, Froggit WH3500, Froggit WH3600 or SAINLOGIC WS3500 [24].

Figure 3.10 – *Froggit WH3000 SE*

3.6 Weather API

Since the weather station makes a measurement of the weather where the station is located, that is why we resort to the idea of obtaining weather information from a location through a weather API.

APIs allow interacting with a computer or a system to **obtain data** or execute a function, so that the system understands the request and fulfils it. With them, enterprises can share resources and information while retaining security, control and authentication, allowing them to determine what content each user can access [11, 54].

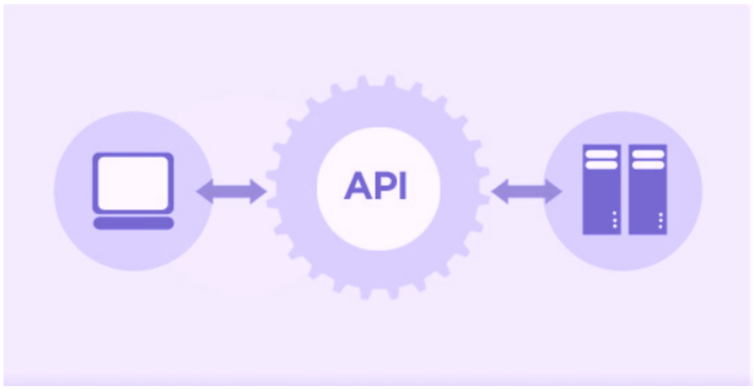


Figure 3.11 – *API*

In this case the weather APIs are REST API since when the client sends an HTTP request it transfers a representation of the state of the requested resource to the requester or endpoint. The information is delivered via HTTP usually in JSON, HTML or XML format.

In the HTTP request, the headers and the parameters of the HTTP request tell the API values to return the information you are requesting, for example in the weather APIs you will have to specify the geolocation or the coordinates of where you want to get the weather measurements [41].

We are going to make a comparison of a few weather APIs, to see what they have to offer:

Features	Openweathermap	Accuweather	Weatherbit
Nº of Request	1000000 calls per day	50 calls per day	50 calls per day
API Key	Yes	Yes	Yes
Response type	JSON, XML, HTML	JSON	JSON
Measurements	Outdoor and apparent temperature Pressure and humidity Wind direction and speed Cloudiness and visibility Rain and snow	Temperature Pressure Wind speed Cloudiness and visibility Rain	Outdoor and apparent temperature Pressure and humidity Wind direction and speed Cloudiness and visibility Rain and snow UV index Air Quality Index Solar Radiation
Future forecasts	Yes	Yes	No
Price	Free	Free	Free
Election	✓ (1º)	✗ (3º)	✗ (2º)

Table 3.7 – *Comparative table of weather APIs*

Openweathermap **has been chosen**, as we have worked with it in previous projects and we know its reliability, and OpenHAB has a binding to facilitate its integration into the system. In addition, it has also been chosen for its number of calls per day, as it is the highest among its competitors, so we can update the weather data every less time.

Weatherbit has very interesting variables such as air quality, UV index or solar radiation, which can be useful when making calculations for solar panels in the future, but 50 calls per day is not a good number to have frequently updated data. Also, integration with OpenHAB would be more complicated as it does not have a specific binding for it.

Figure 3.12 – *OpenWeatherMap*

3.7 Solar inverter

Solar energy is growing, because of the increased price of electricity and gas, so many people are installing solar photovoltaic panels to produce their own energy or at least reduce the consumption of the traditional electricity grid.

A **solar inverter** is a converter that transforms the direct **current energy** coming from the photovoltaic generator into **alternating current**, it is the most important part of the solar installation. This energy converted into alternating current is the energy used by the electrical devices that we have at home [58, 48].

In addition to **energy conversion**, it has other functions [20]:

- **Energy efficiency**: Maximises the power generation of the solar panels.
- **Monitoring and protection**: A solar inverter monitors the photovoltaic system's energy yields, electrical activity and signals when problems occur.
- **Constant operation**: The photovoltaic inverter offers constant operation by dissipating heat in a consistent manner .

There are different **types** of inverters, the most common are [20]:

- **Grid-connected inverters**: These are used in solar installations that are connected to the general electricity grid. They also maintain the voltage of the energy generated by the solar panels a little higher than that of the grid. In this way, the use of solar energy is prioritised, saving as much as possible by using self-consumption first.
- **Inverters for off-grid installations with batteries**: 100 % independent of the general electricity grid. These types of inverters usually have integrated functions to charge the batteries, control the charging and provide safety.
- **Hybrid inverters**: include the possibility of operating with or without batteries.
- **Microinverters**: These are very small solar inverters connected directly to individual solar panels. Since each micro inverter and panel operate independently, they are an excellent choice for complex roof designs and locations with shading problems .

Within each type, there are also **single-phase** and **three-phase** photovoltaic inverters, the difference being that single-phase energy has only one phase and one alternating current (220-230V) and three-phase energy is that which has 3 phases and 3 alternating currents (380-400V). Single-phase installations are the most common in homes and three-phase installations are more common in shops, industrial buildings and factories.

Photovoltaic inverters are produced in a wide range of power ratings, always expressed in kilowatts (kW). The power and type of inverter must be chosen according to the installation of each house [25].

We are going to make a comparison of the **hybrid solar inverters** of some of the most famous brands in the market:

Features	SolarEdge SE5000 HD-Wave	Fronius Primo 5.0-1 5kW	SMA SunnyBoy 5.0	Kostal PIKO MP PLUS 4.6-2 hibrido
Country	Israel	Austria	Germany	Germany
Efficiency	99,2 %	98,1 %	97 %	97,4 %
Nº of MPPT	1 for panel	2	2	2
Warranty	12 years extendable to 25	5 years extendable to 20	5 years extendable to 20	5 years extendable to 20
Quality technical support	1º	2º	4º	3º
Quality monitoring	1º	3º	2º	4º
Level of protection	IP65	IP65	IP65	IP65
Price	1.499 €	1.292 €	1.255 €	1315 €
Election	✗ (3º)	✗ (2º)	✓ (1º)	✗ (4º)

Table 3.8 – *Comparative table of solar inversor [8]*

The best option is **SolarEdge**, as it excels in everything, the problem is the price, which is more expensive than the other companies. As far as the other 3 inverters are concerned, they share many of the same capabilities and do not differ much.

We have chosen the **SMA Sunny Boy inverter** because its efficiency in converting energy into alternating current is good, very close to that of its competitors, but above all we have chosen it for the quality of monitoring, as this is one of the fundamental points, and although its quality of technical support is the worst in the comparison, it is still very good.

This selection includes a complete solar power kit from SMA, solar panels and the SMA SunnyBoy solar inverter. In addition, SMA offers a SMA Energy Meter (optional) and SMA Sunny Island batteries (optional as it is a hybrid inverter).

Figure 3.13 – *Sunny Boy*

3.8 Project budget

The aim of this section is to analyse the costs of both labour and materials used to carry out the project. This will allow us to estimate the cost that a domotics company would have to pay to carry out the same project.

A division into 3 parts has been made to separate the different elements used in the project. On the one hand, the material or hardware costs, on the other hand, the software costs of the services or programs used and, finally, the labour costs.

3.8.1 Physical resources

This section will show the hardware resources used for the project, such as the different smart devices, like the value of the server or host that hosts OpenHAB, such as the solar inverter and solar panels, all the material has been provided by the GranaSAT group and the tutor A.Roldan. In addition, I have provided my own laptop to be able to access the server on a daily basis and configure everything both at home and in the lab.

Item	**Quantity**	**Total cost [€]**
Personal laptop	1	600,00
Echo dot 4º generacion	1	22,66
Amazon Smart Plug	1	14,99
Moes Tuya Zigbee Smart Gateway Hub	2	44,68
TP-Link Tapo P100	4	36,04
Sonoff ZBMINI Zigbee	4	39,16
Froggit WH3000 SE	1	149,99
Solar Panels	10	1.750,00
Sunny Boy 3.0	1	670,00
Home Manager 3 phases	1	279,00
Computer (server)	1	150,00
BUDGET		3.756,52

Table 3.9 – *Project physical resources*

3.8.2 Software resources

Throughout the project, I have been using different software to carry out the project. I would like to add that all the software used has not entailed any additional cost to the project, as free versions have been used.

Item	Total cost [€]
Visual Studio Code	0,00
Gitlab	0,00
Docker	0,00
ClickUp	0,00
Telegram	0,00
OpenHAB	0,00
OpenHAB mobile App	0,00
Team Viewer	0,00
Amazon Alexa App	0,00
Tapo App	0,00
Tuya Smart Life App	0,00
OpenWeatherMap API	0,00
WeatherUnderground API	0,00
Ambient weather app	0,00
OpenHAB Cloud	0,00
BUDGET	0,00

Table 3.10 – *Project software resources*

3.8.3 Human resources

In addition to the material used, when making a correct estimate, we must include the price of the work carried out by the people involved, i.e. the labour. Looking at different websites such as Glassdoor or Indeed, which offer salary data in relation to the job position consulted, the hourly rate after consulting the websites gives an average of more or less 15,00 €/hour for a junior software engineer. I have invested a total of 608 hours in the project. These sources are considered reliable for the information they provide, as it is the employees or ex-employees themselves, or directly from the company, who give the salary range of each one. Also, websites such as Autosolar have been consulted to see how much it costs to install everything related to solar energy.

Item	Total cost [€]
Solar energy installation	1.800,00
Junior Software Engineer	9.120,00
BUDGET	10.920,00

Table 3.11 – *Project human resources*

3.8.4 Final project price

Finally, after having calculated separately the costs of each of the groups proposed for estimation. The last step would be to add them together to obtain the final estimated value of the total project costs.

Sub-project	Budget [€]
Physical resources	3.756,52
Software resources	0,00
Human resources	10.920,00
TOTAL BUDGET	14.676,52

Table 3.12 – *Project total budget*

3.9 Project planning

The aim of this section is to show the guidelines to follow in order to carry out the project planning. To do so, a Gantt chart will be elaborated, in order to see the time invested in the different tasks as well as the total time of the project and its progression.

This diagram has been made with the ClickUp tool, which has already been used for the management of tasks in university subjects. The Gantt chart without the subtasks, as it was not possible to include all the subtasks, can be seen in the following image or to see the subtasks of each one you can do it by accessing from this **link**:

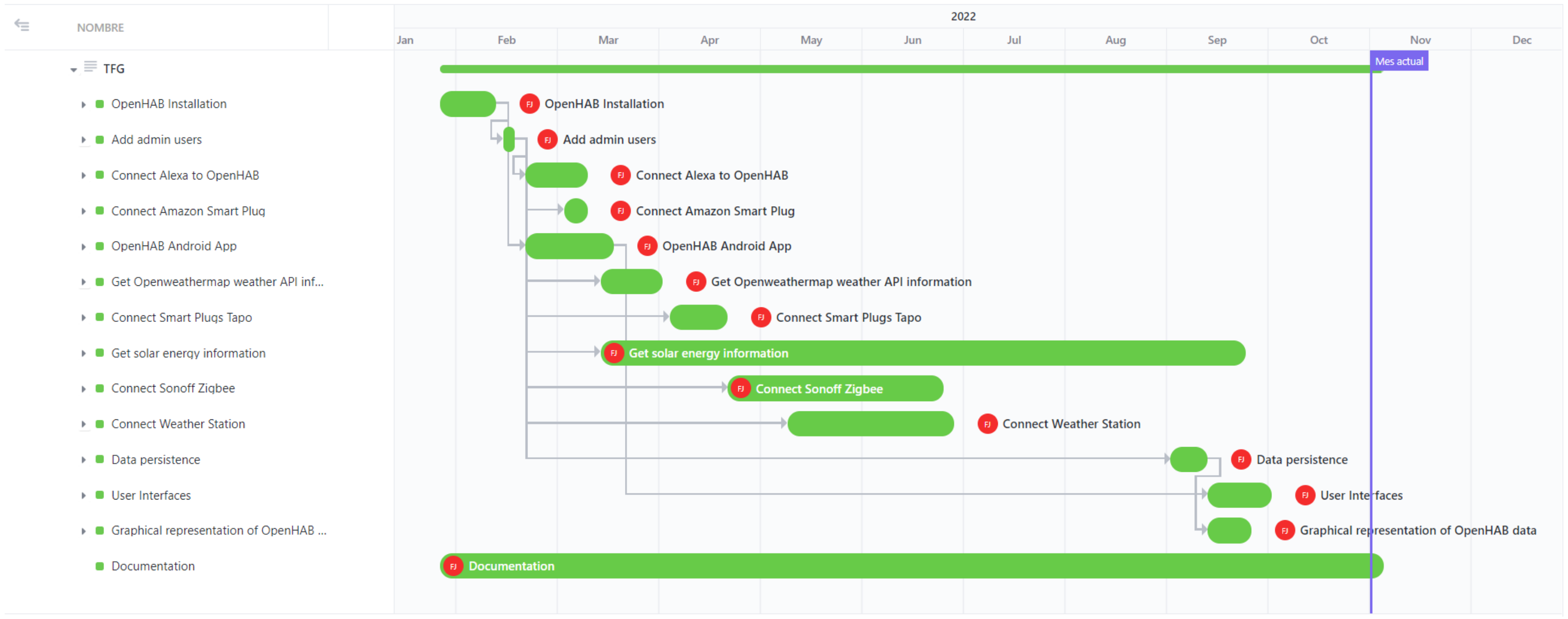


Figure 3.14 – *Project Planning*

Chapter 4

House design

In the following section, once all the comparisons and decisions from the analysis phase have been made, we will detail the designs for each product and how they will be integrated into the house, as explained by the client in the figure 2.1, in order to meet the objectives of the project.

As the client asked us to respect the network architecture already present in the house, which can be seen in the figure 2.2, we will try to integrate everything with it.

4.1 Virtual Assistant

The virtual assistant that has been chosen is Amazon Alexa, as can be seen in the analysis of the virtual assistants in the section 3.2. This will help the customer to facilitate the use of the devices connected to OpenHAB.

This image shows how the communication and operation in the house would look like.

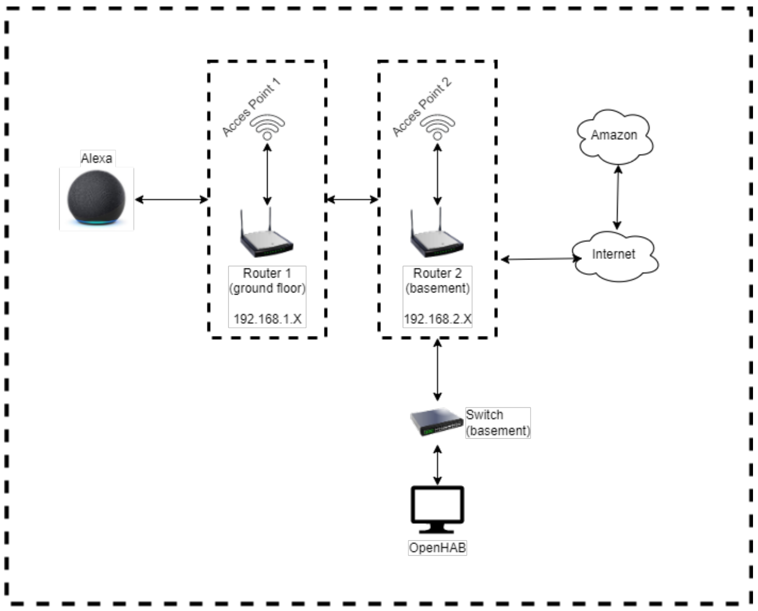


Figure 4.1 – *In-house assistant design*

As can be seen in the image, the echo dot with Alexa is connected to router 1 located on the ground floor of the house and OpenHAB is connected to router 2 located in the basement of the house. The information that Alexa transmits must pass through the two routers until it reaches the Internet so that it reaches Amazon's servers and sends the order to be executed. OpenHAB receives this information via Amazon's binding in order to update the corresponding items in the OpenHAB system in case the device app is used and the OpenHAB app is not used.

4.2 Smart Plugs

The plugs that have been chosen, will be used to control different electronic devices to activate and deactivate them as the customer wishes. As we have seen in the section 3.3 we have chosen two types of sockets.

4.2.1 Amazon Smart Plug

This plug is part of the Amazon ecosystem, so the app from which it is controlled is the same as the Alexa app, Alexa App.

The main function of this plug will be to control the outside irrigation of the house.

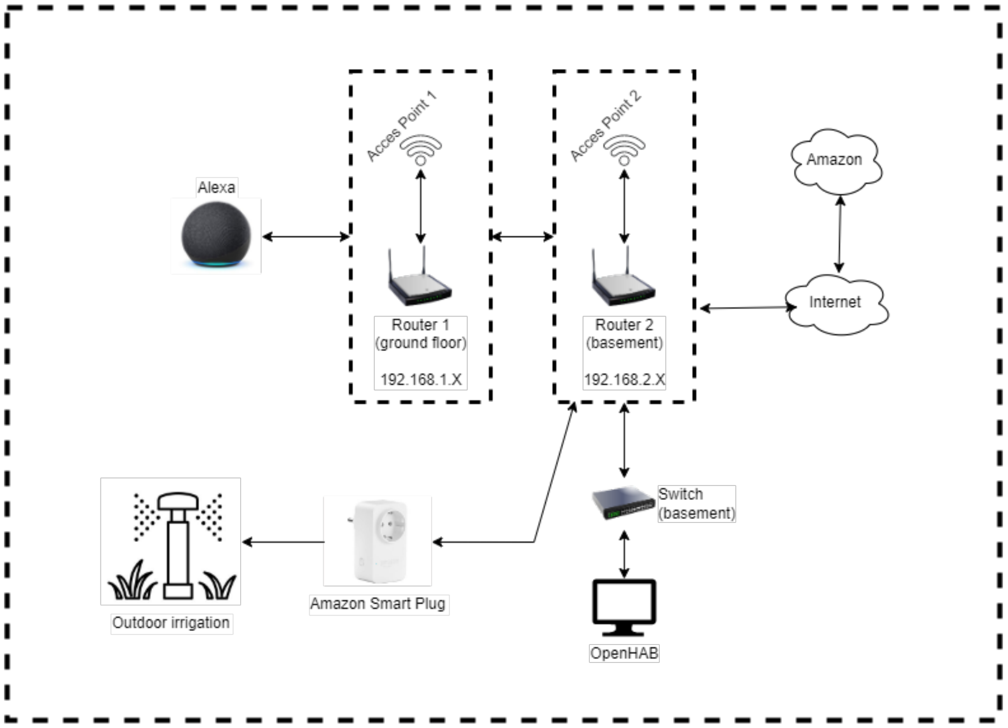


Figure 4.2 – *Design amazon smart plug in the house*

As can be seen in the image, the plug is connected to router 2 and communicates with Alexa via WiFi. Alexa receives the voice command to activate the plug, and it communicates with the amazon servers through the routers, Amazon responds with the answer and the action to be taken. As this plug works through Alexa, OpenHAB receives the information through the Amazon binding of the system, as explained in the previous section 4.1.

4.2.2 Tp-Link Tapo P100

These plugs from the Tp-Link Tapo ecosystem have their own servers and app. However, they can be integrated with Alexa through Amazon skills (additional functions), so that the plugs can be operated with it.

These plugs will control electrical appliances, such as washing machines and dishwashers.

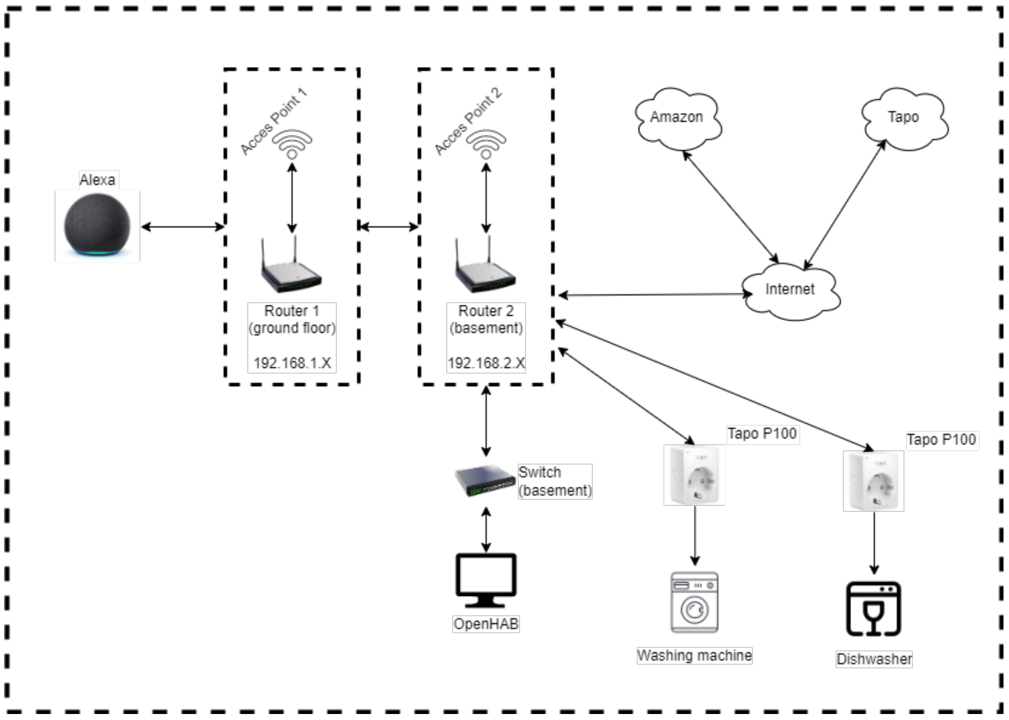


Figure 4.3 – *Design plugs Tapo in the house*

As they are integrable with Alexa, we can see in the diagram, these Tapo plugs are connected to the router 2 through which they communicate with the Tapo servers and this in turn with Amazon, so that Alexa can know the status of these plugs and can change their value.

4.3 Relays

The smart relays will be in charge of controlling the lights in a room of the house, which will be located behind the traditional switches already installed, following the steps of the manufacturer or a video on Youtube.

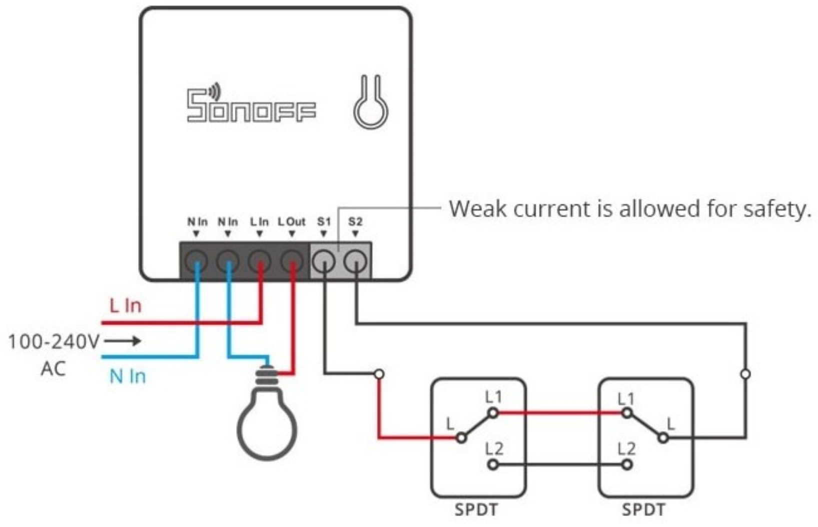


Figure 4.4 – *Sonoff installation*

These relays have a different communication protocol called Zigbee, you can see the differences with WiFi in the comparison table 3.4 between WiFi vs Zigbee. Therefore, these Zigbee relays have to be connected to a Zigbee hub that controls them.

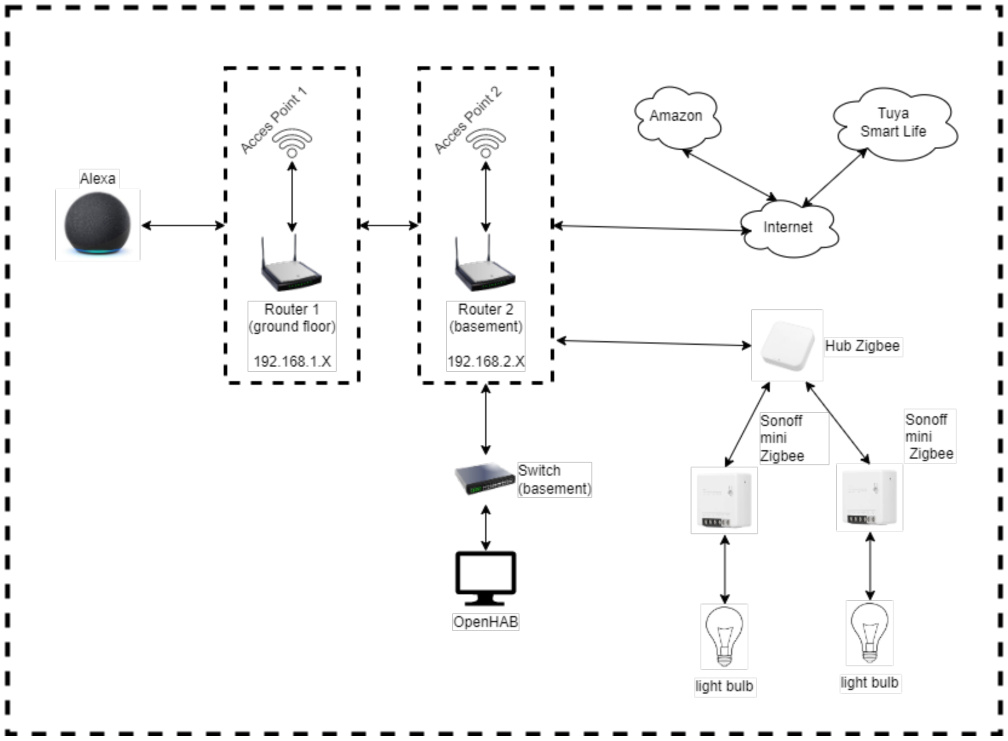


Figure 4.5 – *Design sonoff relays in the house*

In the image you can see Alexa again as these relays are also compatible with this virtual assistant. These relays are connected via Zigbee to the hub and this hub is connected to the router 2 through which it communicates with the Smart Life servers. In addition, if the Sonoff Mini is activated by means of the traditional switch, as we have said that it goes behind these, as they are connected to the hub, if they change state, it sends the information to the servers.

4.4 Weather Station

The weather station is located on the roof of the GranaSAT laboratory building.

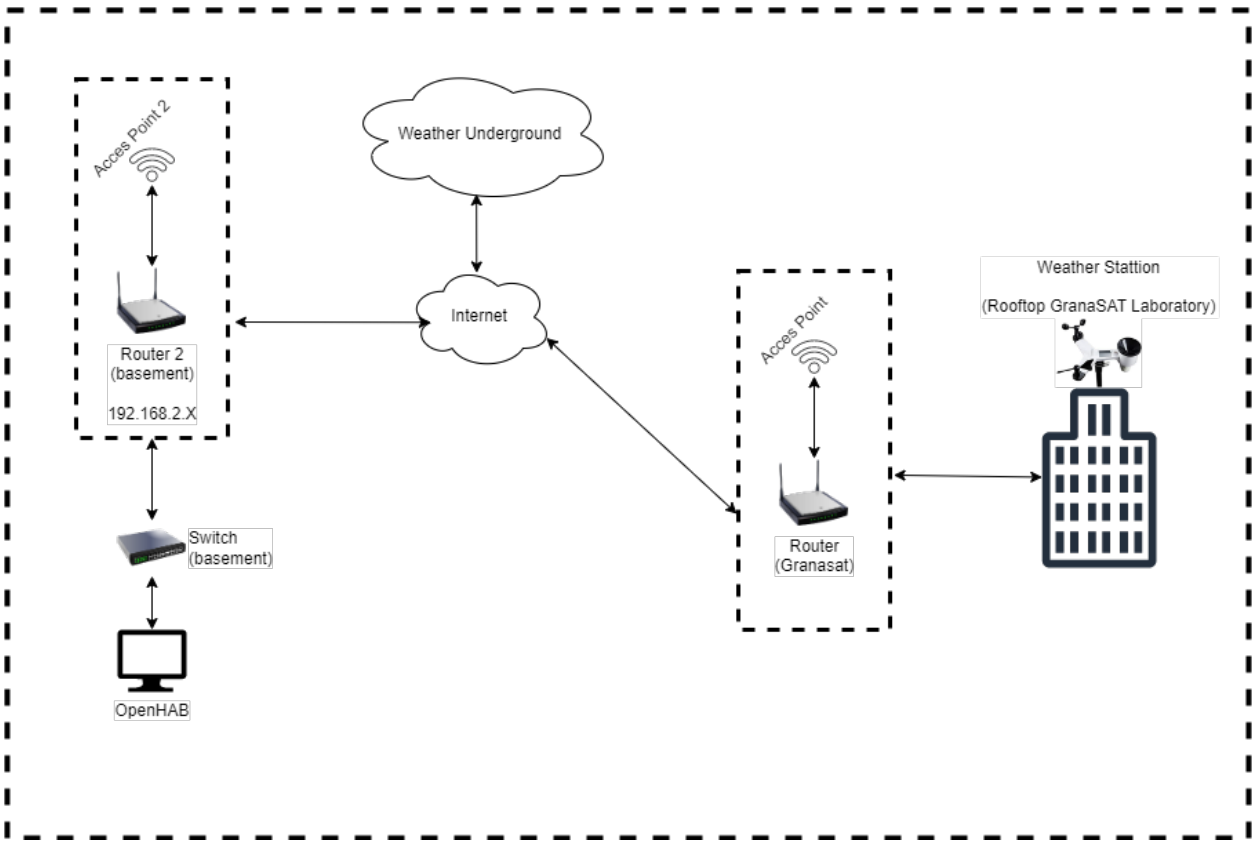


Figure 4.6 – *Weather station design*

In the image we can see that the station is connected to the GranaSAT router via WiFi. In order to obtain the data in the OpenHAB system it is necessary to publish the data in a web and make use of its API (WeatherUnderground), to obtain the data from the system with OpenHAB.

4.5 Weather API

Weather API is more or less the same as the previous section, but we only take the data from the Openweathermap API, weather forecast page, from some coordinates, to return the values closest to that location, and identified by the Openweathermap API key.

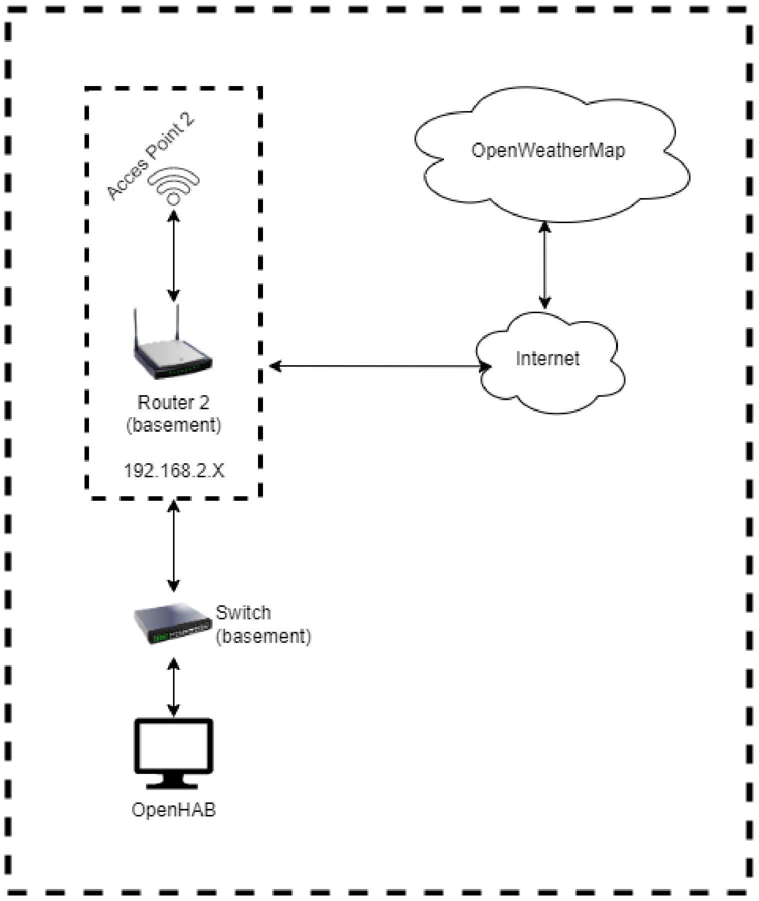


Figure 4.7 – *Weather API design*

OpenHAB will request the weather data through the specified binding, using the coordinates and API key of the Openweathermap account.

4.6 Solar Inverter

The solar inverter is in charge of converting the energy produced by the solar panels into alternating energy that can be used by the devices in the house. It is located in the basement where it is connected to the solar panels. SMA gives us a diagram of how it is works and its different connections.

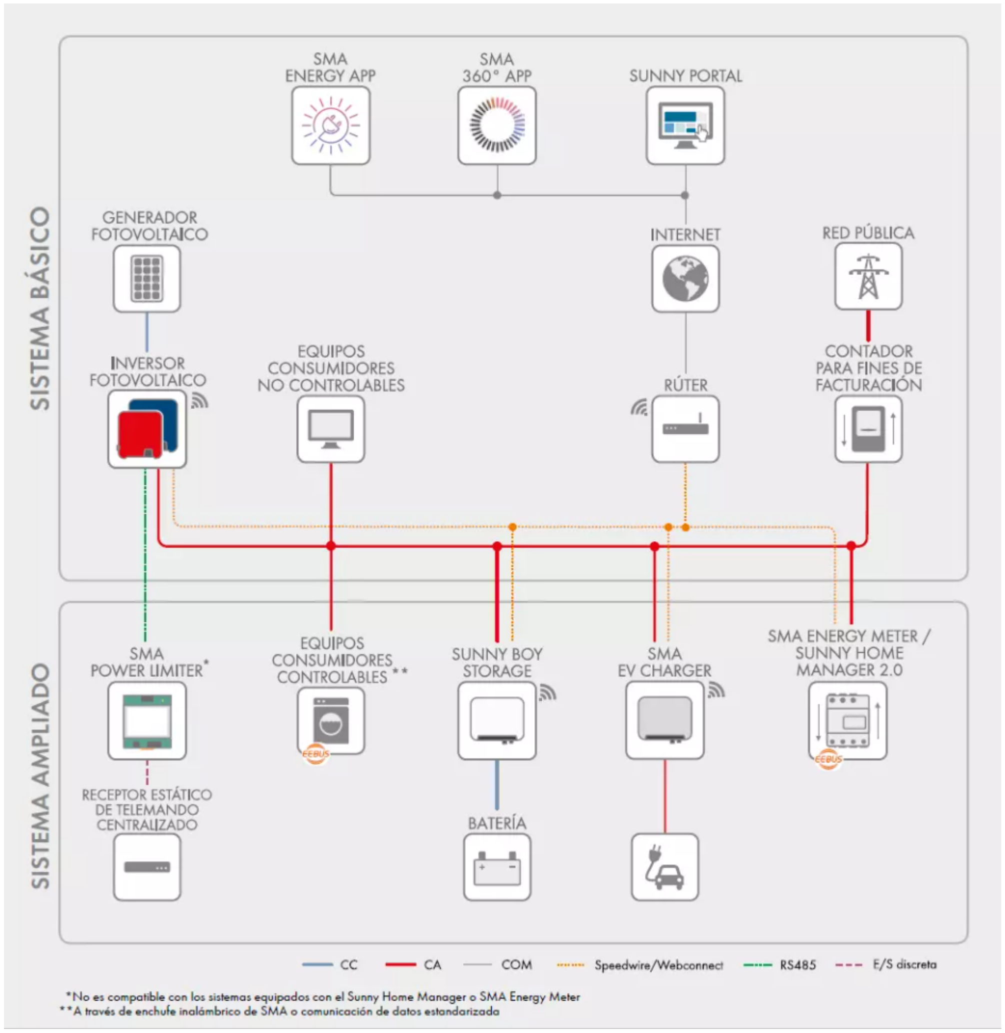

SISTEMA BÁSICO
SMA ENERGY APP
SMA 360° APP
SUNNY PORTAL
GENERADOR FOTOVOLTAICO
INTERNET
RED PÚBLICA
INVERSOR FOTOVOLTAICO
EQUIPOS CONSUMIDORES NO CONTROLABLES
RÚTER
CONTADOR PARA FINES DE FACTURACIÓN
SISTEMA AMPLIADO
SMA POWER LIMITER*
EQUIPOS CONSUMIDORES CONTROLABLES**
SUNNY BOY STORAGE
SMA EV CHARGER
SMA ENERGY METER / SUNNY HOME MANAGER 2.0
RECEPTOR ESTÁTICO DE TELEMANDO CENTRALIZADO
BATERÍA
CC
CA
COM
Speedwire/Webconnect
RS485
E/S discreta
*No es compatible con los sistemas equipados con el Sunny Home Manager o SMA Energy Meter
**A través de enchufe inalámbrico de SMA o comunicación de datos estandarizada

Figure 4.8 – *How the solar system works [45]*

As we can see it is divided into two parts, the **basic part** and the **optional part**. In the basic part we can see the electrical connection that must be in place, the electricity of the house works through the electricity generated by the solar panels, but it can also be supplied from the public grid, that is why we can see the electricity meter of the house to measure this consumption.

But the customer has in addition to the basic installation, the SMA Energy Meter, to control the injection of energy to the public grid, so the scheme would be like this.

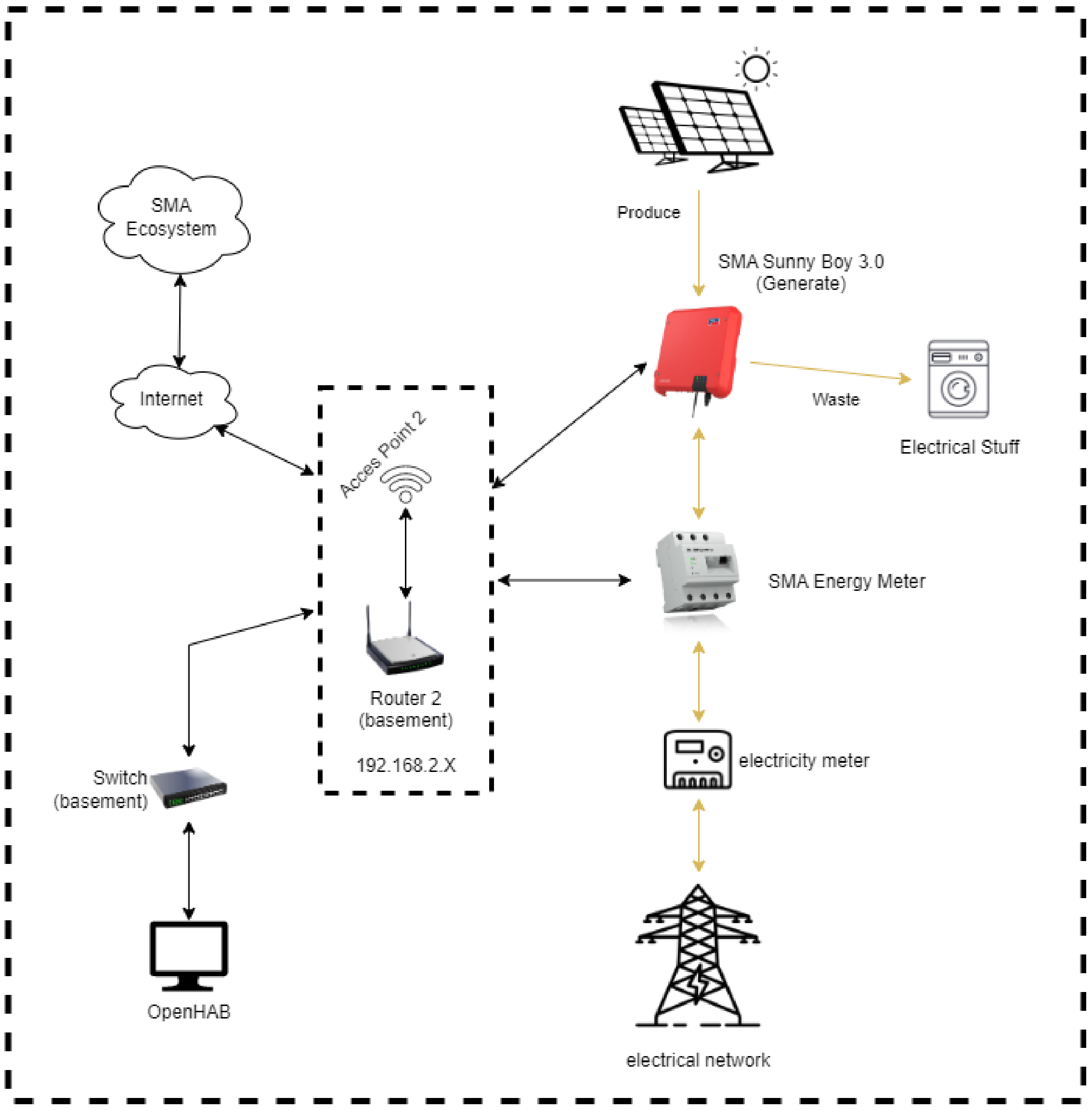


Figure 4.9 – *Diseño asistente en la casa*

As you can see, both the inverter and the energy meter are connected to router 2. SMA gives you access to various information portals where the data of the installed devices, inverter and energy meter are extracted. In these portals we can see the generation of the panels, the approximate solar radiation and the consumption of the house. We will get the data from these devices in our OpenHAB system through Python scripts.

Chapter 5

Implementation and configuration

In this chapter, we will explain how the implementation of the different devices or services in the OpenHAB system has been carried out. Many of these implementations, have been thanks to the Telegram group of OpenHAB España and Luis, a friend of the tutor Andres.

5.1 OpenHAB

In this project, OpenHAB will be the main element of the architecture, collecting and displaying, through a customized interface, the data obtained by the different scripts and interacting with the actuators of the devices.

Although the **learning curve** is balanced, it can be quite steep at the beginning, as the concepts are not very well established and you can get confused with the order in which you have to do things, but you can help yourself with the support of the OpenHAB community through the numerous guides and tutorials available.

Figure 5.1 – *Openhab logo*

OpenHAB is a very good bet because the **security and privacy of users' data** is under their control. In other words, users decide what information leaves the network. The solution can be used without remote access, so all the data collected by the different elements that make up the home automation system are stored securely on the local network.

5.1.1 Concepts

To begin with, the basic concepts of how OpenHAB works will be explained [5, 35]:

- **Things**: are the entities that can be added to the system, they can be as many **physical devices** or sensors, as they can also represent a **web service** or any other manageable source of information and functionality.

- **Bindings**: can be considered as **software adapters**, which makes the Things available to your home automation system, in other words it is the communication link between the Thing and your system.
- **Channels**: is where the things publish their **functionalities**, which are defined for each of them in the Binding. That is to say that for each Thing we have specific functionalities (channels) for that thing and that these functionalities are defined by the Binding. It is not necessary to define all the channels that have each Thing included in the Binding.
- **Items**: represent **capabilities that can be used by applications**, either in the user interfaces or in the automation logic. Items have a State and can receive commands.
- **Links**: are an **association** between exactly one channel and one item. If a channel is linked to an item, it is "enabled", which means that the capability represented by the item can be accessed through that channel. Channels can be linked to several items and items can be linked to several channels.
- **Rules**: are rules that are used for **process automation**. A rule can be activated in many ways, for example when an item changes state (plug goes from ON to OFF) or at a specific time expressed by a cron expression.
- **Persistence**: is to **save the information** of the different items of the OpenHAB system over time, so that it can be consulted later, for example in the form of graphical representations of this information. To save the information, a database is needed to store the different states of the items .
- **Sitemaps**: are a way to select and compose items in a user-oriented representation through **user interfaces (UI)**, including the OpenHAB application for Android. That is, it is a UI for the user to interact with the items, being able to modify the status of the items or view data that has been set, such as weather, humidity, etc.
- **Pages**: is the **new user interface** released in OpenHAB version 3, to interact with the items. It is not very explored yet but you can create custom widgets or use the default ones to represent the items .

A possible graphical representation of the different elements of OpenHAB would be:

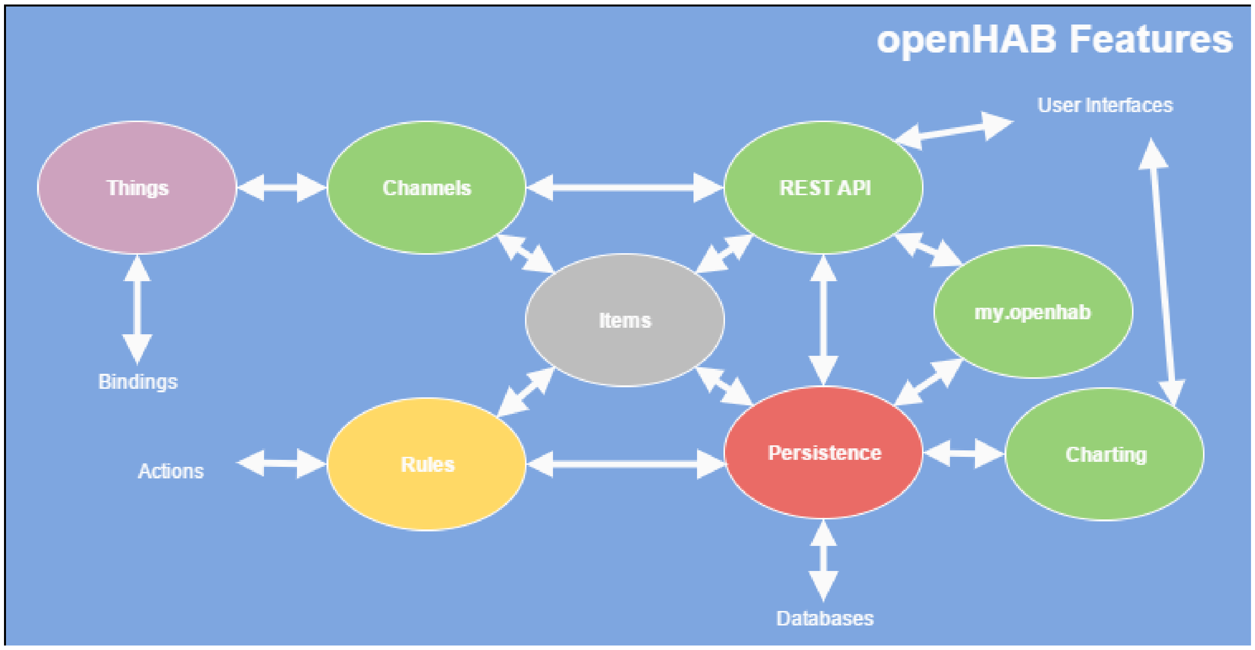


Figure 5.2 – *OpenHAB operation scheme* [9]

Through the OpenHAB REST API other programs can easily access most aspects of the OpenHAB system. This includes, for example, access to all data related to items, things and bindings, as well as the capabilities to invoke actions that can change the state of items or influence the behavior of other OpenHAB elements. Interactions with the REST API are based on the HTTP protocol. Internet access to the REST API is possible, but this represents a significant security risk. Users are encouraged to ensure secure connections.

In this project we make use of the OpenHAB REST API through OpenHAB Cloud as it allows remote access to local instances of OpenHAB without having to expose ports to the Internet or require a complex VPN configuration and thus guarantee a secure connection.

5.1.1.1 Things

As mentioned in the previous section 5.1.1 are the entities that can be added to the system, both physical and software services.

These things have an associated **status** to help identify any type of problem with the thing. The possible states are:

Status	Description
UNINITIALIZED	This is the **initial status** of a Thing when it is added or the framework is being started. This status is also assigned if the initializing process failed or the binding is not available. Commands sent to Channels will not be processed.
INITIALIZING	This status is assigned while the **binding initializes the Thing**. It depends on the binding how long the initializing process takes. Commands sent to Channels will not be processed.
UNKNOWN	The handler is **fully initialized** but due to the nature of the represented device/service it cannot really tell yet whether the Thing is ONLINE or OFFLINE. Therefore the Thing potentially might **be working correctly** already and may or may not process commands.
ONLINE	The device/service represented by a Thing is assumed to be **working correctly** and can **process commands**.
OFFLINE	The device/service represented by a Thing is assumed to be **not working** correctly and may not process commands. But the framework is allowed to send commands, because some radio-based devices may go back to ONLINE if a command is sent to them.
REMOVING	The device/service represented by a Thing **should be removed**, but the binding has not confirmed the deletion yet. Some bindings need to communicate with the device to unpair it from the system.
REMOVED	This status indicates that the device/service represented by a Thing **was removed** from the external system after the REMOVING was initiated by the framework.

Table 5.1 – *States of things [29]*

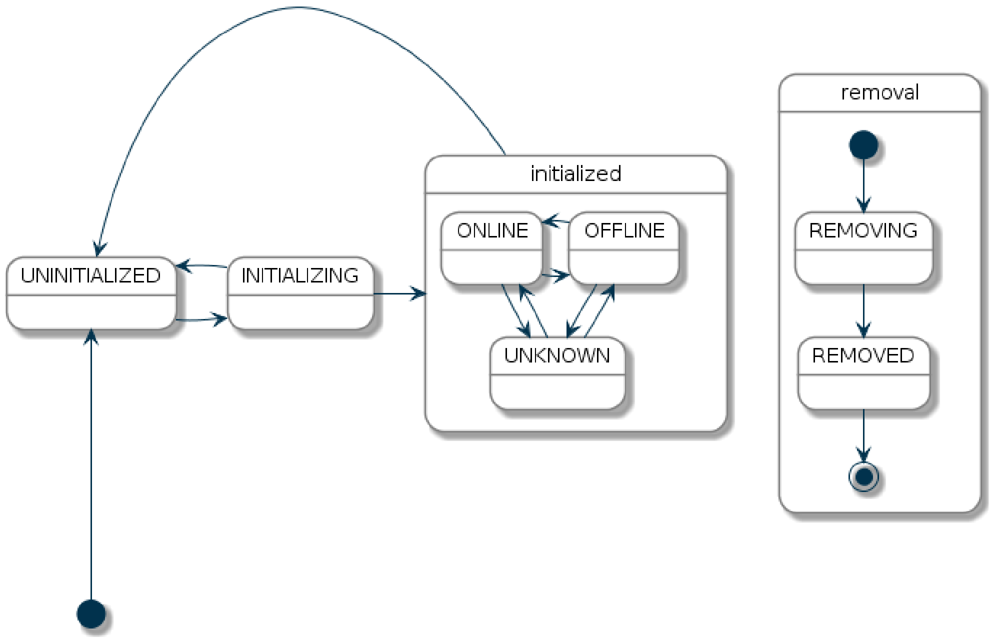


Figure 5.3 – *State diagram of the things [38]*

5.1.1.2 Items

The items have a **type** that facilitates its understanding to the programmer and to know how they act: The items at the time of being defined its type is indicated according to the functionality it represents, so it helps the programmer to understand how it works. The types that there are at the moment in OpenHAB are:

Types	Description	Command types
Color	Color information (RGB)	On/Off, Increase/Decrease, Percent
Contact	Item storing status of e.g. door/window contacts.	Open/Closed
DateTime	Stores date and time	-
Dimmer	Item carrying a percentage value for dimmers	On/Off, Increase/Decrease, Percent
Group	Item to nest other Items / collect them in Groups. They are used in the semantic model	-
Image	Holds the binary data of an image	-
Location	Stores GPS coordinates	Point
Number	Stores values in number format, takes an optional dimension suffix	Decimal
Number:<Measurement>	like Number, additional dimension information for unit support e.g. Frequency or energy	Quantity
Player	Allows to control players (e.g. audio players)	Play/Pause, Next/Previous, Rewind/Fastforward
Rollershutter	Typically used for blinds	Up/Down, Stop/Move, Percent
String	Stores texts	String
Switch	Typically used for lights (on/off)	On/Off

Table 5.2 – *Items types [33]*

5.1.2 Servidor

In this section we are going to explain the characteristics of the computer that will act as a server and will host the OpenHAB software and the changes that have been made to it.

- **CPU**: Intel(R) Pentium(R) Dual CPU T3400 @ 2.16 GHz.
- **RAM memory**: 4 GB.
- **GPU**: Mobile 4 Series Chipset Integrated Graphics Controller.
- **Non-volatile storage**: 1 TB hard disk.
- **Operating system**: Ubuntu 20.04.4 LTS 64 bits

Although it is not a very powerful computer and is a bit old, it will be able to support the use of OpenHAB.

5.1.2.1 Additional programs and changes

To **work remotely**, since the server was located at the client's house, we used a program called TeamViewer, which allows remote access to computers and mobile devices so that you can use them as if you were right there. [49].

Figure 5.4 – *TeamViewer logo*

This program uses an **ID and a password to access the computer**, this password changes every time the program is opened or a session is terminated, so a fixed password was configured, looking at this video.

In addition, the **computer's sleep and hibernation settings were changed** because otherwise the server could not be accessed via TeamViewer [51].

```
amroldan@openhab:~$ sudo systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
[sudo] contraseña para amroldan:
Created symlink /etc/systemd/system/sleep.target -> /dev/null.
Created symlink /etc/systemd/system/suspend.target -> /dev/null.
Created symlink /etc/systemd/system/hibernate.target -> /dev/null.
Created symlink /etc/systemd/system/hybrid-sleep.target -> /dev/null.
```

Listado 5.1 – *Config suspension*

5.1.3 Installation

OpenHAB supports different installation media and has a different software for each one, to be as optimized as possible, for example the most common are directly as a program on the host, in a Docker container or on a Raspberry Pi.

One of the customer's requirements was that the system be installed in a Docker container inside an old computer to take advantage of its power for this project. Also to take advantage of its **portability** and to be **isolated from the host**.

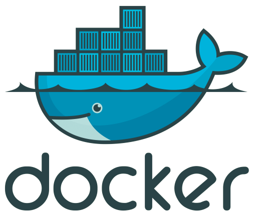


Figure 5.5 – *Docker Logo*

Docker packages software into standardized units called **containers** that include everything needed for the software to run, including libraries, system tools, code and runtime.

Docker acts similarly to a **virtual machine**, although while the virtual machine virtualizes a computer's **hardware and software**, it becomes very difficult for computers with few resources. A container only **packages what is needed to run the required software** and runs it on the computer's hardware and software. Docker is installed on each server and provides **simple commands** that you can use to create, start or stop containers [7].

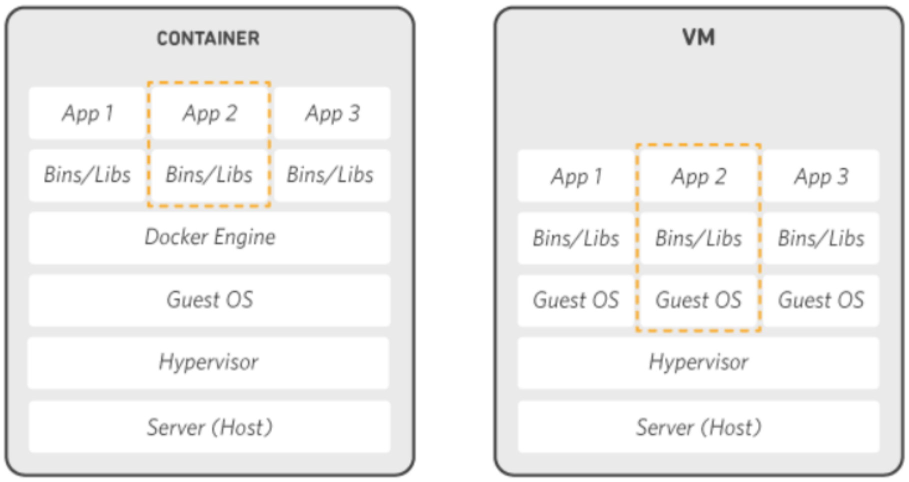


Figure 5.6 – *Docker vs Virtual Machine [7]*

Installing OpenHAB in a Docker container helps to easily **test different versions** of OpenHAB, easily **assign ports**, **isolate the software** from the host and the possibility of **portability** of the container to other computers. But all this flexibility comes at a cost, as OpenHAB runs in an isolated container with only what is necessary for its operation, it is very likely that the binding **Exec**, which is responsible for running custom scripts, created by us, will not work because they will not have access to the host programs, i.e. Python. But we will see how to solve and modify the Docker for its operation in the section 5.9.

5

5.1.3.1 Installing Docker and Docker Compose

For the installation of Docker follow this link:

1. Update your existing list of packages: **sudo apt update**.
2. Install some prerequisite packages that allow APT to use packages over HTTPS: **sudo apt install apt-transport-https ca-certificates curl software-properties-common**
3. Add the GPG key for the official Docker repository to your system: **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**
4. Add Docker repository to APT sources: **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"**
5. Install Docker: **sudo apt install docker-ce**
6. Docker should now be installed, the daemon started and the process enabled to start booting the system. Check that it is running: **sudo systemctl status docker**

And to install Docker Compose this link will help us to turn the container on and off more easily.

1. We confirm the most recent version available on your version page. The following command will download version 1.26.0 and save the executable file to /usr/local/bin/docker-compose, which will make this software globally accessible as docker-compose: **sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose**

2. We will set the correct permissions to make the docker-compose command executable: **sudo chmod +x /usr/local/bin/docker-compose**

3. Verify that the installation was performed correctly: **docker-compose –version**.

Once Docker is installed, we will follow the steps provided by OpenHAB from the Docker repository [13] to install OpenHAB in Docker

5.1.3.2 Create user and group openhab

The **group ID** will default to the same value as the **user ID**. By default, the OpenHAB user in the container runs with:

- **uid**=9001(openhab) **gid**=9001(openhab) **groups**=9001(openhab)

Therefore we have to make sure that the user and group we create has that ID:

- Create group: **groupadd -g 9001 openhab**
- Create the user with ID 9001 and add it to the openhab group: **useradd -u 9001 -g openhab -r -s /sbin/nologin openhab**
- Add our system user to the openhab group: **usermod -a -G openhab myownuser**

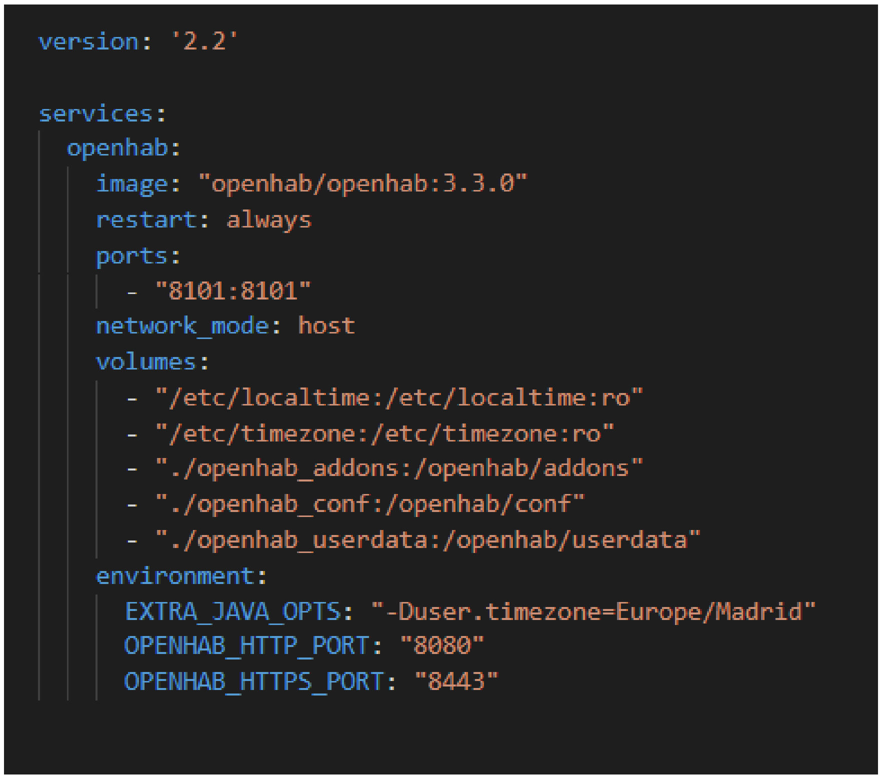
5.1.3.3 Create OpenHAB directories

These directories will be **mounted** on the running Docker container and is where the configurations and persistence data will be stored. Note that software running inside a Docker container cannot follow symbolic links located on a mounted volume. Ensure that the **user openhab** is the **owner of these directories**:

- Create the directories in the preferred path: **mkdir -p ./openhab/openhab_{conf,userdata,addons}**
- Change the owner of the files and directories:: **chown -R openhab:openhab ./openhab**

5.1.3.4 Create docker-compose.yml file

This file will make it easier to turn on and off the container that contains OpenHAB, this way the server can be turned on and off in an easier way.



```
version: '2.2'

services:
  openhab:
    image: "openhab/openhab:3.3.0"
    restart: always
    ports:
      - "8101:8101"
    network_mode: host
    volumes:
      - "/etc/localtime:/etc/localtime:ro"
      - "/etc/timezone:/etc/timezone:ro"
      - "./openhab_addons:/openhab/addons"
      - "./openhab_conf:/openhab/conf"
      - "./openhab_userdata:/openhab/userdata"
    environment:
      EXTRA_JAVA_OPTS: "-Duser.timezone=Europe/Madrid"
      OPENHAB_HTTP_PORT: "8080"
      OPENHAB_HTTPS_PORT: "8443"
```

Figure 5.7 – *Docker-compose.yml file*

I will explain the contents of the file:

- **openhab**, name for the container.
- **image: openhab/openhab:<version>**, OpenHAB image to be downloaded into the container.
- **restart:always**, if the container fails or the system restarts, the container restarts.
- **network_mode: host**, OpenHAB requires the discovery of UPnP, so this parameter causes the Docker container to use the host's network stack.
- **ports**, linking ports with the host.
  - **8101:8101**, SSH port of the OpenHAB console.
- **volumes**, linking files or folders to the host.
  - **/etc/localtime:/etc/localtime:ro**, links the container time to the host time, read-only.
  - **/etc/timezone:/etc/timezone:ro**, links the container time zone to the host time zone, read only.
  - **./openhab_addons:/openhab/addons**, is only necessary if add-on (bindings) installation is not available through the user interface or the console.
  - **./openhab_conf:/openhab/conf**, location of the conf folder for OpenHAB configurations.
  - **./openhab_userdata:/openhab/userdata**, location for logs, cache, persistence, databases, etc.
- **enviroment**, environment variables for the container.

- **EXTRA_JAVA_OPTS: "-Duser.timezone=Europe/Madrid"**, add extra options to the container, in this case we indicate the time zone of the host.
- **OPENHAB_HTTP_PORT: "8080"**, default port for the HTTP protocol connection.
- **OPENHAB_HTTPS_PORT: "8443"**, default port for the HTTPS protocol connection.

Before its initialization, the most used commands for container control will be explained:

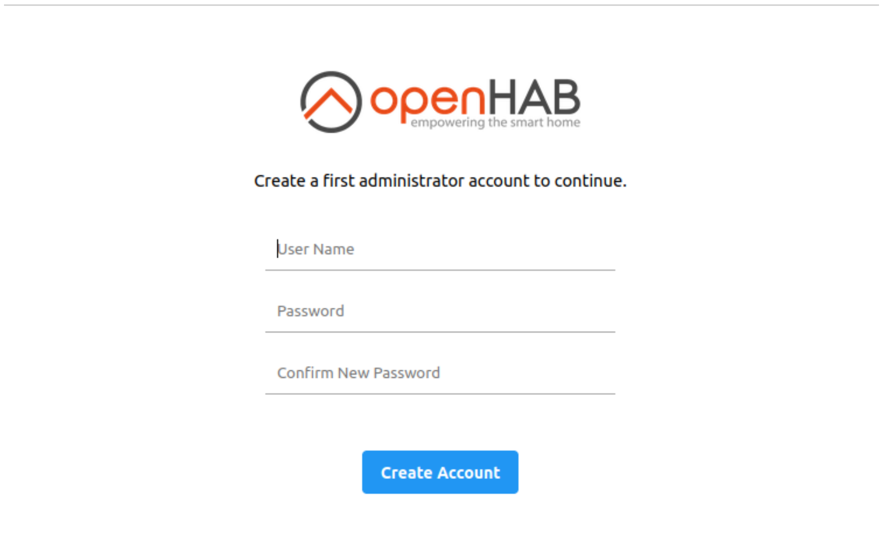
- **docker-compose up -d**, turn on the container in the background
- **docker-compose down**, turn off the container
- **docker ps**, list the containers, see their name or id
- **docker exec -it container_name or container_id /openhab/runtime/bin/client**, enter the OpenHAB console

To **initialize** the OpenHAB server we execute **docker-compose up -d** to activate the container.

5.1.3.5 OpenHAB Initialization

Once the container is initialized, we wait a couple of minutes for everything necessary to be created and we connect to `http://localhost:8080`, to start configuring OpenHAB.

When you log in, you will be prompted to create a user account and, by default, you will only be able to access the administration pages if you are logged in with an administrator account.


Create a first administrator account to continue.

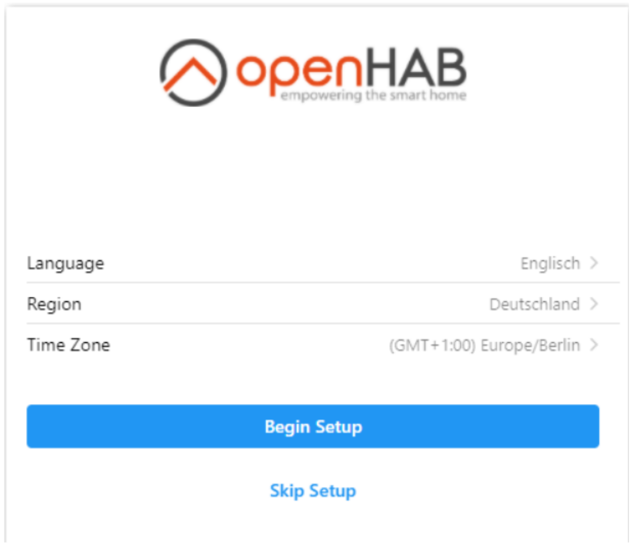
User Name

Password

Confirm New Password

Figure 5.8 – *Create admin user*

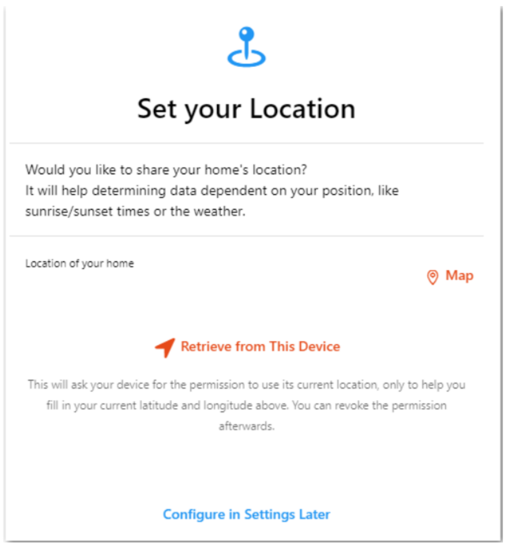
After creating an **administrator user**, you will be guided through an initial setup wizard, starting with setting the language, region and time zone. Once completed it will give us the ability to set our locale or complete it later.



openHAB
empowering the smart home

Language	Englisch
Region	Deutschland
Time Zone	(GMT+1:00) Europe/Berlin

(a) *Set language*



Set your Location

Would you like to share your home's location?
It will help determining data dependent on your position, like sunrise/sunset times or the weather.

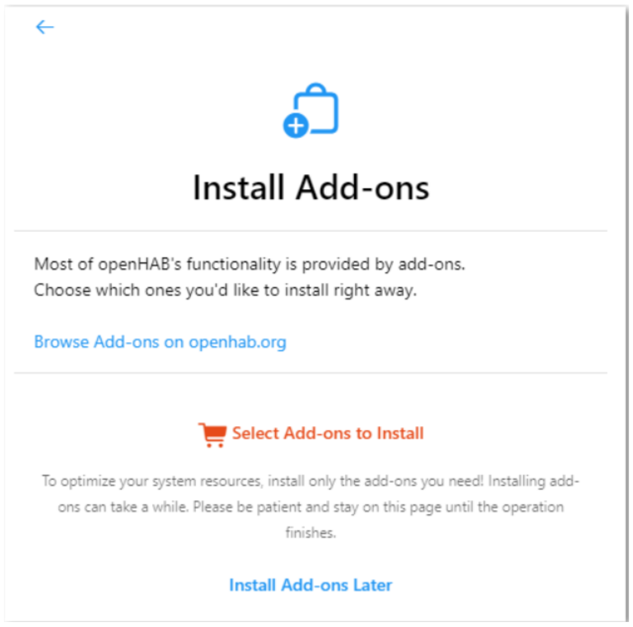
Location of your home

This will ask your device for the permission to use its current location, only to help you fill in your current latitude and longitude above. You can revoke the permission afterwards.

(b) *Set location*

Figure 5.9 – *Initial configuration*

Now we will be given the option to **install**OpenHAB bindings. This is done if we already know which specific bindings we have to install, this is used if you are upgrading a system from a previous version of OpenHAB. As we still do not know which bindings we are going to install we skip this step and will install the plugins later.



Install Add-ons

Most of openHAB's functionality is provided by add-ons.
Choose which ones you'd like to install right away.

To optimize your system resources, install only the add-ons you need! Installing add-ons can take a while. Please be patient and stay on this page until the operation finishes.

Figure 5.10 – *Intall Add-ons in initial configuration*

After finishing the initial configuration, you will be redirected to the **main control panel**, where you will **start configuring the selected devices**.

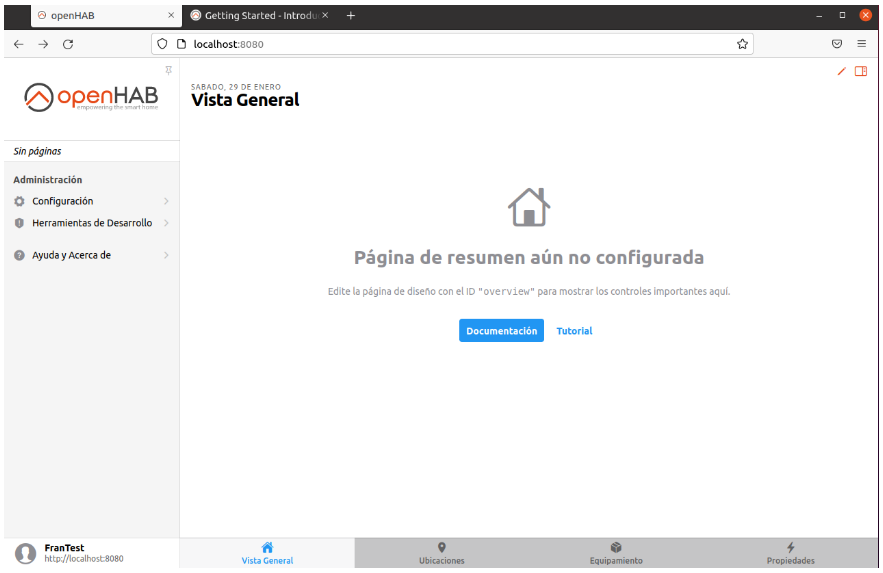
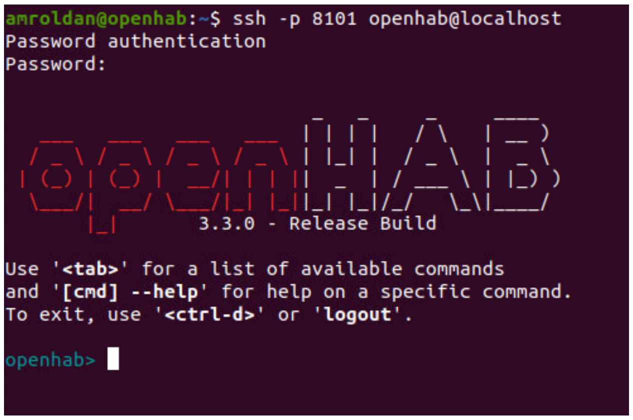


Figure 5.11 – *MainUI*

5.1.4 Console and logs

To access the OpenHAB console in Docker there are two ways:

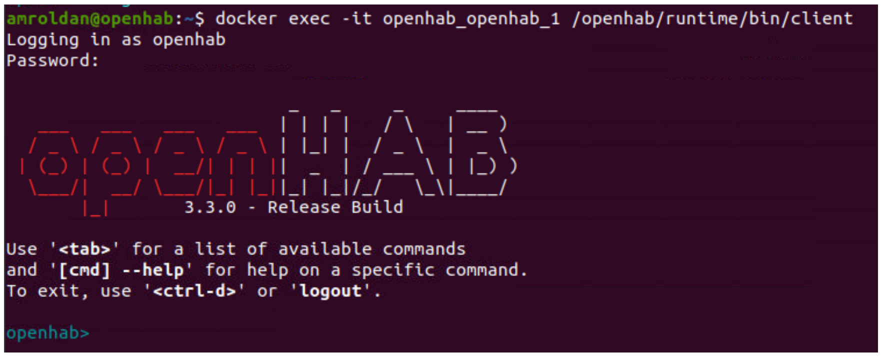
- Using SSH, with the Docker container started, we run from a terminal **ssh -p 8101 openhab@localhost**. By default, connections are only allowed from localhost, that is, only from the machine running OpenHAB. Connections are intentionally not allowed from remote hosts due to security concerns. If we have problems accessing locally we must open the corresponding port in Ubuntu with the following command **sudo ufw allow 8101/tcp**



```
amroldan@openhab:~$ ssh -p 8101 openhab@localhost
Password authentication
Password:

                          _   _     _     ____
  ___   ___   ___   ___  | | | |   / \   | __ )
 / _ \ / _ \ / _ \ / _ \ | |_| |  / _ \  |  _ \
| (_) | (_) |  __/| | | ||  _  | / ___ \ | |_) )
 \___/|  __/ \___/|_| |_||_| |_|/_/   \_\|____/
       |_|       3.3.0 - Release Build

Use '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
To exit, use '<ctrl-d>' or 'logout'.

openhab>
```

Figure 5.12 – *Console access via SSH*

- Running an instance of the console with Docker with the command seen in the previous section, **docker exec -it container_name or container_id /openhab/runtime/bin/client**.



```
amroldan@openhab:~$ docker exec -it openhab_openhab_1 /openhab/runtime/bin/client
Logging in as openhab
Password:

                          _   _     _     ____
  ___   ___   ___   ___  | | | |   / \   | __ )
 / _ \ / _ \ / _ \ / _ \ | |_| |  / _ \  |  _ \
| (_) | (_) |  __/| | | ||  _  | / ___ \ | |_) )
 \___/|  __/ \___/|_| |_||_| |_|/_/   \_\|____/
       |_|       3.3.0 - Release Build

Use '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
To exit, use '<ctrl-d>' or 'logout'.

openhab>
```

Figure 5.13 – *Console access via docker command*

The default user:password when logging into the console is **openhab:habopen**. To change the default password is with **sudo sed -i -e "s/openhab = .*,/openhab = newPassword,/g" /var/lib/openhab/etc/users.properties**, eplacing newPassword by the password that we want to put.

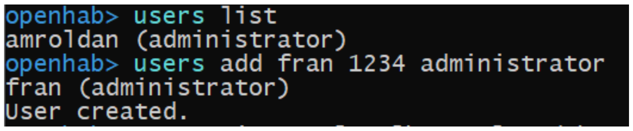
The OpenHAB console offers the possibility of [30]:

- **Monitor the log** in realtime, which are two files openhab.log and events.log. The **events.log** file shows the events that have happened in the system, such as updating item statuses, and the **openhab.log** file shows the faults that occur in the system (Commands).
- **Manage bundles**, not necessary in normal use, but may be useful for advanced users (More info).
- **Execute runtime commands** (Examples).

Another way of accessing the system logs can be done from the system text files in the path ***openhab/openhab/userdata/logs/***.

Currently, the only way to add additional users or manage existing ones (including changing passwords and assigning roles) is with the console. To create users we use the following commands:

- List system users with **users list**.
- Add user with **users add name password (role)**, the current roles are administrator and user, the role user cannot access the configuration menu when logged in. It can only access the pages created in MainUI to change the status of the devices.



```
openhab> users list
amroldan (administrator)
openhab> users add fran 1234 administrator
fran (administrator)
User created.
```

Figure 5.14 – *Add administrator to the system from the OpenHAB console*

5.1.5 UI vs text files

There are several ways to configure OpenHAB via text files, via graphical interface or via the OpenHAB console.

Configuration Task	Text files	Main UI	OpenHAB console
Auto-Discover Things and Items	✗	✓	✓
Define and manage Things	✓	✓	✓
Define and manage Groups and Items	✓	✓	✓
Define GUI	✓ (Sitemaps only)	✓ (Sitemaps and Pages)	✓
Define Transformations	✓	✗	✗
Define Persistence	✓	✗	✗
Define Rules	✓	✓	✗
Modify openHAB Settings/Services	✓	✓	✓
Install Add-ons	✓	✓	✓
Eleccion	✓	✓	✗

Table 5.3 – *Text files vs Mainui vs Console config [31]*

As we can see the OpenHAB console has almost the same options as the MainUI, but the console is much **less intuitive** than the MainUI and there is less information in the community forum.

OpenHAB recommends new users to start with a configuration through the **MainUI**, as it is much more intuitive for users who have never handled OpenHAB before, especially for the **Auto-Discover Things and add items** in a clearer way. In addition, OpenHAB 3 incorporates the **semantic model** that helps to group the items by groups, to have it more orderly and clear, and to be able to add items to the user interfaces in an easier way through the MainUI. Also, both configurations can be used, both text files and MainUI, we have made clear that MainUI is better, but when creating rules and scripts, defining persistence and transformations, text files win. For all this, my **recommendation** is to use the MainUI to install bindings, things and items and the text files to configure the persistence files (save data) and define rules and scripts, although you can also create things and items and add bindings.

5.2 OpenHAB Cloud

OpenHAB Cloud allows to **connect the local OpenHAB runtime to a remote OpenHAB Cloud**, for example as `myopenHAB.org`, which is an instance of the OpenHAB Cloud service hosted by OpenHAB Foundation. That is, we can **access our OpenHAB server remotely** without the need to be on the local network, so we can use the **OpenHAB mobile App** and as **MainUI from another computer** outside the network, such as the one at work [34].

5.2.1 Install binding of OpenHAB Cloud

As we have said in the section , we are going to install the bindings from the MainUI. For it from the MainUI we go to **Settings -> Bindings**.

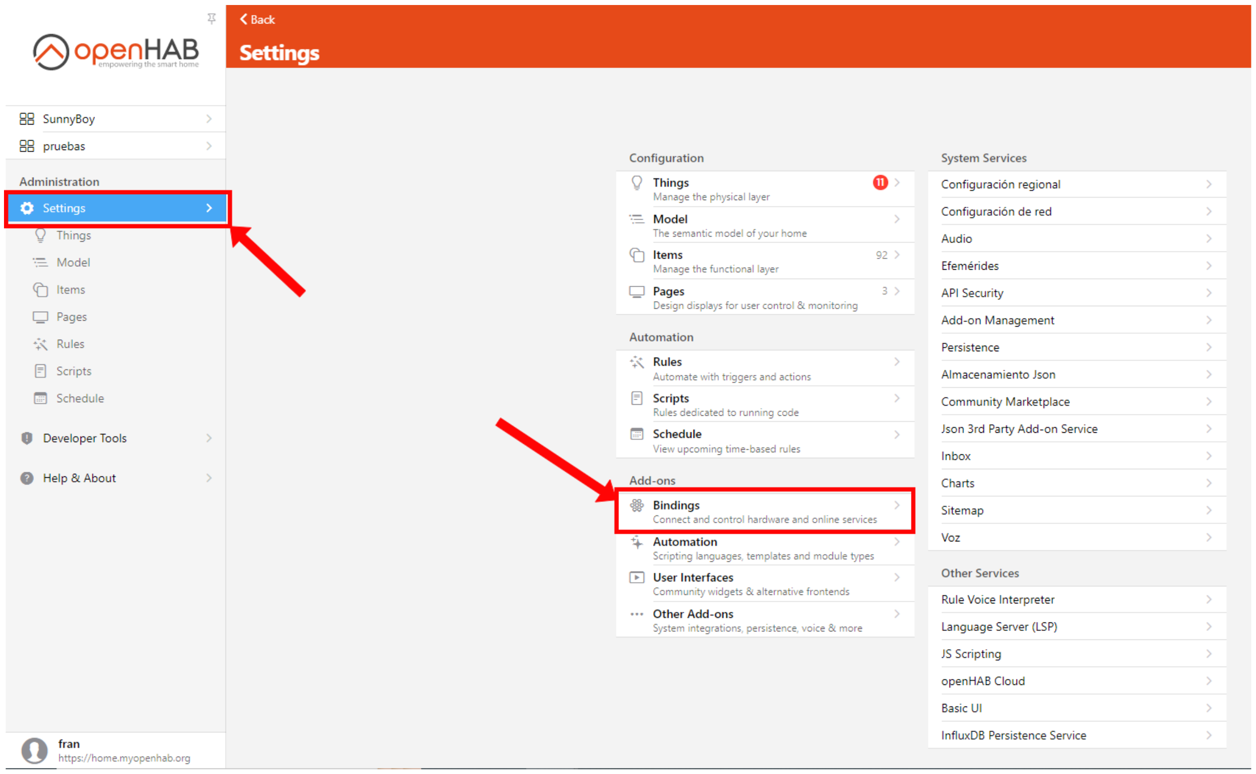
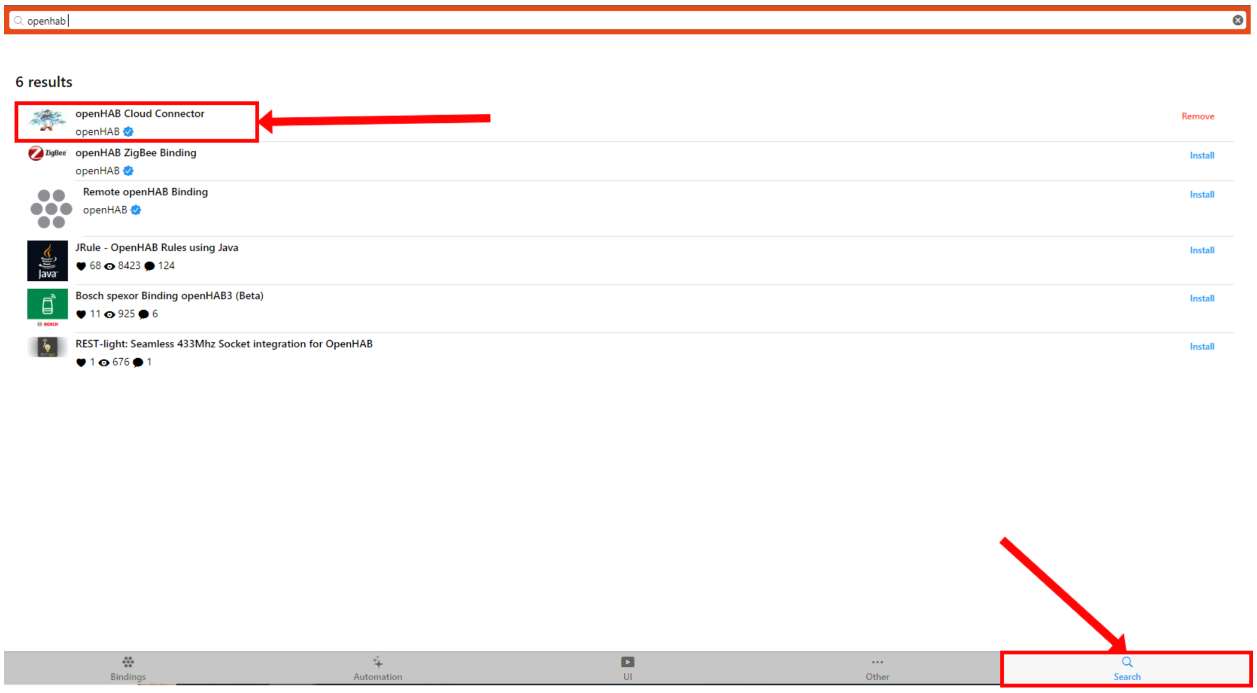

openHAB
empowering the smart home
SunnyBoy
pruebas
Administration
Settings
Things
Model
Items
Pages
Rules
Scripts
Schedule
Developer Tools
Help & About
fran
https://home.myopenhab.org
Back
Settings
Configuration
Things
Manage the physical layer
11
Model
The semantic model of your home
Items
Manage the functional layer
92
Pages
Design displays for user control & monitoring
3
Automation
Rules
Automate with triggers and actions
Scripts
Rules dedicated to running code
Schedule
View upcoming time-based rules
Add-ons
Bindings
Connect and control hardware and online services
Automation
Scripting languages, templates and module types
User Interfaces
Community widgets & alternative frontends
Other Add-ons
System integrations, persistence, voice & more
System Services
Configuración regional
Configuración de red
Audio
Efemérides
API Security
Add-on Management
Persistence
Almacenamiento Json
Community Marketplace
Json 3rd Party Add-on Service
Inbox
Charts
Sitemap
Voz
Other Services
Rule Voice Interpreter
Language Server (LSP)
JS Scripting
openHAB Cloud
Basic UI
InfluxDB Persistence Service

Figure 5.15 – *Access bindings tab*

Once here, go to the **Search tab**, search for **Openhab cloud connector** and **install the binding**.


openhab

6 results

openHAB Cloud Connector
openHAB
Remove

openHAB ZigBee Binding
openHAB
Install

Remote openHAB Binding
openHAB
Install

JRule - OpenHAB Rules using Java
68 8423 124
Install

Bosch spexor Binding openHAB3 (Beta)
11 925 6
Install

REST-light: Seamless 433Mhz Socket integration for OpenHAB
1 676 1
Install

Figure 5.16 – *Install binding*

5.2.2 Configure OpenHAB Cloud binding

Once installed, a section will be created in **Settings -> Other Services -> OpenHAB Cloud**, to configure the portal, but we will not change anything.

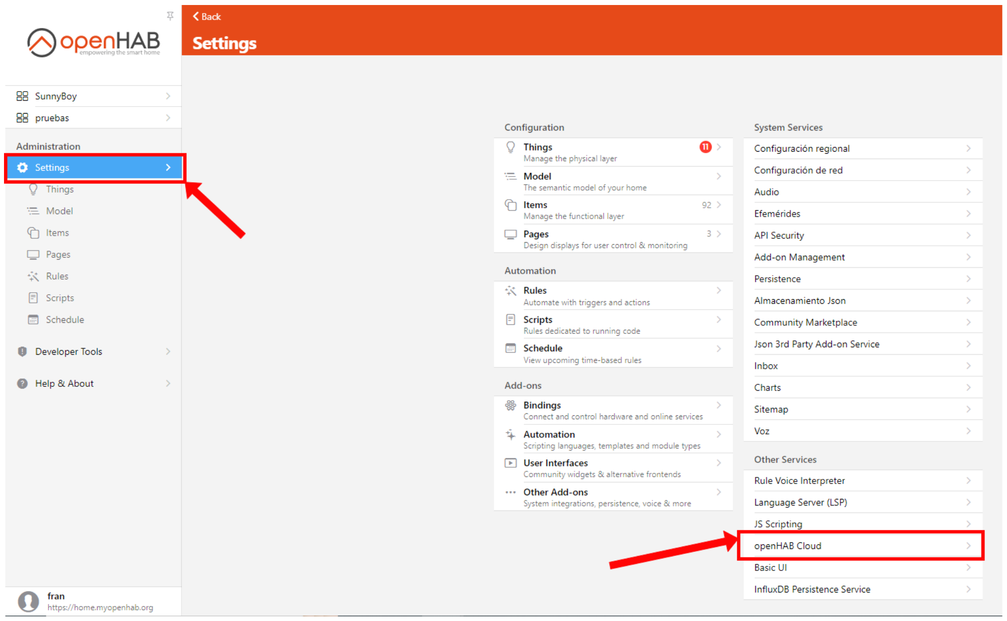
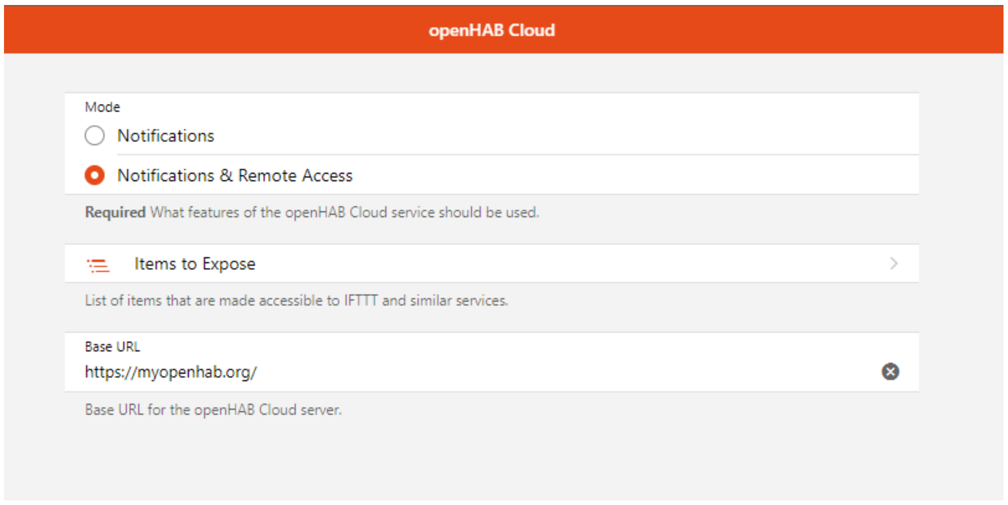


Figure 5.17 – *OpenhHAB cloud configuration localization in mainui*


openHAB Cloud

Mode
Notifications
[✓ Notifications & Remote Access]
Required What features of the openHAB Cloud service should be used.

Items to Expose
List of items that are made accessible to IFTTT and similar services.

Base URL
https://myopenhab.org/
Base URL for the openHAB Cloud server.

Figure 5.18 – *Config Openhab cloud via main ui*

It is not necessary to expose all items in the "Items to expose" tab, only those that IFTTT can access, as in our case there is no need to do so, we omit the step.

You can also configure the OpenHAB Cloud binding from the text files by modifying the **openhab_conf/services/openhab_cloud.cfg** file. But it already has the default values preassigned that we have seen in the MainUI.

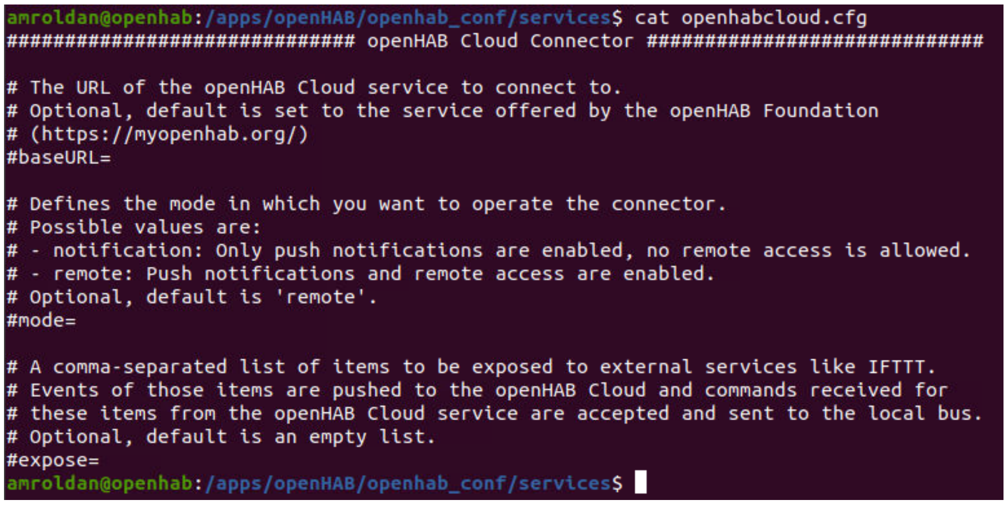
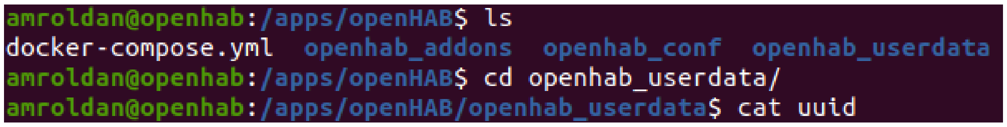

```
amroldan@openhab:/apps/openHAB/openhab_conf/services$ cat openhabcloud.cfg
############################## openHAB Cloud Connector #############################

# The URL of the openHAB Cloud service to connect to.
# Optional, default is set to the service offered by the openHAB Foundation
# (https://myopenhab.org/)
#baseURL=

# Defines the mode in which you want to operate the connector.
# Possible values are:
# - notification: Only push notifications are enabled, no remote access is allowed.
# - remote: Push notifications and remote access are enabled.
# Optional, default is 'remote'.
#mode=

# A comma-separated list of items to be exposed to external services like IFTTT.
# Events of those items are pushed to the openHAB Cloud and commands received for
# these items from the openHAB Cloud service are accepted and sent to the local bus.
# Optional, default is an empty list.
#expose=
amroldan@openhab:/apps/openHAB/openhab_conf/services$
```

Figure 5.19 – *Config Openhab cloud via text files*

5.2.3 Create OpenHAB Cloud account

The next step is to **create the account** in https://myopenhab.org/, but first we must obtain **two flags** from text files so that when creating it OpenHAB Cloud knows which is our server, for security reasons I will not show the number:

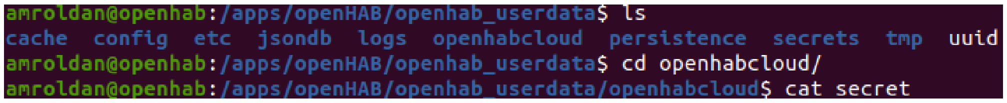
- **UUID**, can be found in **openhab_userdata/uuid**



```
amroldan@openhab:/apps/openHAB$ ls
docker-compose.yml  openhab_addons  openhab_conf  openhab_userdata
amroldan@openhab:/apps/openHAB$ cd openhab_userdata/
amroldan@openhab:/apps/openHAB/openhab_userdata$ cat uuid
```

Figure 5.20 – *UUID*

- **Secret**, can be found in **openhab_userdata/openhabcloud/secret**



```
amroldan@openhab:/apps/openHAB/openhab_userdata$ ls
cache  config  etc  jsondb  logs  openhabcloud  persistence  secrets  tmp  uuid
amroldan@openhab:/apps/openHAB/openhab_userdata$ cd openhabcloud/
amroldan@openhab:/apps/openHAB/openhab_userdata/openhabcloud$ cat secret
```

Figure 5.21 – *Secret*

Now that we have the flags, we go to the https://myopenhab.org/ page and create the account.

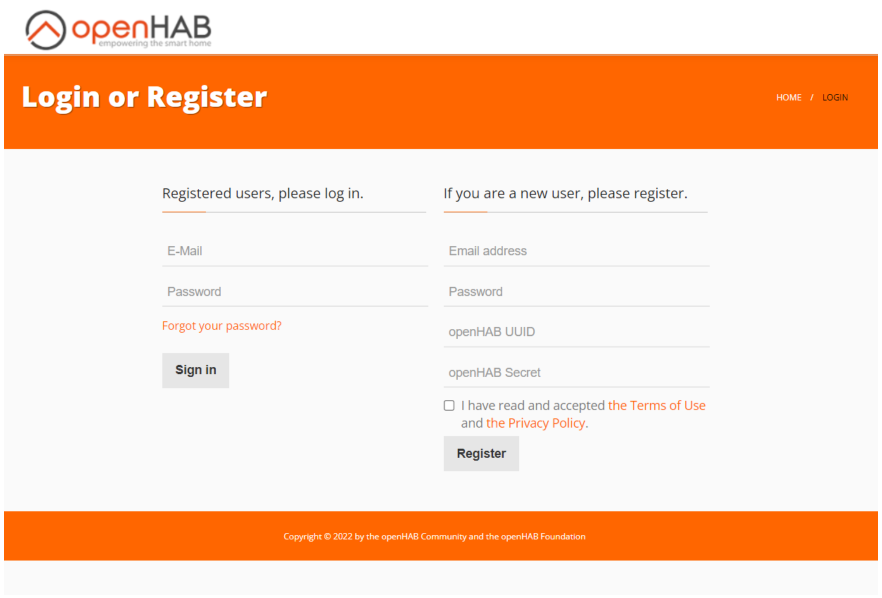
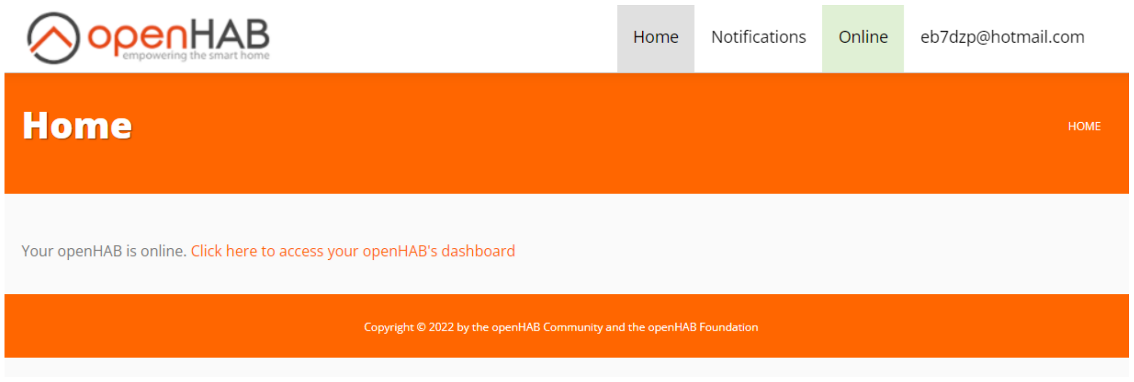

openHAB
empowering the smart home
Login or Register
HOME / LOGIN
Registered users, please log in.
E-Mail
Password
Forgot your password?
Sign in
If you are a new user, please register.
Email address
Password
openHAB UUID
openHAB Secret
I have read and accepted the Terms of Use and the Privacy Policy.
Register
Copyright © 2022 by the openHAB Community and the openHAB Foundation

Figure 5.22 – *Create account in openHAB Cloud*

5.2.4 Access to our OpenHAB server

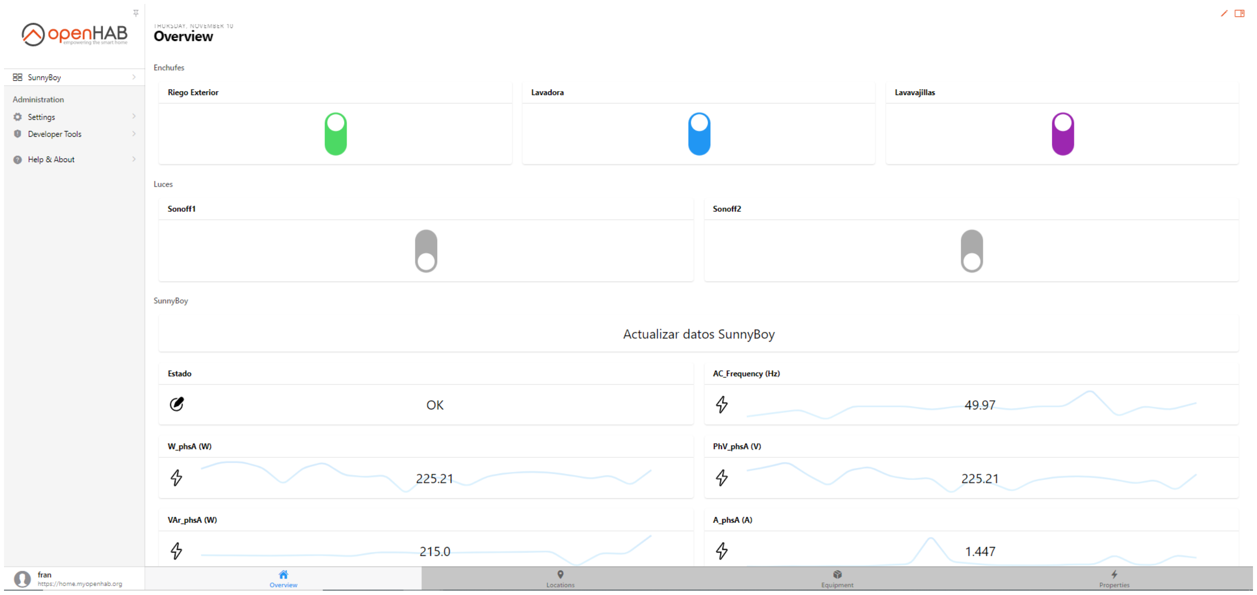
Once the account is created, we will be **redirected to a tab**, which will indicate whether our server is online or offline and a link to access the MainUI of our server if it is online, in order to control the different items that we have configured.



Your openHAB is online. Click here to access your openHAB's dashboard

Copyright © 2022 by the openHAB Community and the openHAB Foundation

Figure 5.23 – *Status tab*



Estado	AC_Frequency (Hz)
OK	49.97
W_phsA (W)	PhV_phsA (V)
225.21	225.21
VAr_phsA (W)	A_phsA (A)
215.0	1.447

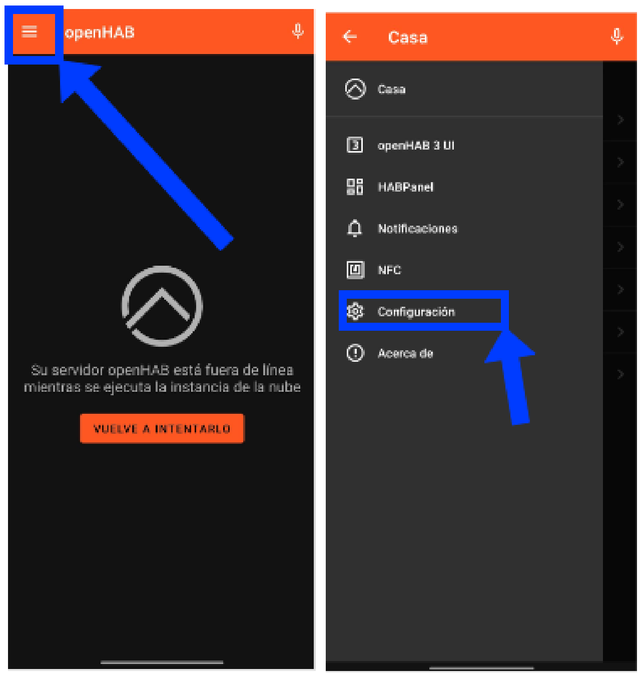
Figure 5.24 – *Inside server*

5.3 OpenHAB mobile app

OpenHAB has a mobile app for both Android and IOS. This app connects to our OpenHAB server through OpenHAB Cloud. With the app we can **modify the items** configured on our server through the different **user interfaces**, which we will see later.

To configure the app correctly, we must follow the steps below:

1. Have an **account** and configured in OpenHAB Cloud
2. Download the **OpenHAB app** from the Play Store or Apple Store, depending on the operating system of your mobile phone.
3. Once we have the app, we must **set up** the OpenHAB Cloud account created earlier.



Su servidor openHAB está fuera de línea mientras se ejecuta la instancia de la nube

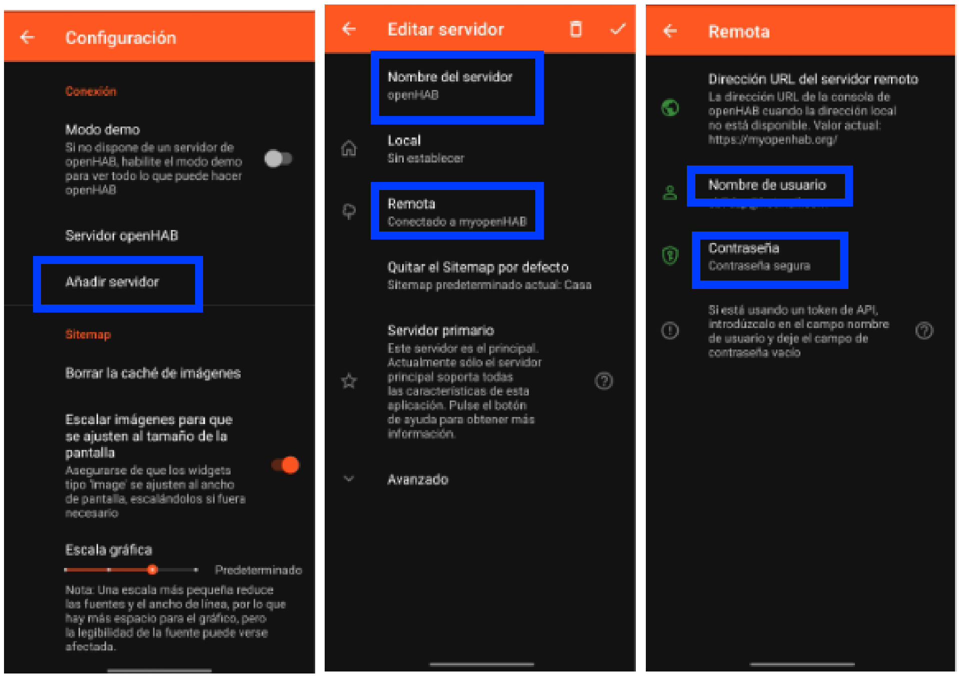
(a) *Init app* **(b)** *Config*

Figure 5.25 – *Access to configuration*

To configure it, we must go to the **three lines at the top left -> configuration -> connection -> add server**. We put the name we want and click on **Remote**, here we must put the following:

- **URL address**: https://myopenhab.org/
- **User** : email address of the OpenHAB Cloud account.
- **Password**: OpenHAB Cloud account password.

5


Configuración

Conexión

Modo demo
Si no dispone de un servidor de openHAB, habilite el modo demo para ver todo lo que puede hacer openHAB

Servidor openHAB

Añadir servidor

Sitemap

Borrar la caché de imágenes

Escalar imágenes para que se ajusten al tamaño de la pantalla
Asegurarse de que los widgets tipo 'image' se ajusten al ancho de pantalla, escalándolos si fuera necesario

Escala gráfica
Predeterminado
Nota: Una escala más pequeña reduce las fuentes y el ancho de línea, por lo que hay más espacio para el gráfico, pero la legibilidad de la fuente puede verse afectada.

Editar servidor

Nombre del servidor
openHAB

Local
Sin establecer

Remota
Conectado a myopenHAB

Quitar el Sitemap por defecto
Sitemap predeterminado actual: Casa

Servidor primario
Este servidor es el principal. Actualmente sólo el servidor principal soporta todas las características de esta aplicación. Pulse el botón de ayuda para obtener más información.

Avanzado

Remota

Dirección URL del servidor remoto
La dirección URL de la consola de openHAB cuando la dirección local no está disponible. Valor actual: https://myopenhab.org/

Nombre de usuario

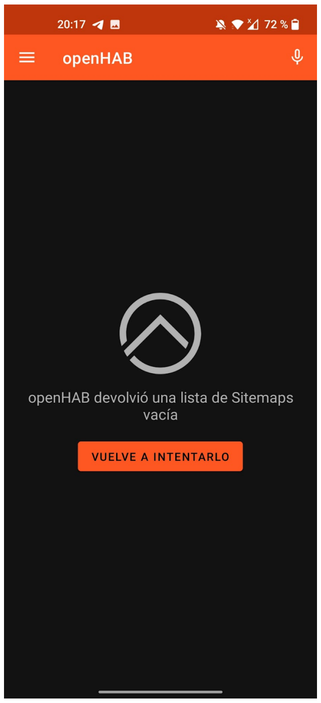
Contraseña
Contraseña segura

Si está usando un token de API, introdúzcalo en el campo nombre de usuario y deje el campo de contraseña vacío

(a) *Connection settings* **(b)** *Add server* **(c)** *Check values*

Figure 5.26 – *Server config*

Click on the **check at the top right** and you will be linked to our server.


20:17 72 %

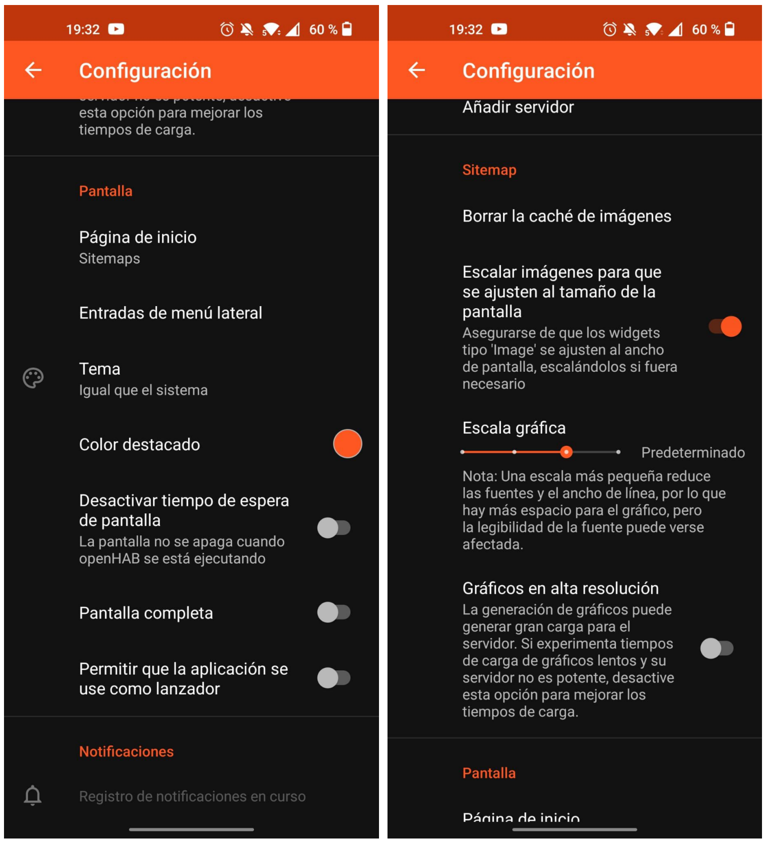
openHAB

openHAB devolvió una lista de Sitemaps vacía

VUELVE A INTENTARLO

Figure 5.27 – *Configured app*

In the **configuration** section in the app, there are quite a few **options** depending on what you want to do with the app. For example, you can change the scale of the graphics, choose what kind of interface you want to display by default, the colour of the app, send information from the device to the server, and so on.



19:32 60 %

Configuración

esta opción para mejorar los tiempos de carga.

Pantalla

Página de inicio
Sitemaps

Entradas de menú lateral

Tema
Igual que el sistema

Color destacado ●

Desactivar tiempo de espera de pantalla
La pantalla no se apaga cuando openHAB se está ejecutando

Pantalla completa

Permitir que la aplicación se use como lanzador

Notificaciones

Registro de notificaciones en curso

(a) *Display options*

19:32 60 %

Configuración

Añadir servidor

Sitemap

Borrar la caché de imágenes

Escalar imágenes para que se ajusten al tamaño de la pantalla
Asegurarse de que los widgets tipo 'Image' se ajusten al ancho de pantalla, escalándolos si fuera necesario

Escala gráfica
Predeterminado
Nota: Una escala más pequeña reduce las fuentes y el ancho de línea, por lo que hay más espacio para el gráfico, pero la legibilidad de la fuente puede verse afectada.

Gráficos en alta resolución
La generación de gráficos puede generar gran carga para el servidor. Si experimenta tiempos de carga de gráficos lentos y su servidor no es potente, desactive esta opción para mejorar los tiempos de carga.

Pantalla

Página de inicio

(b) *Sitemaps options*

Figure 5.28 – *Different options of the app*

5.4 Amazon Alexa

In this section, we will install and configure the **Amazon Alexa virtual assistant** on our OpenHAB server through the echo dot speaker:

5.4.1 Add and configure echo dot speaker in the Alexa App

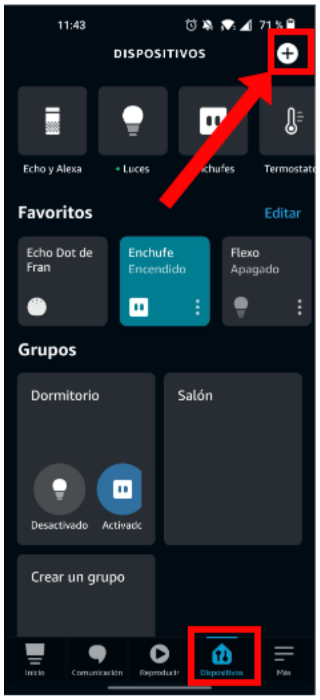
- Download the **Amazon Alexa application** from the Play Store or Apple Store, depending on the operating system of your mobile phone.

Figure 5.29 – *Amazon Alexa App*

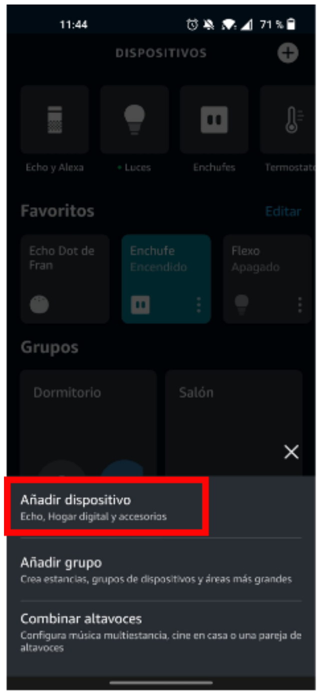
- To add the **echo dot**:

a. We open the Amazon Alexa application.

b. Go to the **devices tab** and in the **top right corner** click on the **+ icon** and **add device**.



(a)


11:44
DISPOSITIVOS
Añadir dispositivo
Echo, Hogar digital y accesorios
Añadir grupo
Crea estancias, grupos de dispositivos y áreas más grandes
Combinar altavoces
Configura música multiestancia, cine en casa o una pareja de altavoces

(b)

Figure 5.30 – *Add device*

c. We indicate the **type of device** we want to configure, in this case an Amazon Echo.

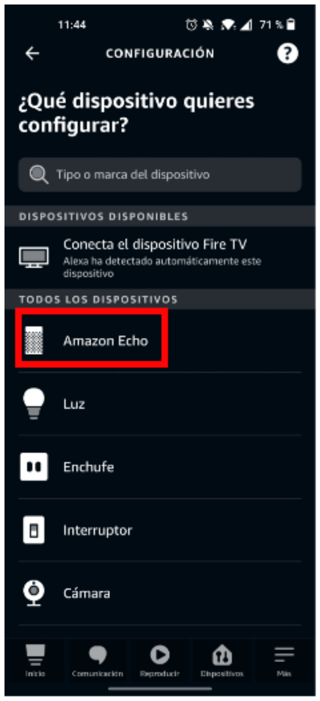
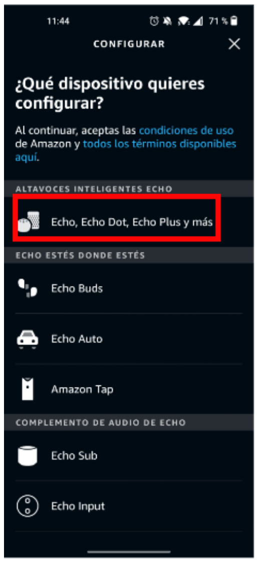

11:44
CONFIGURACIÓN
¿Qué dispositivo quieres configurar?
Tipo o marca del dispositivo
DISPOSITIVOS DISPONIBLES
Conecta el dispositivo Fire TV
Alexa ha detectado automáticamente este dispositivo
TODOS LOS DISPOSITIVOS
Amazon Echo
Luz
Enchufe
Interruptor
Cámara
Inicio
Comunicación
Reproducir
Dispositivos
Más

Figure 5.31 – *Select device type*

d. We click on **echo dot**.


11:44 71 %

CONFIGURAR

¿Qué dispositivo quieres configurar?

Al continuar, aceptas las condiciones de uso de Amazon y todos los términos disponibles aquí.

ALTAVOCES INTELIGENTES ECHO

Echo, Echo Dot, Echo Plus y más

ECHO ESTÉS DONDE ESTÉS

Echo Buds

Echo Auto

Amazon Tap

COMPLEMENTO DE AUDIO DE ECHO

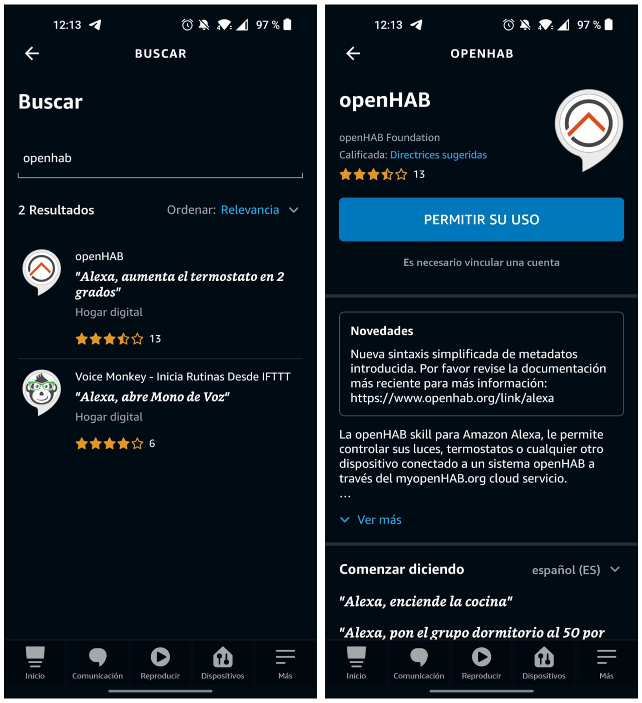
Echo Sub

Echo Input

Figure 5.32 – *Select device*

e. **Follow the steps indicated** by the application to complete the installation of the echo dot speaker.

Once the echo dot is added, we add the **OpenHAB skill**, which will allow us to control the items of the devices connected to our OpenHAB server, this is possible thanks to the OpenHAB Cloud service seen in the section 5.2 through myopenhab.org. For install the skill go to the **Amazon Alexa App -> More -> Skills and games -> search OpenHAB**. Once the skill has been added, you will have to **log in** with the **email and password** of the **OpenHAB Cloud** account previously created.


12:13 97 %

BUSCAR

Buscar

openhab

2 Resultados Ordenar: Relevancia

openHAB
"Alexa, aumenta el termostato en 2 grados"
Hogar digital
★★★⯪☆ 13

Voice Monkey - Inicia Rutinas Desde IFTTT
"Alexa, abre Mono de Voz"
Hogar digital
★★★★☆ 6

Inicio Comunicación Reproducir Dispositivos Más

12:13 97 %

OPENHAB

openHAB

openHAB Foundation
Calificada: Directrices sugeridas
★★★⯪☆ 13

PERMITIR SU USO

Es necesario vincular una cuenta

Novedades

Nueva sintaxis simplificada de metadatos introducida. Por favor revise la documentación más reciente para más información: https://www.openhab.org/link/alexa

La openHAB skill para Amazon Alexa, le permite controlar sus luces, termostatos o cualquier otro dispositivo conectado a un sistema openHAB a través del myopenHAB.org cloud servicio.
...

Ver más

Comenzar diciendo español (ES)

"Alexa, enciende la cocina"

"Alexa, pon el grupo dormitorio al 50 por

Inicio Comunicación Reproducir Dispositivos Más

(a) *Search skill* (b) *Install skill*

Figure 5.33 – *OpenHAB skill*

5.4.2 Install binding Amazon Echo Control

Once the echo dot speaker with alexa has been configured in the Amazon Alexa application, we go on to configure it in OpenHAB, to do this we will install the binding to be able to control it, called **Amazon Echo Dot Control**. As in previous sections we go to **Settings -> Bindings -> Search tab -> install binding Amazon Echo Control**.

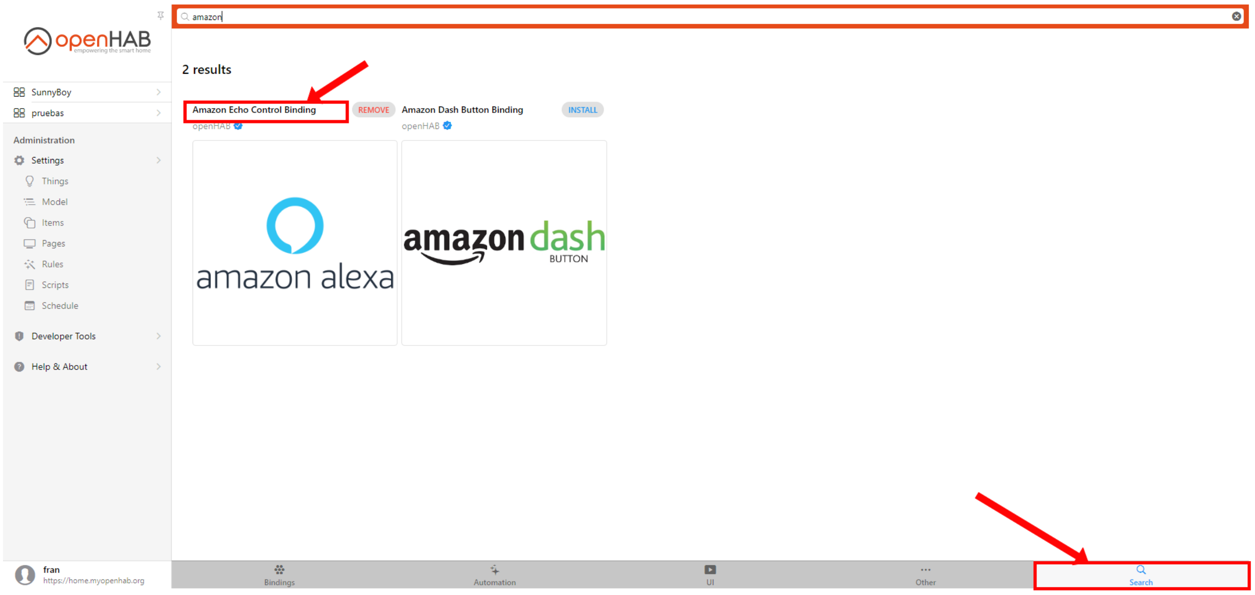
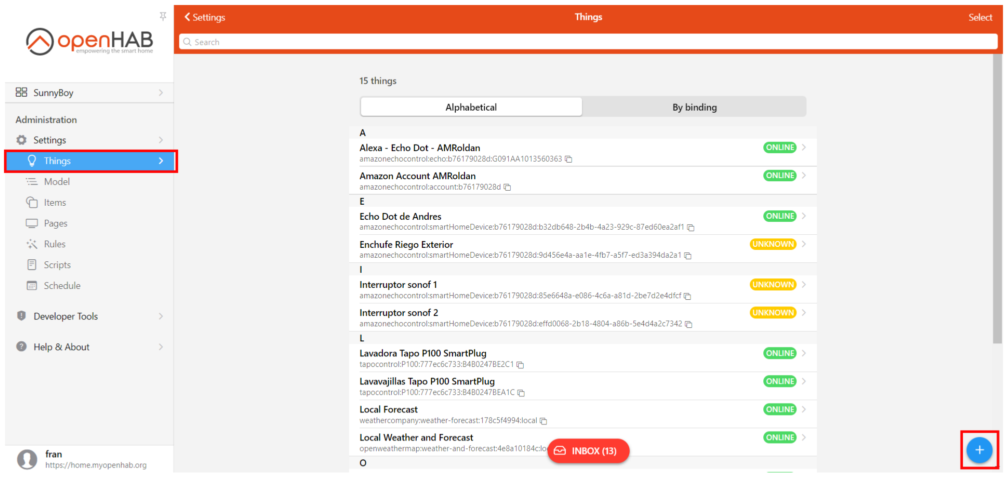


Figure 5.34 – *Add Amazon binding*

5.4.3 Create and configure things

Once the binding has been added, we are going to create the relevant things to control Amazon Alexa. To create the things, go to the **Settings -> Things tab**, then click on the **+ button** in the bottom right corner. Select the **Amazon Echo Control** binding and **install the necessary Things**.



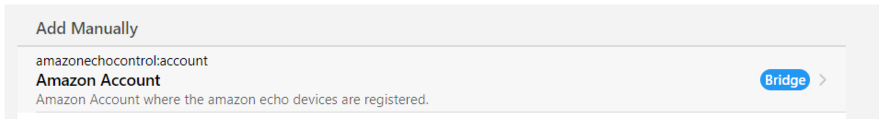
15 things

Alphabetical | By binding

A
Alexa - Echo Dot - AMRoldan — ONLINE
amazonechocontrol:echo:b76179028d:G091AA1013560363
Amazon Account AMRoldan — ONLINE
amazonechocontrol:account:b76179028d
E
Echo Dot de Andres — ONLINE
amazonechocontrol:smartHomeDevice:b76179028d:b32db648-2b4b-4a23-929c-87ed60ea2af1
Enchufe Riego Exterior — UNKNOWN
amazonechocontrol:smartHomeDevice:b76179028d:9d456e4a-aa1e-4fb7-a5f7-ed3a394da2a1
I
Interruptor sonof 1 — UNKNOWN
amazonechocontrol:smartHomeDevice:b76179028d:85e6648a-e086-4c6a-a81d-2be7d2e4dfcf
Interruptor sonof 2 — UNKNOWN
amazonechocontrol:smartHomeDevice:b76179028d:effd0068-2b18-4804-a86b-5e4d4a2c7342
L
Lavadora Tapo P100 SmartPlug — ONLINE
tapocontrol:P100:777ec6c733:B4B0247BE2C1
Lavavajillas Tapo P100 SmartPlug — ONLINE
tapocontrol:P100:777ec6c733:B4B0247BEA1C
Local Forecast — ONLINE
weathercompany:weather-forecast:178c5f4994:local
Local Weather and Forecast — ONLINE
openweathermap:weather-and-forecast:4e8a10184c:lo
O

Figure 5.35 – *Add things*

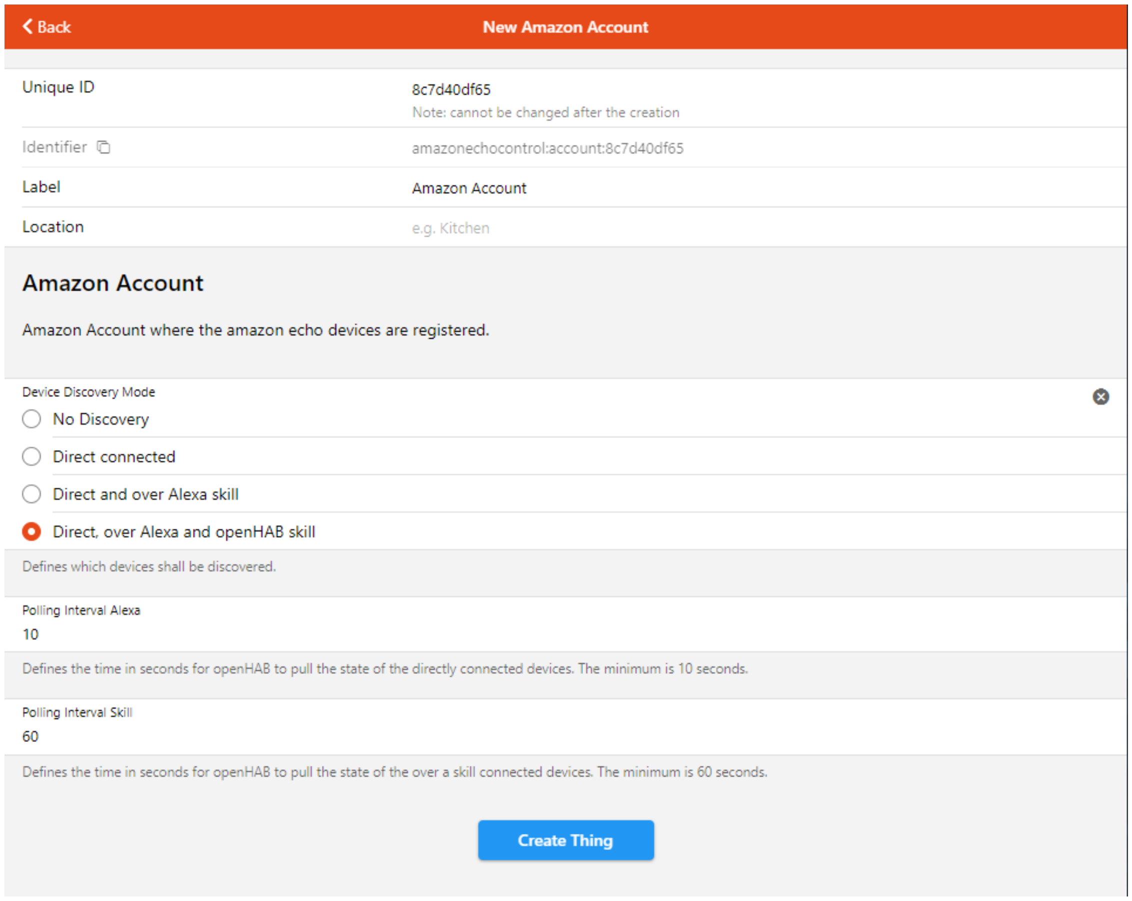
1. **Amazon account**, which acts as a bridge between Amazon and OpenHAB. In its configuration it will ask for a unique identifier, a label and parameters for how often Amazon will get the status of directly connected devices and vice versa, and how the devices will be discovered by OpenHAB and Amazon.



Add Manually

amazonechocontrol:account
Amazon Account
Amazon Account where the amazon echo devices are registered.

Figure 5.36 – *add thing amazon account*



Unique ID: 8c7d40df65
Note: cannot be changed after the creation
Identifier: amazonechocontrol:account:8c7d40df65
Label: Amazon Account
Location: e.g. Kitchen

Amazon Account

Amazon Account where the amazon echo devices are registered.

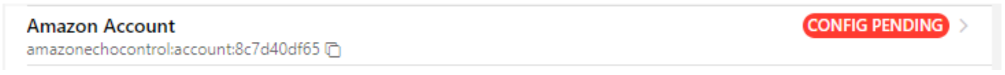
Device Discovery Mode
No Discovery
Direct connected
Direct and over Alexa skill
[✓ Direct, over Alexa and openHAB skill]
Defines which devices shall be discovered.

Polling Interval Alexa
10
Defines the time in seconds for openHAB to pull the state of the directly connected devices. The minimum is 10 seconds.

Polling Interval Skill
60
Defines the time in seconds for openHAB to pull the state of the over a skill connected devices. The minimum is 60 seconds.

Figure 5.37 – *Config amazon account*

2. Once created, a thing will appear, like this one, indicating that we **still have to configure the account**.

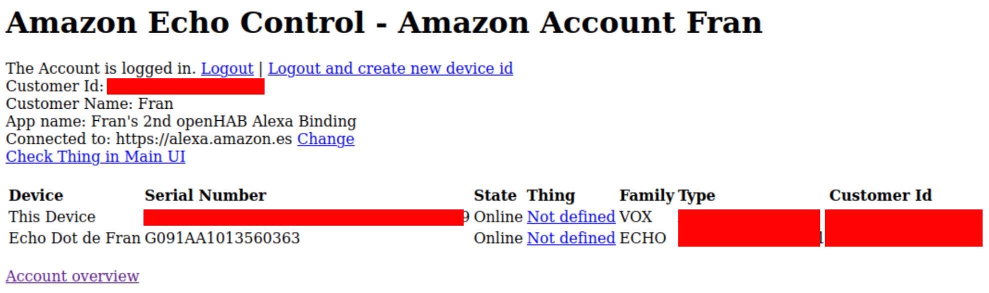


Amazon Account
amazonechocontrol:account:8c7d40df65

CONFIG PENDING

Figure 5.38 – *Amazon account thing*

We will have to access the **URL** http://localhost:8080/amazonechocontrol/ that indicates us from the local computer and it will ask us to **log in** with Amazon. Once logged in we will get something like this:



Amazon Echo Control - Amazon Account Fran

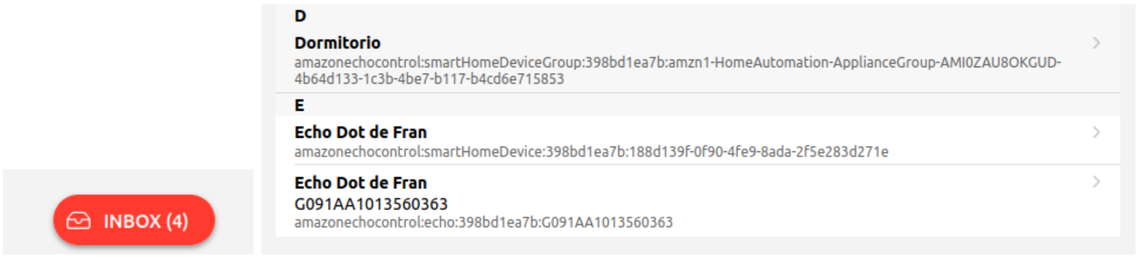
The Account is logged in. Logout | Logout and create new device id
Customer Id: [redacted]
Customer Name: Fran
App name: Fran's 2nd openHAB Alexa Binding
Connected to: https://alexa.amazon.es Change
Check Thing in Main UI

Device	Serial Number	State	Thing	Family	Type	Customer Id
This Device	[redacted]	Online	Not defined	VOX	[redacted]	[redacted]
Echo Dot de Fran	G091AA1013560363	Online	Not defined	ECHO	[redacted]	[redacted]

Account overview

Figure 5.39 – *Login Amazon Account in bridge*

3. Once the account is associated, the account thing will go **online** and we will see an **inbox button** at the bottom, which tells us that things connected to Alexa have been **autodiscovered**. In this case, the group where the echo dot has been added to the Alexa app and two echo dots have been discovered, one is the speaker with the functions to play songs and the other is the alarm function.



D
Dormitorio
amazonechocontrol:smartHomeDeviceGroup:398bd1ea7b:amzn1-HomeAutomation-ApplianceGroup-AMI0ZAU8OKGUD-4b64d133-1c3b-4be7-b117-b4cd6e715853
E
Echo Dot de Fran
amazonechocontrol:smartHomeDevice:398bd1ea7b:188d139f-0f90-4fe9-8ada-2f5e283d271e
Echo Dot de Fran
G091AA1013560363
amazonechocontrol:echo:398bd1ea7b:G091AA1013560363

(a) *inbox* **(b)** *Things discovered*

Figure 5.40 – *Configured account thing*

4. We select all the things we want to add to the system, in this case we only want the thing corresponding to the music player. To do so, click on it and a window will pop up, click on **"Add as a Thing"**, and it will be added to the system by itself. We wait a while and then it will go **online**.

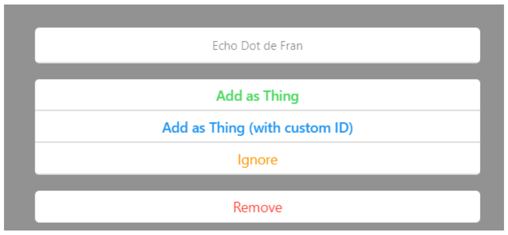
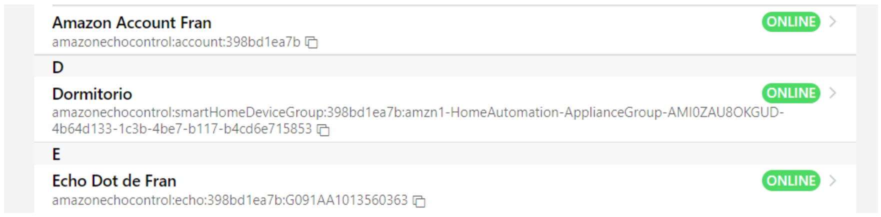


Figure 5.41 – *Add as Thing tab*



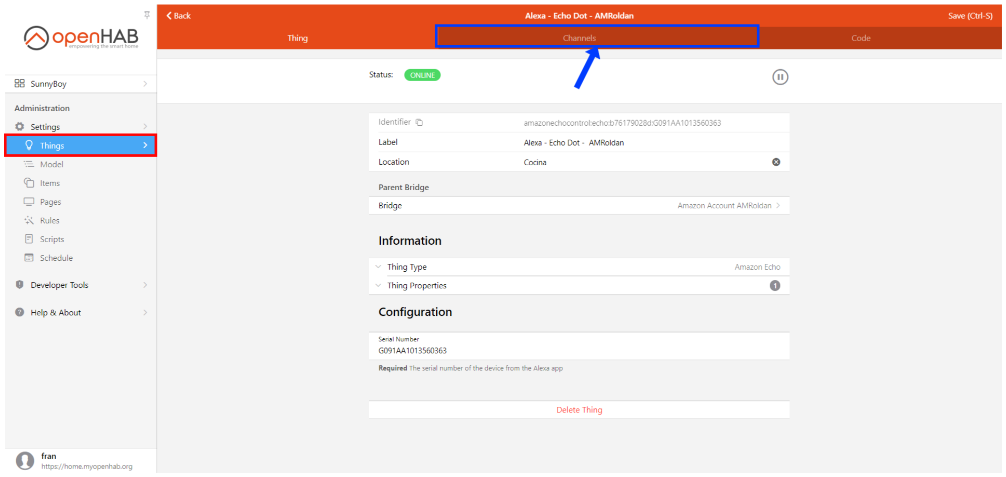
Amazon Account Fran ONLINE
amazonechocontrol:account:398bd1ea7b
D
Dormitorio ONLINE
amazonechocontrol:smartHomeDeviceGroup:398bd1ea7b:amzn1-HomeAutomation-ApplianceGroup-AMI0ZAU8OKGUD-4b64d133-1c3b-4be7-b117-b4cd6e715853
E
Echo Dot de Fran ONLINE
amazonechocontrol:echo:398bd1ea7b:G091AA1013560363

Figure 5.42 – *Things online*

5.4.4 Create and configure items

As we have explained before, the things cannot be managed, only the items that we create with the functions of the things. Therefore, once the things are configured, we will create the items to control their functions:

1. From the things tab, we **access the thing** of the device we want to add its functionalities. Once here we go to the **channels tab** to see all the functions that the thing can do.



Alexa - Echo Dot - AMRoldan

Status: ONLINE

Identifier	amazonechocontrol:echo:b76179028d:G091AA1013560363
Label	Alexa - Echo Dot - AMRoldan
Location	Cocina

Parent Bridge

Bridge: Amazon Account AMRoldan

Information

Thing Type: Amazon Echo

Thing Properties

Configuration

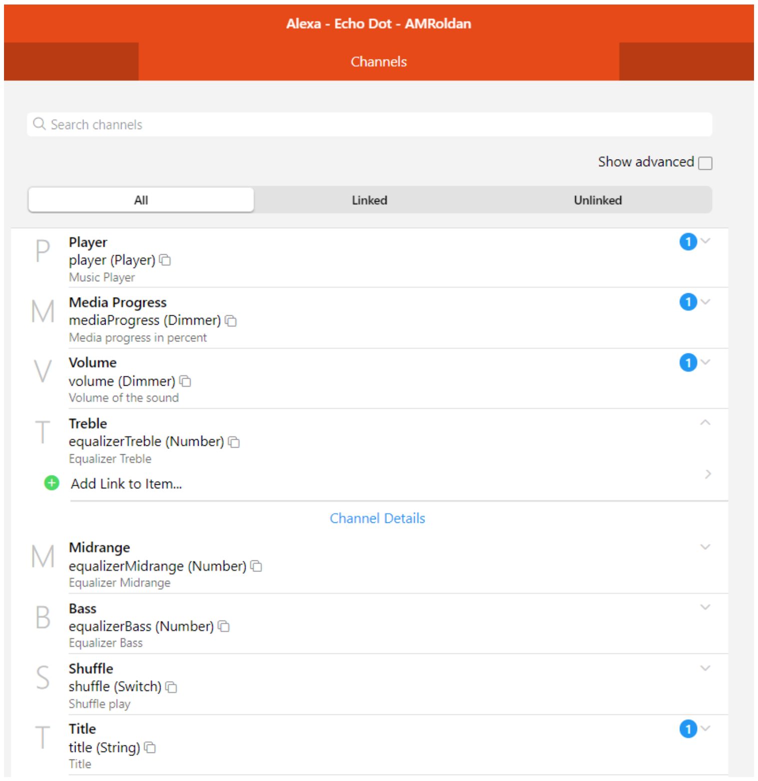
Serial Number
G091AA1013560363

Required The serial number of the device from the Alexa app

Delete Thing

Figure 5.43 – *Channels tab*

2. Once in the channels tab, click on the function you want to create and click on **"Link to item"**.



Alexa - Echo Dot - AMRoldan

Player
player (Player)
Music Player

Media Progress
mediaProgress (Dimmer)
Media progress in percent

Volume
volume (Dimmer)
Volume of the sound

Treble
equalizerTreble (Number)
Equalizer Treble

Midrange
equalizerMidrange (Number)
Equalizer Midrange

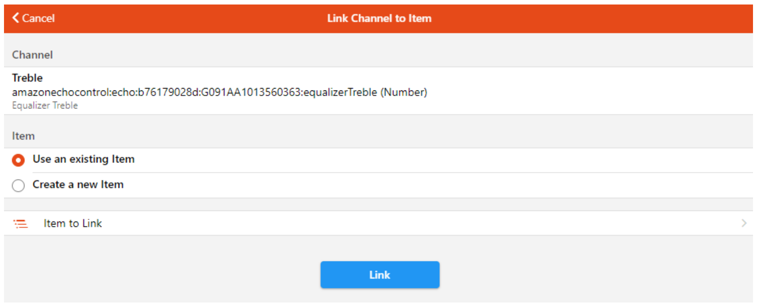
Bass
equalizerBass (Number)
Equalizer Bass

Shuffle
shuffle (Switch)
Shuffle play

Title
title (String)
Title

Figure 5.44 – *Link channel to item*

3. It will redirect us to another page that will give us the option to **link this channel** to an **existing item** or to **create an item** from scratch. In the item we have to put a unique id, a label, indicate the type of the variable if it is a number, a string, a switch, etc, we can also put an icon to the item. We can also indicate to OpenHAB the type of semantic class it belongs to, for example if it is a location (kitchen, dining room...), an equipment (projector, dishwasher, etc.) or a point (point, measurement, switch...) and the type of profile we prefer, which define the joint operation of the channels and the elements, in default the system adjusts the best profile in an autonomous way. These parameters will help us when creating the user interfaces in the MainUI, grouping the items according to their semantic class or group.



Channel

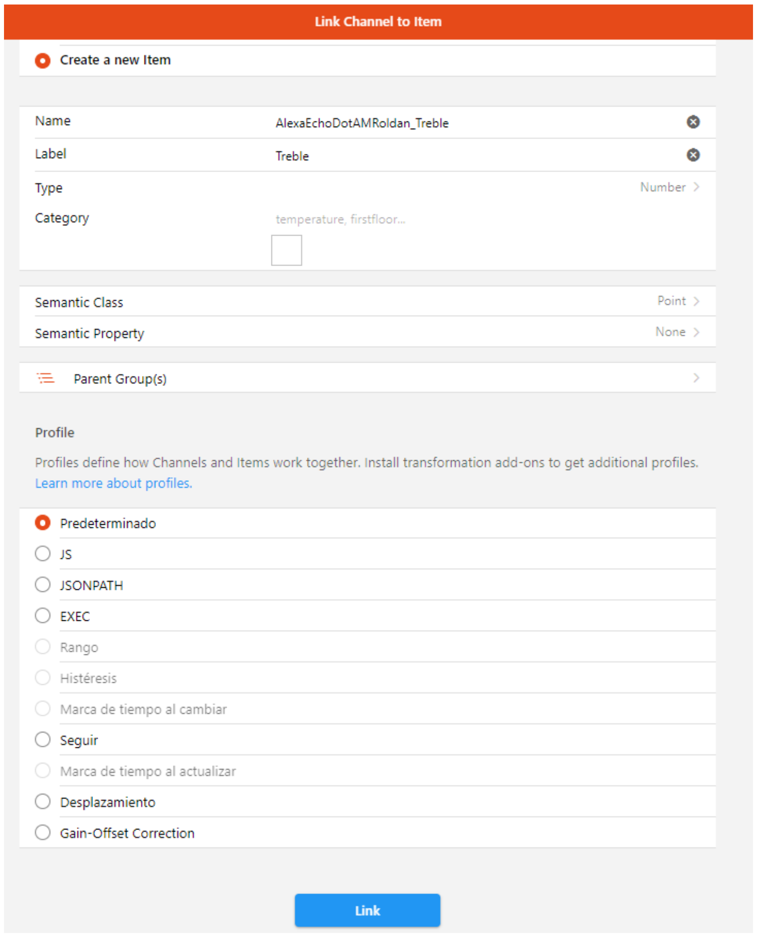
Treble
amazonechocontrol:echo:b76179028d:G091AA1013560363:equalizerTreble (Number)
Equalizer Treble

Item

[✓ Use an existing Item]
Create a new Item

Item to Link

Figure 5.45 – *Create on existing item*



[✓ Create a new Item]

Name	AlexaEchoDotAMRoldan_Treble
Label	Treble
Type	Number
Category	temperature, firstfloor...
Semantic Class	Point
Semantic Property	None

Parent Group(s)

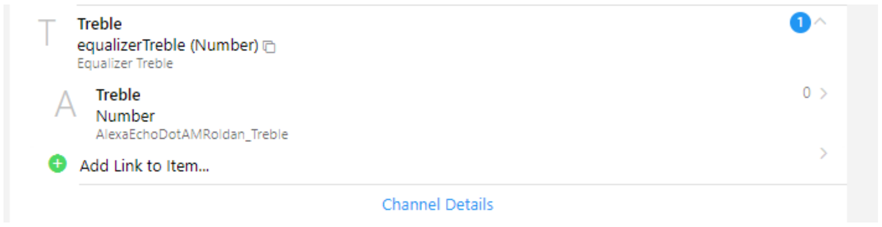
Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

[✓ Predeterminado]
JS
JSONPATH
EXEC
Rango
Histéresis
Marca de tiempo al cambiar
Seguir
Marca de tiempo al actualizar
Desplazamiento
Gain-Offset Correction

Figure 5.46 – *Create new item*

4. And finally the **item will be linked to the chosen channel**. If we access it, it will show us information about the current status of the item, the channel from which it has been created and the item created, in case we want to edit its parameters.



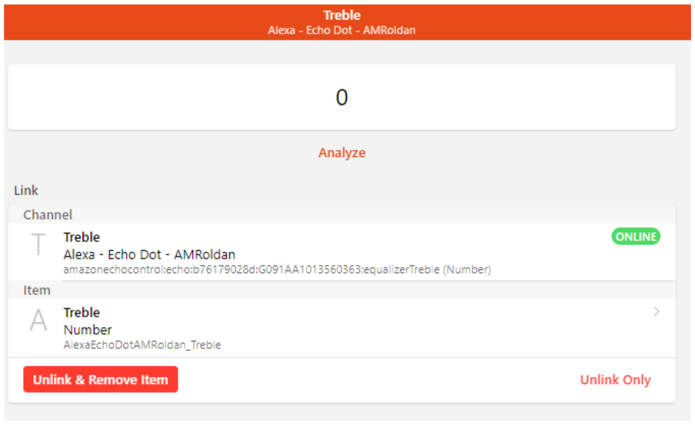
Treble
equalizerTreble (Number)
Equalizer Treble

Treble
Number
AlexaEchoDotAMRoldan_Treble
0

Add Link to Item...

Channel Details

Figure 5.47 – *Created item*



Treble
Alexa - Echo Dot - AMRoldan

0

Analyze

Link

Channel

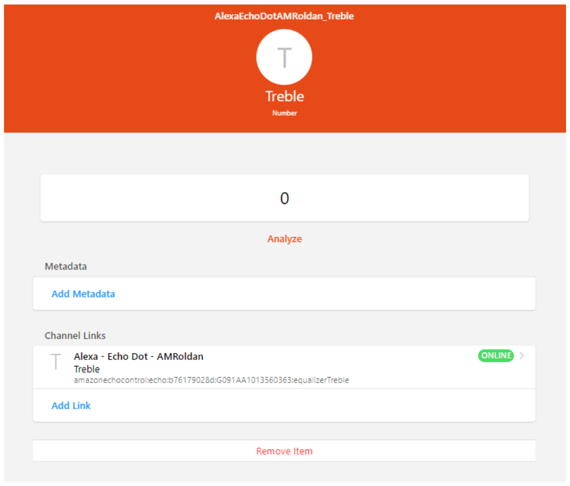
Treble
Alexa - Echo Dot - AMRoldan
amazonechocontrol:echo:b76179028d:G091AA1013560363:equalizerTreble (Number)
ONLINE

Item

Treble
Number
AlexaEchoDotAMRoldan_Treble

Unlink & Remove Item
Unlink Only

(a) *link info*



AlexaEchoDotAMRoldan_Treble
Treble
Number

0

Analyze

Metadata

Add Metadata

Channel Links

Alexa - Echo Dot - AMRoldan
Treble
amazonechocontrol:echo:b76179028d:G091AA1013560363:equalizerTreble
ONLINE

Add Link

Remove Item

(b) *item info*

Figure 5.48 – *Display information of item*

5.5 Smart Plugs

In this section we will see how to include the smart plugs chosen in the section 3.3. These plugs will be dedicated to turn on and off electronic devices as desired by the customer.

5.5.1 Amazon Smart Plug

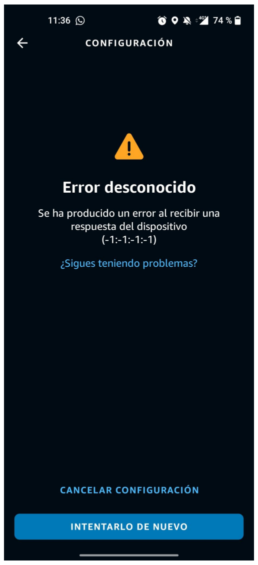
5.5.1.1 Add and configure the plug to Amazon Alexa

First of all, the plug has to be added to the Amazon Alexa application, as the plug is from the Amazon ecosystem. To add it we follow the same steps as in the section 5.4, as it is added in the same way as the echo dot speaker.

- **Download the Amazon Alexa application** from the Play Store or Apple Store, depending on the operating system of your mobile phone.

- To add the **plug**:
  a. We open the Amazon Alexa application.
  b. Go to the **devices tab** and in the **top right corner** click on the + **icon** and **add device**.
  c. We indicate that what we want to configure is a **Plug**.
  d. We will be asked about the **brand of the device**, as in this case it is Amazon, click on the corresponding brand.
  e. **Follow the steps indicated** by the application to complete the installation of the plug.

If you get an **error screen**, press and hold the button on the side of the device to **reset the plug** configuration. This screen may be because the WiFi has been changed.


11:36

CONFIGURACIÓN

Error desconocido

Se ha producido un error al recibir una respuesta del dispositivo (-1:-1:-1:-1)

¿Sigues teniendo problemas?

CANCELAR CONFIGURACIÓN

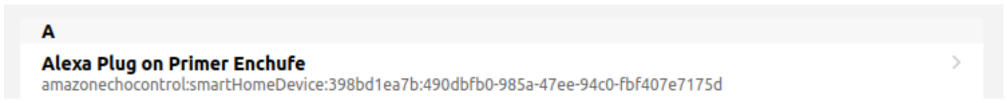
INTENTARLO DE NUEVO

Figure 5.49 – *Error screen add plug*

5.5.1.2 Create and configure things

Once the plug is added to Amazon Alexa, it will be **automatically discovered** thanks to the Amazon Echo Control binding that we have installed and configured in the 5.4 section to control the echo dot speaker. Therefore, a thing will have to be created as it has been done in the same section as the echo dot speaker, but we will no longer have to create the amazon account because it has already been configured previously.

1. To add the thing, go to the inbox tab and you should see the new thing that has been auto-discovered.

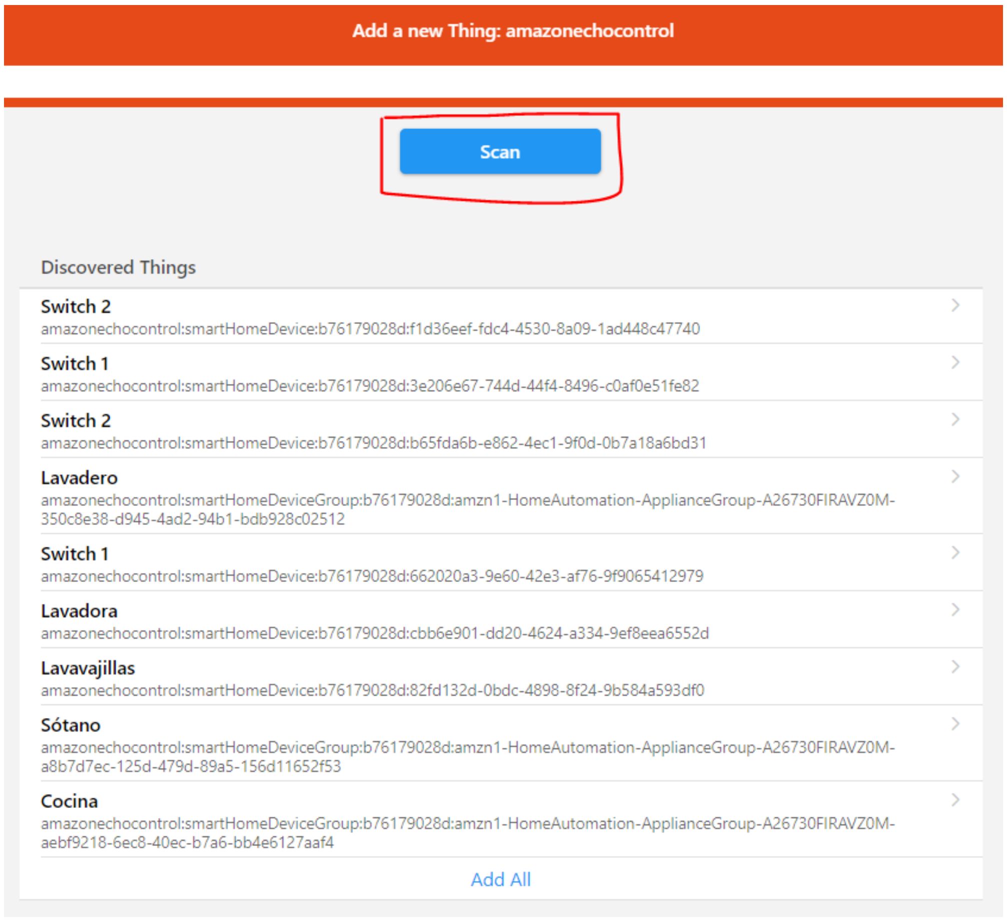

A

Alexa Plug on Primer Enchufe

amazonechocontrol:smartHomeDevice:398bd1ea7b:490dbfb0-985a-47ee-94c0-fbf407e7175d

Figure 5.50 – *Thing autodiscovered*

Otherwise it can also be viewed from the **Things -> + icon -> Amazon Echo Control -> Scan button**, this will perform a scan to auto-discover new things added in the Amazon Alexa application.



Discovered Things

Switch 2
amazonechocontrol:smartHomeDevice:b76179028d:f1d36eef-fdc4-4530-8a09-1ad448c47740

Switch 1
amazonechocontrol:smartHomeDevice:b76179028d:3e206e67-744d-44f4-8496-c0af0e51fe82

Switch 2
amazonechocontrol:smartHomeDevice:b76179028d:b65fda6b-e862-4ec1-9f0d-0b7a18a6bd31

Lavadero
amazonechocontrol:smartHomeDeviceGroup:b76179028d:amzn1-HomeAutomation-ApplianceGroup-A26730FIRAVZ0M-350c8e38-d945-4ad2-94b1-bdb928c02512

Switch 1
amazonechocontrol:smartHomeDevice:b76179028d:662020a3-9e60-42e3-af76-9f9065412979

Lavadora
amazonechocontrol:smartHomeDevice:b76179028d:cbb6e901-dd20-4624-a334-9ef8eea6552d

Lavavajillas
amazonechocontrol:smartHomeDevice:b76179028d:82fd132d-0bdc-4898-8f24-9b584a593df0

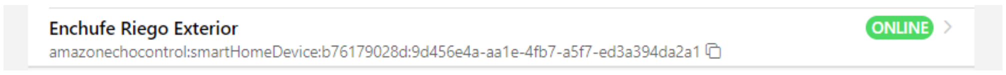
Sótano
amazonechocontrol:smartHomeDeviceGroup:b76179028d:amzn1-HomeAutomation-ApplianceGroup-A26730FIRAVZ0M-a8b7d7ec-125d-479d-89a5-156d11652f53

Cocina
amazonechocontrol:smartHomeDeviceGroup:b76179028d:amzn1-HomeAutomation-ApplianceGroup-A26730FIRAVZ0M-aebf9218-6ec8-40ec-b7a6-bb4e6127aaf4

Add All

Figure 5.51 – *Alternative not autodiscover thing*

2. Once the thing is discovered, we add it to the system, to do this we click on it and a window will pop up, we click on **"Add as a Thing"**, and it will be added to the system by itself, as explained in section 5.4.

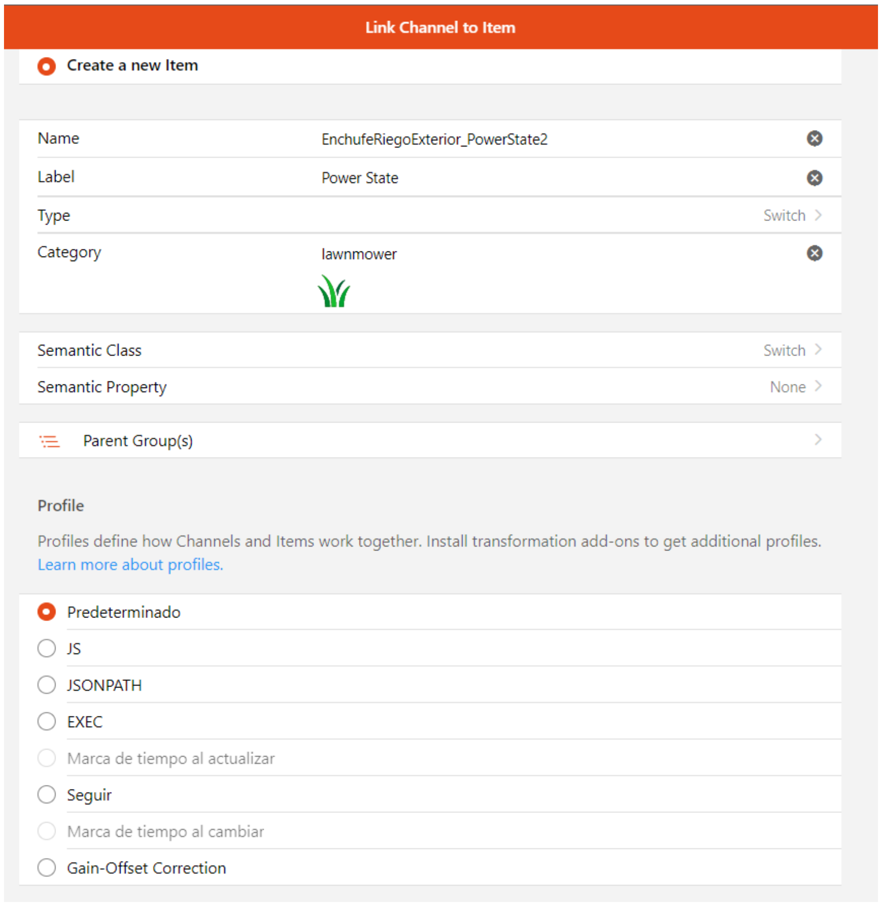


Enchufe Riego Exterior
amazonechocontrol:smartHomeDevice:b76179028d:9d456e4a-aa1e-4fb7-a5f7-ed3a394da2a1

Figure 5.52 – *Thing online*

5.5.1.3 Create and configure item

Once the thing is created and configured, we create the corresponding item to turn the plug on and off. It is created in the same way as we have seen in the section 5.4 of creating items. In this case, as it is a plug, we will use the semantic class Point and type Swicth.


Link Channel to Item

[✓ Create a new Item]

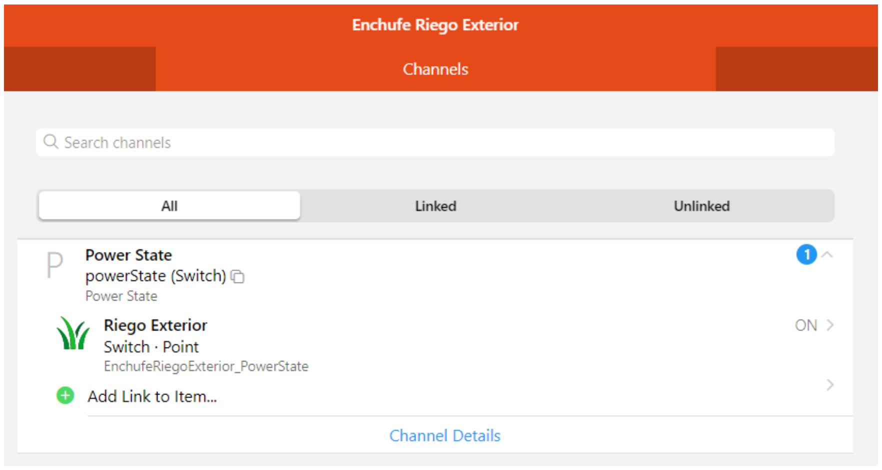
Name	EnchufeRiegoExterior_PowerState2
Label	Power State
Type	Switch
Category	lawnmower
Semantic Class	Switch
Semantic Property	None
Parent Group(s)	

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

[✓ Predeterminado]
JS
JSONPATH
EXEC
Marca de tiempo al actualizar
Seguir
Marca de tiempo al cambiar
Gain-Offset Correction

Figure 5.53 – *Create new item*


Enchufe Riego Exterior

Channels

Search channels

All | Linked | Unlinked

Power State
powerState (Switch)
Power State

Riego Exterior
Switch · Point
EnchufeRiegoExterior_PowerState
ON

Add Link to Item...

Channel Details

Figure 5.54 – *Created Item*

5.5.2 TP-Link Tapo P100

These plugs are from the TP-Link Tapo ecosystem so they have another implementation in the system.

5.5.2.1 Add plugs to Tapo app

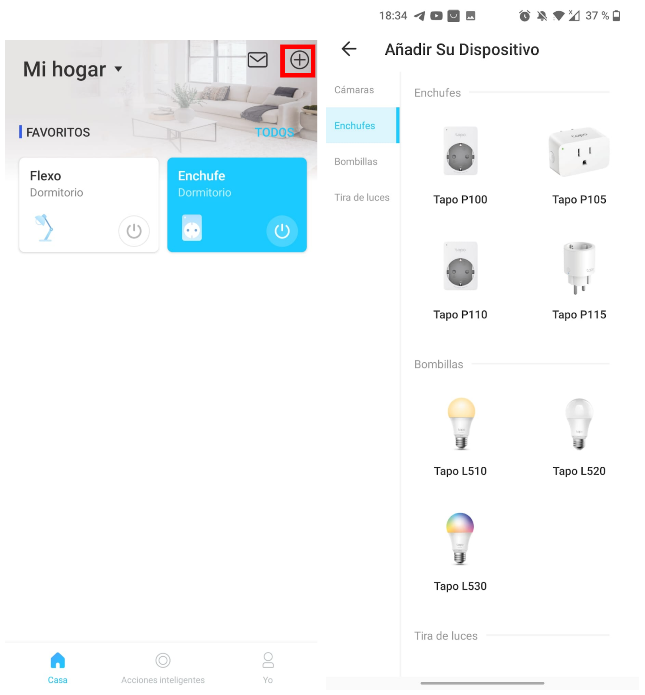
- Download the **Tapo App** from the Play Store or Apple Store, depending on the operating system of your mobile and create an account.

Figure 5.55 – *Tapo logo*

- To add the different plugs Tapo P100:

a. Open the Tapo application.

b. On the main screen in the top right corner click on the + **icon** and indicate which device it is and the **model**, in this case Plug and P100.



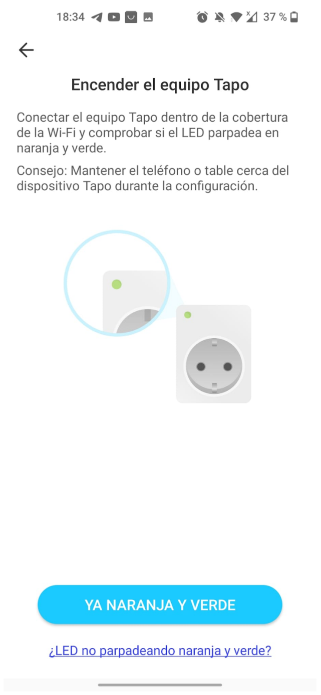
(a) *Home* **(b)** *Select device*

Figure 5.56 – *Add new device*

c. We connect the equipment to the power supply, so that it turns on and activates the **linking protocol**.

d. **Follow the steps** indicated by the application to complete the installation of the Tapo P100 plug:

  - It will ask us if the device has the light flashing between orange and green (it goes straight on when it is plugged in).


18:34 37 %

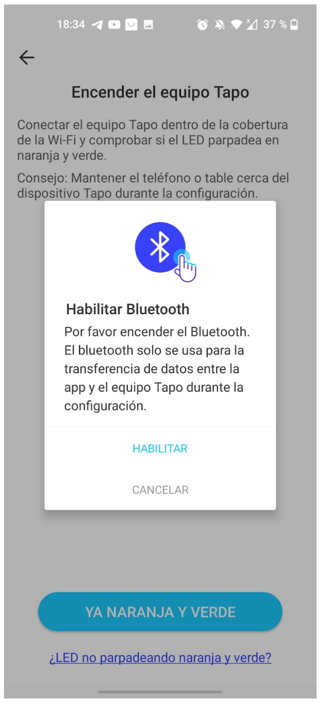
Encender el equipo Tapo

Conectar el equipo Tapo dentro de la cobertura de la Wi-Fi y comprobar si el LED parpadea en naranja y verde.

Consejo: Mantener el teléfono o table cerca del dispositivo Tapo durante la configuración.

Figure 5.57 – *Blink linkage*

- You will need to have blutooth and location activated on your mobile phone.


18:34 37 %

Encender el equipo Tapo

Conectar el equipo Tapo dentro de la cobertura de la Wi-Fi y comprobar si el LED parpadea en naranja y verde.

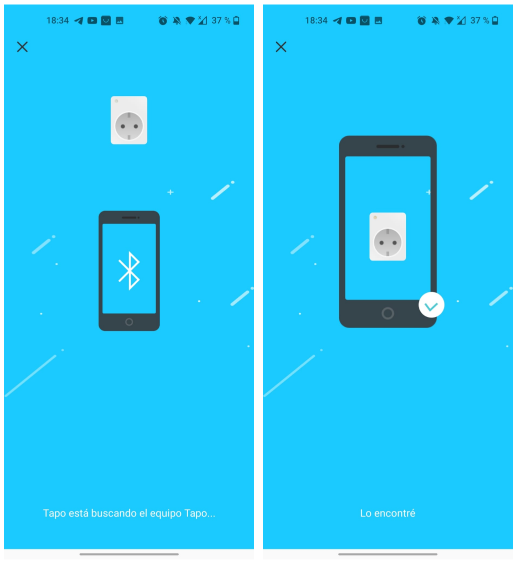
Consejo: Mantener el teléfono o table cerca del dispositivo Tapo durante la configuración.

Habilitar Bluetooth

Por favor encender el Bluetooth. El bluetooth solo se usa para la transferencia de datos entre la app y el equipo Tapo durante la configuración.

Figure 5.58 – *Enable bluetooth*

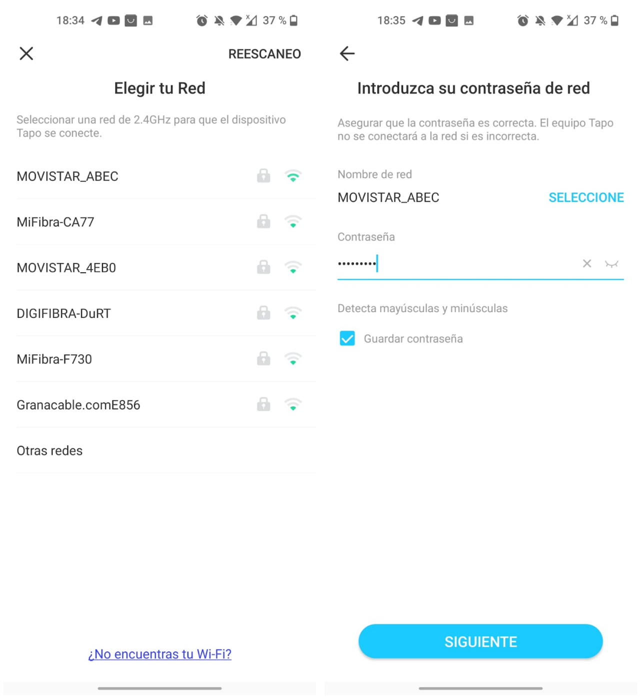
- It will start looking for the plug.



(a) **(b)**

Figure 5.59 – *Looking for device*

- Once found, we will have to enter the wifi network to connect the plug to it.


18:34 37 %

REESCANEO

Elegir tu Red

Seleccionar una red de 2.4GHz para que el dispositivo Tapo se conecte.

MOVISTAR_ABEC

MiFibra-CA77

MOVISTAR_4EB0

DIGIFIBRA-DuRT

MiFibra-F730

Granacable.comE856

Otras redes

¿No encuentras tu Wi-Fi?

18:35 37 %

Introduzca su contraseña de red

Asegurar que la contraseña es correcta. El equipo Tapo no se conectará a la red si es incorrecta.

Nombre de red

MOVISTAR_ABEC SELECCIONE

Contraseña

•••••••••

Detecta mayúsculas y minúsculas

☑ Guardar contraseña

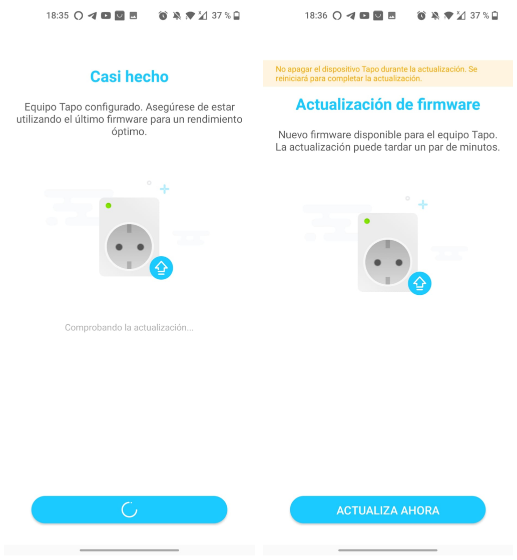
SIGUIENTE

(a) **(b)**

Figure 5.60 – *Connect device to WiFi*

- It will ask us to give a name and icon to the device.

- Finally, it will check for updates to the device's firmware, if it finds any it is recommended to update.


18:35 37 %

Casi hecho

Equipo Tapo configurado. Asegúrese de estar utilizando el último firmware para un rendimiento óptimo.

Comprobando la actualización...

18:36 37 %

No apagar el dispositivo Tapo durante la actualización. Se reiniciará para completar la actualización.

Actualización de firmware

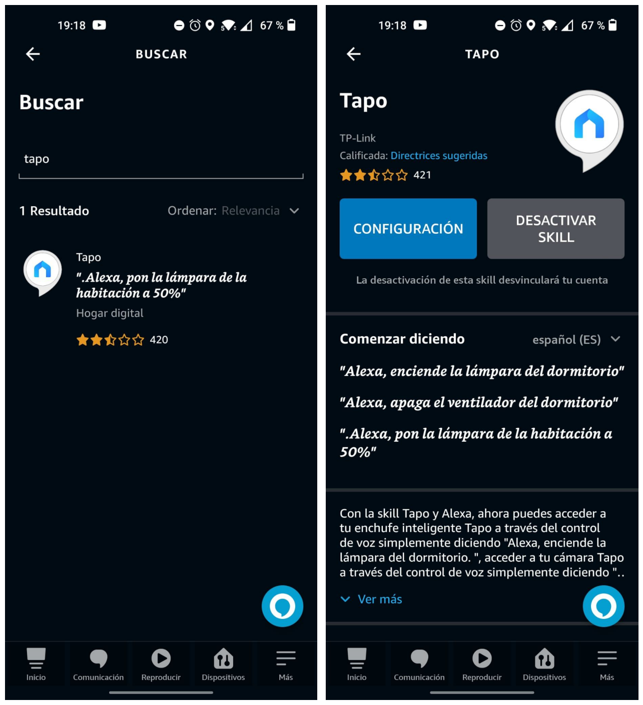
Nuevo firmware disponible para el equipo Tapo. La actualización puede tardar un par de minutos.

ACTUALIZA AHORA

(a) **(b)**

Figure 5.61 – *Update firmware*

If we have Alexa installed, it will notify us that it has found a Tapo device on the network and will install the Tapo skill automatically and we will be able to operate it with the Amazon Alexa app as well. If we do not get the message we can install the skill from the **Amazon Alexa application -> more -> skills and games -> search for Tapo and install the skill**, it will ask us for the account details that we had to create when we entered the Tapo app.


19:18 67 %

BUSCAR

Buscar

tapo

1 Resultado Ordenar: Relevancia

Tapo
".Alexa, pon la lámpara de la habitación a 50%"
Hogar digital
420

Inicio Comunicación Reproducir Dispositivos Más

19:18 67 %

TAPO

Tapo

TP-Link
Calificada: Directrices sugeridas
421

CONFIGURACIÓN DESACTIVAR SKILL

La desactivación de esta skill desvinculará tu cuenta

Comenzar diciendo español (ES)

"Alexa, enciende la lámpara del dormitorio"

"Alexa, apaga el ventilador del dormitorio"

".Alexa, pon la lámpara de la habitación a 50%"

Con la skill Tapo y Alexa, ahora puedes acceder a tu enchufe inteligente Tapo a través del control de voz simplemente diciendo "Alexa, enciende la lámpara del dormitorio. ", acceder a tu cámara Tapo a través del control de voz simplemente diciendo "..

Ver más

Inicio Comunicación Reproducir Dispositivos Más

(a) *add* **(b)** *bindings*

Figure 5.62 – *Skill tapo*

5.5.2.2 Install and configure binding

Once the devices have been added in their respective application, we can add them to our system. These plugs can be added in two different ways:

- Through the **Amazon Echo Control binding**, as we have installed it to connect to Alexa.
- Through its own **Tapo Control binding**.

It will be installed using Tapo Control's own binding. As we have the Amazon Echo Control binding installed, the corresponding things will be auto-discovered, but we won't add them, but we'll leave the possibility there in case the other binding fails.

As we have seen in the other sections of the other devices, to install the bindings we go to the **Settings -> Bindings tab**, look for **Tapo Control** and install the binding.

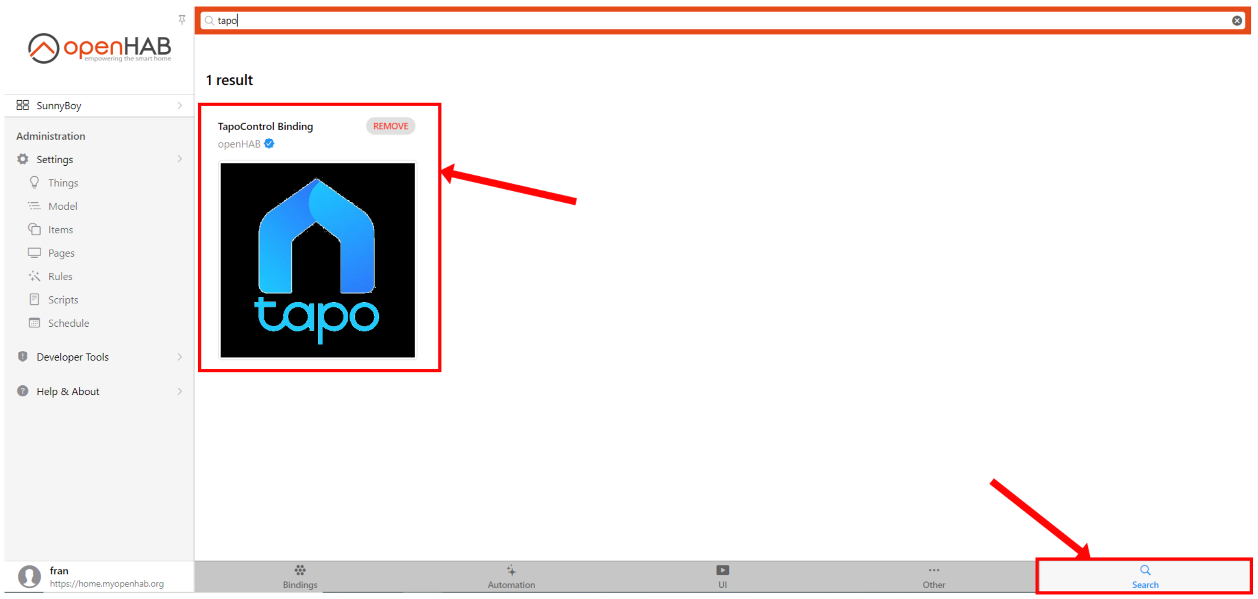
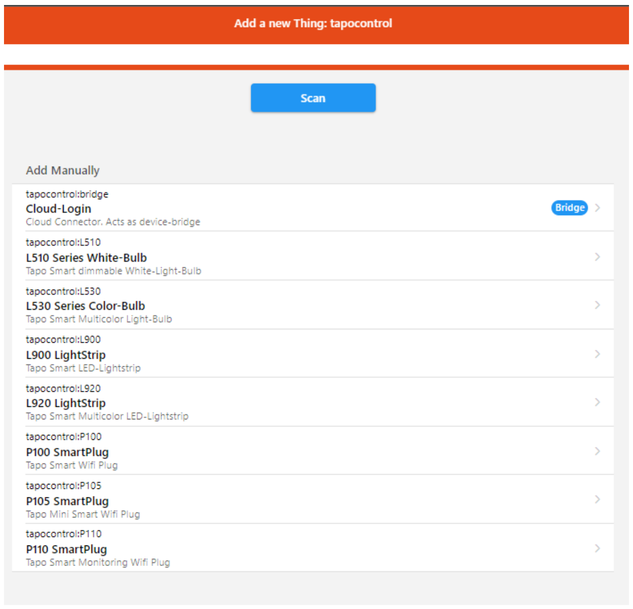


Figure 5.63 – *Install Tapo binding*

5.5.2.3 Create and configure Thing

Once the binding is installed, go to the **Settings -> Things tab**.

1. As with the Amazon Echo Control binding, we need a thing bridge to connect the two platforms. So from the **Things tab -> + icon -> select the Tapo Control binding**. Once inside, select **Cloud-Login** and create the thing by entering the email and password of the Tapo account that we created when we logged into the Tapo mobile app.



Add a new Thing: tapocontrol

Scan

Add Manually

tapocontrol:bridge
Cloud-Login
Cloud Connector. Acts as device-bridge
Bridge

tapocontrol:L510
L510 Series White-Bulb
Tapo Smart dimmable White-Light-Bulb

tapocontrol:L530
L530 Series Color-Bulb
Tapo Smart Multicolor Light-Bulb

tapocontrol:L900
L900 LightStrip
Tapo Smart LED-Lightstrip

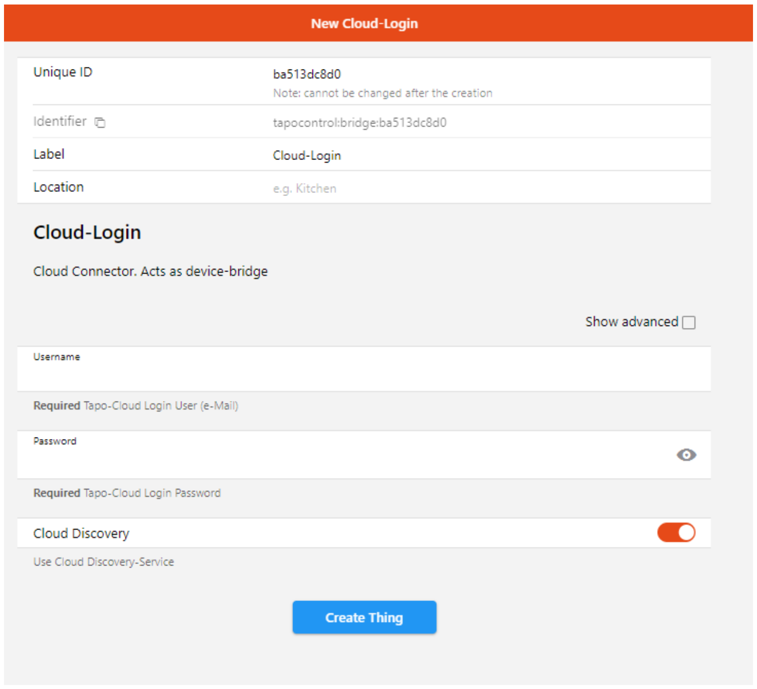
tapocontrol:L920
L920 LightStrip
Tapo Smart Multicolor LED-Lightstrip

tapocontrol:P100
P100 SmartPlug
Tapo Smart Wifi Plug

tapocontrol:P105
P105 SmartPlug
Tapo Mini Smart Wifi Plug

tapocontrol:P110
P110 SmartPlug
Tapo Smart Monitoring Wifi Plug

Figure 5.64 – *Things we can add with tapo binding*



New Cloud-Login

Unique ID	ba513dc8d0 Note: cannot be changed after the creation
Identifier	tapocontrol:bridge:ba513dc8d0
Label	Cloud-Login
Location	e.g. Kitchen

Cloud-Login

Cloud Connector. Acts as device-bridge

Show advanced

Username

Required Tapo-Cloud Login User (e-Mail)

Password

Required Tapo-Cloud Login Password

Cloud Discovery

Use Cloud Discovery-Service

Create Thing

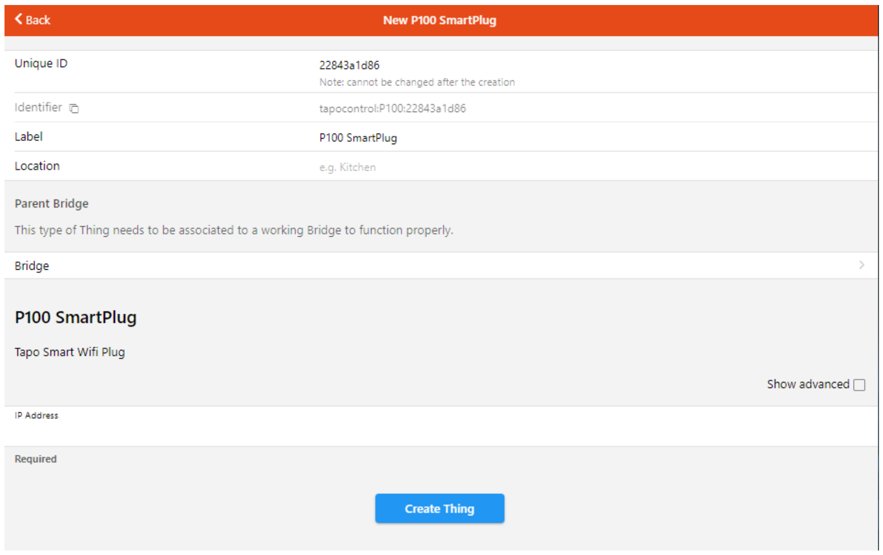
Figure 5.65 – *Configure Tapo account thing*

2. Once the bridge has been created, the plug we have added to the application will be **autodiscovered**, as with Alexa. To see if it has been auto-discovered, we go to the inbox tab and there the plug should appear with the same name that we have given it in the Tapo application. If it is not autodiscovered, we can perform the scan that we have explained previously in the section 5.4.

3. If it still does not appear we can add it manually as we have created the Cloud-Login thing, and we choose to add a **P100 Smartplug thing**. To configure the thing is only to indicate the thing of the

bridge, the account that we have created before, and to put the IP of the device, to see that IP has the plug:

- Go to the Tapo app.
- Click on the device.
- Click on the wheel at the top right to open the configuration.
- Select device information, in this section we will see information related to the device, time zone, WiFi network to which the device is connected, model, IP address, MAC address ...


< Back
New P100 SmartPlug

Unique ID 22843a1d86
Note: cannot be changed after the creation
Identifier tapocontrol:P100:22843a1d86
Label P100 SmartPlug
Location e.g. Kitchen

Parent Bridge
This type of Thing needs to be associated to a working Bridge to function properly.
Bridge

P100 SmartPlug
Tapo Smart Wifi Plug
Show advanced
IP Address
Required
Create Thing

Figure 5.66 – *Add plug manually*

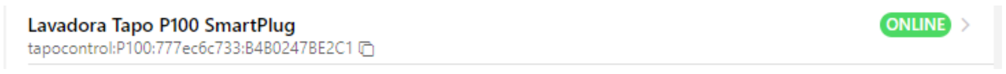
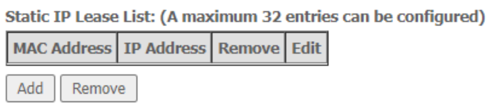

Lavadora Tapo P100 SmartPlug
tapocontrol:P100:777ec6c733:B4B0247BE2C1
ONLINE

Figure 5.67 – *Thing online*

Once the thing is added, a recommendation is to **assign a static IP to the plug according to its MAC address**, with this we get that if there is a power outage or internet connection, the device has the same IP, otherwise it would have to change it every time there is a power outage or internet connection.

To set a static IP according to a MAC address, it is very different depending on your internet company and type of router, as to assign the static IP we must access the router to be able to assign the static IP to the device. For example in Movistar:

- Access 192.168.1.1 from the browser.
- Once inside, it will ask for a password which is on the back of the router, where the default passwords are located when the router is installed.
- Menu->Advanced Settings -> LAN -> Static IP Lease List, add the MAC address and the static IP that the device has, as we have seen before from the Tapo app.


Static IP Lease List: (A maximum 32 entries can be configured)

MAC Address	IP Address	Remove	Edit

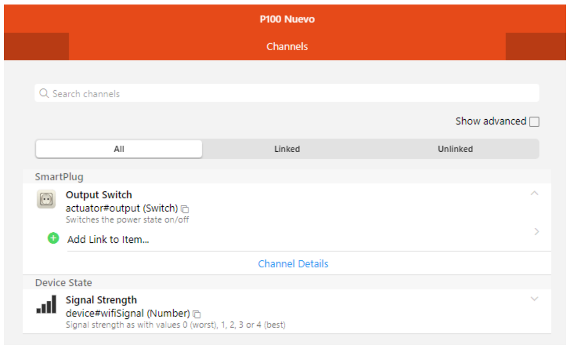
Add Remove

Figure 5.68 – *Add static IP*

5.5.2.4 Create and configure item

Once the corresponding things have been configured, the items for controlling the plug will be added.

1. From the **things tab**, we access the Tapo P100 plug thing to add its functionalities. Once here we go to the **channels tab** to see all the functions that the thing can perform. Once in the channels tab, click on the function we want to create, in this case the power state to control the on and off, and click on **"Link to item"**.



P100 Nuevo
Channels

Search channels

Show advanced ☐

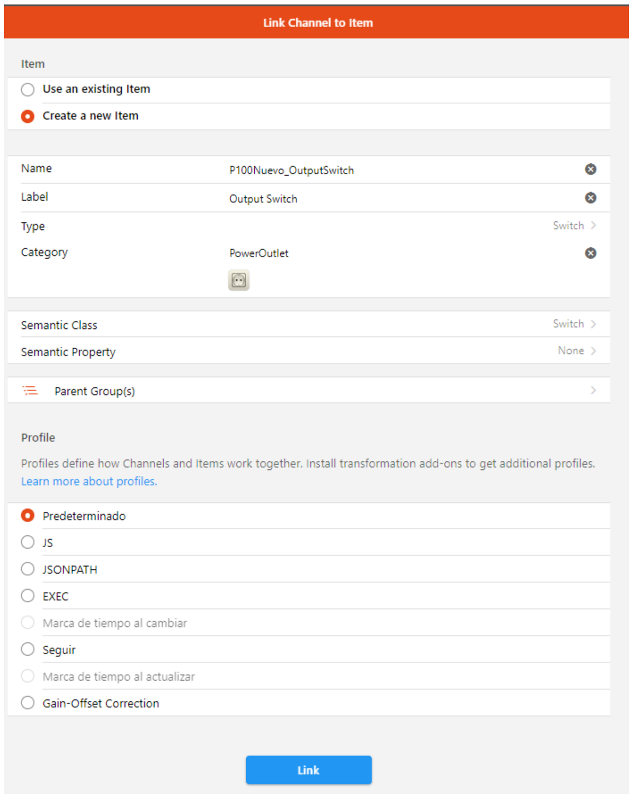
All | Linked | Unlinked

SmartPlug
Output Switch
actuator#output (Switch)
Switches the power state on/off
Add Link to Item...
Channel Details

Device State
Signal Strength
device#wifiSignal (Number)
Signal strength as with values 0 (worst), 1, 2, 3 or 4 (best)

Figure 5.69 – *Channels*

2. It will redirect us to another page that will give us the option to **link this channel** to an **existing item** or to **create an item from zero**. In the item we have to put a unique id, a label, indicate the type of the variable if it is a number, a string, a switch, etc, in this case put switch and we can also put an icon to the item. In addition, indicate the semantic class swicth and the profile type in default.



Link Channel to Item

Item
○ Use an existing Item
● Create a new Item

Name	P100Nuevo_OutputSwitch
Label	Output Switch
Type	Switch
Category	PowerOutlet
Semantic Class	Switch
Semantic Property	None

Parent Group(s)

Profile
Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

● Predeterminado
○ JS
○ JSONPATH
○ EXEC
○ Marca de tiempo al cambiar
○ Seguir
○ Marca de tiempo al actualizar
○ Gain-Offset Correction

Link

Figure 5.70 – *Create item*

3. And finally the item will be linked to the chosen channel. If we access it, it will show us **information** about the current status of the item, the channel from which it has been created and the item created, in case we want to edit its parameters.

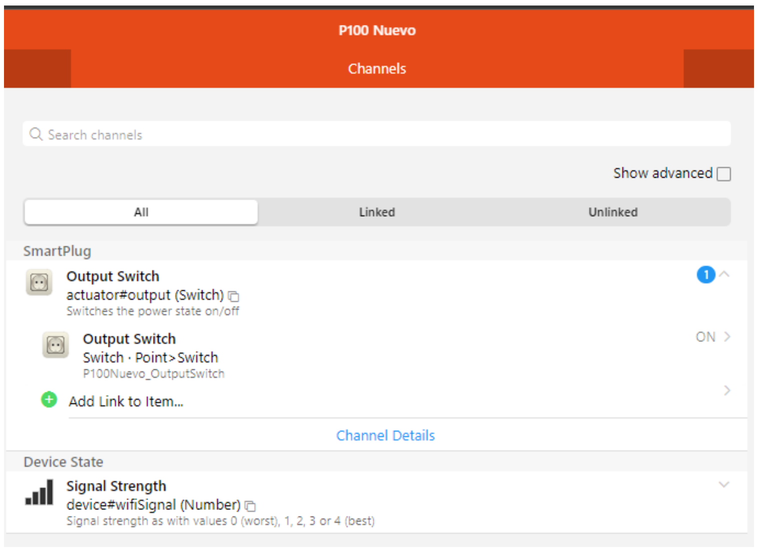
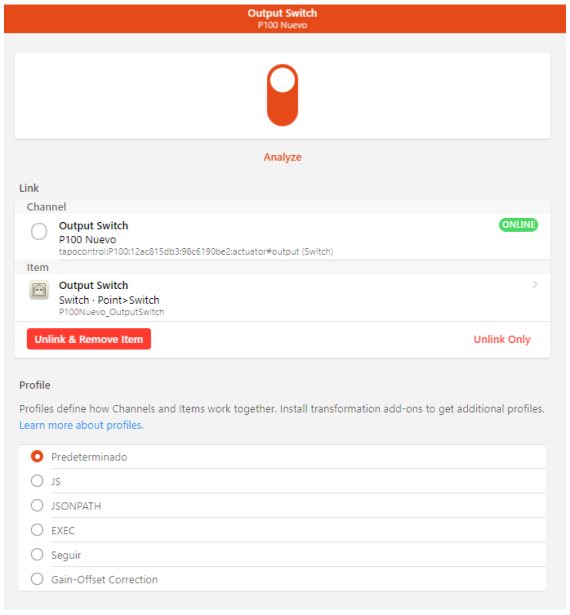
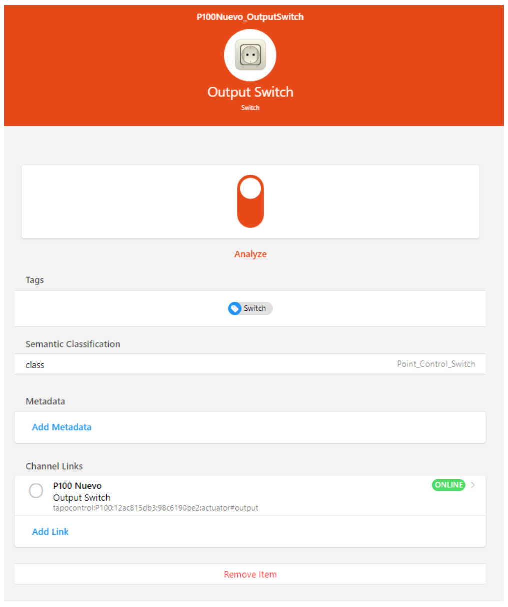


Figure 5.71 – *Created Item*



(a) *Info link*



(b) *Info item*

Figure 5.72 – *Display information of item*

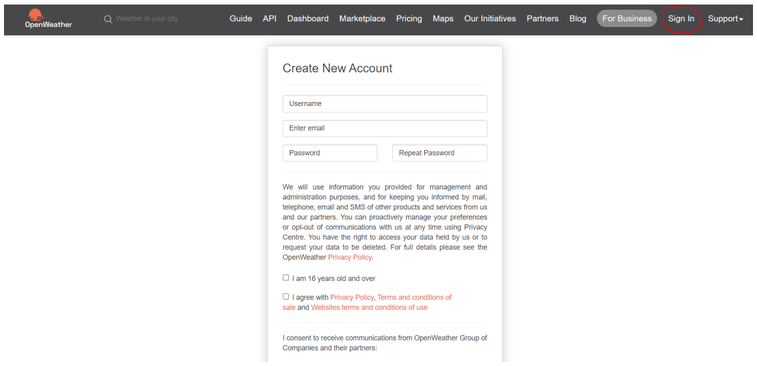
5.6 OpenWeatherMap API

In this section, we will explain how to implement a service, such as weather forecasting, to OpenHAB to obtain weather measurements for a given area. As you can see in the comparison of the weather APIs in the section 3.6, we decided to stay with OpenWeatherMap because of its number of calls to the API per day.

5.6.1 Create account and get API key

First of all, get an account on the OpenWeatherMap page and get the API key, which allows the service to identify the requests you make to the API. To do this:

1. Access the OpenWeatherMap page and **register** by clicking on Sign In in the top right-hand corner. Once inside, click on Create account and enter the data requested.



Create New Account

Username

Enter email

Password

Repeat Password

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather Privacy Policy.

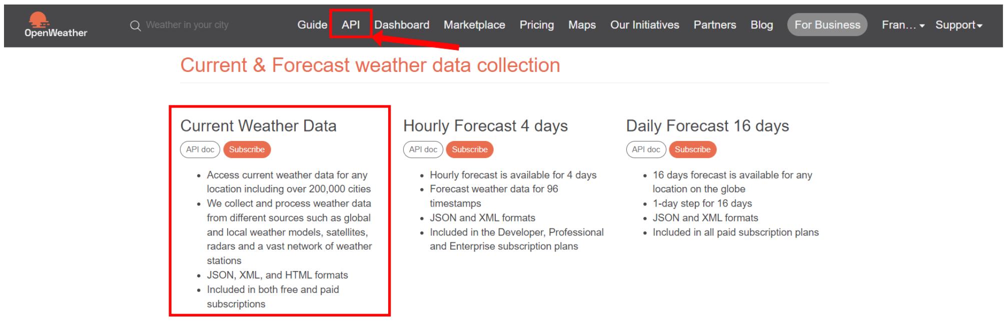
☐ I am 16 years old and over

☐ I agree with Privacy Policy, Terms and conditions of sale and Websites terms and conditions of use

I consent to receive communications from OpenWeather Group of Companies and their partners:

Figure 5.73 – *Register on the website*

2. Once registered, go to the **API tab** at the top of the menu, go to the **Current & Forecast weather data** collection section, the one we are interested in is Current Weather Data and click on **subscribe**.



Current & Forecast weather data collection

Current Weather Data

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations
- JSON, XML, and HTML formats
- Included in both free and paid subscriptions

Hourly Forecast 4 days

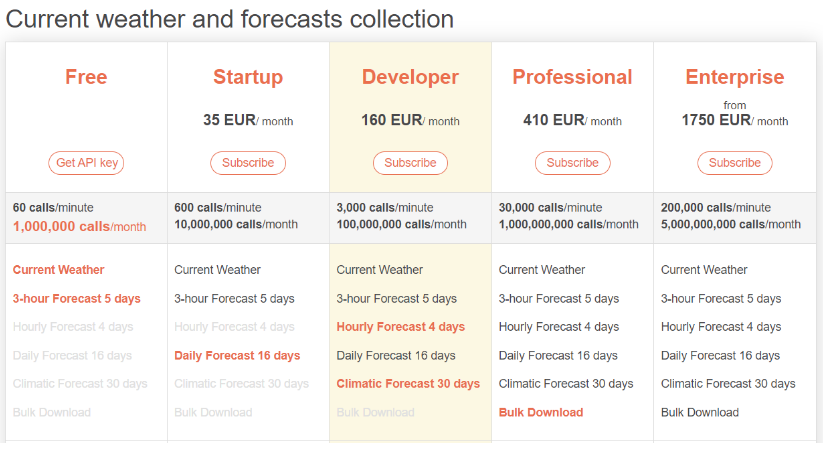
- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- JSON and XML formats
- Included in the Developer, Professional and Enterprise subscription plans

Daily Forecast 16 days

- 16 days forecast is available for any location on the globe
- 1-day step for 16 days
- JSON and XML formats
- Included in all paid subscription plans

Figure 5.74 – *Select API*

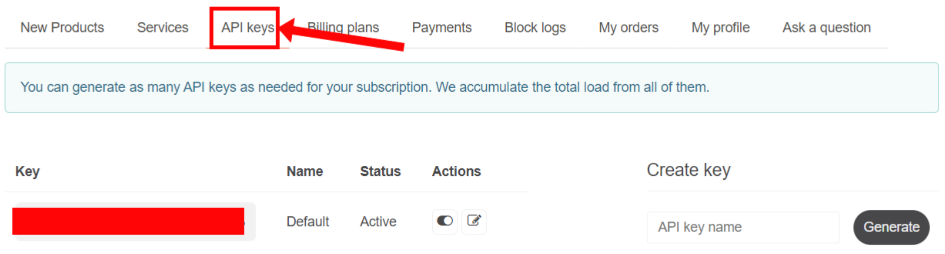
3. It will take us to the **pricing page**, go down a little and we will see a table comparing the types of plans available. As we are going to use the free plan, we click on get **API key** of the free plan. We redirect to a page and select the API key tab, there we will get the API key that we have been given, you can generate as many as you want.



Current weather and forecasts collection

Free	Startup	Developer	Professional	Enterprise
	35 EUR/ month	160 EUR/ month	410 EUR/ month	from 1750 EUR/ month
Get API key	Subscribe	Subscribe	Subscribe	Subscribe
60 calls/minute 1,000,000 calls/month	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
Current Weather	Current Weather	Current Weather	Current Weather	Current Weather
3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days
Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days
Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days
Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days
Bulk Download	Bulk Download	Bulk Download	Bulk Download	Bulk Download

Figure 5.75 – *Pricing API*



You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
[redacted]	Default	Active	

Create key

Figure 5.76 – *Get API key*

5.6.2 Install and configure binding

Once the account has been created and the API key has been obtained. As we have seen in the other sections of the other devices, to install the bindings we go to the **Settings -> Bindings tab**, look for **OpenWeatherMap binding** and install the binding.

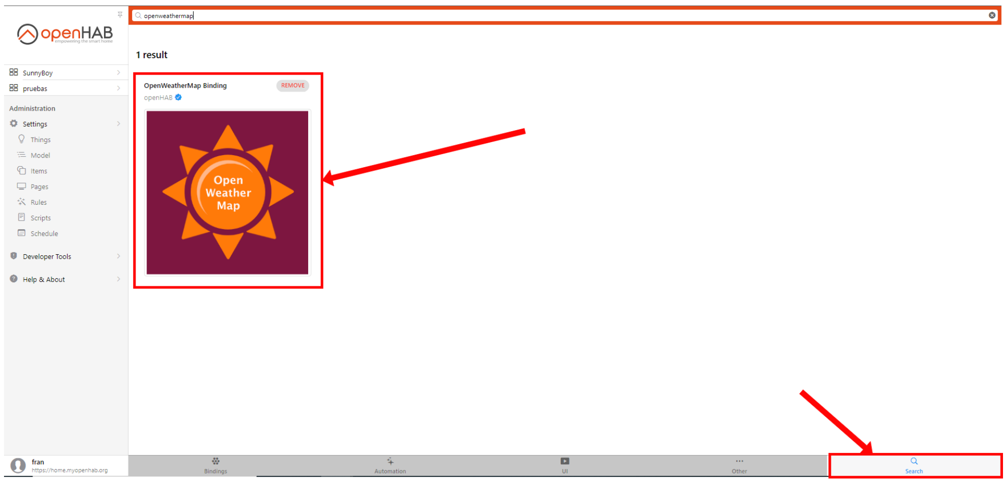
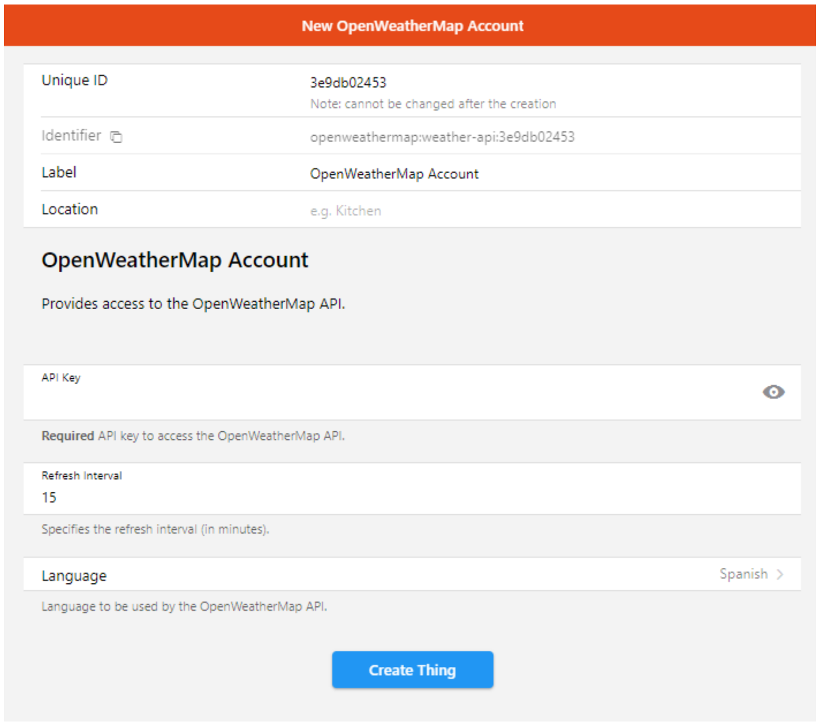


Figure 5.77 – *Install binding*

5.6.3 Create and configure things

Once the binding is installed, go to the **Settings -> Things tab** to start adding and configuring things.

1. As with the other bindings, we also need a thing bridge to connect the two platforms. So from the **Things tab -> + icon -> select the OpenWeatherMap binding**. Once inside, select **OpenWeatherMap account** and create the thing putting the **API key** we have obtained, the time interval we want the data to be updated and the language.



New OpenWeatherMap Account

Unique ID	3e9db02453 Note: cannot be changed after the creation
Identifier	openweathermap:weather-api:3e9db02453
Label	OpenWeatherMap Account
Location	e.g. Kitchen

OpenWeatherMap Account

Provides access to the OpenWeatherMap API.

API Key

Required API key to access the OpenWeatherMap API.

Refresh Interval
15

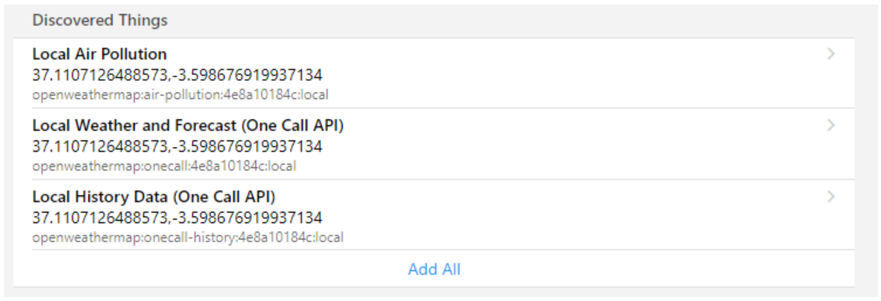
Specifies the refresh interval (in minutes).

Language Spanish

Language to be used by the OpenWeatherMap API.

Figure 5.78 – *Create bridge thing*

2. Once the bridge is created, it will **autodiscover** a few things, like the **weather forecast**, air quality and One Call API (but this one you had to put your credit card to get it, although it is free if you do not exceed the number of calls per day) you can see it from the inbox tab or from the binding tab itself to add things manually. If it doesn't auto discover, we can do the scan that we have explained above in the section 5.4.



Discovered Things

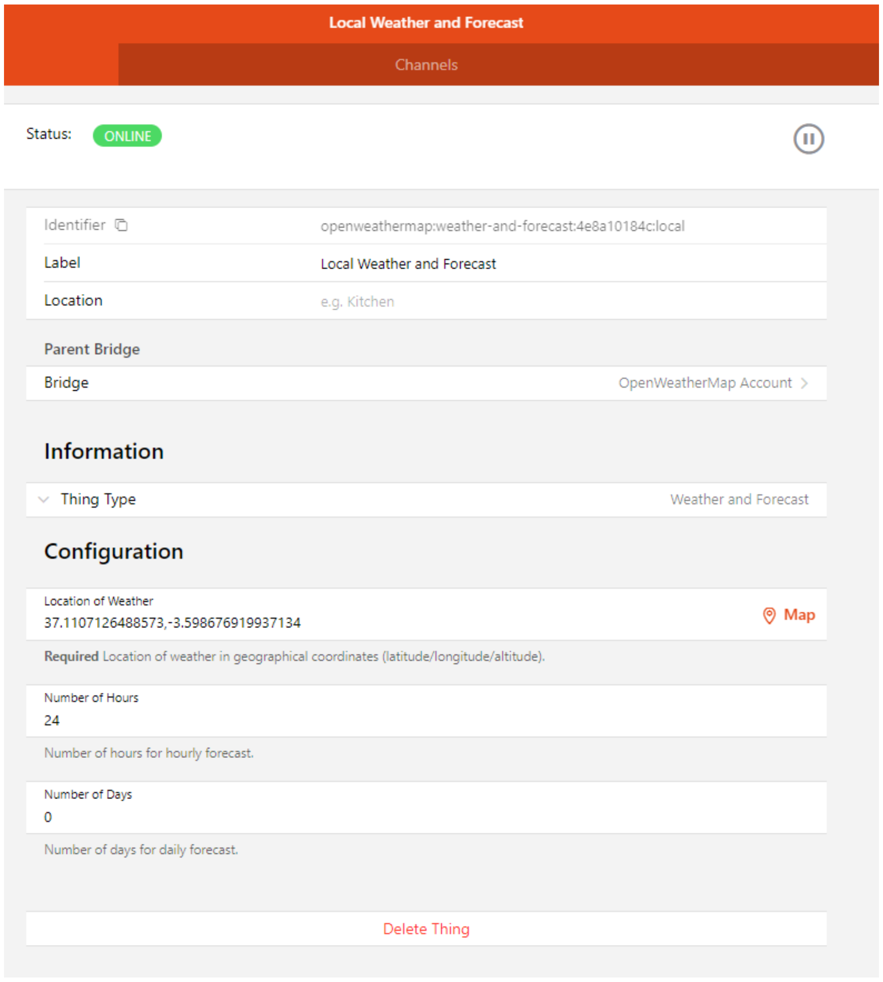
Local Air Pollution
37.1107126488573,-3.598676919937134
openweathermap:air-pollution:4e8a10184c:local

Local Weather and Forecast (One Call API)
37.1107126488573,-3.598676919937134
openweathermap:onecall:4e8a10184c:local

Local History Data (One Call API)
37.1107126488573,-3.598676919937134
openweathermap:onecall-history:4e8a10184c:local

Figure 5.79 – *Autodiscovered things*

3. If it still does not appear we can add it manually as we have done in the bridge, and choose **Local Weather and Forecast**. To configure the thing is only to indicate the bridge thing to indicate the account, the location from which we want to get the data, by default it puts the location that we put in the installation, and number of hours for the hourly forecast and number of days for the daily forecast, once you change the parameters forecastHours or forecastDays, the groups of channels available in the thing will be created or deleted accordingly.



Identifier	openweathermap:weather-and-forecast:4e8a10184c:local
Label	Local Weather and Forecast
Location	e.g. Kitchen

Parent Bridge

Bridge	OpenWeatherMap Account

Information

Thing Type	Weather and Forecast

Configuration

Location of Weather
37.1107126488573,-3.598676919937134
Map
Required Location of weather in geographical coordinates (latitude/longitude/altitude).

Number of Hours
24
Number of hours for hourly forecast.

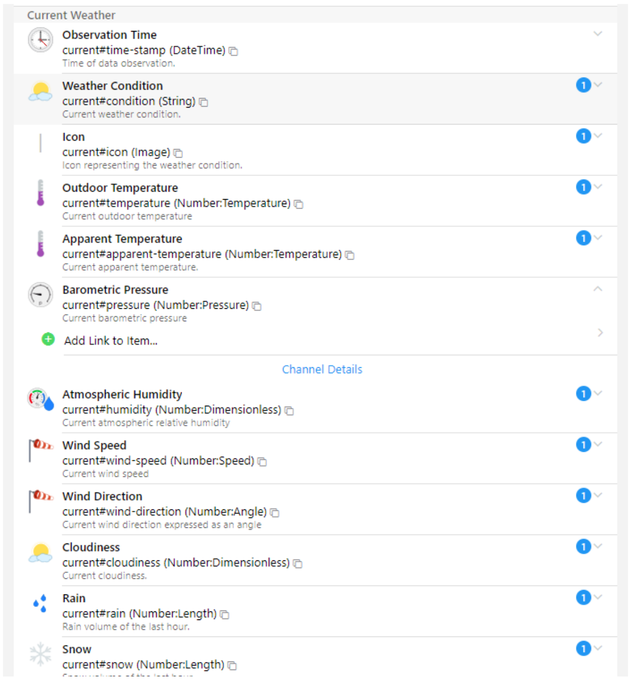
Number of Days
0
Number of days for daily forecast.

Figure 5.80 – *Configure thing*

5.6.4 Create and configure items

Once the things have been created, in order to obtain the meteorological data we access the **"Local Weather and Forecast" thing** that we have added before, once there in the channels tab, we will see the meteorological measurements that we can obtain and add to the system, according to the values established in the configuration parameters of the thing

1. From the **channels tab**, click on the **variable** you want to obtain its measurement in the system, and click on **"Link to item"**.



Current Weather

Observation Time
current#time-stamp (DateTime)
Time of data observation.

Weather Condition
current#condition (String)
Current weather condition.

Icon
current#icon (Image)
Icon representing the weather condition.

Outdoor Temperature
current#temperature (Number:Temperature)
Current outdoor temperature

Apparent Temperature
current#apparent-temperature (Number:Temperature)
Current apparent temperature.

Barometric Pressure
current#pressure (Number:Pressure)
Current barometric pressure

Add Link to Item...

Channel Details

Atmospheric Humidity
current#humidity (Number:Dimensionless)
Current atmospheric relative humidity

Wind Speed
current#wind-speed (Number:Speed)
Current wind speed

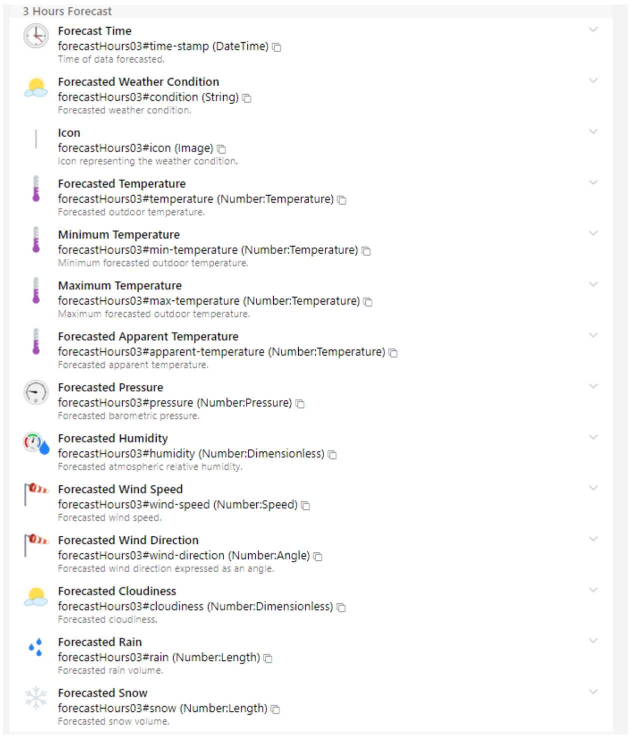
Wind Direction
current#wind-direction (Number:Angle)
Current wind direction expressed as an angle

Cloudiness
current#cloudiness (Number:Dimensionless)
Current cloudiness.

Rain
current#rain (Number:Length)
Rain volume of the last hour.

Snow
current#snow (Number:Length)

Figure 5.81 – *Some channels available*



3 Hours Forecast

Forecast Time
forecastHours03#time-stamp (DateTime)
Time of data forecasted.

Forecasted Weather Condition
forecastHours03#condition (String)
Forecasted weather condition.

Icon
forecastHours03#icon (Image)
Icon representing the weather condition.

Forecasted Temperature
forecastHours03#temperature (Number:Temperature)
Forecasted outdoor temperature.

Minimum Temperature
forecastHours03#min-temperature (Number:Temperature)
Minimum forecasted outdoor temperature.

Maximum Temperature
forecastHours03#max-temperature (Number:Temperature)
Maximum forecasted outdoor temperature.

Forecasted Apparent Temperature
forecastHours03#apparent-temperature (Number:Temperature)
Forecasted apparent temperature.

Forecasted Pressure
forecastHours03#pressure (Number:Pressure)
Forecasted barometric pressure.

Forecasted Humidity
forecastHours03#humidity (Number:Dimensionless)
Forecasted atmospheric relative humidity.

Forecasted Wind Speed
forecastHours03#wind-speed (Number:Speed)
Forecasted wind speed.

Forecasted Wind Direction
forecastHours03#wind-direction (Number:Angle)
Forecasted wind direction expressed as an angle.

Forecasted Cloudiness
forecastHours03#cloudiness (Number:Dimensionless)
Forecasted cloudiness.

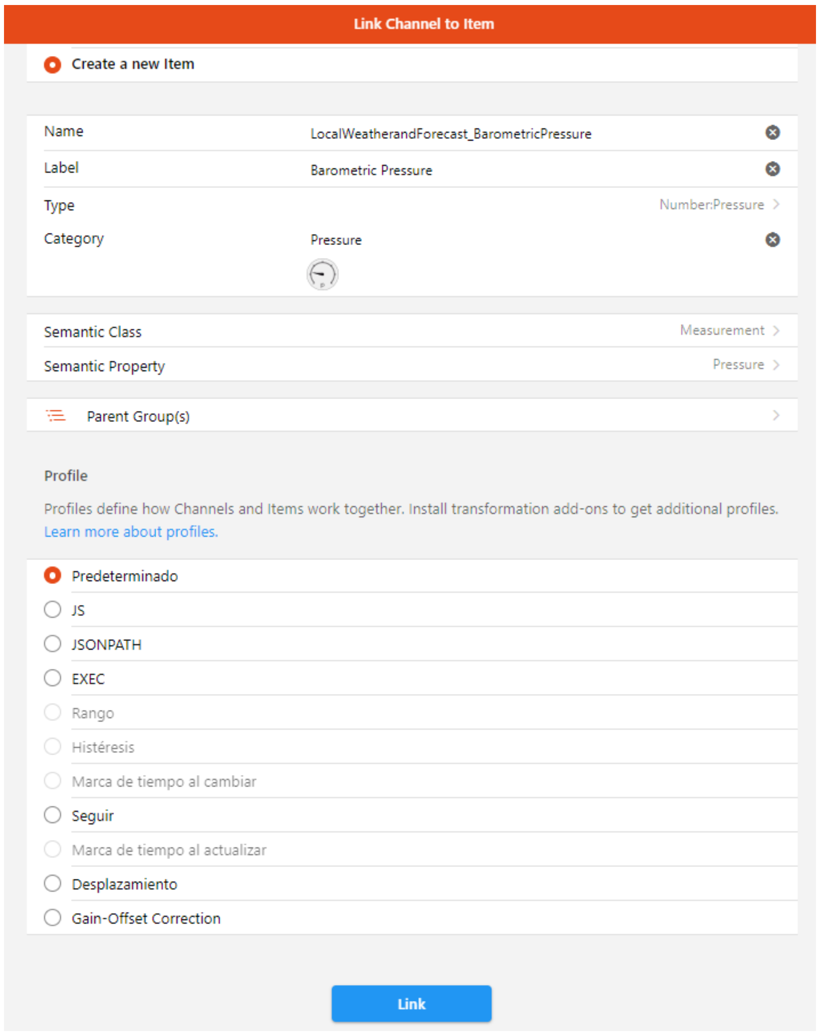
Forecasted Rain
forecastHours03#rain (Number:Length)
Forecasted rain volume.

Forecasted Snow
forecastHours03#snow (Number:Length)
Forecasted snow volume.

Figure 5.82 – *Some channels available 2*

2. It will redirect us to another page that will give us the option to link this channel to an **existing item** or to **create an item from zero**. In the item we have to put a unique id, a label, indicate the type of the variable if it is a number, a string, a switch, etc, we can also put an icon to the item. Also,

indicate the semantic class Measurement, the semantic property that corresponds to the variable and the type of profile in default.


Link Channel to Item

Create a new Item [✓]

Name	LocalWeatherandForecast_BarometricPressure
Label	Barometric Pressure
Type	Number:Pressure
Category	Pressure
Semantic Class	Measurement
Semantic Property	Pressure

Parent Group(s)

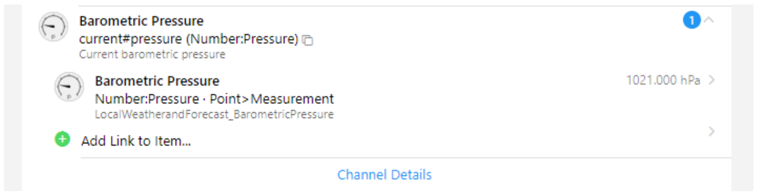
Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles. Learn more about profiles.

- [✓ Predeterminado]
- JS
- JSONPATH
- EXEC
- Rango
- Histéresis
- Marca de tiempo al cambiar
- Seguir
- Marca de tiempo al actualizar
- Desplazamiento
- Gain-Offset Correction

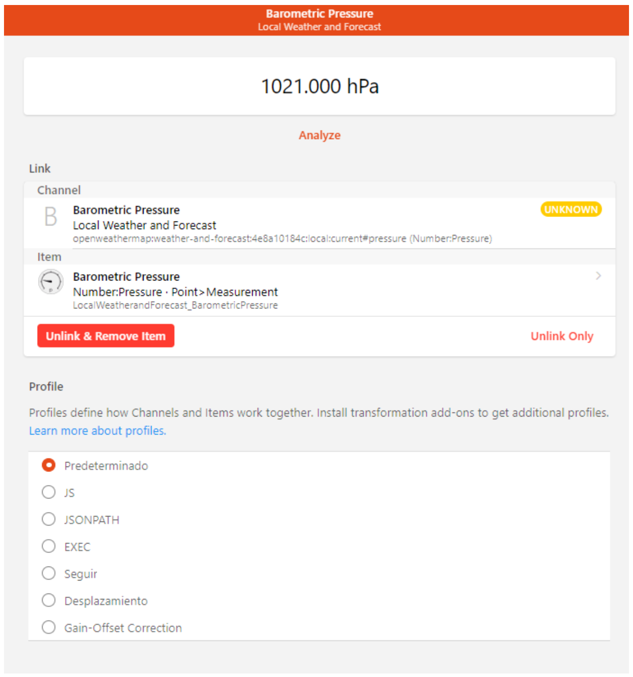
Figure 5.83 – *Create item*

3. And finally the **item will be linked to the chosen channel**. If we access it, it will show us information about the current value of the item, i.e. the weather variable, the channel from which it has been created and the item created, in case we want to edit its parameters.


Barometric Pressure
current#pressure (Number:Pressure)
Current barometric pressure

Barometric Pressure
Number:Pressure · Point>Measurement
LocalWeatherandForecast_BarometricPressure
1021.000 hPa

Figure 5.84 – *Created item*


Barometric Pressure
Local Weather and Forecast

1021.000 hPa

Analyze

Link

Channel

Barometric Pressure
Local Weather and Forecast
openweathermap:weather-and-forecast:4e8a10184c:local:current#pressure (Number:Pressure)
UNKNOWN

Item

Barometric Pressure
Number:Pressure · Point>Measurement
LocalWeatherandForecast_BarometricPressure

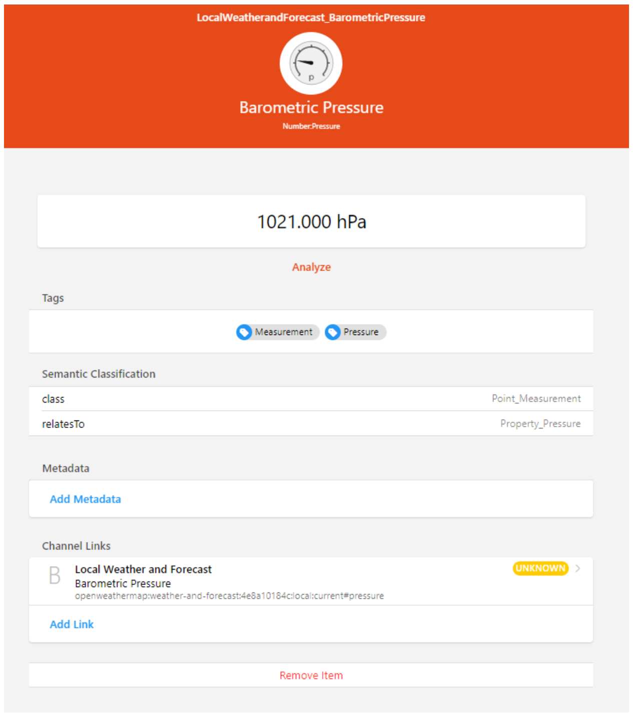
Unlink & Remove Item
Unlink Only

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

- [✓ Predeterminado]
- JS
- JSONPATH
- EXEC
- Seguir
- Desplazamiento
- Gain-Offset Correction

(a) *Link info*


LocalWeatherandForecast_BarometricPressure

Barometric Pressure
Number:Pressure

1021.000 hPa

Analyze

Tags

Measurement
Pressure

Semantic Classification

class	Point_Measurement
relatesTo	Property_Pressure

Metadata

Add Metadata

Channel Links

Local Weather and Forecast
Barometric Pressure
openweathermap:weather-and-forecast:4e8a10184c:local:current#pressure
UNKNOWN

Add Link

Remove Item

(b) *Item info*

Figure 5.85 – *Display information of item*

5.7 Sonoff Mini Zigbee Relays

In this section we will see how to include the Zigbee smart relays chosen in the section 3.4. These relays will be dedicated to turn on and off electrical circuits taking advantage of traditional devices. For this case, they will be installed in electrical circuits to control the lights in a room. The electrical installation of the device can be seen in section 4.3.

5.7.1 Add and configure Zigbee hub with your App

As already mentioned in the section 3.4, Zigbee devices need a Zigbee Hub to work. So the first thing to add to the application is the Zigbee hub. Our chosen Zigbee hub is from the Tuya brand and its corresponding application is Smart Life.

Figure 5.86 – *Tuya Zigbee hub*

Normally Zigbee devices must go to the hub of the same brand to **avoid incompatibilities**, but sometimes they can be compatible between different brands as in our case. To add the hub, follow the steps below:

1. Download the **Smart Life** application from the Play Store or Apple Store, depending on the operating system of your mobile phone, and register in the application by creating an account.

Figure 5.87 – *Smart Life logo*

2. Connect the hub to the power supply and turn it on by pressing the **back button** until the blue LED stays steady and the red LED flashes slowly, or it can also flash quickly (but I couldn't get the pairing to work this way), to activate the pairing mode.

3. Once the **red LED is flashing**, we go into the Smart Life application, click on the **+ icon** in the top right corner and then add device. We go to the **Gateway Control** section and select the first Zigbee model, as I have had problems with the others with the pairing.

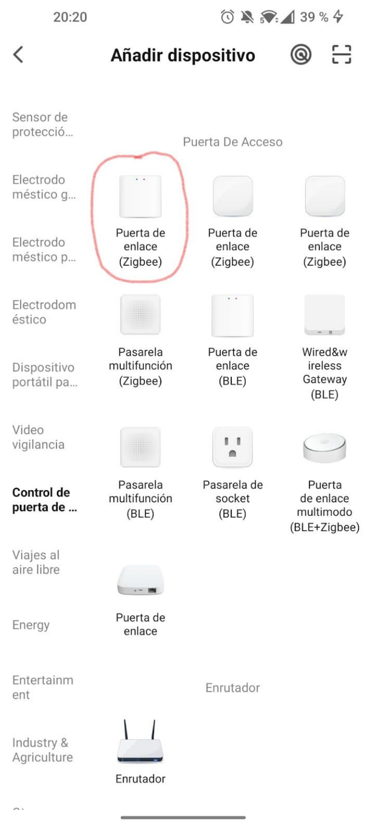
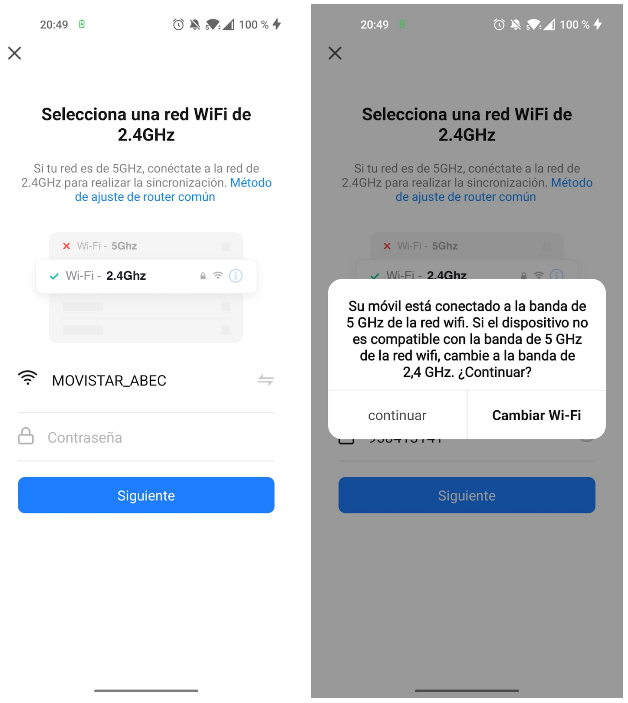


Figure 5.88 – *Select device*

4. Once the gateway model has been selected, follow the steps indicated by the application to connect the device:

   a. Enter the password of the WiFi network to which you are connected, a warning will appear if you are connected to a 5 GHz network, as it indicates that the WiFi network has a frequency of 2.4 GHz.


20:49 100 %

Selecciona una red WiFi de 2.4GHz

Si tu red es de 5GHz, conéctate a la red de 2.4GHz para realizar la sincronización. Método de ajuste de router común

Wi-Fi - 5Ghz
Wi-Fi - 2.4Ghz

MOVISTAR_ABEC

Contraseña

Siguiente

20:49 100 %

Selecciona una red WiFi de 2.4GHz

Si tu red es de 5GHz, conéctate a la red de 2.4GHz para realizar la sincronización. Método de ajuste de router común

Wi-Fi - 5Ghz
Wi-Fi - 2.4Ghz

Su móvil está conectado a la banda de 5 GHz de la red wifi. Si el dispositivo no es compatible con la banda de 5 GHz de la red wifi, cambie a la banda de 2,4 GHz. ¿Continuar?

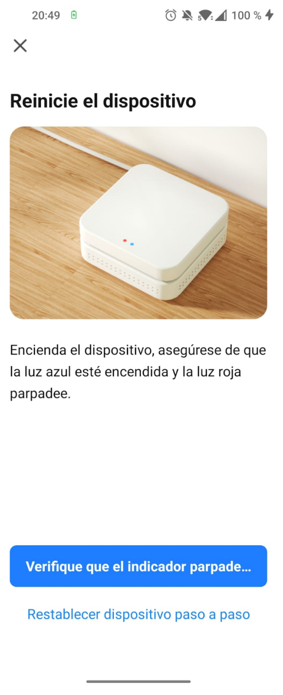
continuar | Cambiar Wi-Fi

Siguiente

(a) (b)

Figure 5.89 – *Connect WiFi*

b. We verify that the device is still in pairing mode (the red LED is flashing and the blue LED is steady) we indicate the speed at which the LED flashes so that the application activates the best pairing protocol. In my case, I select slow flashing.


20:49 100 %

Reinicie el dispositivo

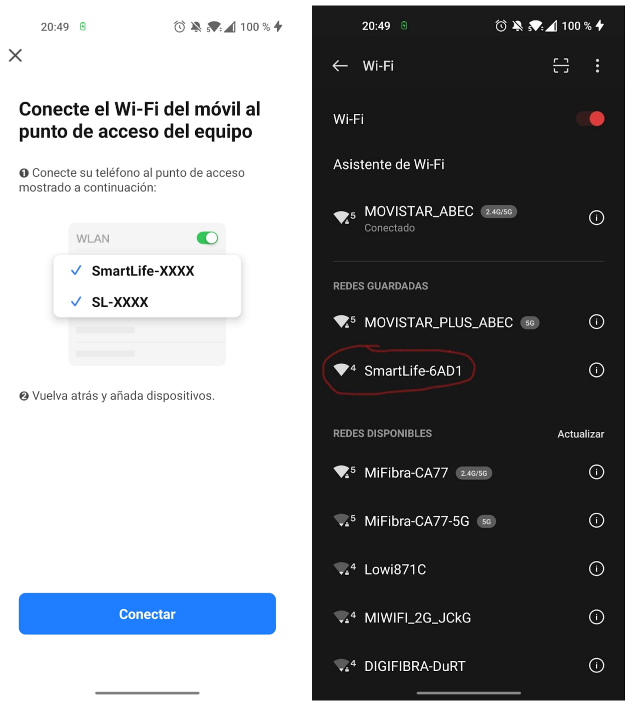
Encienda el dispositivo, asegúrese de que la luz azul esté encendida y la luz roja parpadee.

Verifique que el indicador parpade...

Restablecer dispositivo paso a paso

Figure 5.90 – *Check blinking*

c. For pairing with **slow flashing**, the hub creates a WiFi network to which we have to connect.



20:49 100 %

Conecte el Wi-Fi del móvil al punto de acceso del equipo

❶ Conecte su teléfono al punto de acceso mostrado a continuación:

WLAN
✓ SmartLife-XXXX
✓ SL-XXXX

❷ Vuelva atrás y añada dispositivos.

Conectar

20:49 100 %

Wi-Fi

Wi-Fi

Asistente de Wi-Fi

MOVISTAR_ABEC 2.4G/5G
Conectado

REDES GUARDADAS

MOVISTAR_PLUS_ABEC 5G

SmartLife-6AD1

REDES DISPONIBLES Actualizar

MiFibra-CA77 2.4G/5G

MiFibra-CA77-5G 5G

Lowi871C

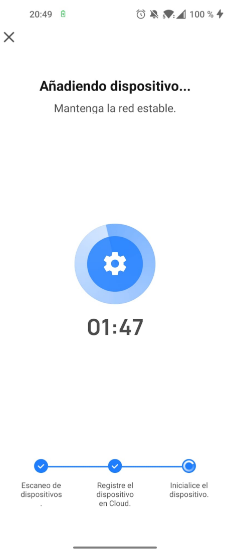
MIWIFI_2G_JCkG

DIGIFIBRA-DuRT

(a) (b)

Figure 5.91 – *Connect WiFi*

d. Once we have connected to the device's network, we go back to the application and the pairing will start automatically.



20:49 100 %

Añadiendo dispositivo...

Mantenga la red estable.

01:47

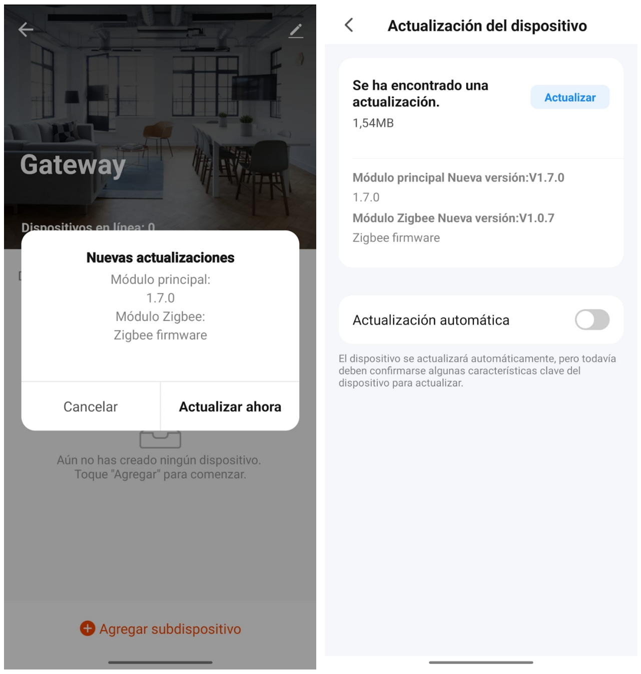
Escaneo de dispositivos

Registre el dispositivo en Cloud.

Inicialice el dispositivo.

Figure 5.92 – *Search device*

e. Once pairing is complete, it will check for firmware updates for the device.


Gateway

Dispositivos en línea: 0

Nuevas actualizaciones

Módulo principal:
1.7.0
Módulo Zigbee:
Zigbee firmware

Cancelar | Actualizar ahora

Aún no has creado ningún dispositivo.
Toque "Agregar" para comenzar.

Agregar subdispositivo

Actualización del dispositivo

Se ha encontrado una actualización.
1,54MB

Actualizar

Módulo principal Nueva versión:V1.7.0
1.7.0
Módulo Zigbee Nueva versión:V1.0.7
Zigbee firmware

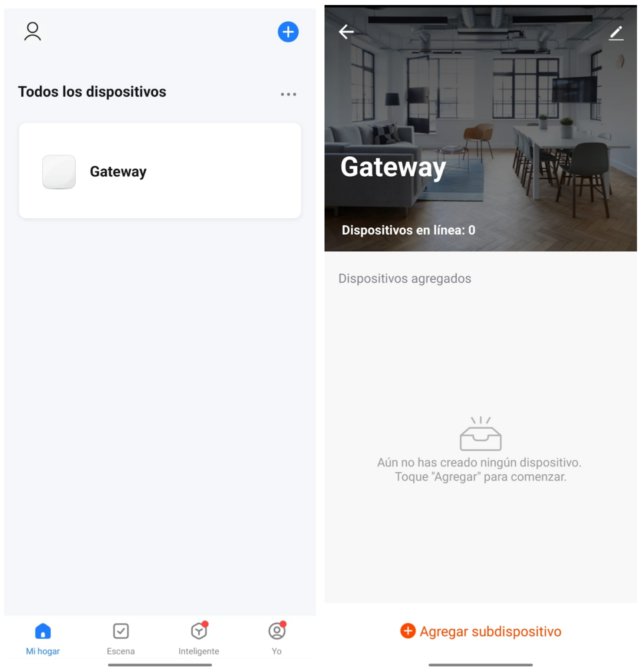
Actualización automática

El dispositivo se actualizará automáticamente, pero todavía deben confirmarse algunas características clave del dispositivo para actualizar.

(a) **(b)**

Figure 5.93 – *Update firmware*

5. Once the Zigbee hub is added, the red light will turn solid red, and we can add devices to it.


Todos los dispositivos

Gateway

Gateway

Dispositivos en línea: 0

Dispositivos agregados

Aún no has creado ningún dispositivo.
Toque "Agregar" para comenzar.

Agregar subdispositivo

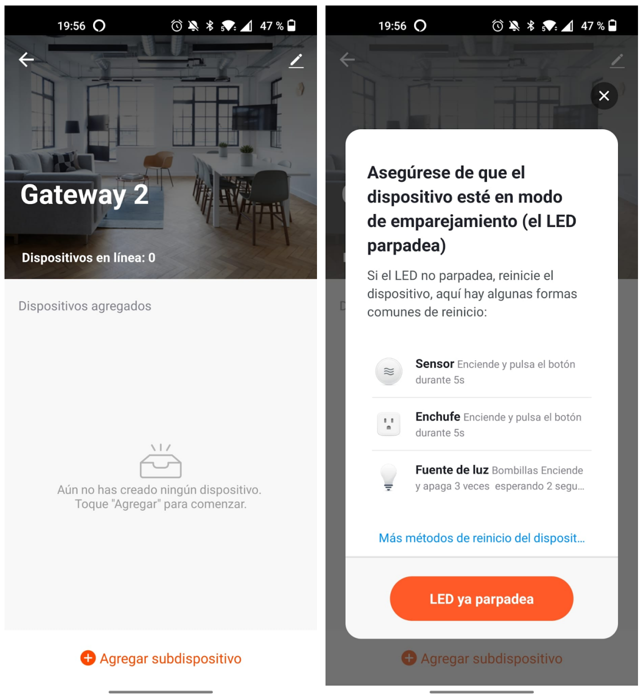
(a) **(b)**

Figure 5.94 – *Hub Connected*

5.7.2 Adding and configuring sonoff mini with Zigbee hub

For the installation of the sonoff mini in the **electrical circuit** we can see the electrical diagram in the figure 4.4. Once installed in the electrical circuit, it will have power and the internal LED will start flashing and the pairing mode will be activated. The device has to be connected to the Zigbee hub in its respective application, in this case Smart Life, to do so:

- Open the **Smart Life** application and **select the hub** we have previously added and click on **add device**. A warning will pop up to make sure that the device we want to connect is **flashing**, indicating that it is in pairing mode.



Aún no has creado ningún dispositivo.
Toque "Agregar" para comenzar.

Asegúrese de que el dispositivo esté en modo de emparejamiento (el LED parpadea)

Si el LED no parpadea, reinicie el dispositivo, aquí hay algunas formas comunes de reinicio:

Sensor Enciende y pulsa el botón durante 5s

Enchufe Enciende y pulsa el botón durante 5s

Fuente de luz Bombillas Enciende y apaga 3 veces esperando 2 segu...

Más métodos de reinicio del disposit...

(a) (b)

Figure 5.95 – *Add sonoff on hub*

- It will start searching for the device, recognise it and add it to the hub.

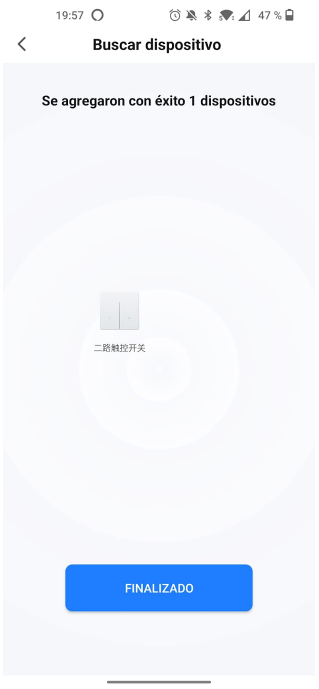
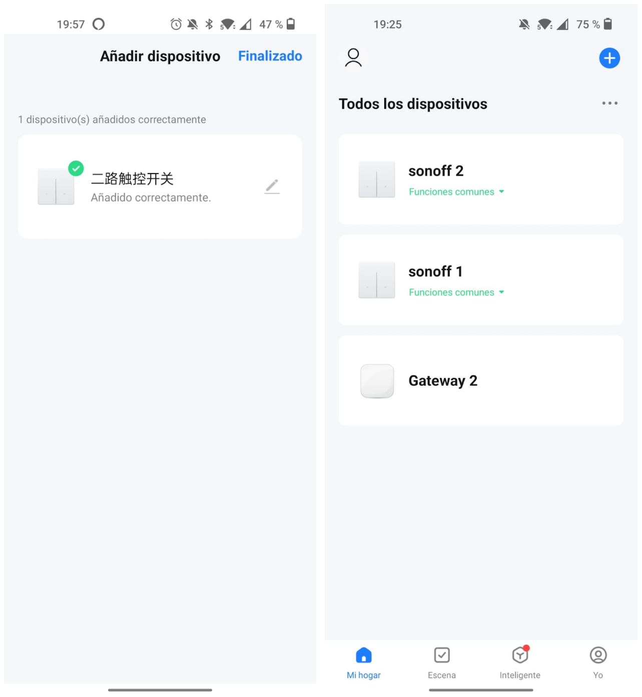


Figure 5.96 – *Search device*

- Once the device has been found and linked, it will be placed in the hub from which it was discovered. We can change the name, as mine, for example, came with the name in Chinese.



(a) **(b)**
Figure 5.97 – *Sonoff instaled*

- Inside the **hub**, by **clicking on the device**, we can access its controls, which are two buttons and a timer, to programme when to turn the light on or off, but the two buttons have the same function.

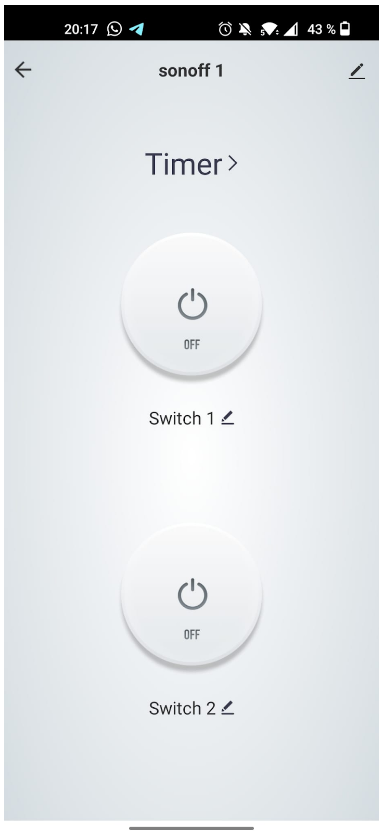
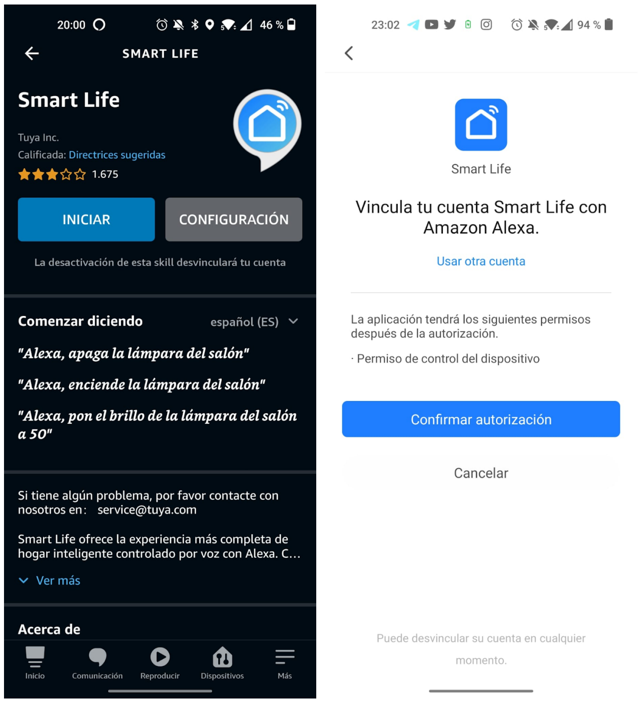


Figure 5.98 – *Device functions*

These devices are also compatible with Amazon Alexa, to connect them to it we must install the **Smart Life skill** in the Amazon Alexa application:

- Enter the Amazon Alexa app.
- Go to more, in the bottom **menu -> skills and games**
- **Search for Smart Life** and **install the skill**, it will ask us to log in with the account we have created in Smart Life previously and it will ask us to search for devices.
- Once found it will add them to the devices as switches. It will create three devices: a general one with the device name, one for switch 1 and one for switch 2.



Smart Life

Tuya Inc.

Calificada: Directrices sugeridas

1.675

La desactivación de esta skill desvinculará tu cuenta

Comenzar diciendo español (ES)

"Alexa, apaga la lámpara del salón"

"Alexa, enciende la lámpara del salón"

"Alexa, pon el brillo de la lámpara del salón a 50"

Si tiene algún problema, por favor contacte con nosotros en: service@tuya.com

Smart Life ofrece la experiencia más completa de hogar inteligente controlado por voz con Alexa. C...

Acerca de

Vincula tu cuenta Smart Life con Amazon Alexa.

La aplicación tendrá los siguientes permisos después de la autorización.

· Permiso de control del dispositivo

Puede desvincular su cuenta en cualquier momento.

(a) (b)

Figure 5.99 – *Install skill Smart Life on Amazon Alexa*

5.7.3 Install and configure binding

For this case, we are going to reuse the **Amazon Echo Control binding** that we have already installed in the 5.4 section. Since there is **no specific binding for sonoff or smart life**, we have opted for this option as it is compatible with alexa.

There was another way but you had to flash the device to change the firmware so that it could be recognised and added to OpenHAB via MQTT. But there was a risk that the device would be damaged and become unusable.

5.7.4 Create and configure things

As we said, we are going to use the Amazon Echo Control binding that we already had installed in the 5.4 section. As we explained before, the things will be auto-discovered in the **Settings->Things-> Inbox tab or from the Things -> + -> Amazon Echo Control -> Scan tab**, this will perform a scan to autodiscover the new things added in the Amazon Alexa app thanks to the Smart Life skill.

It will detect 3 new devices, all of them being of the switch type:

- **Device name**, in my case sonoff 1
- **Switch 1**, name of the first button.
- **Switch 2**, name of the second button.

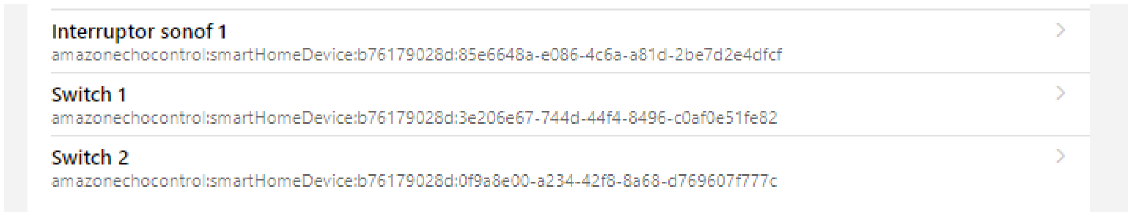

Interruptor sonof 1
amazonechocontrol:smartHomeDevice:b76179028d:85e6648a-e086-4c6a-a81d-2be7d2e4dfcf
Switch 1
amazonechocontrol:smartHomeDevice:b76179028d:3e206e67-744d-44f4-8496-c0af0e51fe82
Switch 2
amazonechocontrol:smartHomeDevice:b76179028d:0f9a8e00-a234-42f8-8a68-d769607f777c

Figure 5.100 – *Autodiscovered things*

Adding one of the 3 is enough, as they do exactly the same thing. We add the thing we want from the 3 to the system, I have used the one with the device name. Once discovered the thing, we add it to the system any of the 3 discovered, I have used the one with the complete name. To add it to the system, click on the **sonoff thing** and a window will pop up, click on **"Add as a Thing"**, and it will be added to the system by itself, as explained in the section 5.4.

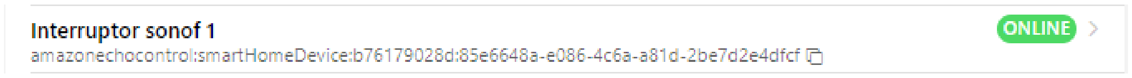

Interruptor sonof 1
amazonechocontrol:smartHomeDevice:b76179028d:85e6648a-e086-4c6a-a81d-2be7d2e4dfcf
ONLINE

Figure 5.101 – *Thing online*

5.7.5 Create and configure items

Once the thing is created and configured, we create the corresponding item to turn the lights on and off. It is created in the same way as we have seen in the other sections of creating items.

1. From the **Settings-> Things tab**, click on the **thing we have created**, go to the **channels tab**, click on the **Power State** channel, and click on **"Link to item"**.

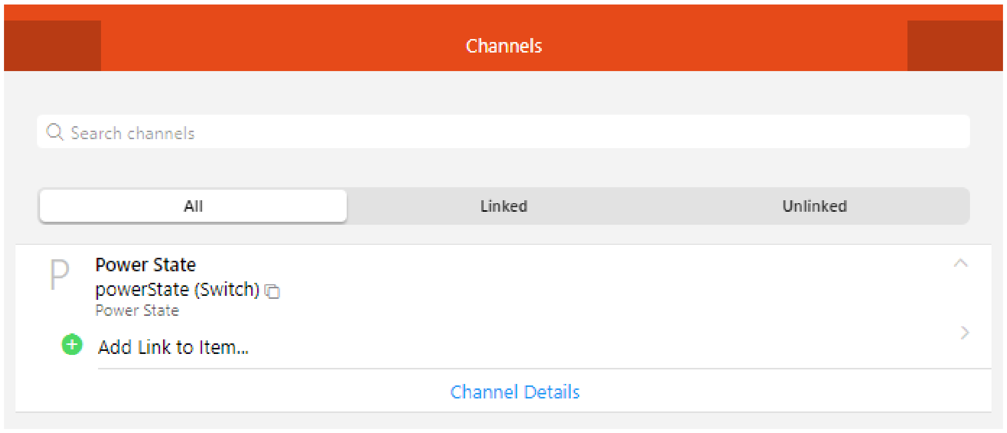
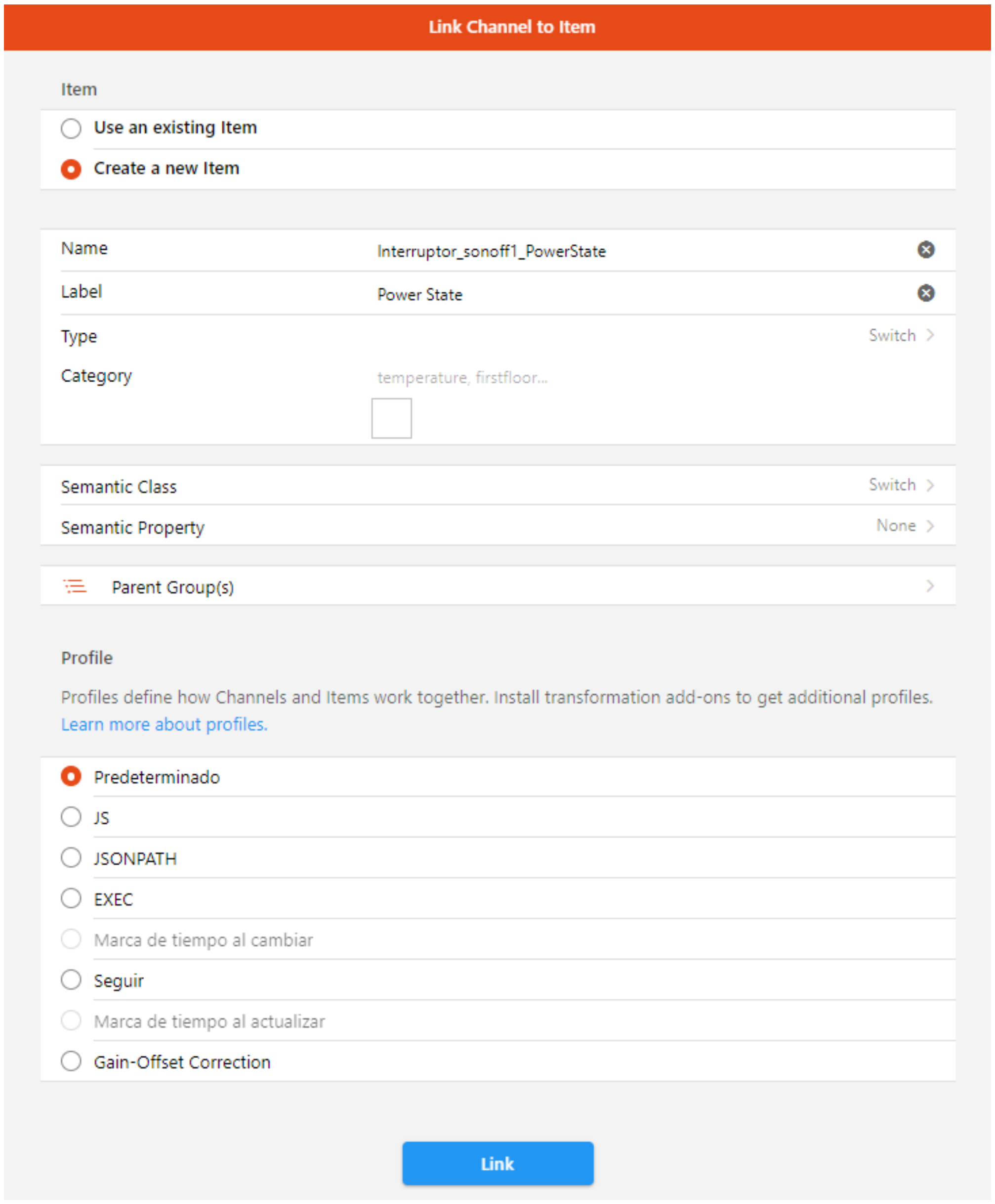


Figure 5.102 – *Channels*

2. It will redirect us to another page that will give us the option to **link this channel** to an **existing item** or to **create an item from zero**. In the item we have to put a unique id, a label, indicate the type of the variable if it is a number, a string, a switch, etc, we can also put an icon to the item. Also, indicate the semantic class Switch.


Link Channel to Item

Item

Use an existing Item [✓ Create a new Item]

Name	Interruptor_sonoff1_PowerState
Label	Power State
Type	Switch
Category	temperature, firstfloor...
Semantic Class	Switch
Semantic Property	None

Parent Group(s)

Profile

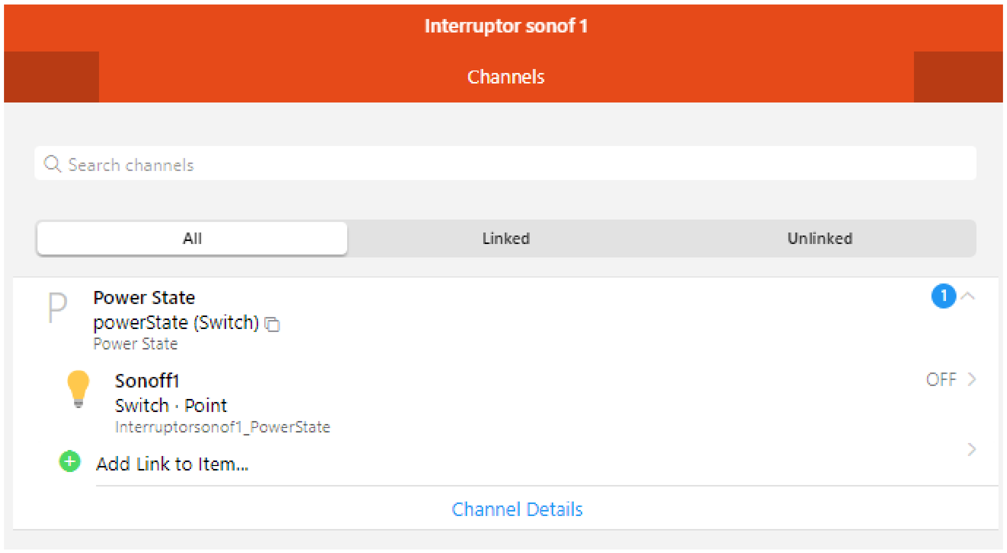
Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles. Learn more about profiles.

[✓ Predeterminado] JS JSONPATH EXEC Marca de tiempo al cambiar Seguir Marca de tiempo al actualizar Gain-Offset Correction

Link

Figure 5.103 – *Create item*

3. And finally the item will be **linked to the chosen channel**. If we access it, it will show us information about the current value of the item, i.e. the weather variable, the channel from which it has been created and the item created, in case we want to edit its parameters.



Interruptor sonof 1

Channels

Search channels

All | Linked | Unlinked

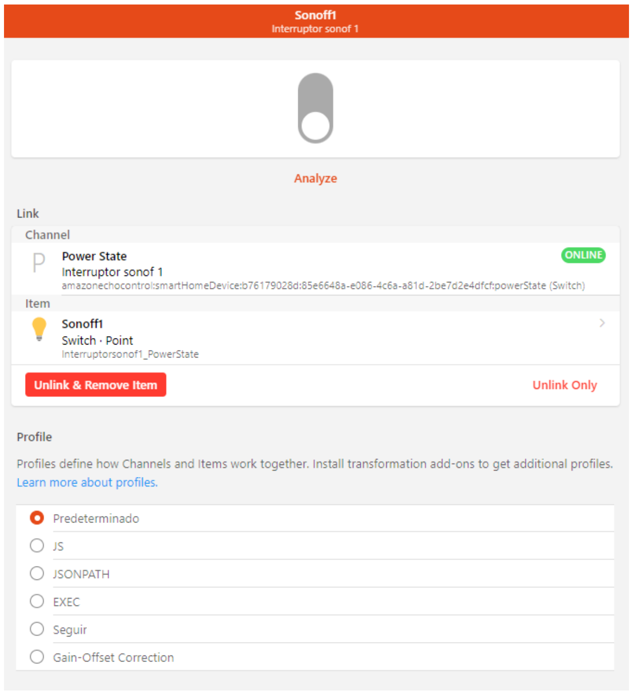
Power State
powerState (Switch)
Power State

Sonoff1 OFF
Switch · Point
Interruptorsonof1_PowerState

Add Link to Item...

Channel Details

Figure 5.104 – *Created Item*



Sonoff1
Interruptor sonof 1

Analyze

Link

Channel

Power State ONLINE
Interruptor sonof 1
amazonechocontrol:smartHomeDevice:b76179028d:85e6648a-e086-4c6a-a81d-2be7d2e4dfcf:powerState (Switch)

Item

Sonoff1
Switch · Point
Interruptorsonof1_PowerState

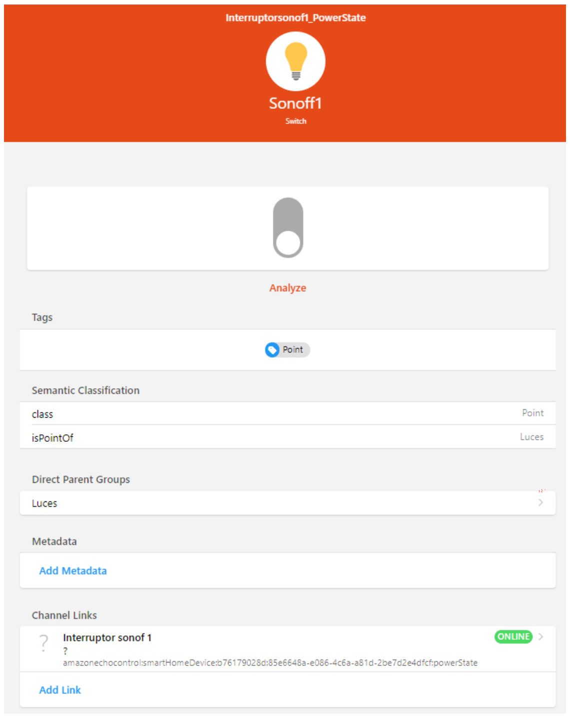
Unlink & Remove Item | Unlink Only

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

- [✓ Predeterminado]
- JS
- JSONPATH
- EXEC
- Seguir
- Gain-Offset Correction

(a) *Link info*



Interruptorsonof1_PowerState

Sonoff1
Switch

Analyze

Tags

Point

Semantic Classification

class	Point
isPointOf	Luces

Direct Parent Groups

Luces

Metadata

Add Metadata

Channel Links

Interruptor sonof 1 ONLINE
?
amazonechocontrol:smartHomeDevice:b76179028d:85e6648a-e086-4c6a-a81d-2be7d2e4dfcf:powerState

Add Link

(b) *Item info*

Figure 5.105 – *Display information of item*

5.8 Weather station

In this section we will see how to connect the **Froggit WH3000 SE** weather station, which was chosen as you can see in the 3.5 section. It was a bit complicated because this model of weather station has a different name depending on the company that imports it.

5.8.1 Add and configure weather station

As mentioned above, the name of the product depends on the distributor who imports it, the different names are:

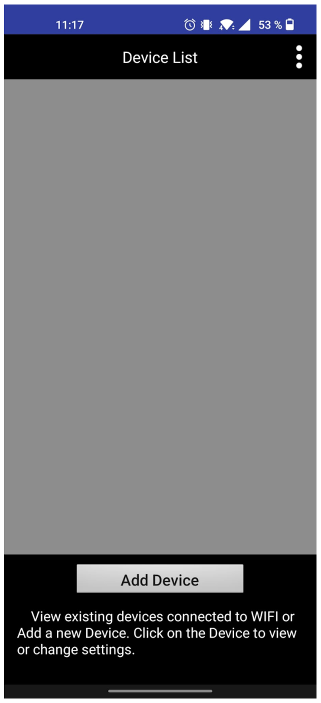
- **Froggit WH3000 SE**
- **Ambient Weather ws-2902**
- **Sainlogic WS3500**

Following the Froggit WH3000 SE manual, I couldn't connect the station to the application, so I tried the other brands' applications, with which I was successful with (Ambient Weather manual).

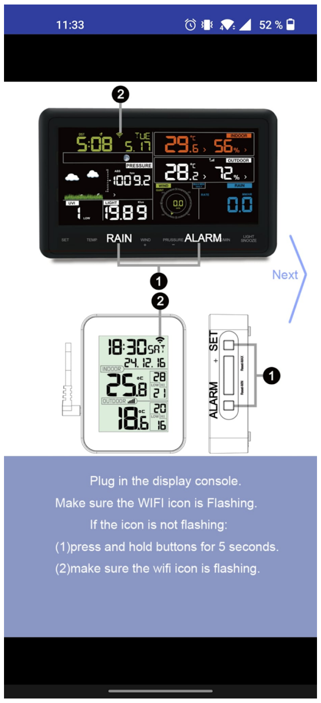
1. Following the section **Connecting the Weather Station Console to WiFi**. Download the **Ambient Weather Osprey Tool application**, which is the application described in the Froggit manual, so we'll go between the two manuals.

Figure 5.106 – *Ambient Weather Osprey Tool icon*

2. Open the application and click on **Add Device** and follow the steps explained in the application:
   - **Connect the console** (station display) to the power supply.
   - Make sure that the **WiFi icon on the console is flashing**. If it is not flashing, press the Rain and Alarm buttons for 5 seconds.



View existing devices connected to WIFI or Add a new Device. Click on the Device to view or change settings.

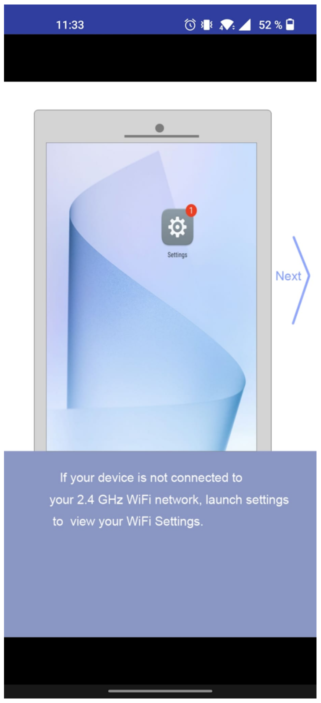


Plug in the display console.
Make sure the WIFI icon is Flashing.
If the icon is not flashing:
(1)press and hold buttons for 5 seconds.
(2)make sure the wifi icon is flashing.

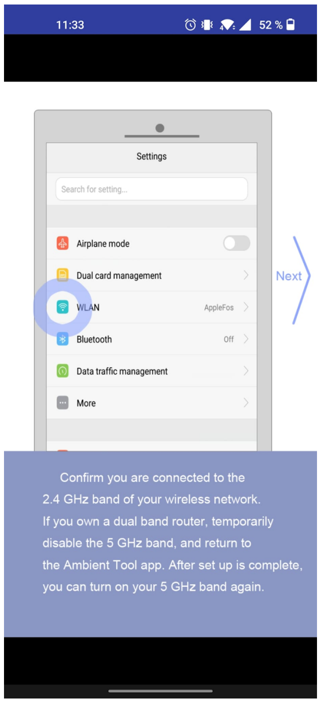
(a) **(b)**

Figure 5.107 – *Add device*

- Connecting the mobile phone to a 2.4 GHz WiFi network.



If your device is not connected to your 2.4 GHz WiFi network, launch settings to view your WiFi Settings.

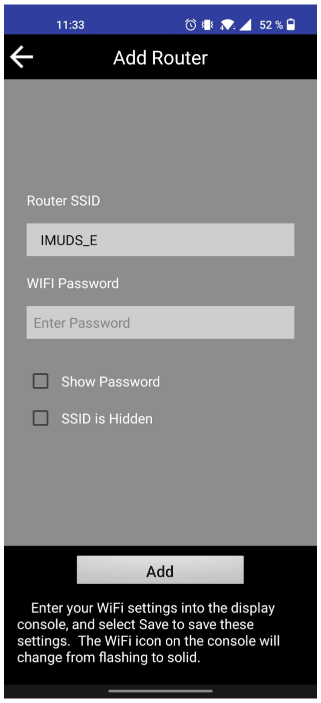


Confirm you are connected to the 2.4 GHz band of your wireless network.
If you own a dual band router, temporarily disable the 5 GHz band, and return to the Ambient Tool app. After set up is complete, you can turn on your 5 GHz band again.

(a) **(b)**

Figure 5.108 – *Connect to WiFi*

- Once everything is done, the application will ask you to enter the **WiFi password** and press Add.


11:33 52 %

Add Router

Router SSID

IMUDS_E

WIFI Password

Enter Password

Show Password

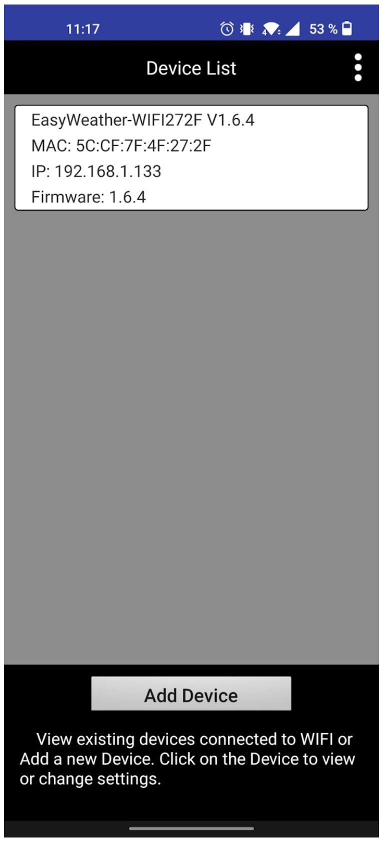
SSID is Hidden

Add

Enter your WiFi settings into the display console, and select Save to save these settings. The WiFi icon on the console will change from flashing to solid.

Figure 5.109 – *Introduce WiFi password*

- After enter the WiFi password, it will start searching for and adding the station.


11:17 53 %

Device List

EasyWeather-WIFI272F V1.6.4
MAC: 5C:CF:7F:4F:27:2F
IP: 192.168.1.133
Firmware: 1.6.4

Add Device

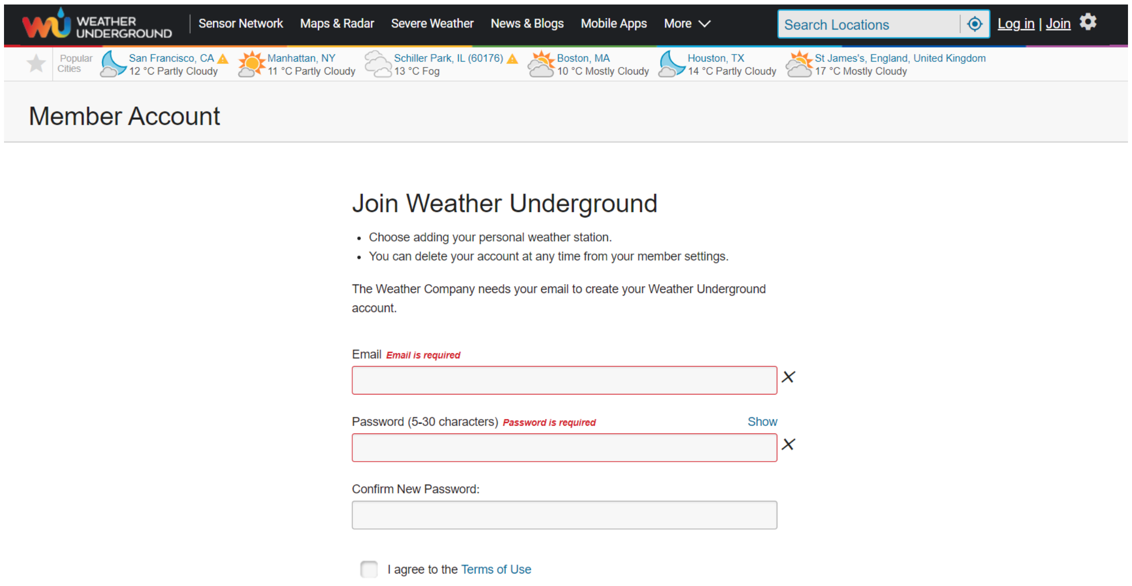
View existing devices connected to WIFI or Add a new Device. Click on the Device to view or change settings.

Figure 5.110 – *Weather station found*

5.8.2 Publish station data to WeatherUnderground

Once the station has been added to the application, the next step is to **publish the station data on a weather website**, so that it can be obtained in OpenHAB. The chosen website is WeatherUnderground, but before publishing the data you have to create an account and add the device:

1. Register at WeatherUnderground.


WEATHER UNDERGROUND
Sensor Network
Maps & Radar
Severe Weather
News & Blogs
Mobile Apps
More
Search Locations
Log in | Join
Popular Cities
San Francisco, CA 12 °C Partly Cloudy
Manhattan, NY 11 °C Partly Cloudy
Schiller Park, IL (60176) 13 °C Fog
Boston, MA 10 °C Mostly Cloudy
Houston, TX 14 °C Partly Cloudy
St James's, England, United Kingdom 17 °C Mostly Cloudy

Member Account

Join Weather Underground

- Choose adding your personal weather station.
- You can delete your account at any time from your member settings.

The Weather Company needs your email to create your Weather Underground account.

Email *Email is required*

Password (5-30 characters) *Password is required* Show

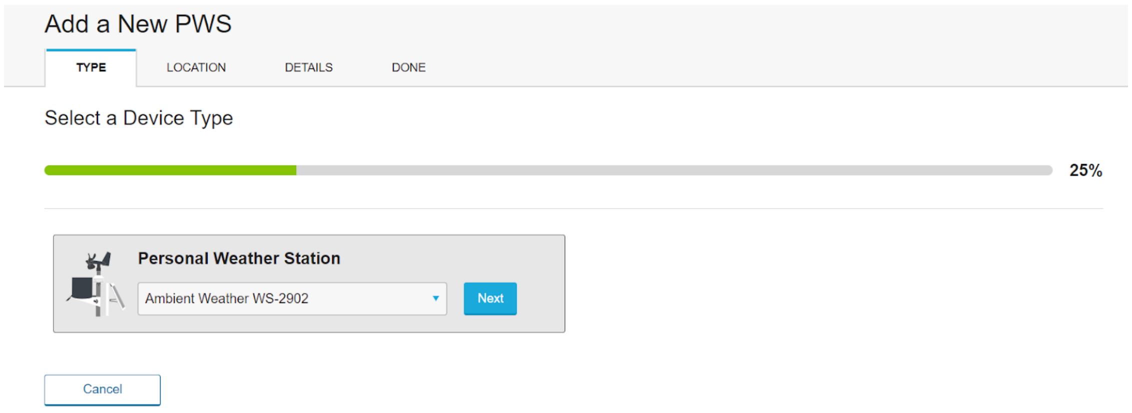
Confirm New Password:

I agree to the Terms of Use

Figure 5.111 – *Register at WeatherUnderground*

2. To add the device go to **My profile -> My Devices -> Add new device** and follow the steps to add the device:

   - **Select the model of the station**, in my case it is a Froggit WH3000 SE, but I am going to register it as the equivalent model of the ambient weather brand, which is the app that has worked for me.



Add a New PWS

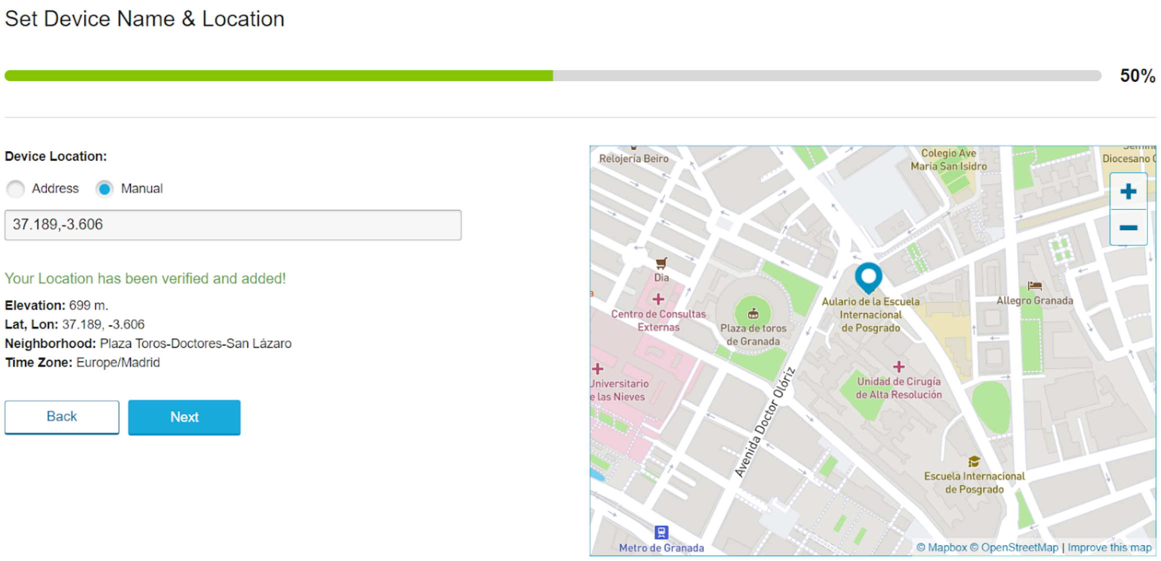
Select a Device Type

Personal Weather Station

Ambient Weather WS-2902

Figure 5.112 – *Select device type*

- We indicate the address where the station is **located**.



Set Device Name & Location

50%

Device Location:

Address [✓ Manual]

37.189,-3.606

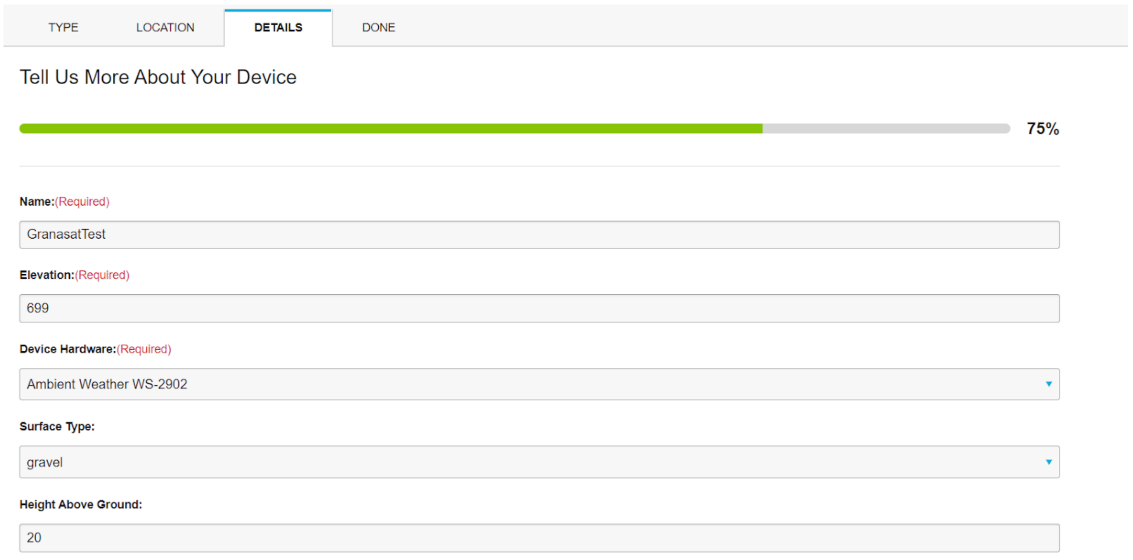
Your Location has been verified and added!

Elevation: 699 m.
Lat, Lon: 37.189, -3.606
Neighborhood: Plaza Toros-Doctores-San Lázaro
Time Zone: Europe/Madrid

Back
Next
Relojería Beiro
Colegio Ave María San Isidro
Dia
Centro de Consultas Externas
Plaza de toros de Granada
Aulario de la Escuela Internacional de Posgrado
Allegro Granada
Universitario las Nieves
Unidad de Cirugía de Alta Resolución
Avenida Doctor Olóriz
Escuela Internacional de Posgrado
Metro de Granada
© Mapbox © OpenStreetMap | Improve this map

Figure 5.113 – *Set location*

- Then we have to **give more information** about the station, such as name, elevation, the material and the height from the ground at which it is placed.



Tell Us More About Your Device

75%

Name:(Required)
GranasatTest

Elevation:(Required)
699

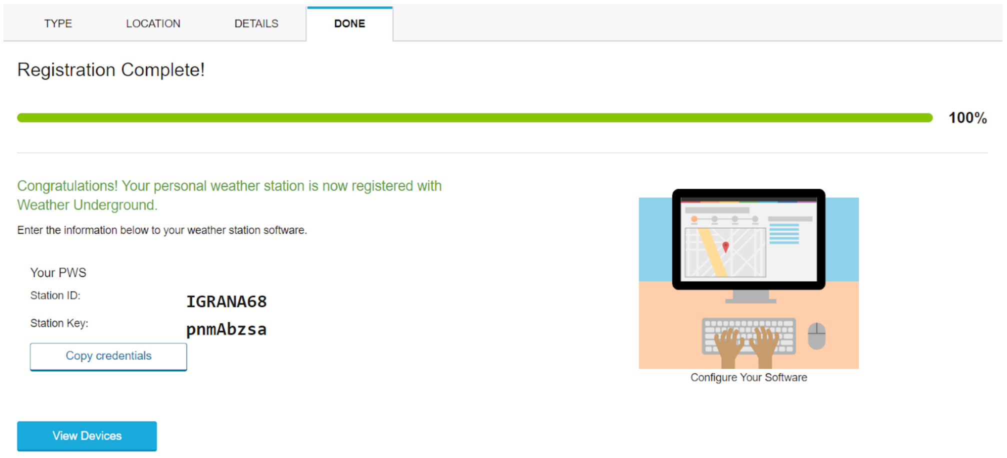
Device Hardware:(Required)
Ambient Weather WS-2902

Surface Type:
gravel

Height Above Ground:
20

Figure 5.114 – *More info about weather station*

- With everything configured, we will get a **station id and a station key (password)**, to identify the device, although the device will be offline as no data is being published yet (the following image is not the real data, for security reasons.).



TYPE LOCATION DETAILS DONE

Registration Complete!

100%

Congratulations! Your personal weather station is now registered with Weather Underground.

Enter the information below to your weather station software.

Your PWS

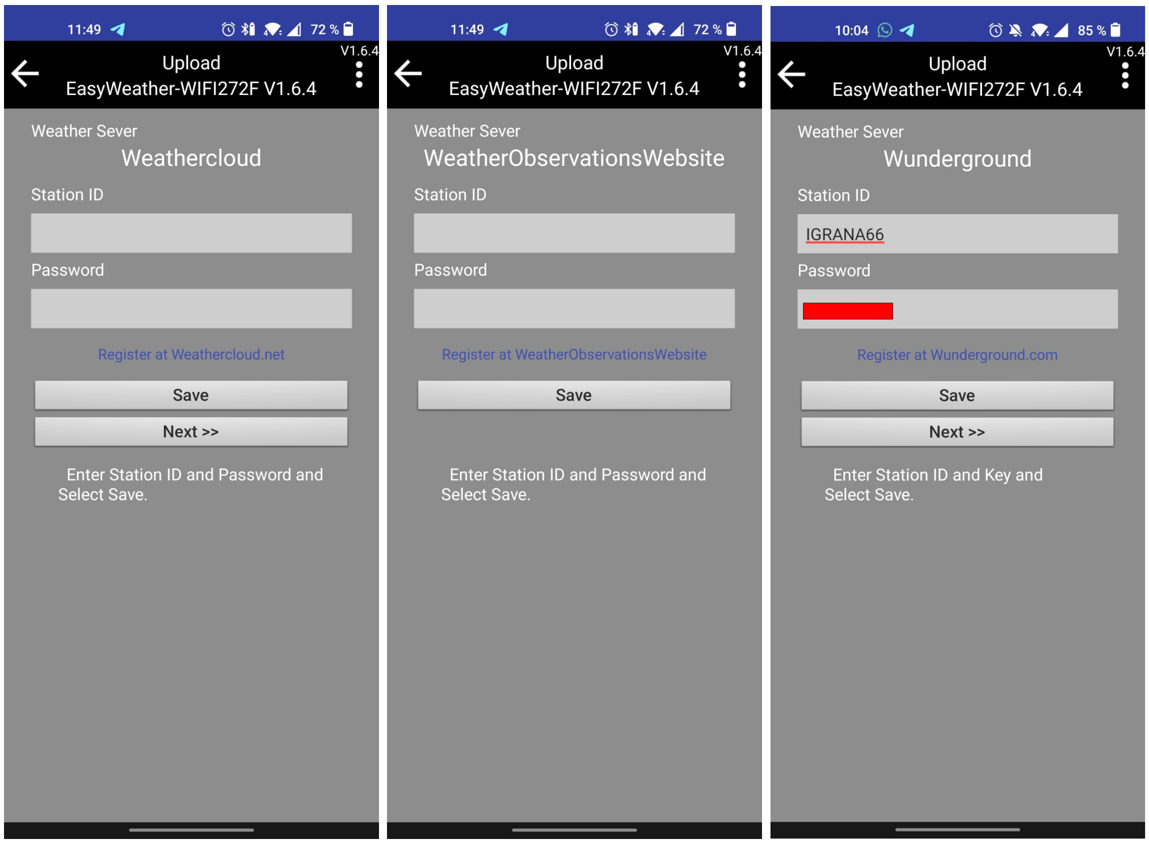
Station ID:	IGRANA68
Station Key:	pnmAbzsa

Copy credentials

View Devices

Figure 5.115 – *Registration complete*

3. Once the device is created, go back to the **Ambient Weather Osprey Tool application** on the mobile, select the device and different websites will appear where you can **upload the data**, press next until WeatherUnderground appears and enter the station ID and password (station key) that we were given when we created the device on the website. After a while the device will appear online on the WeatherUnderground website.



Upload EasyWeather-WIFI272F V1.6.4 — V1.6.4

Weather Sever: Weathercloud

Station ID:

Password:

Register at Weathercloud.net

Save

Next >>

Enter Station ID and Password and Select Save.

Upload EasyWeather-WIFI272F V1.6.4 — V1.6.4

Weather Sever: WeatherObservationsWebsite

Station ID:

Password:

Register at WeatherObservationsWebsite

Save

Enter Station ID and Password and Select Save.

Upload EasyWeather-WIFI272F V1.6.4 — V1.6.4

Weather Sever: Wunderground

Station ID: IGRANA66

Password: [redacted]

Register at Wunderground.com

Save

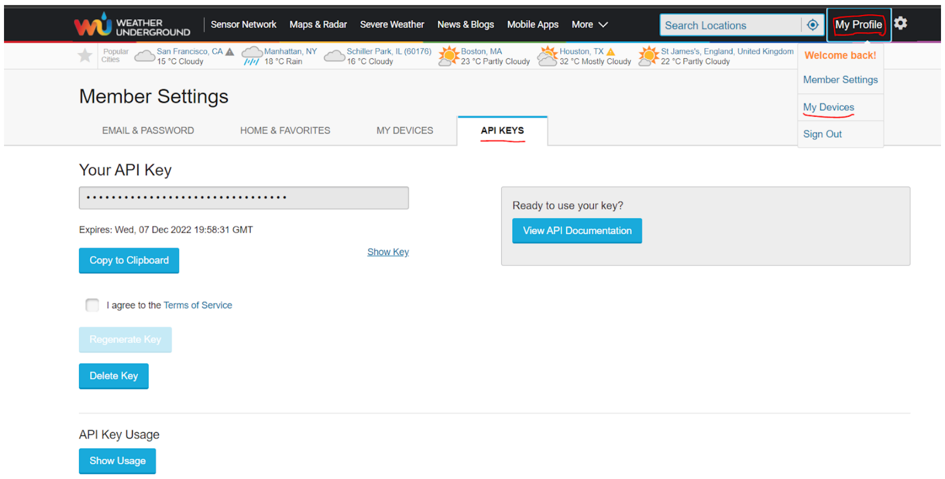
Next >>

Enter Station ID and Key and Select Save.

(a) *Weathercloud* (b) *WeatherObservationsWebsite* (c) *WeatherUnderground*

Figure 5.116 – *Upload data a website*

Once everything is created, we will have to obtain an API key to get the data in OpenHAB through it. To do this we go to **My profile-> My devices -> Api Keys**, by default it will be created automatically when you create the account and add a device.



Expires: Wed, 07 Dec 2022 19:58:31 GMT

Figure 5.117 – *Get API key*

5.8.3 Install and configure binding

To get the data from the WeatherUnderground website, we will have to install a new binding, in this case there were two possibilities:

- Through the **WeatherUnderground binding**.
- Through the **Weather Company binding**.

The WeatherUnderground binding stopped working on 31/12/2018, this binding is no longer supported. So we will use the Weather Company binding, which has the same functionality as the WeatherUnderground binding, this service is only available to PWS users who upload their weather data from PWS to WeatherUnderground using the API.

To install the binding, go to **Settings -> Binding -> Search -> search for Weather Company** and install the binding.

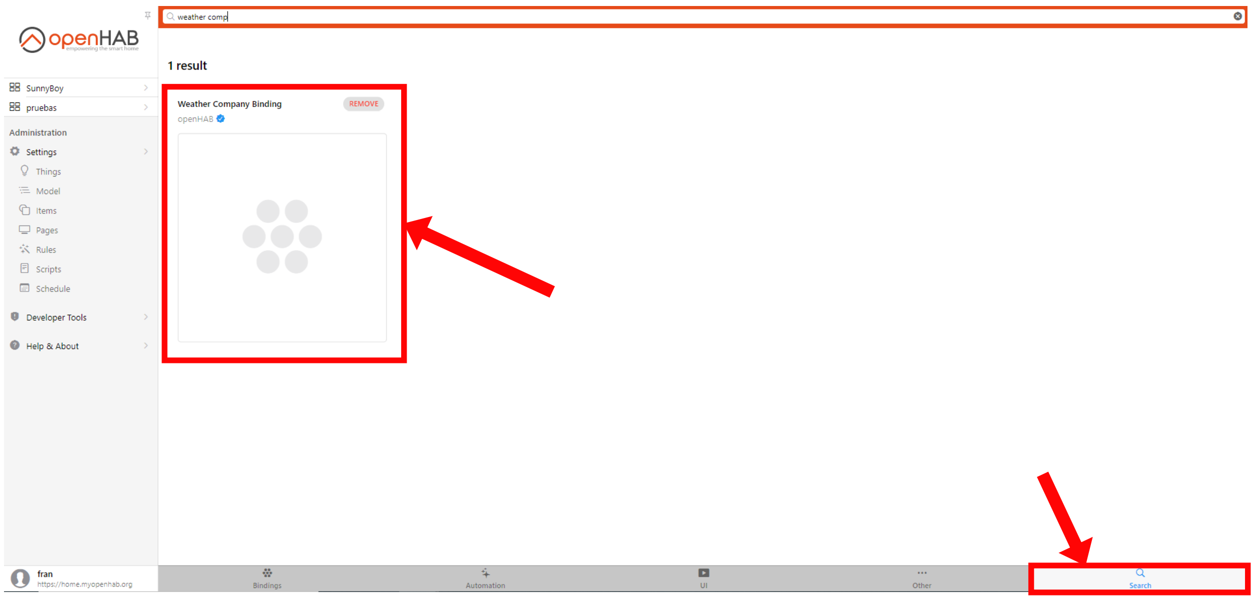
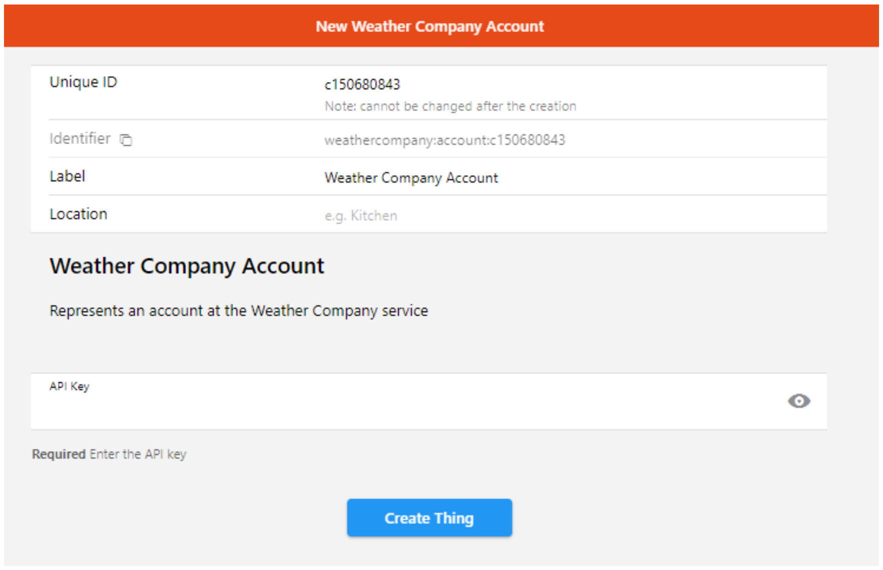


Figure 5.118 – *Install binding*

5.8.4 Create and configure things

Once the binding is installed, go to the **Settings -> Things tab** to start adding and configuring things.

1. As with the other bindings, we also need a thing bridge to connect the two platforms. So from the **Things tab -> + icon -> select the Weather Company binding**. Once inside, we **select Weather Company account** and create the thing putting the API key that we have obtained before.



New Weather Company Account

Unique ID	c150680843 Note: cannot be changed after the creation
Identifier	weathercompany:account:c150680843
Label	Weather Company Account
Location	e.g. Kitchen

Weather Company Account

Represents an account at the Weather Company service

API Key

Required Enter the API key

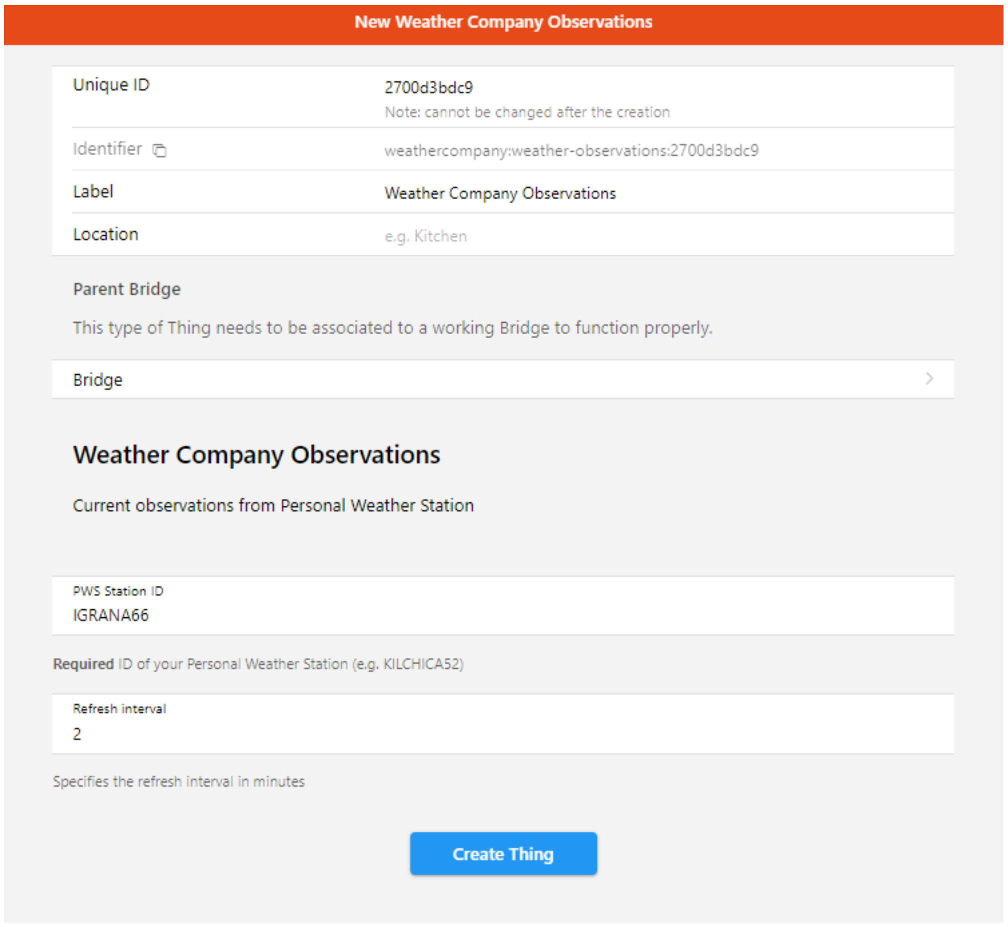
Figure 5.119 – *Create bridge thing*

2. Once the bridge is created, it will autodiscover some things, the current weather (station data) and a forecast weather (like an API) , which can be viewed from the inbox tab or from the binding tab itself

to add things manually. If it is not auto-discovered, we can perform the scan that we have explained previously in the section 5.4.

3. If it still does not appear we can add it manually as we have done the bridge:

   - **Weather Company Observations**, to configure the thing is only to indicate the bridge thing to indicate the WeatherUnderground account, set the ID of the station and the time interval we want to refresh the data.



Unique ID	2700d3bdc9 Note: cannot be changed after the creation
Identifier	weathercompany:weather-observations:2700d3bdc9
Label	Weather Company Observations
Location	e.g. Kitchen

Parent Bridge

This type of Thing needs to be associated to a working Bridge to function properly.

Bridge

Weather Company Observations

Current observations from Personal Weather Station

PWS Station ID
IGRANA66

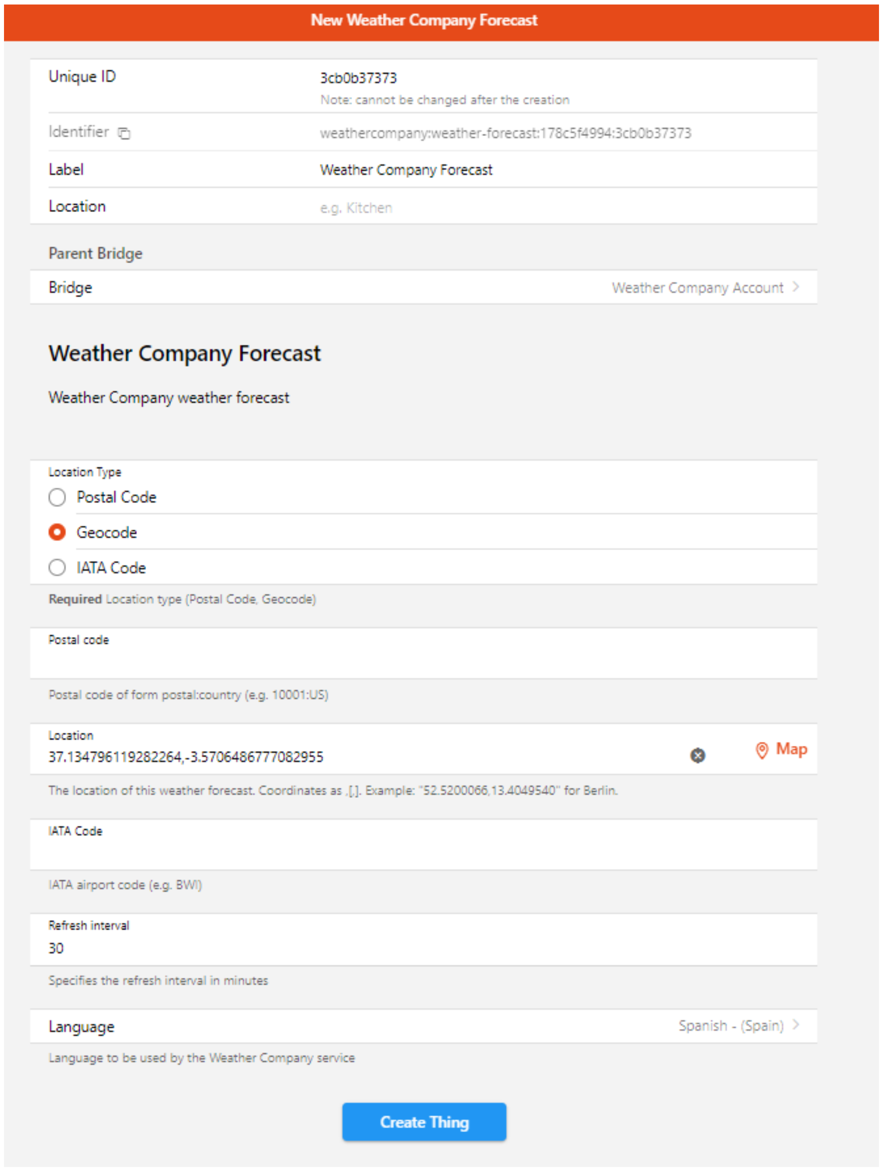
Required ID of your Personal Weather Station (e.g. KILCHICA52)

Refresh interval
2

Specifies the refresh interval in minutes

Figure 5.120 – *Create current weather data thing*

- **Weather Company Forecast**, indicate the account that acts as a bridge, set the geolocation, the time interval how often we want the data to be updated and the language.



New Weather Company Forecast

Unique ID	3cb0b37373 Note: cannot be changed after the creation
Identifier	weathercompany:weather-forecast:178c5f4994:3cb0b37373
Label	Weather Company Forecast
Location	e.g. Kitchen

Parent Bridge

Bridge	Weather Company Account

Weather Company Forecast

Weather Company weather forecast

Location Type
Postal Code [✓ Geocode] IATA Code

Required Location type (Postal Code, Geocode)

Postal code

Postal code of form postal:country (e.g. 10001:US)

Location
37.134796119282264,-3.5706486777082955

The location of this weather forecast. Coordinates as ,[,]. Example: "52.5200066,13.4049540" for Berlin.

IATA Code

IATA airport code (e.g. BWI)

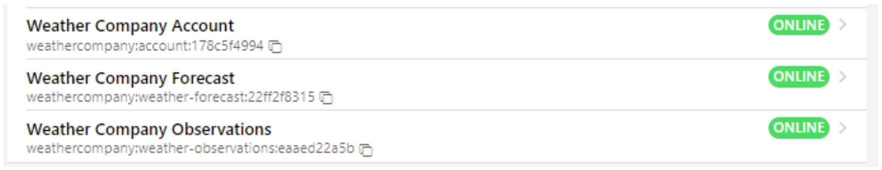
Refresh interval
30

Specifies the refresh interval in minutes

Language: Spanish - (Spain)

Language to be used by the Weather Company service

Figure 5.121 – *Create forecast weather data thing*



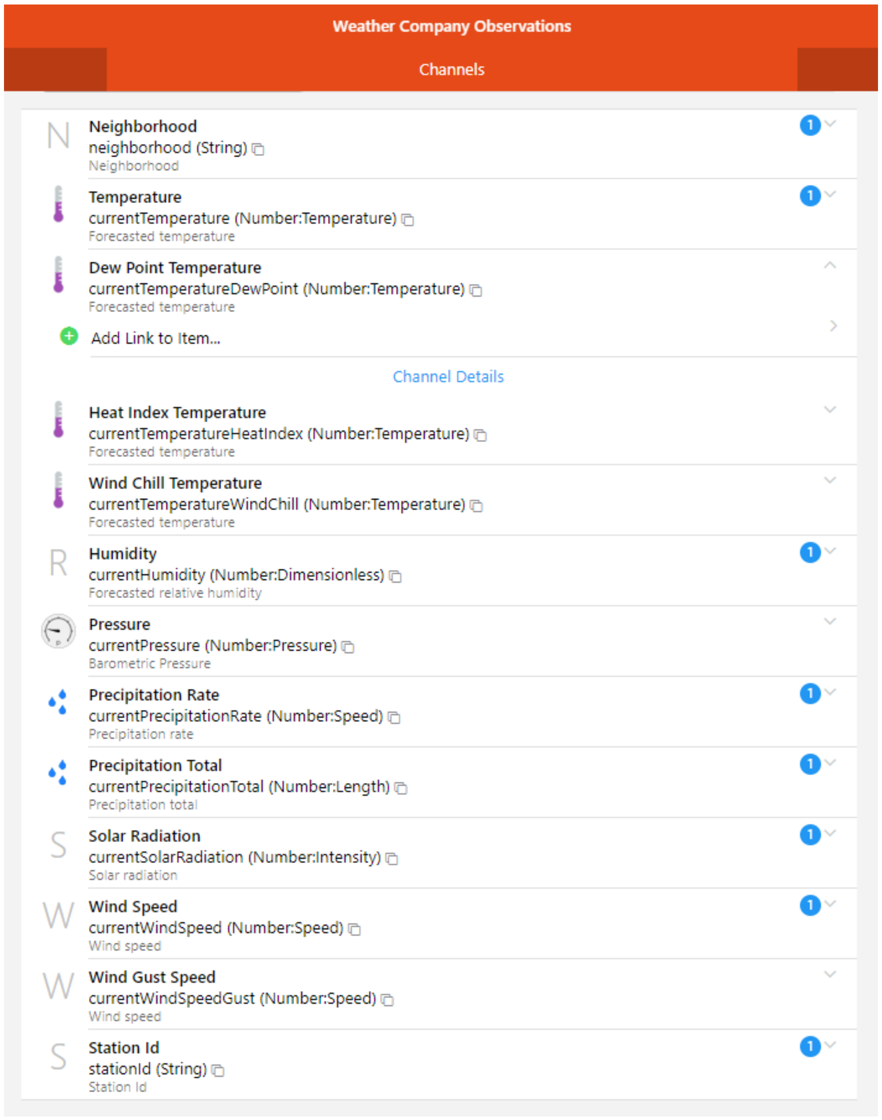
Weather Company Account — weathercompany:account:178c5f4994 — ONLINE
Weather Company Forecast — weathercompany:weather-forecast:22ff2f8315 — ONLINE
Weather Company Observations — weathercompany:weather-observations:eaaed22a5b — ONLINE

Figure 5.122 – *Things online*

5.8.5 Create and configure items

Once the things are created, in order to obtain the meteorological data we access the **"Weather Company Observations"** thing that we have added before, once there in the channels tab, we will see the meteorological measurements that we can obtain and add to the system, according to what is published in WeatherUnderground.

1. From the **channels tab**, click on the **variable** you want to obtain its measurement in the system, and click on **"Link to item"**.


Weather Company Observations
Channels

Neighborhood
neighborhood (String)
Neighborhood

Temperature
currentTemperature (Number:Temperature)
Forecasted temperature

Dew Point Temperature
currentTemperatureDewPoint (Number:Temperature)
Forecasted temperature
Add Link to Item...
Channel Details

Heat Index Temperature
currentTemperatureHeatIndex (Number:Temperature)
Forecasted temperature

Wind Chill Temperature
currentTemperatureWindChill (Number:Temperature)
Forecasted temperature

Humidity
currentHumidity (Number:Dimensionless)
Forecasted relative humidity

Pressure
currentPressure (Number:Pressure)
Barometric Pressure

Precipitation Rate
currentPrecipitationRate (Number:Speed)
Precipitation rate

Precipitation Total
currentPrecipitationTotal (Number:Length)
Precipitation total

Solar Radiation
currentSolarRadiation (Number:Intensity)
Solar radiation

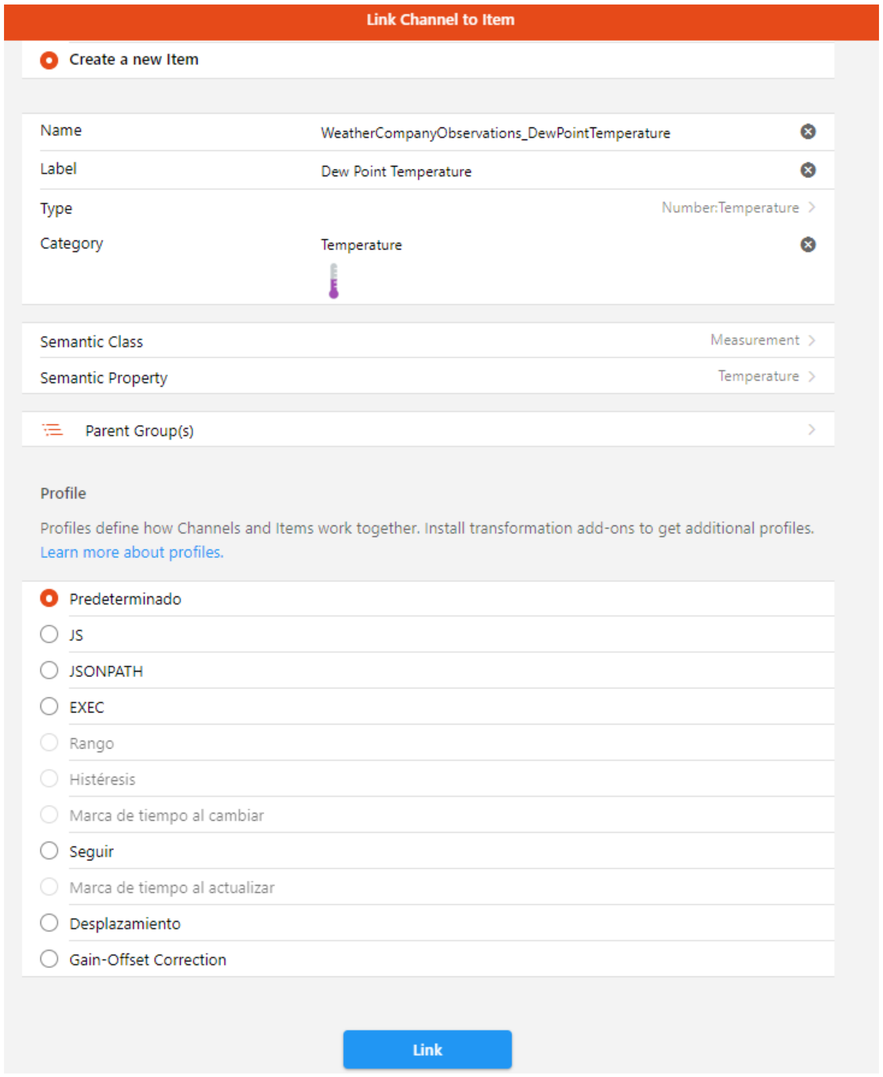
Wind Speed
currentWindSpeed (Number:Speed)
Wind speed

Wind Gust Speed
currentWindSpeedGust (Number:Speed)
Wind speed

Station Id
stationId (String)
Station Id

Figure 5.123 – *Channels*

2. It will redirect us to another page that will give us the option to **link this channel** to an **existing item** or to **create an item from zero**. In the item we have to put a unique id, a label, indicate the type of the variable if it is a number, a string, a switch, etc, we can also put an icon to the item. Also, indicate the semantic class Measurement, the semantic property that corresponds to the variable and the type of profile in default.



Link Channel to Item

[✓ Create a new Item]

Name	WeatherCompanyObservations_DewPointTemperature
Label	Dew Point Temperature
Type	Number:Temperature
Category	Temperature
Semantic Class	Measurement
Semantic Property	Temperature

Parent Group(s)

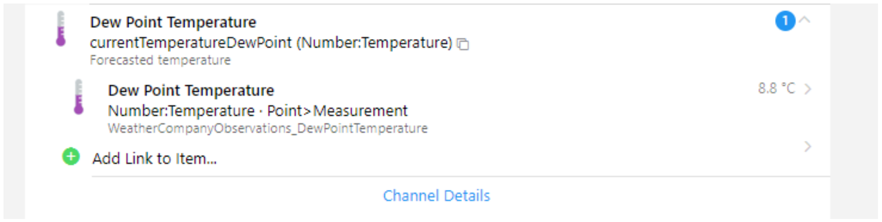
Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles. Learn more about profiles.

- [✓ Predeterminado]
- JS
- JSONPATH
- EXEC
- Rango
- Histéresis
- Marca de tiempo al cambiar
- Seguir
- Marca de tiempo al actualizar
- Desplazamiento
- Gain-Offset Correction

Figure 5.124 – *Create item*

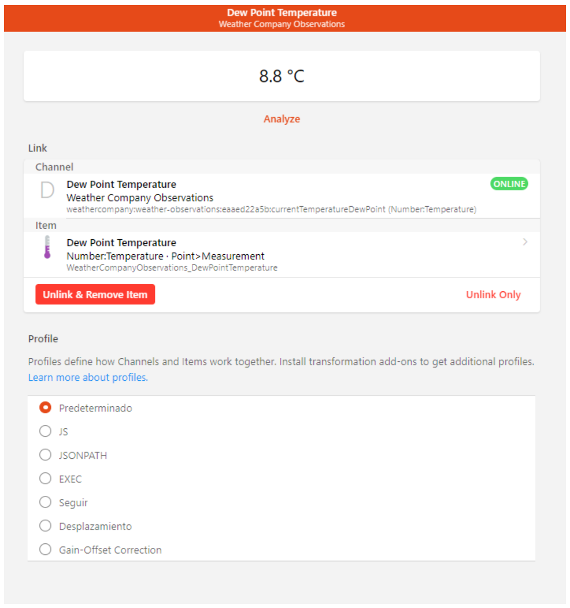
3. And finally the item will be **linked to the chosen channel**. If we access it, it will show us information about the current value of the item, i.e. the weather variable, the channel from which it has been created and the item created, in case we want to edit its parameters.



Dew Point Temperature
currentTemperatureDewPoint (Number:Temperature)
Forecasted temperature

Dew Point Temperature 8.8 °C
Number:Temperature · Point>Measurement
WeatherCompanyObservations_DewPointTemperature

Figure 5.125 – *Created item*



Dew Point Temperature
Weather Company Observations

8.8 °C

Analyze

Link

Channel

Dew Point Temperature
Weather Company Observations
weathercompany:weather-observations:eaaed22a5b:currentTemperatureDewPoint (Number:Temperature)

ONLINE

Item

Dew Point Temperature
Number:Temperature · Point>Measurement
WeatherCompanyObservations_DewPointTemperature

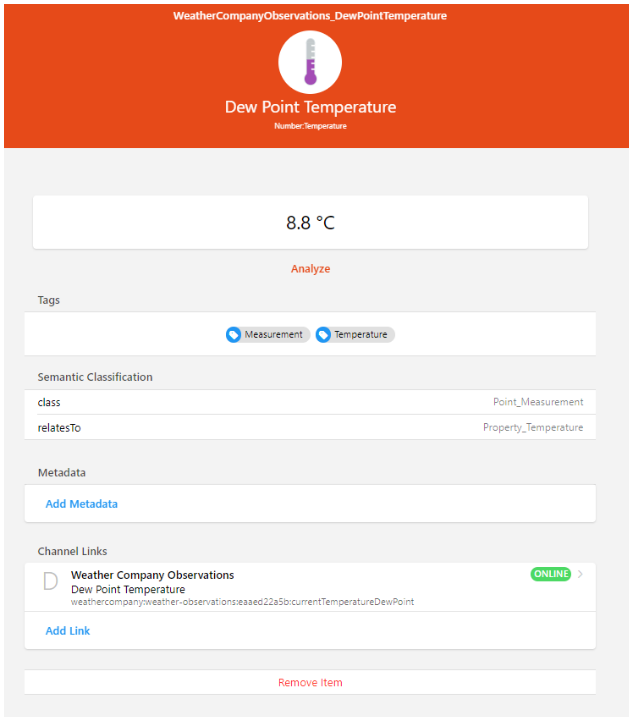
Unlink & Remove Item   Unlink Only

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
Learn more about profiles.

- [✓ Predeterminado]
- JS
- JSONPATH
- EXEC
- Seguir
- Desplazamiento
- Gain-Offset Correction

(a) *Link info*



WeatherCompanyObservations_DewPointTemperature

Dew Point Temperature
Number:Temperature

8.8 °C

Analyze

Tags

Measurement   Temperature

Semantic Classification

class	Point_Measurement
relatesTo	Property_Temperature

Metadata

Add Metadata

Channel Links

Weather Company Observations
Dew Point Temperature
weathercompany:weather-observations:eaaed22a5b:currentTemperatureDewPoint

ONLINE

Add Link

Remove Item

(b) *Item info*

Figure 5.126 – *Display information of item*

5.9 Solar inverter and Home Manager

The customer wanted to obtain data related to solar energy, the solar inverter and energy meter worked with another communication protocol such as Modbus RTU, based on master/slave or client/server (TCP/IP) architecture.

5.9.1 Options to implement solar energy devices

Searching for ways to get the information from solar energy, I found two bindings, which could be useful for the implementation of these devices:

- **SMA Energy Meter binding**, which is responsible for collecting the information obtained from the device SMA Energy Meter.
- **Modbus binding**, for obtaining data from the single-phase solar inverter

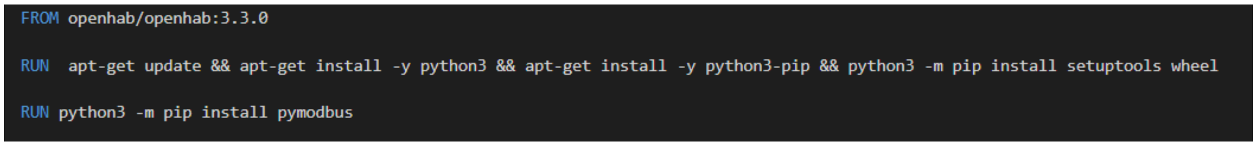
The SMA Energy Meter binding worked correctly, creating the thing by entering the IP address of the Energy Meter device, obtaining data such as PowerIn, PowerOut and EnergyIn and EnergyOut, close to reality. But when installing and configuring the inverter items with the modbus binding, I found that the values obtained had nothing to do with reality. So I decided to look for other options.

Seeing the situation Andres, my tutor, modified some scripts made in Python, to obtain data from solar energy devices, the **SunnyBoy 3.0 solar inverter** and the **SMA energy Meter**, which obtained values very close to reality, so it was decided to implement these **scripts**, in order to explain another way to obtain data in OpenHAB from some scripts.

5.9.2 Create and configure Dockerfile for execute scripts

To get the scripts to work inside the Docker, we had to **modify the Docker** by making a new image, because if we install Python from a console in the Docker, when the Docker was shut down, these changes would not be persistent. This new image would contain the OpenHAB image and we would also have to install Python with its respective additional modules (libraries), which we need to run the scripts inside the Docker. To do this we must:

1. **Create a Dockerfile** in which the base image is the OpenHAB image and Python is added with the modules we are going to use.



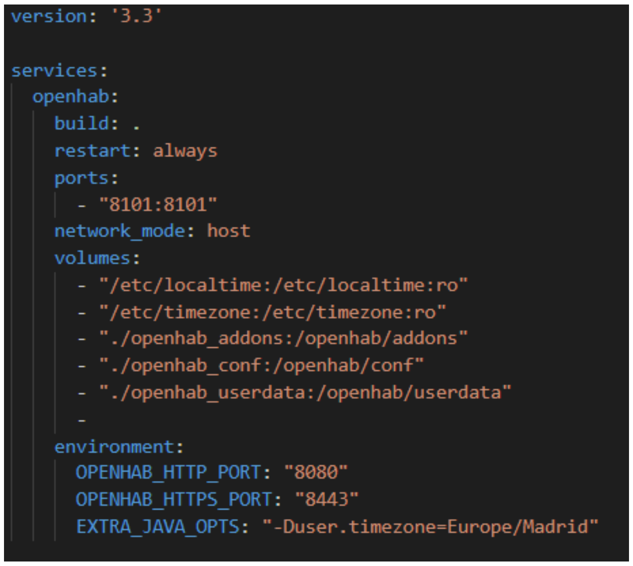
```
FROM openhab/openhab:3.3.0

RUN  apt-get update && apt-get install -y python3 && apt-get install -y python3-pip && python3 -m pip install setuptools wheel

RUN python3 -m pip install pymodbus
```

Figure 5.127 – *Dockerfile*

A short explanation of the commands used in the Dockerfile(more info):

- **FROM** -> Specified the base image for the container.
- **RUN** -> Executes the specified command. Used to install packages inside the container normally, although it can be run as a bash command.

2. Once the Dockerfile is created, with the container off (**docker-compose down**), you have to **modify the docker-compose.yml file**, to specify that you want to create the Docker container from the created Dockerfile and not from the default OpenHAB image of Docker. Just change the image parameter to **build** and **indicate the path** from the docker-compose.yml file to the folder containing the Dockerfile, for example if the docker-compose.yml file and the Dockerfile are in the same folder, the path you would have to put in the build parameter is ".".



```
version: '3.3'

services:
  openhab:
    build: .
    restart: always
    ports:
      - "8101:8101"
    network_mode: host
    volumes:
      - "/etc/localtime:/etc/localtime:ro"
      - "/etc/timezone:/etc/timezone:ro"
      - "./openhab_addons:/openhab/addons"
      - "./openhab_conf:/openhab/conf"
      - "./openhab_userdata:/openhab/userdata"
      -
    environment:
      OPENHAB_HTTP_PORT: "8080"
      OPENHAB_HTTPS_PORT: "8443"
      EXTRA_JAVA_OPTS: "-Duser.timezone=Europe/Madrid"
```

Figure 5.128 – *New docker-compose.yml*

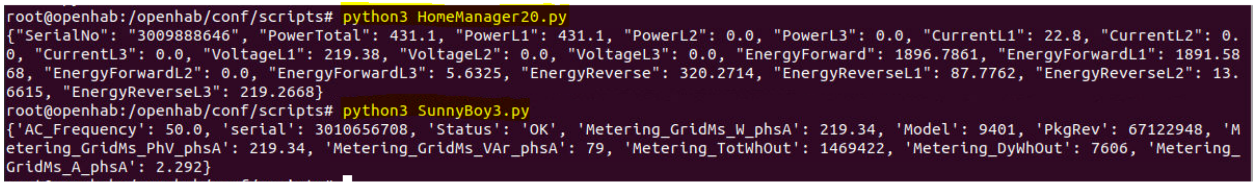
3. Once the Dockerfile has been created and the docker-compose.yml has been modified, we will have to build and deploy the new Docker:

5

- Shut down the Docker container if it is running, with the command **docker-compose down**.
- The first time when you start the container with **docker-compose up -d**, it builds the new image from the Dockerfile automatically. If the container is deleted, when it is activated again, it looks for the image created from the Dockerfile and if it does not find it, it creates it again.

  The best option is to create the image but without activating the container, to avoid problems with building the new image. To build the image in this way we run the command **docker-compose build –no-cache**, because Docker to save space runs each line of the Dockerfile in a separate container and then joins them in case you reuse the same line in several Docker, but if you modify something in the Dockerfile and it does not work, it is possible that it has taken one of those containers that it had before and has not realized that the Dockerfile has been modified (usually it realizes but it may fail).

4. Once the new docker container is created and up, we move the scripts to the **openhab_conf/scripts** folder and access inside the container with **docker exec -it container_name or container_id /bin/bash**, and test to see if the scripts work inside the container.



```
root@openhab:/openhab/conf/scripts# python3 HomeManager20.py
{"SerialNo": "3009888646", "PowerTotal": 431.1, "PowerL1": 431.1, "PowerL2": 0.0, "PowerL3": 0.0, "CurrentL1": 22.8, "CurrentL2": 0.
0, "CurrentL3": 0.0, "VoltageL1": 219.38, "VoltageL2": 0.0, "VoltageL3": 0.0, "EnergyForward": 1896.7861, "EnergyForwardL1": 1891.58
68, "EnergyForwardL2": 0.0, "EnergyForwardL3": 5.6325, "EnergyReverse": 320.2714, "EnergyReverseL1": 87.7762, "EnergyReverseL2": 13.
6615, "EnergyReverseL3": 219.2668}
root@openhab:/openhab/conf/scripts# python3 SunnyBoy3.py
{'AC_Frequency': 50.0, 'serial': 3010656708, 'Status': 'OK', 'Metering_GridMs_W_phsA': 219.34, 'Model': 9401, 'PkgRev': 67122948, 'M
etering_GridMs_PhV_phsA': 219.34, 'Metering_GridMs_VAr_phsA': 79, 'Metering_TotWhOut': 1469422, 'Metering_DyWhOut': 7606, 'Metering_
GridMs_A_phsA': 2.292}
```

Figure 5.129 – *Execute scripts in docker container*

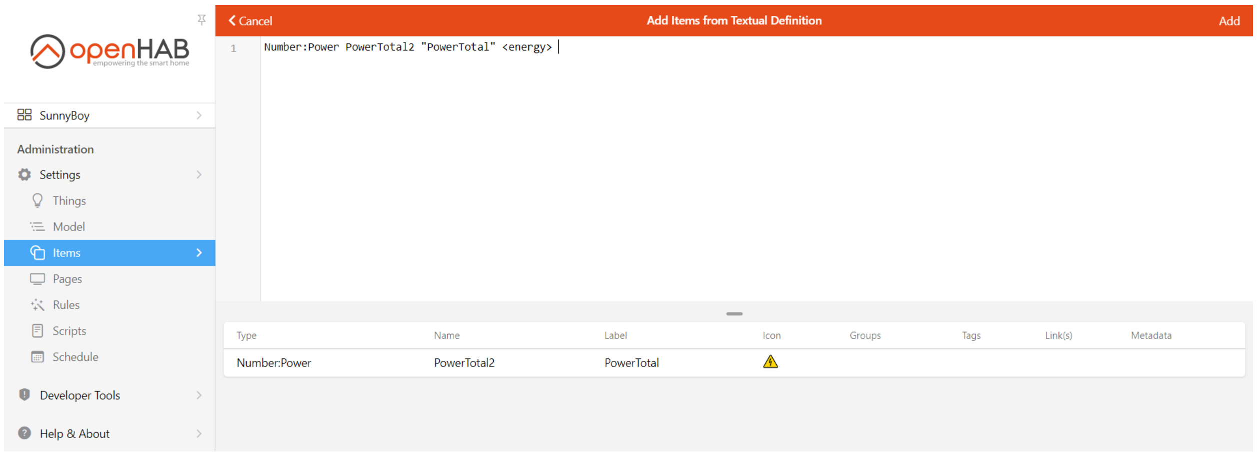
5.9.3 Create and configure items

Once the new Docker image is created and the scripts can be executed correctly in the Docker, the corresponding items will be created to show the data of the variables of the scripts.

These items can be created in various ways:

- **From text files** from a PC terminal, go to **openhab/openhab_conf/items/** and create a file to define the items. The definition of the items in the text files is: **Type:measurement Name "Label" <Icon> (Group)** [**Tags**].
- From the MainUI in the **Settings->Items->+ icon -> Add item tab**, to do it with the modern interface, it is easier but you have to go one by one.
- From the MainUI in the **Settings-> Items-> + icon-> Add item from textual definition**, to do it the same way as the textual form. This way is faster because you can define more than one item at a time.

The way I chose to create the items was from the main ui using the text file form. Here is an example of how to create an item:



```
Number:Power PowerTotal2 "PowerTotal" <energy>
```

Type	Name	Label	Icon	Groups	Tags	Link(s)	Metadata
Number:Power	PowerTotal2	PowerTotal	⚠				

Figure 5.130 – *Add item from textual definition*

5.9.4 Create and run rules for script execution

Until now, we have connected OpenHAB to devices through Things, modeled the devices with Items, but that amounts to home control, not automation. To create home automation we need to define behaviours. In OpenHAB, behaviours are defined by **rules**. You can do almost anything you can think of, as long as you have a relevant event to trigger it and access the necessary data to decide what to do.

The rules are divided into three parts:

- **Trigger** -> events that cause the rule to be executed (triggers list).
- **Condition** -> conditions that must be met before the rule is allowed to run even when triggered (conditions list).
- **Action** -> actions to be taken when the rule is executed (actions list).

A single rule is **not required to have all parts** (although a rule without the action part is not very sensible) and in many cases multiple instances may be required [36].

To **execute the scripts** we need to use the **rules**, as we said in the section 5.1.5 we will use the text files to create the rules. But before defining the rules, we have to install a **transformation binding** to extract the result data from the scripts in JSON format and we will also install another binding to make use of Javascript inside the rules. As in the other sections of the other devices, to install the bindings we go to **Settings ->Bindings -> search** for **JsonPath** and install the binding and search for **JSScripting** and install the binding.

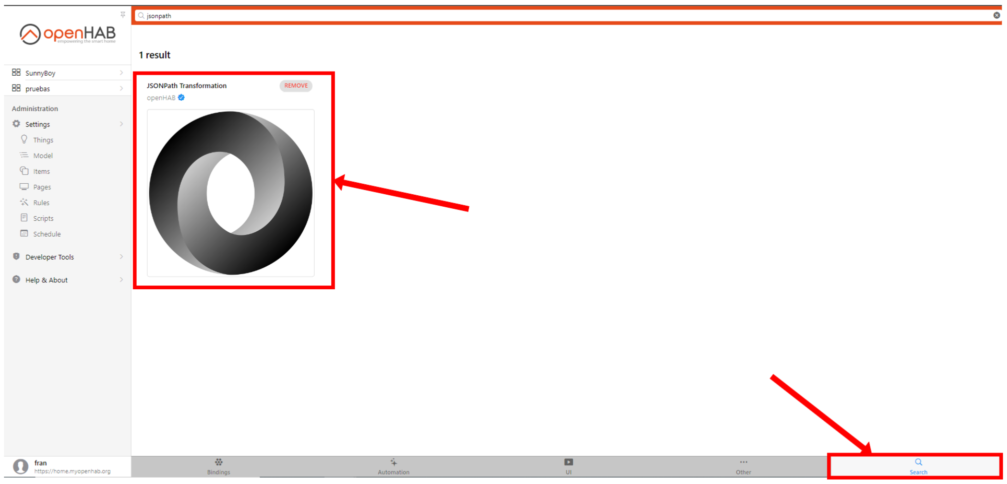


Figure 5.131 – *Install binding JSON*

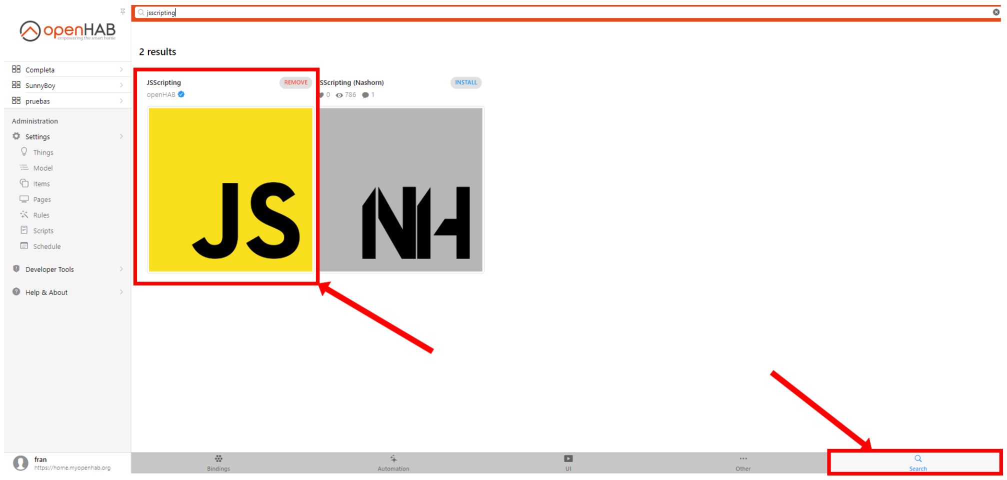


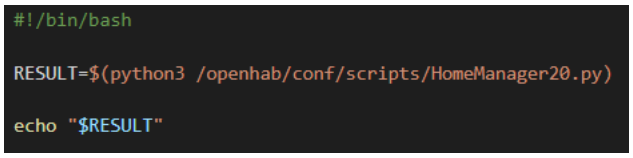
Figure 5.132 – *Install binding JSScripting*

Once the binding is installed, we will **create the rules** to execute the scripts and obtain their values inside the system. The rule will consist of **executing the script and updating the items** with the corresponding value extracted from the result of the script through the JsonPath binding, using a trigger of the rule by time through a cron expression.

The execution of the script gave me a lot of problems because I couldn't execute and get the result, using the action **executeCommandLine** (more info), this command can be used by entering the bash instruction directly or the path to a **bash script**. The latter way was the one that worked for me and is the one I am going to explain:

1. First of all, create the **text file** for the rule in the folder **openhab_conf/rules/** and define which will be the trigger of the rule, in this case we will use the trigger by time through a 6-digit **cron expression**, this expression is used to schedule the execution of automated tasks.

2. As the execution of the script failed, we had to create an intermediary **bash script**, which will only execute the Python command to run the script and display the result on screen, this script must go in the **openHAB/openhab_conf/scripts/** folder. The only thing that the path of the script must be the path to the file inside the container (**/openhab/conf/scripts/**).

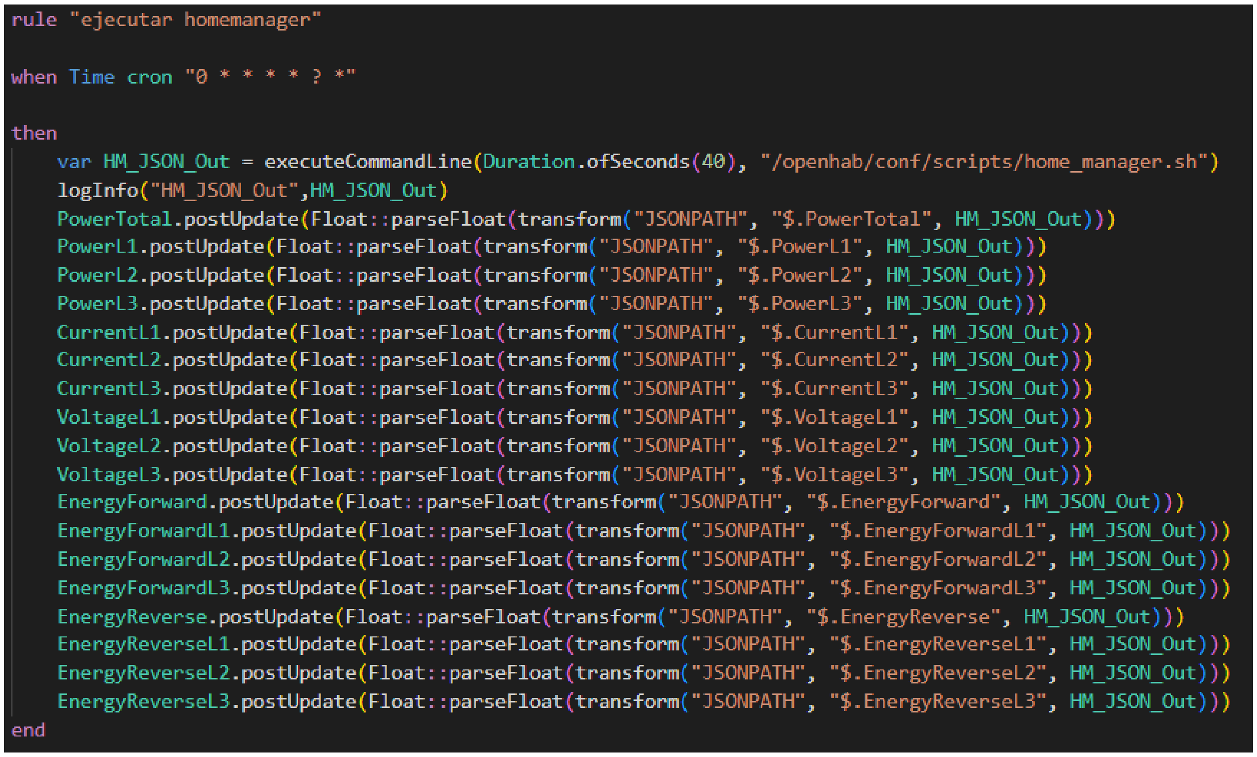


```
#!/bin/bash

RESULT=$(python3 /openhab/conf/scripts/HomeManager20.py)

echo "$RESULT"
```

Figure 5.133 – *Script bash*

3. Execute the intermediary bash script using the **executeCommandLine** command in the rule.
4. Once executed, we split the JSON result obtained through the JsonPath binding and update the value of the items created through the postUpdate action of each item. (more info).

Here we can see a rule for a script, already finished.:



```
rule "ejecutar homemanager"

when Time cron "0 * * * * ? *"

then
    var HM_JSON_Out = executeCommandLine(Duration.ofSeconds(40), "/openhab/conf/scripts/home_manager.sh")
    logInfo("HM_JSON_Out",HM_JSON_Out)
    PowerTotal.postUpdate(Float::parseFloat(transform("JSONPATH", "$.PowerTotal", HM_JSON_Out)))
    PowerL1.postUpdate(Float::parseFloat(transform("JSONPATH", "$.PowerL1", HM_JSON_Out)))
    PowerL2.postUpdate(Float::parseFloat(transform("JSONPATH", "$.PowerL2", HM_JSON_Out)))
    PowerL3.postUpdate(Float::parseFloat(transform("JSONPATH", "$.PowerL3", HM_JSON_Out)))
    CurrentL1.postUpdate(Float::parseFloat(transform("JSONPATH", "$.CurrentL1", HM_JSON_Out)))
    CurrentL2.postUpdate(Float::parseFloat(transform("JSONPATH", "$.CurrentL2", HM_JSON_Out)))
    CurrentL3.postUpdate(Float::parseFloat(transform("JSONPATH", "$.CurrentL3", HM_JSON_Out)))
    VoltageL1.postUpdate(Float::parseFloat(transform("JSONPATH", "$.VoltageL1", HM_JSON_Out)))
    VoltageL2.postUpdate(Float::parseFloat(transform("JSONPATH", "$.VoltageL2", HM_JSON_Out)))
    VoltageL3.postUpdate(Float::parseFloat(transform("JSONPATH", "$.VoltageL3", HM_JSON_Out)))
    EnergyForward.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyForward", HM_JSON_Out)))
    EnergyForwardL1.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyForwardL1", HM_JSON_Out)))
    EnergyForwardL2.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyForwardL2", HM_JSON_Out)))
    EnergyForwardL3.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyForwardL3", HM_JSON_Out)))
    EnergyReverse.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyReverse", HM_JSON_Out)))
    EnergyReverseL1.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyReverseL1", HM_JSON_Out)))
    EnergyReverseL2.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyReverseL2", HM_JSON_Out)))
    EnergyReverseL3.postUpdate(Float::parseFloat(transform("JSONPATH", "$.EnergyReverseL3", HM_JSON_Out)))
end
```

Figure 5.134 – *Rule Home manager*

If any problems occur we should always look at the system logs in the **openhab_userdate/logs/openhab.log** file, explained in the section 5.1.4 to tell us what has happened and look for bugs in the OpenHAB community.

Also in the rules we can put log functions to debug variables, states or values of the items by displaying messages in the log files.

5.10 Data persistence

Once all items are created and running, the different values of these items are not being saved, for example to represent the stored data in a graph. To be able to save the data we need a database, the most popular among the community is InfluxDB.

5.10.1 InfluxDB

5.10.1.1 Install and configure InfluxDB

The idea was to create another Docker container to be able to transport it with the OpenHAB container, but I had problems with the connection between one container and another, so I decided to install InfluxDB on the host, InfluxDB has released version 2 but version 1 has been installed, to avoid problems with new configurations and implementations with OpenHAB. We have followed this link for the installation and configuration of InfluxDB:

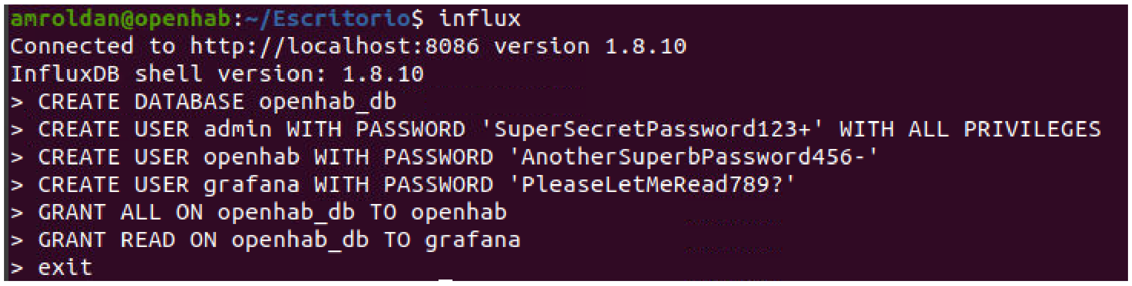
1. Add InfluxDB repository, with the following commands:

   - curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
   - source /etc/lsb-release
   - echo "deb https://repos.influxdata.com/ ${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

2. Upgrade host packages and install and start InfluxDB.

   - sudo apt-get update && sudo apt-get install influxdb
   - sudo systemctl start influxdb

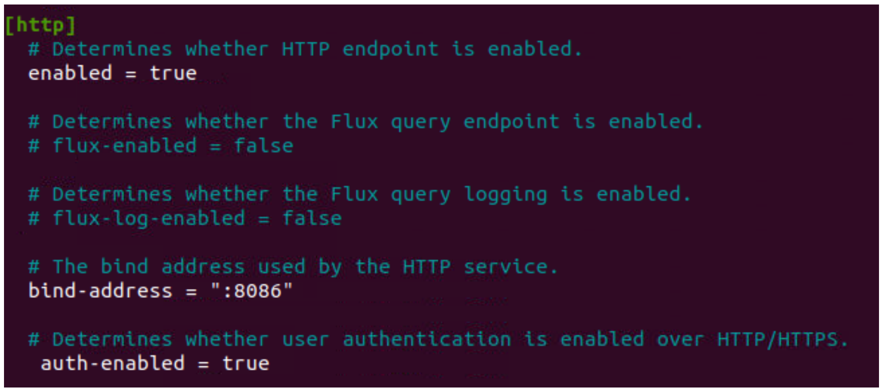
3. Now you have to create the database and the necessary users (InfluxDB documentation).



```
amroldan@openhab:~/Escritorio$ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> CREATE DATABASE openhab_db
> CREATE USER admin WITH PASSWORD 'SuperSecretPassword123+' WITH ALL PRIVILEGES
> CREATE USER openhab WITH PASSWORD 'AnotherSuperbPassword456-'
> CREATE USER grafana WITH PASSWORD 'PleaseLetMeRead789?'
> GRANT ALL ON openhab_db TO openhab
> GRANT READ ON openhab_db TO grafana
> exit
```

Figure 5.135 – *Create and configure data base*

4. Enable authentication in the [http] section of the **/etc/influxdb/influxdb.conf** configuration file.

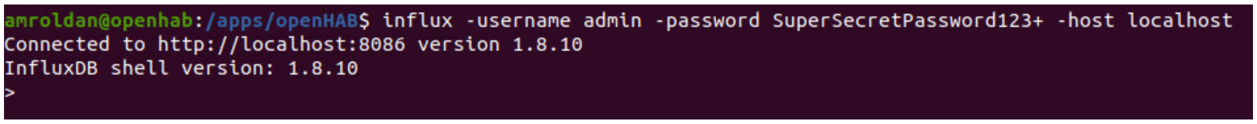


```
[http]
  # Determines whether HTTP endpoint is enabled.
  enabled = true

  # Determines whether the Flux query endpoint is enabled.
  # flux-enabled = false

  # Determines whether the Flux query logging is enabled.
  # flux-log-enabled = false

  # The bind address used by the HTTP service.
  bind-address = ":8086"

  # Determines whether user authentication is enabled over HTTP/HTTPS.
   auth-enabled = true
```

Figure 5.136 – *Http config of data base*

5. Finally, restart the InfluxDB service: **sudo systemctl restart influxdb.service**.

6. Now the terminal interface asks for authentication, to access the data base: **influx -username admin -password SuperSecretPassword123+ -host localhost**.



```
amroldan@openhab:/apps/openHAB$ influx -username admin -password SuperSecretPassword123+ -host localhost
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
```

Figure 5.137 – *Data base access*

5.10.1.2 Connecting InfluxDB to OpenHAB

Once InfluxDB is installed, we need to connect the database to OpenHAB. To do this we will install the InfluxDB persistence binding:

1. As we already know to install the binding we go to the tab **Settings-> Bindings -> Search -> search for InfluxDB and install the binding**.

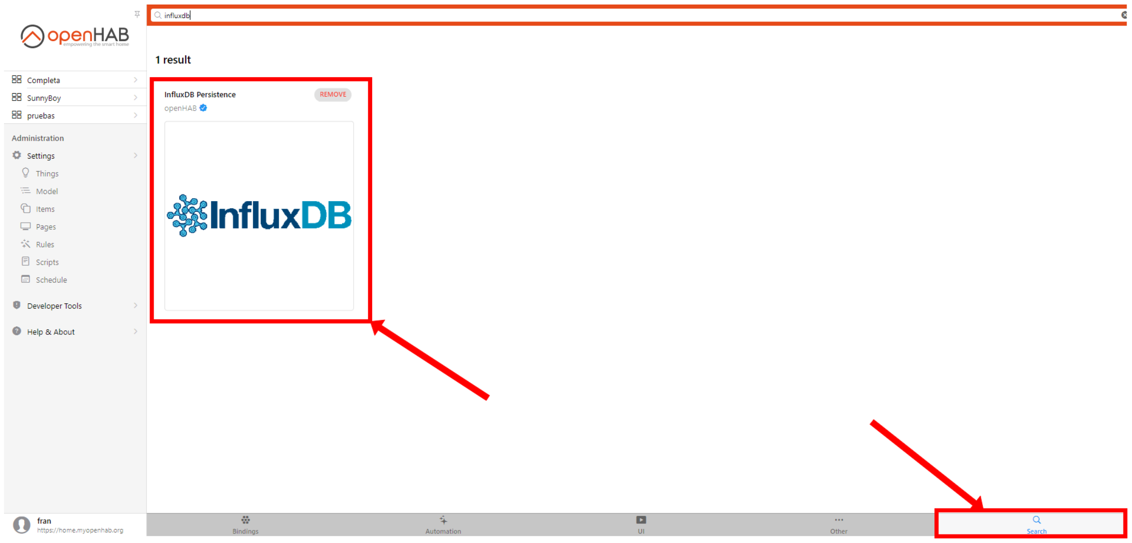
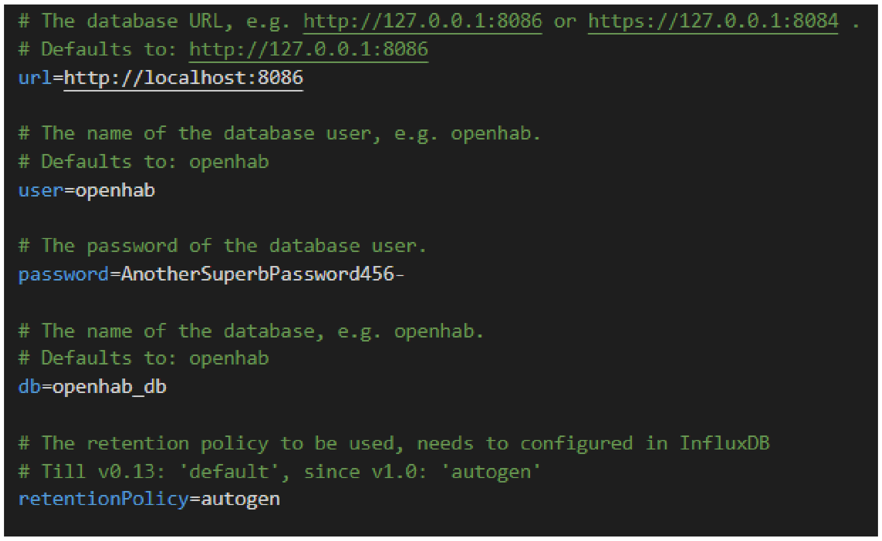


Figure 5.138 – *Install InfluxDB binding*

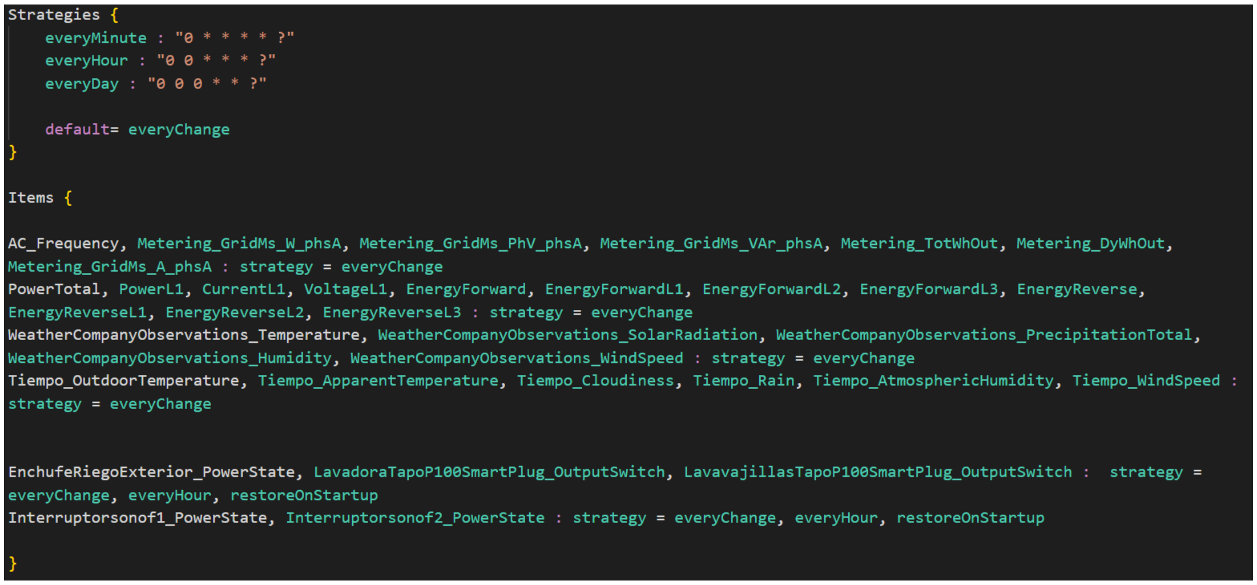
2. Once the binding is installed, we will add the InfluxDB connection details to the persistence service configuration file **/openhab_conf/services/influxdb.conf**.



```
# The database URL, e.g. http://127.0.0.1:8086 or https://127.0.0.1:8084 .
# Defaults to: http://127.0.0.1:8086
url=http://localhost:8086

# The name of the database user, e.g. openhab.
# Defaults to: openhab
user=openhab

# The password of the database user.
password=AnotherSuperbPassword456-

# The name of the database, e.g. openhab.
# Defaults to: openhab
db=openhab_db

# The retention policy to be used, needs to configured in InfluxDB
# Till v0.13: 'default', since v1.0: 'autogen'
retentionPolicy=autogen
```

Figure 5.139 – *Influxdb.conf*

3. Once we have added the configuration for the connection, we must **add a persistence job** for OpenHAB to send the status of its items to InfluxDB, in this file in **/openhba_conf/persistence/influxdb.persist** we must **add all the items or groups of items** that we want to be persistent by means of some saving strategies (cron expressions). The restoreOnStartup strategy consists of restoring the sockets and switches to the last state they were in when the Docker container was restarted when the server went down or when the power or Internet connection went down.



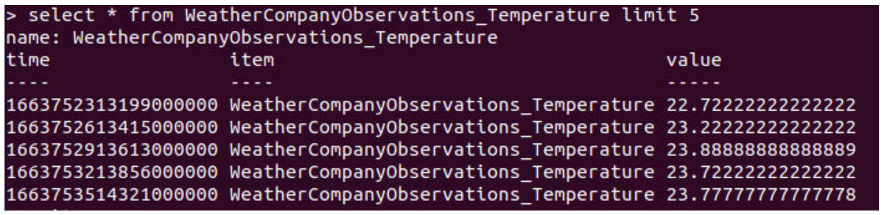
```
Strategies {
    everyMinute : "0 * * * * ?"
    everyHour : "0 0 * * * ?"
    everyDay : "0 0 0 * * ?"

    default= everyChange
}

Items {

AC_Frequency, Metering_GridMs_W_phsA, Metering_GridMs_PhV_phsA, Metering_GridMs_VAr_phsA, Metering_TotWhOut, Metering_DyWhOut,
Metering_GridMs_A_phsA : strategy = everyChange
PowerTotal, PowerL1, CurrentL1, VoltageL1, EnergyForward, EnergyForwardL1, EnergyForwardL2, EnergyForwardL3, EnergyReverse,
EnergyReverseL1, EnergyReverseL2, EnergyReverseL3 : strategy = everyChange
WeatherCompanyObservations_Temperature, WeatherCompanyObservations_SolarRadiation, WeatherCompanyObservations_PrecipitationTotal,
WeatherCompanyObservations_Humidity, WeatherCompanyObservations_WindSpeed : strategy = everyChange
Tiempo_OutdoorTemperature, Tiempo_ApparentTemperature, Tiempo_Cloudiness, Tiempo_Rain, Tiempo_AtmosphericHumidity, Tiempo_WindSpeed :
strategy = everyChange


EnchufeRiegoExterior_PowerState, LavadoraTapoP100SmartPlug_OutputSwitch, LavavajillasTapoP100SmartPlug_OutputSwitch :  strategy =
everyChange, everyHour, restoreOnStartup
Interruptorsonof1_PowerState, Interruptorsonof2_PowerState : strategy = everyChange, everyHour, restoreOnStartup

}
```

Figure 5.140 – *Items persistence file*

5.10.1.3 Using and managing InfluxDB

Once InfluxDB is installed and connected to OpenHAB, we will explain how to manage the database in order to visualise the data being stored. InfluxDB works with the SQL language, so if you know the language it will be easier for you to use it.

- We login to InfluxDB with the data we created earlier, **influx -username admin -password SuperSecretPassword123+ -host localhost**.
- We go into the database that we have created with the command **use openhab_db**.
- Once inside the database, we can use for example:
  - **Show measurements**, to show all the tables of the saved items.
  - **SQL Select queries**, to display data from tables. Influx stores data according to Unix time, we can use this page to convert Unix time to date.



```
> select * from WeatherCompanyObservations_Temperature limit 5
name: WeatherCompanyObservations_Temperature
time                item                                   value
----                ----                                   -----
1663752313199000000 WeatherCompanyObservations_Temperature 22.72222222222222
1663752613415000000 WeatherCompanyObservations_Temperature 23.22222222222222
1663752913613000000 WeatherCompanyObservations_Temperature 23.88888888888889
1663753213856000000 WeatherCompanyObservations_Temperature 23.72222222222222
1663753514321000000 WeatherCompanyObservations_Temperature 23.77777777777778
```

Figure 5.141 – *Example select query*

5.11 User Interfaces

We already have everything configured, but we don't have a user interface that allows us to manage more easily the values of the items, because until now if we wanted to change the status of an item, we had to go to the item and change it from the **Settings-> Items tab** or from the different applications of the devices. To solve this problem there are different types of user interfaces in OpenHAB:

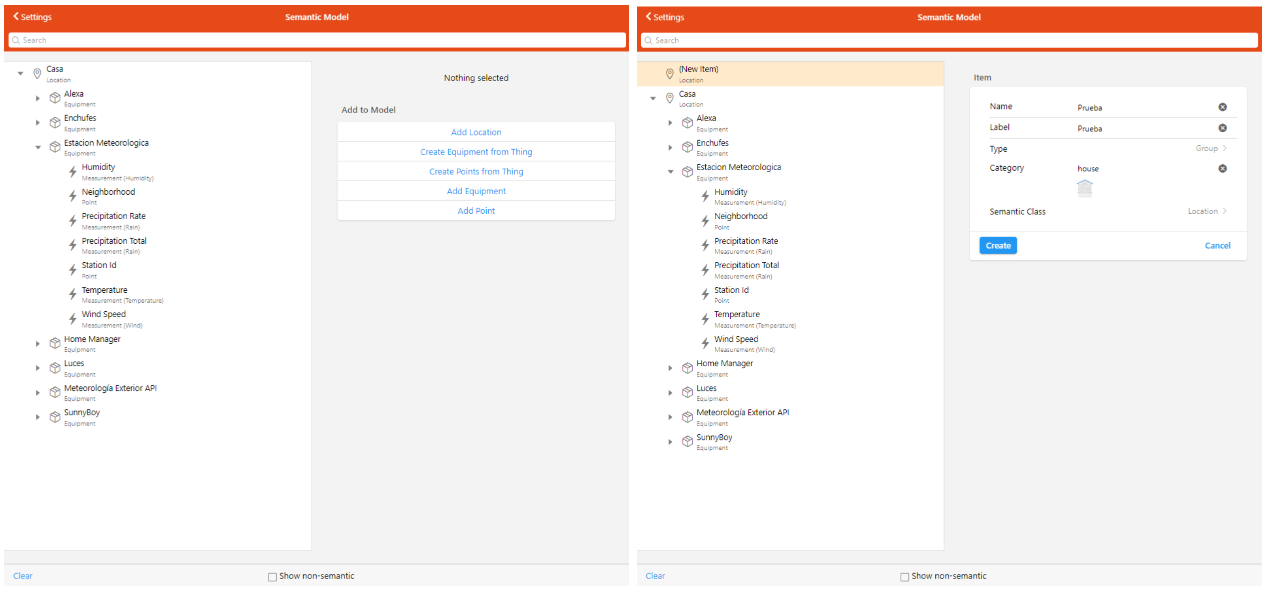
- **Sitemap** -> A simple declarative way to define a simple user interface. Does not require the configuration of the semantic model.
- **HABPanel** -> User interface designed for fixed wall-mounted tablets and similar touch screens using a block interface. Requires configuration of the semantic model
- **Page** -> The latest interface included in OpenHAB 3, built on MainUI. It allows a wide set of options to present your home automation. Requires configuration of the semantic model

In this project, we will use the sitemaps for the OpenHAB mobile application that we have already configured in the section 5.3 and the pages for the web, although both interfaces can be used in both sites.

5.11.1 Semantic model

The **semantic model** is a new concept that has been added in OpenHAB 3, which will make things easier when creating pages, it **represents the physical layout of the home automation devices in the house**, it is also a way that OpenHAB has to organise all the items, by function, by unit of measurement, etc. OpenHAB makes extensive use of the semantic model when creating pages. To create the semantic model go to the **Settings-> Model tab** and follow the steps below:

- **Add the necessary locations** that you want, location of a group of devices, at OpenHAB level they will be represented as groups of items (groups are also items). To create the locations from the tab **Settings-> Model -> Add location** and fill in the data, name, label and icon. If we want to put it inside another location, we must first click on the parent location and click add location.



(a)

Settings — Semantic Model

Search

- Casa (Location)
  - Alexa (Equipment)
  - Enchufes (Equipment)
  - Estacion Meteorologica (Equipment)
    - Humidity (Measurement (Humidity))
    - Neighborhood (Point)
    - Precipitation Rate (Measurement (Rain))
    - Precipitation Total (Measurement (Rain))
    - Station Id (Point)
    - Temperature (Measurement (Temperature))
    - Wind Speed (Measurement (Wind))
  - Home Manager (Equipment)
  - Luces (Equipment)
  - Meteorologia Exterior API (Equipment)
  - SunnyBoy (Equipment)

Nothing selected

Add to Model

Add Location
Create Equipment from Thing
Create Points from Thing
Add Equipment
Add Point

Clear — Show non-semantic

(b)

Settings — Semantic Model

Search

- (New Item) (Location)
- Casa (Location)
  - Alexa (Equipment)
  - Enchufes (Equipment)
  - Estacion Meteorologica (Equipment)
    - Humidity (Measurement (Humidity))
    - Neighborhood (Point)
    - Precipitation Rate (Measurement (Rain))
    - Precipitation Total (Measurement (Rain))
    - Station Id (Point)
    - Temperature (Measurement (Temperature))
    - Wind Speed (Measurement (Wind))
  - Home Manager (Equipment)
  - Luces (Equipment)
  - Meteorologia Exterior API (Equipment)
  - SunnyBoy (Equipment)

Item

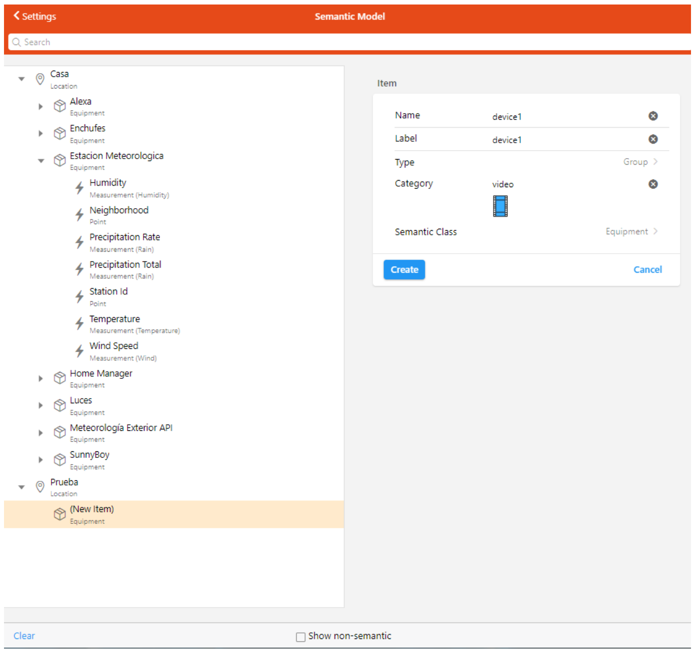
Name	Prueba
Label	Prueba
Type	Group
Category	house
Semantic Class	Location

Create — Cancel

Clear — Show non-semantic

Figure 5.142 – *Add location*

- **Add equipment**, which is the same as locations, to represent groups of items. To create an equipment, we can do it in two ways:
  - **Create equipment**, which will create a group as before with the location, in which we have to fill in the same data as the locations.



Settings — Semantic Model

Search

- Casa (Location)
  - Alexa (Equipment)
  - Enchufes (Equipment)
  - Estacion Meteorologica (Equipment)
    - Humidity (Measurement (Humidity))
    - Neighborhood (Point)
    - Precipitation Rate (Measurement (Rain))
    - Precipitation Total (Measurement (Rain))
    - Station Id (Point)
    - Temperature (Measurement (Temperature))
    - Wind Speed (Measurement (Wind))
  - Home Manager (Equipment)
  - Luces (Equipment)
  - Meteorologia Exterior API (Equipment)
  - SunnyBoy (Equipment)
- Prueba (Location)
  - (New Item) (Equipment)

Item

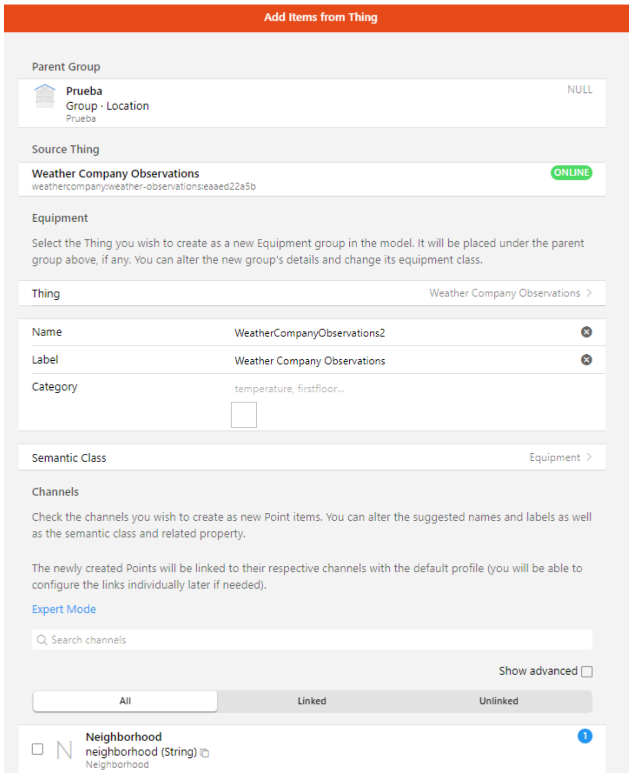
Name	device1
Label	device1
Type	Group
Category	video
Semantic Class	Equipment

Create — Cancel

Clear — Show non-semantic

Figure 5.143 – *Add equipment*

- **Create equipment from a thing**, which from a thing, creates the group and gives us the possibility to select the items that are linked to the channels of that thing.



Add Items from Thing

Parent Group

Prueba NULL
Group · Location
Prueba

Source Thing

Weather Company Observations ONLINE
weathercompany:weather-observations:eaaed22a5b

Equipment

Select the Thing you wish to create as a new Equipment group in the model. It will be placed under the parent group above, if any. You can alter the new group's details and change its equipment class.

Thing	Weather Company Observations
Name	WeatherCompanyObservations2
Label	Weather Company Observations
Category	temperature, firstfloor...
Semantic Class	Equipment

Channels

Check the channels you wish to create as new Point items. You can alter the suggested names and labels as well as the semantic class and related property.

The newly created Points will be linked to their respective channels with the default profile (you will be able to configure the links individually later if needed).

Expert Mode

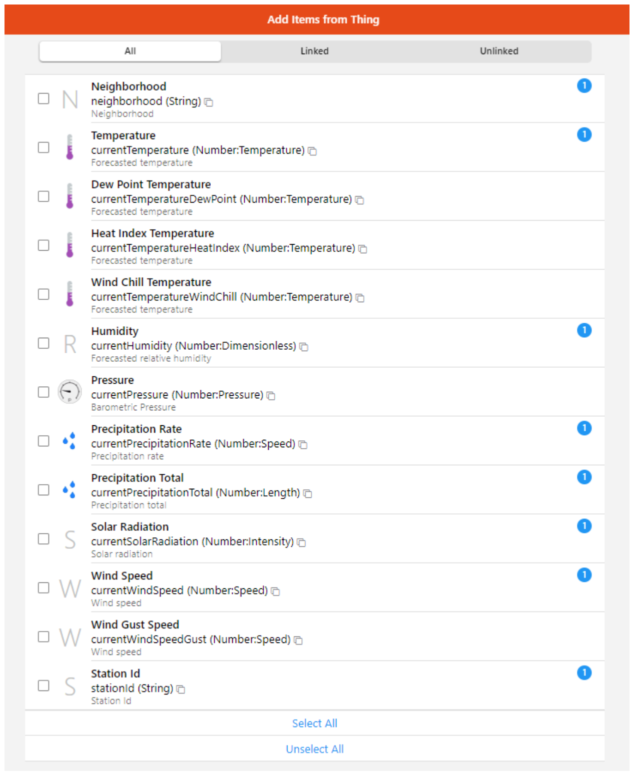
Search channels

Show advanced

All Linked Unlinked

Neighborhood
neighborhood (String)
Neighborhood

(a)



Add Items from Thing

All Linked Unlinked

Neighborhood
neighborhood (String)
Neighborhood

Temperature
currentTemperature (Number:Temperature)
Forecasted temperature

Dew Point Temperature
currentTemperatureDewPoint (Number:Temperature)
Forecasted temperature

Heat Index Temperature
currentTemperatureHeatIndex (Number:Temperature)
Forecasted temperature

Wind Chill Temperature
currentTemperatureWindChill (Number:Temperature)
Forecasted temperature

Humidity
currentHumidity (Number:Dimensionless)
Forecasted relative humidity

Pressure
currentPressure (Number:Pressure)
Barometric Pressure

Precipitation Rate
currentPrecipitationRate (Number:Speed)
Precipitation rate

Precipitation Total
currentPrecipitationTotal (Number:Length)
Precipitation total

Solar Radiation
currentSolarRadiation (Number:Intensity)
Solar radiation

Wind Speed
currentWindSpeed (Number:Speed)
Wind speed

Wind Gust Speed
currentWindSpeedGust (Number:Speed)
Wind speed

Station Id
stationId (String)
Station Id

Select All

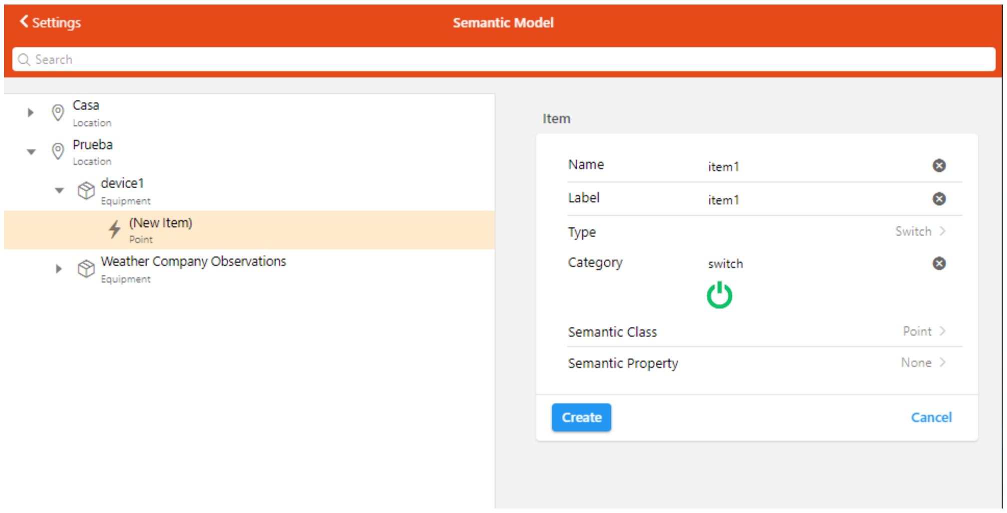
Unselect All

(b)

Figure 5.144 – *Create equipment from a thing*

- **Create points** that represent the items (the functions that each device has) and from which their states can be managed. Like equipment, they can be added in two ways:

  - **Add point**, which will create the points to be able to manage the items. But we will have to edit the created item to link the item to a channel of a thing to manage a functionality of the thing.



Settings Semantic Model

Search

Casa
Location
Prueba
Location
device1
Equipment
(New Item)
Point
Weather Company Observations
Equipment

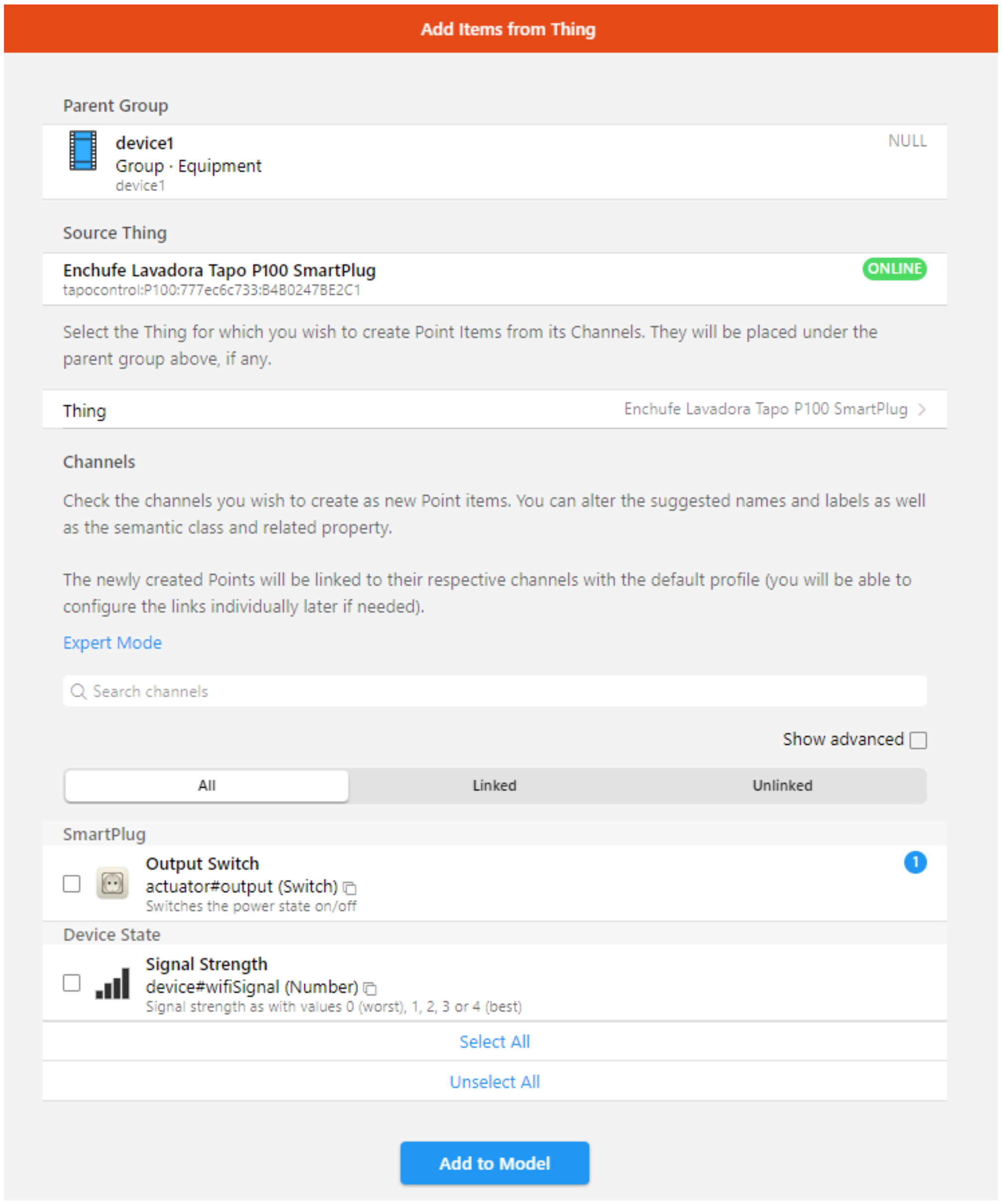
Item

Name	item1
Label	item1
Type	Switch
Category	switch
Semantic Class	Point
Semantic Property	None

Create Cancel

Figure 5.145 – *Add point*

– **Create points from a thing**, which from a thing, gives us the possibility to select the items that are linked to the channels of that thing.



Add Items from Thing

Parent Group

device1
Group · Equipment
device1 NULL

Source Thing

Enchufe Lavadora Tapo P100 SmartPlug ONLINE
tapocontrol:P100:777ec6c733:B4B0247BE2C1

Select the Thing for which you wish to create Point Items from its Channels. They will be placed under the parent group above, if any.

Thing Enchufe Lavadora Tapo P100 SmartPlug

Channels

Check the channels you wish to create as new Point items. You can alter the suggested names and labels as well as the semantic class and related property.

The newly created Points will be linked to their respective channels with the default profile (you will be able to configure the links individually later if needed).

Expert Mode

Search channels

Show advanced

All Linked Unlinked

SmartPlug

Output Switch
actuator#output (Switch)
Switches the power state on/off

Device State

Signal Strength
device#wifiSignal (Number)
Signal strength as with values 0 (worst), 1, 2, 3 or 4 (best)

Select All

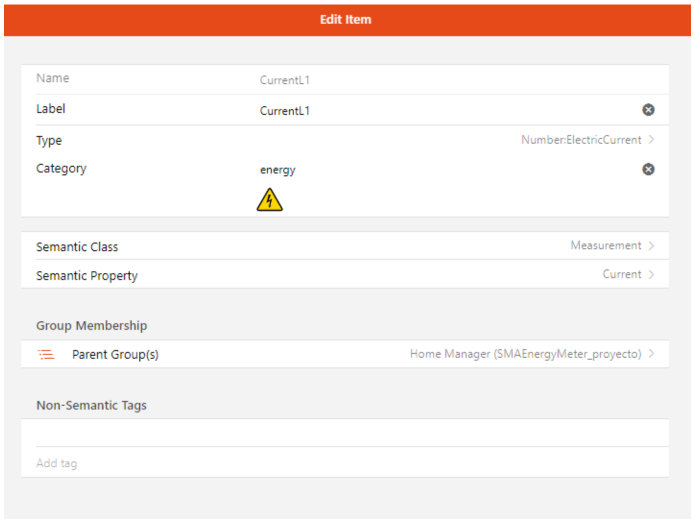
Unselect All

Add to Model

Figure 5.146 – *Create points from a thing*

The second option is easier to add the items of a thing and the first one to add the items that are not derived from a thing, such as the items created for the values of the scripts of the section 5.9.

Equipments and points can be added to the semantic model by **assigning them their corresponding group** from the **Settings-> Items -> Select item -> Edit -> Parent Group tab**, an item can be in more than one group at a time.



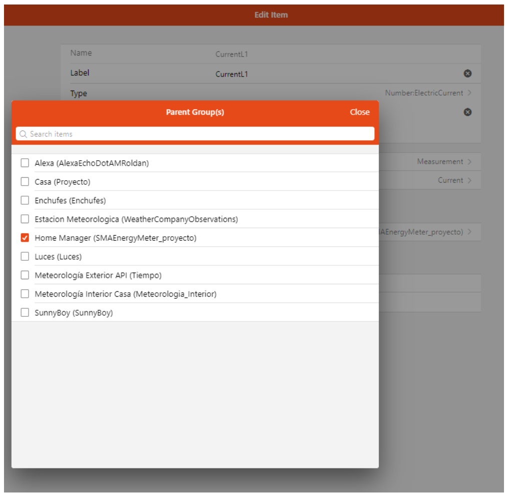
Edit Item

Name: CurrentL1
Label: CurrentL1
Type: Number:ElectricCurrent
Category: energy
Semantic Class: Measurement
Semantic Property: Current

Group Membership
Parent Group(s): Home Manager (SMAEnergyMeter_proyecto)

Non-Semantic Tags
Add tag

(a)



Parent Group(s) — Close

Search items

- [ ] Alexa (AlexaEchoDotAMRoldan)
- [ ] Casa (Proyecto)
- [ ] Enchufes (Enchufes)
- [ ] Estacion Meteorologica (WeatherCompanyObservations)
- [✓] Home Manager (SMAEnergyMeter_proyecto)
- [ ] Luces (Luces)
- [ ] Meteorología Exterior API (Tiempo)
- [ ] Meteorología Interior Casa (Meteorologia_Interior)
- [ ] SunnyBoy (SunnyBoy)

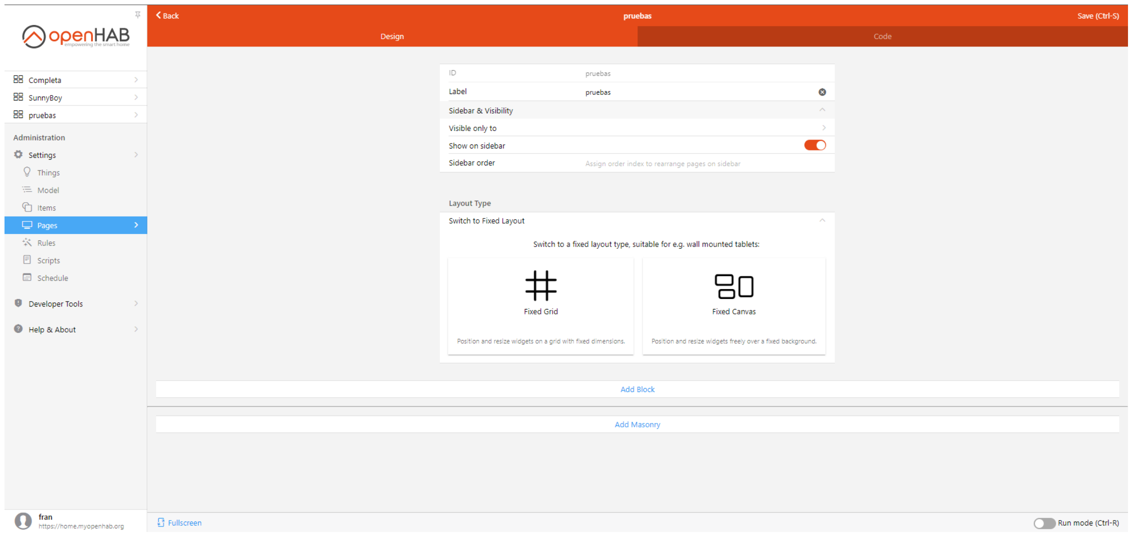
(b)

Figure 5.147 – *Add item to group*

5.11.2 Pages

Pages are the latest type of interface to be added to OpenHAB with the arrival of OpenHAB 3, so there is less explanation and configuration in the community. This interface is **more customizable than sitemaps**, due to the possibility of creating widgets and changing the parameter values to your liking. To create pages go to the **Settings -> Pages -> + icon -> add layout tab**, this will create a page. The main page at the top of the home page is called **Overview**, which is created by default and cannot be removed.

The pages give us the option to choose the layout of the page: **fixed grid**, in which all the widgets we add will have the same size by default, and another **fixed canvas**, in which we will put the widgets in the place and size we want.



ID: pruebas
Label: pruebas
Sidebar & Visibility
Visible only to
Show on sidebar: on
Sidebar order: Assign order index to rearrange pages on sidebar

Layout Type
Switch to Fixed Layout
Switch to a fixed layout type, suitable for e.g. wall mounted tablets:
Fixed Grid — Position and resize widgets on a grid with fixed dimensions.
Fixed Canvas — Position and resize widgets freely over a fixed background.

Figure 5.148 – *Layouts*

Apart from these two layouts, if we do not select any of them, by default we get a grid that is divided into:

- **Masonry**, a layout in which all widgets are together in the same area in no particular order.

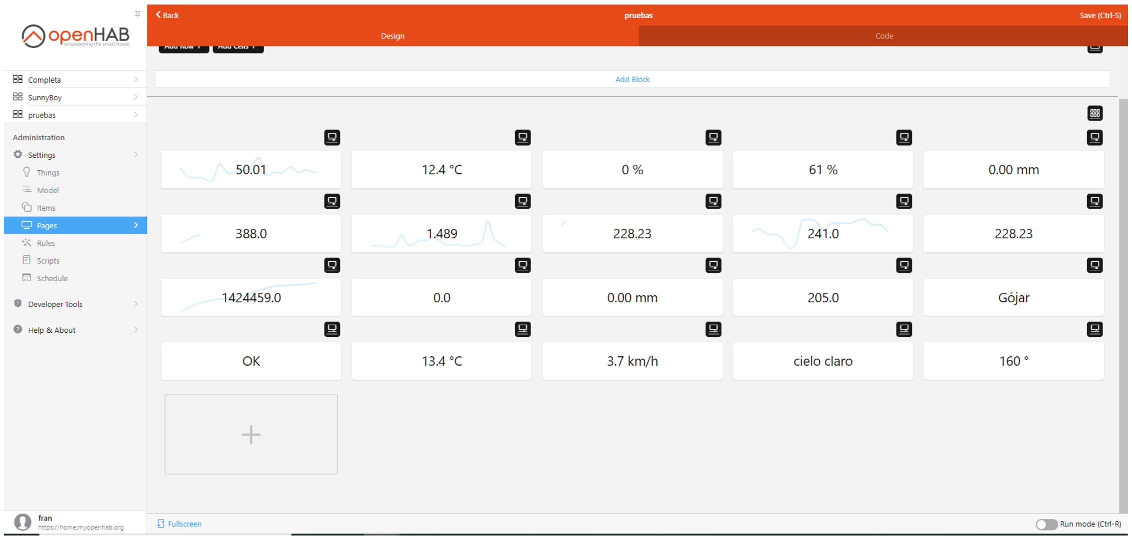


Figure 5.149 – *Masonry*

- **Block**, a block-structured design, which in turn is divided into:
  - **Row**, which contains the **columns**, which is where the widgets of the different items are placed. These columns adapt to the available size of the row, that is to say if there is 1 column it will occupy all the space of the row, if there are 2 columns half for each one and so on.
  - **Cells**, they are like other kind of widgets with a different aesthetic.

5

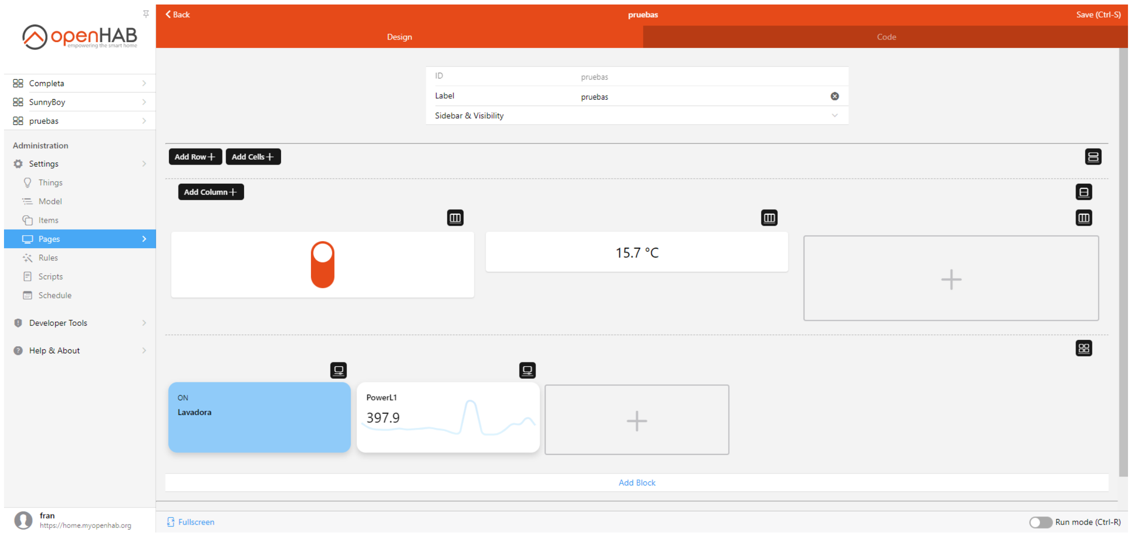
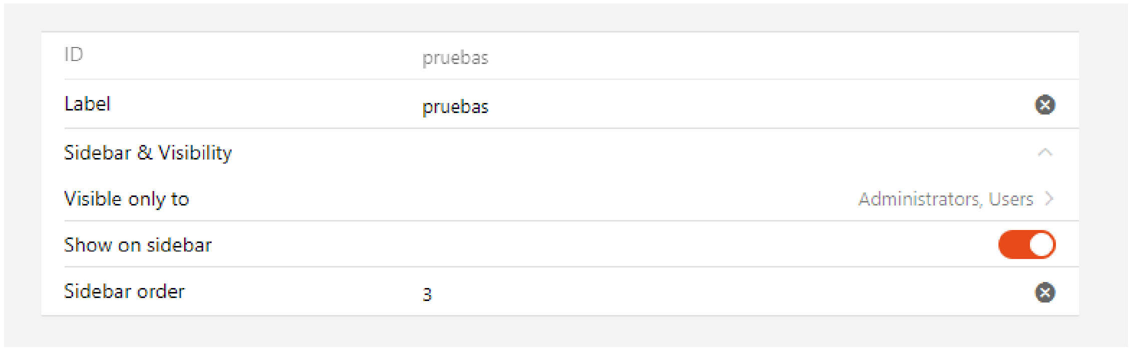


Figure 5.150 – *Block and cells*

Once you have chosen the layout of the page (more info about layouts), you only have to implement the widgets corresponding to each of the items, you can **add them directly from the model** and OpenHAB only takes care of putting the best widget that fits the item. You can also **add manually** from the default widget that brings OpenHAB and configure the widget for each item, here you can see a list of the different widgets that can be added. You can also create your own widgets, in the community you can find some very cool widgets (example). It's up to you to let your imagination run wild and see if you can implement the idea with the tools you have, it helps a lot to look at the community.

The page that I have created for the client, has been simple, I will explain how to add a widget and how to configure it, so that it can be used as an example for others. I have used the block layout, for me it is cleaner and tidier than the others:

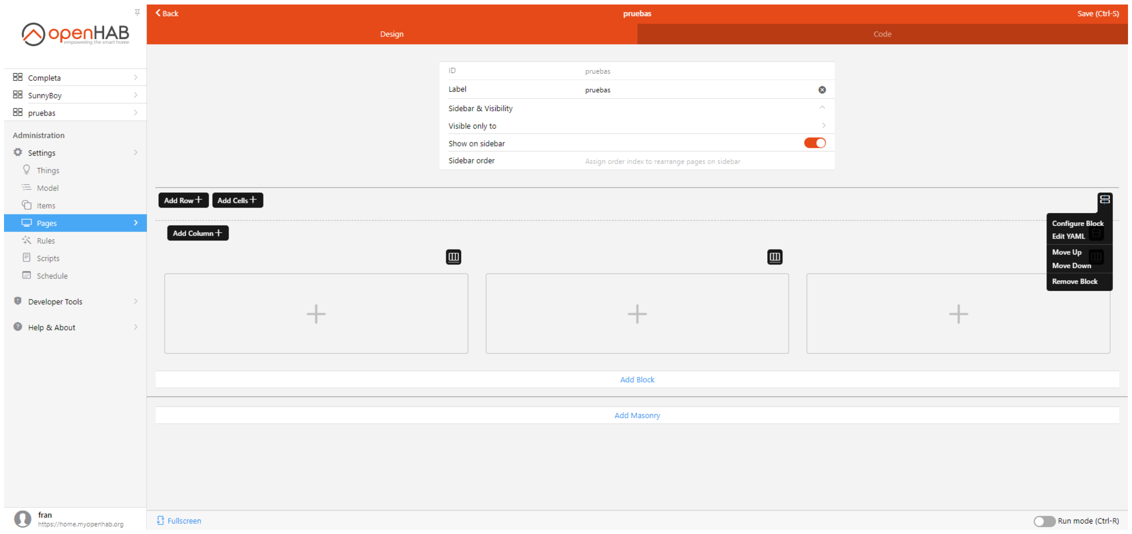
1. Once a page is created **Settings -> Pages -> + icon -> add layout tab**, we configure the name and label of this one, as well as the visibility options for the users of the system, if we want it to be only for users, only for admins or for both. Also if it is not the main page, the pages that we create will ask us if we want to be visible in the sidebar to be able to navigate to them and in what position should be the page created.



ID	pruebas
Label	pruebas
Sidebar & Visibility	
Visible only to	Administrators, Users
Show on sidebar	●
Sidebar order	3

Figure 5.151 – *Configure-page*

2. Once the initial parameters are configured, add a block by **clicking on Add Block**, add rows by clicking on **Add Row** and create columns by clicking on **Add column**. As we can see on the right side there is an icon, where a menu is displayed, where we can configure the blocks, rows and columns.



ID	pruebas
Label	pruebas
Sidebar & Visibility	
Visible only to	
Show on sidebar	●
Sidebar order	Assign order index to rearrange pages on sidebar

Figure 5.152 – *Option block*

3. To **add a widget**, click on the column we have created and a menu will be displayed to select the

widget, we can choose from several options:

- **Default widgets** (list of the different widgets).
- **Widgets created by us** (example).
- **Add from the model**, which puts the widget that best fits that item. This is the option that has been chosen

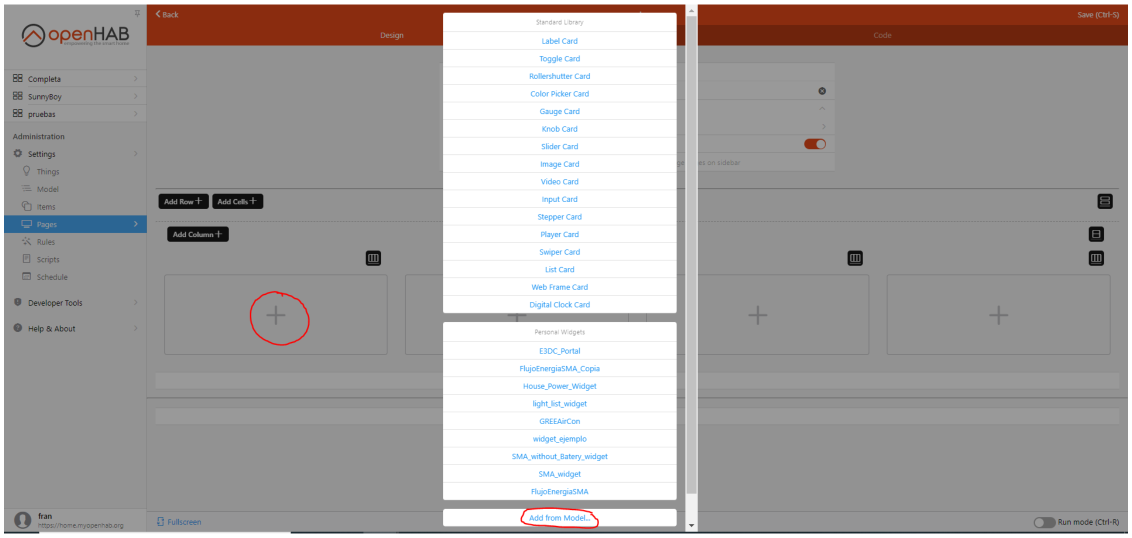


Figure 5.153 – *Add widget from semantic model*

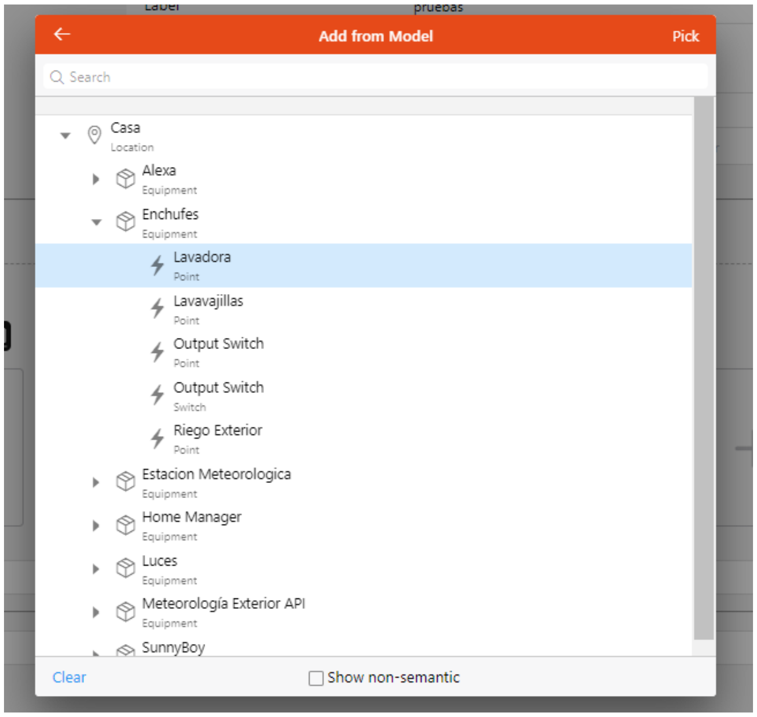
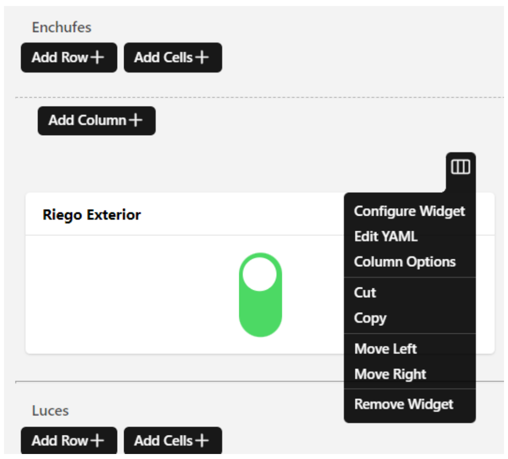


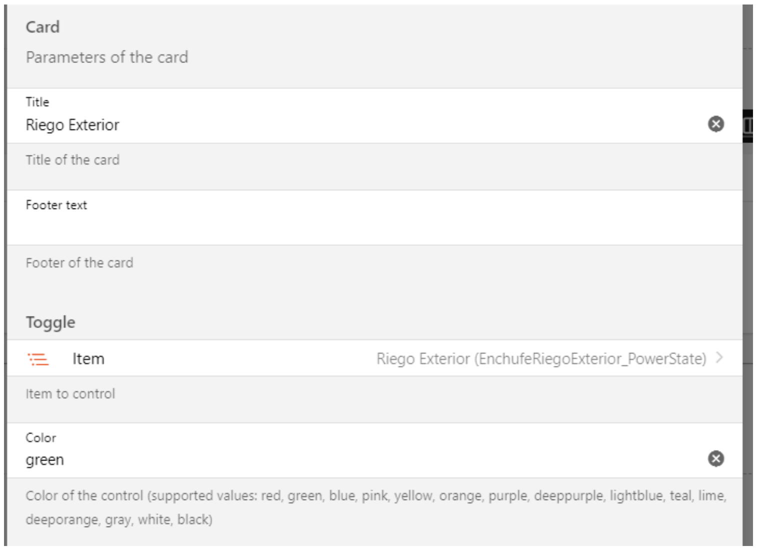
Figure 5.154 – *Select item from semantic model*

4. Once added, if we click on the **icon on the right of the column**, we will see an option to **configure**

widget, to change parameters, such as name, icon, colour, the item it is referring to, etc. Each widget has its own parameters to configure.



(a) *Configure widget*


Card
Parameters of the card

Title
Riego Exterior
Title of the card

Footer text

Footer of the card

Toggle
Item — Riego Exterior (EnchufeRiegoExterior_PowerState)
Item to control

Color
green
Color of the control (supported values: red, green, blue, pink, yellow, orange, purple, deeppurple, lightblue, teal, lime, deeporange, gray, white, black)

(b) *Configuration options*

Figure 5.155 – *Configure widget*

Here you can see a small piece of the **graphical interface page** I have created.

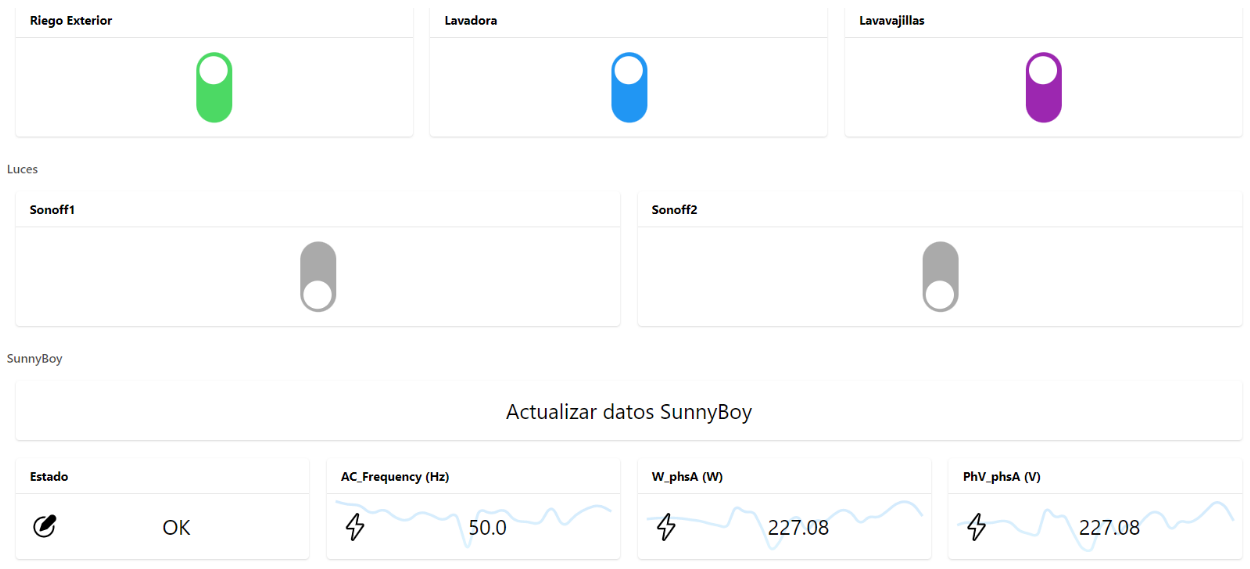


Figure 5.156 – *Page overview*

In terms of **graphs to represent the data stored** in the system, the pages use graphs, which are OpenHAB's native graphs, to represent the data obtained over time by means of a graph. These graphs can be modified to show the maximum, minimum and average values of the elements. The graph can be accessed by clicking on the **"Analyze" button** within the item or from the page created in the widget that is associated with an item showing a measurement, e.g. temperature. If you add the item from the model you will not have to configure anything, but in the widget configuration you will have to indicate the corresponding item in the analyze section.

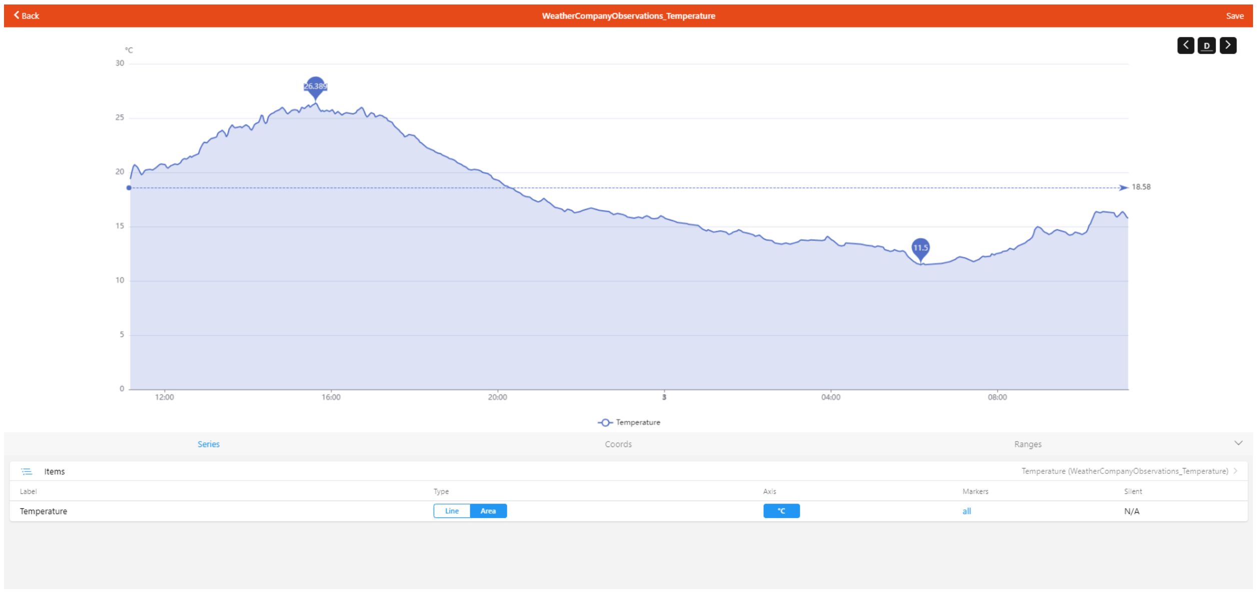


Figure 5.157 – *Item graph*

It is also possible to join **graphs of several items** into one, e.g. to compare the data of one item with another.

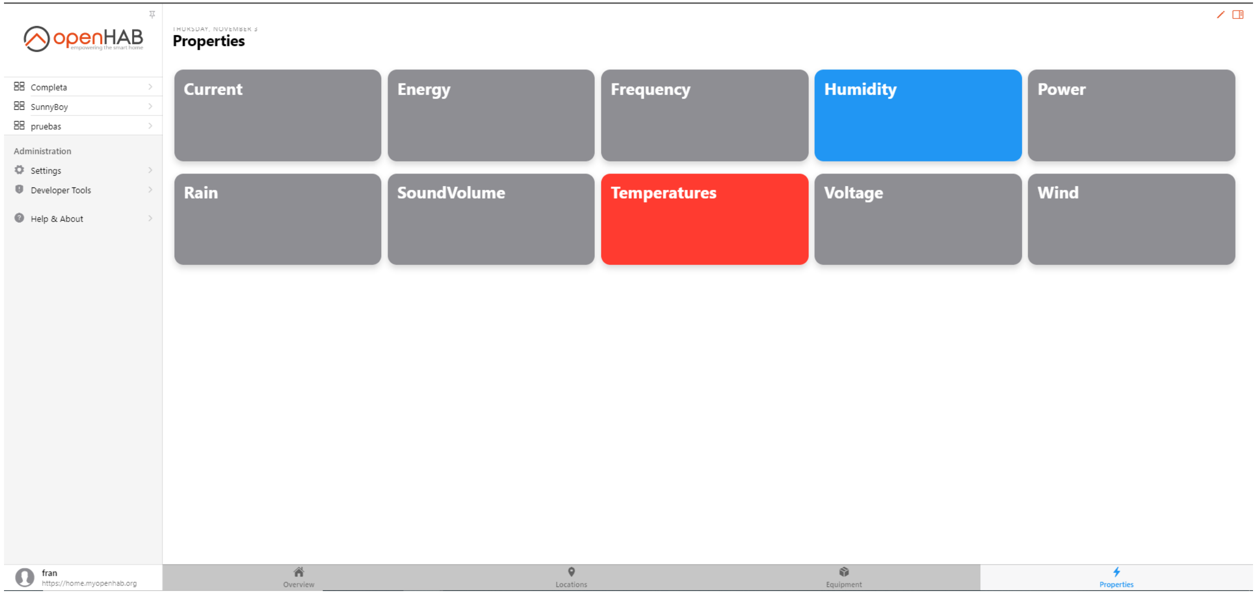
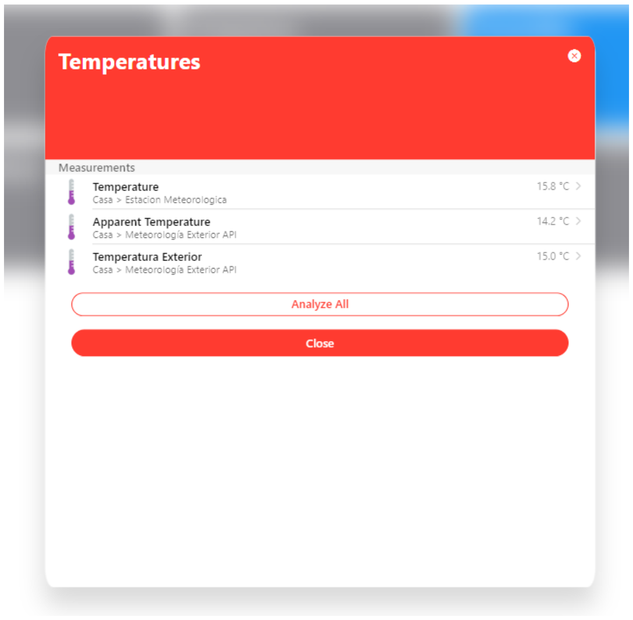
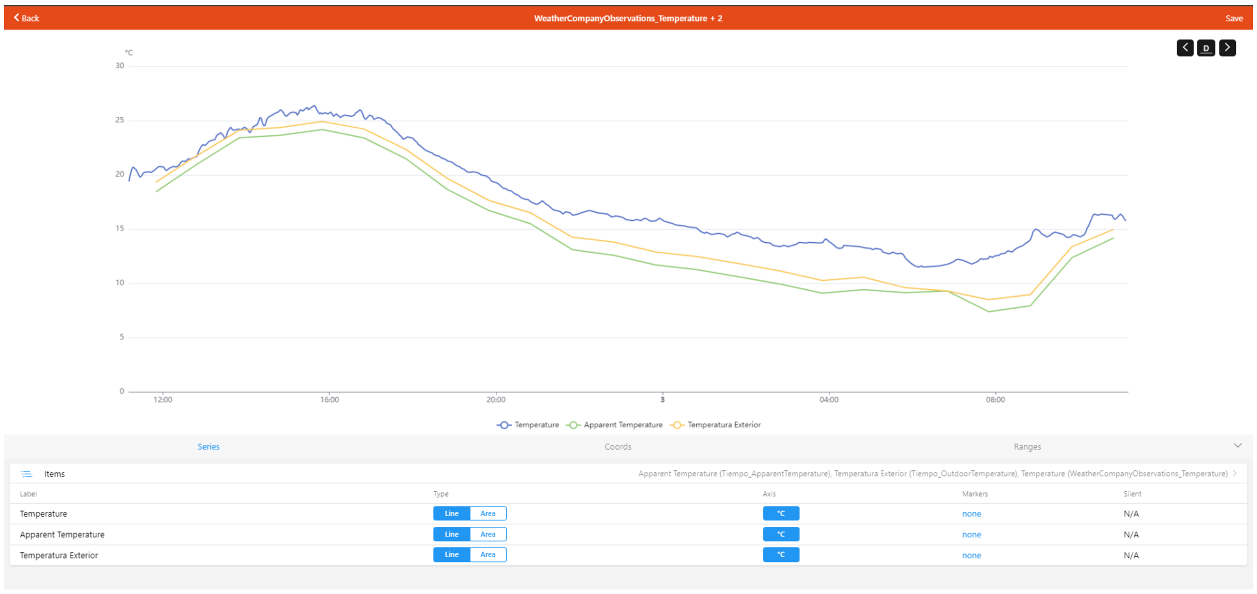


Figure 5.158 – *Groups according to measurements*



Temperature Casa > Estacion Meteorologica	15.8 °C
Apparent Temperature Casa > Meteorología Exterior API	14.2 °C
Temperatura Exterior Casa > Meteorología Exterior API	15.0 °C

Figure 5.159 – *Items in group according to measurement*



Apparent Temperature (Tiempo_ApparentTemperature), Temperatura Exterior (Tiempo_OutdoorTemperature), Temperature (WeatherCompanyObservations_Temperature)

Label	Type	Axis	Markers	Silent
Temperature	Line Area	°C	none	N/A
Apparent Temperature	Line Area	°C	none	N/A
Temperatura Exterior	Line Area	°C	none	N/A

Figure 5.160 – *Graph with various items*

5.11.3 Sitemaps

For the **mobile** it was decided to create a sitemap, to see another user interface that is **simpler and less heavy** for the mobile. Sitemaps exist since the first versions of OpenHAB, so you will find many examples in the documentation and in questions from the community.

Sitemaps can be created **from MainUI** (from **Settings-> Pages -> + icon -> create sitemap**) and from **text files**, creating the file in **/openhab_conf/sitemaps/**. For this development we chose to make the sitemap from text files, since pages can only be created from MainUI. Sitemaps are composed of:

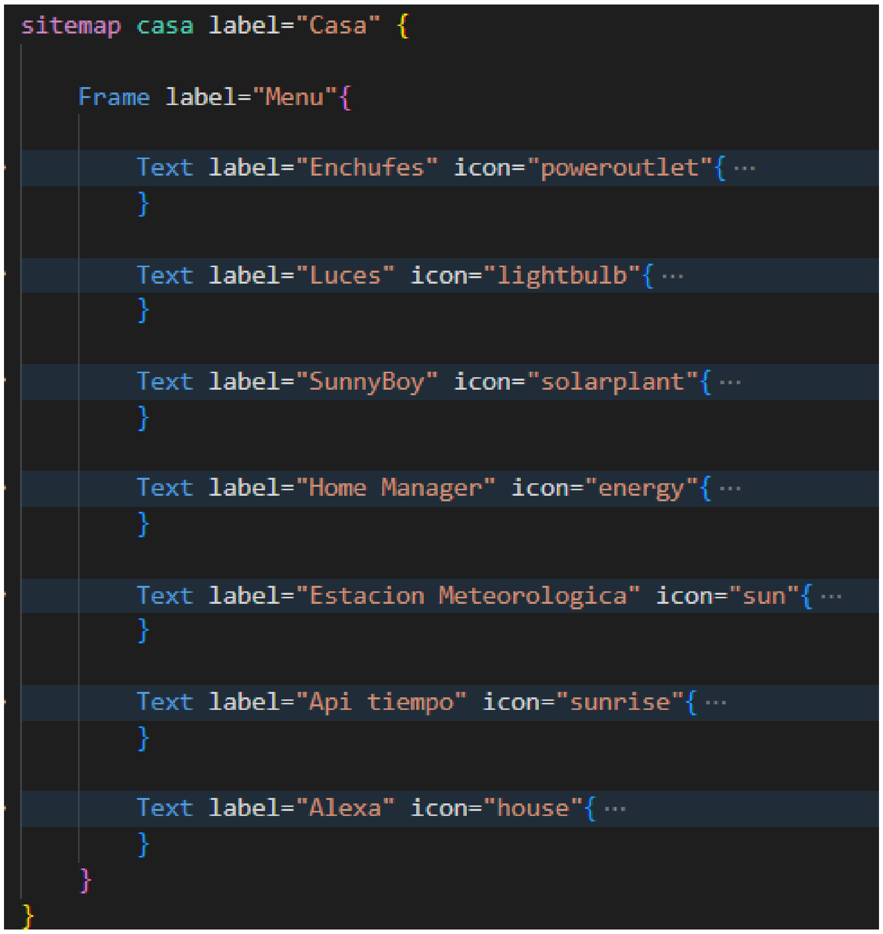
- **Elements** which present information and allow interaction with the item to which they are linked, here you can see a list of the different elements we can choose from.
- These elements have **parameters** that can be configured to customise the appearance of an element, for example to indicate the icon that the element will have, the label to indicate the text to be displayed or the item with which it will be associated.
  - **item** -> defines the name of the item to be submitted.
  - **label** -> sets the **description to be displayed** next to the item data to be processed, for example PowerTotal [%.2f W], this will display the name PowerTotal for the item and with the square brackets we are overwriting the state of the variable, in this case it will set the number to two decimal places and also overwrites the unit of the item.
  - **icon** -> we can put an icon next to the element, here you can see the list of icons that OpenHAB comes with.
- It is also possible to encapsulate several elements in what are known as **blocks**, for this purpose the Frame element is used or we can also use brackets behind the **Text, Group or Default elements**, but only those elements can be given the braces

There are what are called **dynamic sitemaps**, which are sitemaps whose icons are changed according to certain parameters, for example, if a plug is off, the icon is a red plug, and if it is on, the icon is a green plug (more info), or set items to be displayed only when a condition is satisfied (more info) [37].

Once explained all this, each user can create the sitemap as he likes and with the distribution that each one wants, I am going to explain how to create a sitemap:

5

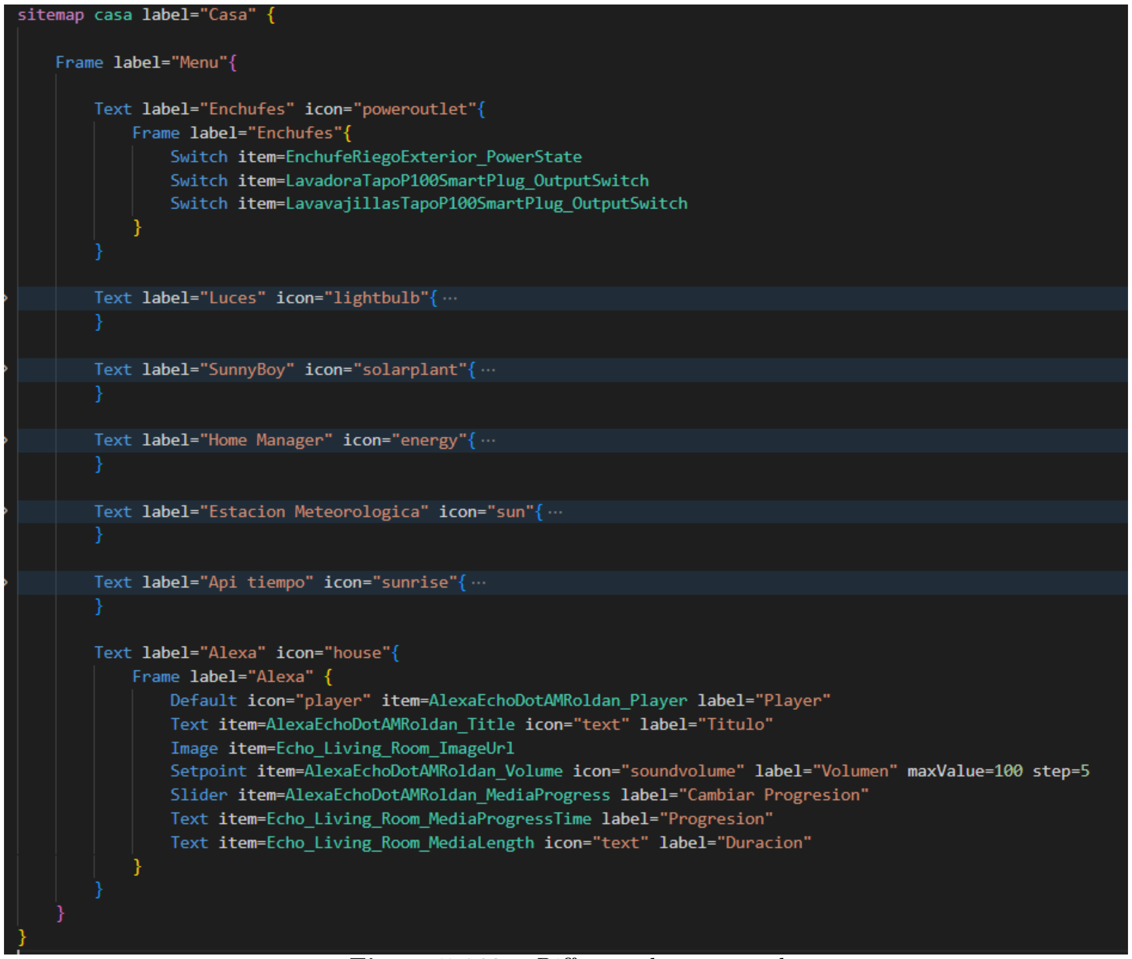
1. Create the text file to contain the sitemap in the folder **openhab_conf/sitemaps/**.
2. The main thing is the **sitemap element**, which will identify the file and give it a name.
3. I made a **menu**, of the different fields that the client wanted, like plugs, lights, SunnyBoy, etc. In this case I used a Frame to put a Menu text and Text elements to group the different items of the devices.



```
sitemap casa label="Casa" {

    Frame label="Menu"{

        Text label="Enchufes" icon="poweroutlet"{ ···
        }

        Text label="Luces" icon="lightbulb"{ ···
        }

        Text label="SunnyBoy" icon="solarplant"{ ···
        }

        Text label="Home Manager" icon="energy"{ ···
        }

        Text label="Estacion Meteorologica" icon="sun"{ ···
        }

        Text label="Api tiempo" icon="sunrise"{ ···
        }

        Text label="Alexa" icon="house"{ ···
        }
    }
}
```

Figure 5.161 – *Menu in sitemap file*

4. To **choose the right sitemap element**, it is best to look at the type of each item we have added by MainUI, and decide which one is the most suitable from the list. I use Switch for plugs and light switches, for number or text variables I use the Text element and for alexa items Setpoint for volume and slider for music progression, among others.



```
sitemap casa label="Casa" {

    Frame label="Menu"{

        Text label="Enchufes" icon="poweroutlet"{
            Frame label="Enchufes"{
                Switch item=EnchufeRiegoExterior_PowerState
                Switch item=LavadoraTapoP100SmartPlug_OutputSwitch
                Switch item=LavavajillasTapoP100SmartPlug_OutputSwitch
            }
        }

        Text label="Luces" icon="lightbulb"{ ···
        }

        Text label="SunnyBoy" icon="solarplant"{ ···
        }

        Text label="Home Manager" icon="energy"{ ···
        }

        Text label="Estacion Meteorologica" icon="sun"{ ···
        }

        Text label="Api tiempo" icon="sunrise"{ ···
        }

        Text label="Alexa" icon="house"{
            Frame label="Alexa" {
                Default icon="player" item=AlexaEchoDotAMRoldan_Player label="Player"
                Text item=AlexaEchoDotAMRoldan_Title icon="text" label="Titulo"
                Image item=Echo_Living_Room_ImageUrl
                Setpoint item=AlexaEchoDotAMRoldan_Volume icon="soundvolume" label="Volumen" maxValue=100 step=5
                Slider item=AlexaEchoDotAMRoldan_MediaProgress label="Cambiar Progresion"
                Text item=Echo_Living_Room_MediaProgressTime label="Progresion"
                Text item=Echo_Living_Room_MediaLength icon="text" label="Duracion"
            }
        }
    }
}
```

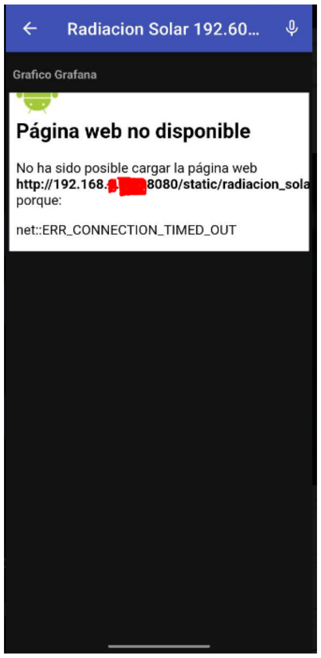
Figure 5.162 – *Different elements used*

5

For the **graphical representation of the data of the different items stored** in the database, there were two proposals:

- **Grafana** is a tool for visualising time series data. It is a very powerful software to make graphs and to be able to customize those graphs as you want. Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored [18].
- **OpenHAB Charts**, but not as page graphics, but more basic ones to reduce the load on the system, especially for sitemaps.

I wanted to test with **Grafana**, so I installed it following the same link as for installing InfluxDB, as they go hand in hand. But when showing the diagram it gave me a **connection error**, it had not been possible to connect to the server.



Radiacion Solar 192.60...

Grafico Grafana

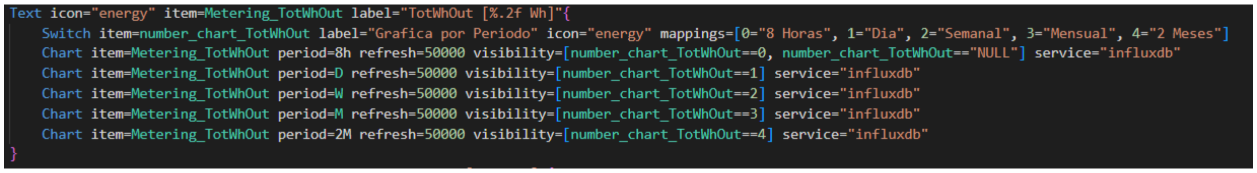
Página web no disponible

No ha sido posible cargar la página web http://192.168.[redacted]8080/static/radiacion_sola porque:

net::ERR_CONNECTION_TIMED_OUT

Figure 5.163 – *Error grafana*

So I decided to try with the **OpenHAB charts**, to reduce the use of RAM memory, because being native of OpenHAB it doesn't need another service running in the background, and these worked, next I'll explain how I implemented them. For each item I put 5 charts with different time intervals and I also included a switch to switch between these different charts.



```
Text icon="energy" item=Metering_TotWhOut label="TotWhOut [%.2f Wh]"{
    Switch item=number_chart_TotWhOut label="Grafica por Periodo" icon="energy" mappings=[0="8 Horas", 1="Dia", 2="Semanal", 3="Mensual", 4="2 Meses"]
    Chart item=Metering_TotWhOut period=8h refresh=50000 visibility=[number_chart_TotWhOut==0, number_chart_TotWhOut=="NULL"] service="influxdb"
    Chart item=Metering_TotWhOut period=D refresh=50000 visibility=[number_chart_TotWhOut==1] service="influxdb"
    Chart item=Metering_TotWhOut period=W refresh=50000 visibility=[number_chart_TotWhOut==2] service="influxdb"
    Chart item=Metering_TotWhOut period=M refresh=50000 visibility=[number_chart_TotWhOut==3] service="influxdb"
    Chart item=Metering_TotWhOut period=2M refresh=50000 visibility=[number_chart_TotWhOut==4] service="influxdb"
}
```

Figure 5.164 – *Chart in sitemap file*

As you can see in the picture, inside each Text element, which shows the item value, I included the switch and the charts:

- This **switch** is associated to an item type number that is created without being linked to any channel, which is in charge of storing which graph to display.
- The **parameter mappings**, serves to map according to the value of the switch, to show one or another graphic.
- The **Chart element** creates the corresponding chart indicating the period and how often it has to be refreshed, as well as the visibility of each one (dynamic sitemap) according to the switch and indicate where to take the data to create the chart, in this case InfluxDB.

A bug that came up during the development of showing the graphics is that all the parameters must be written without space between parameter and = in the definition file, otherwise it will give an error.

The charts we have included are for a **quick overview**, but if we **click on the graph**, we can access all the time intervals that the system offers us to visualise the data.

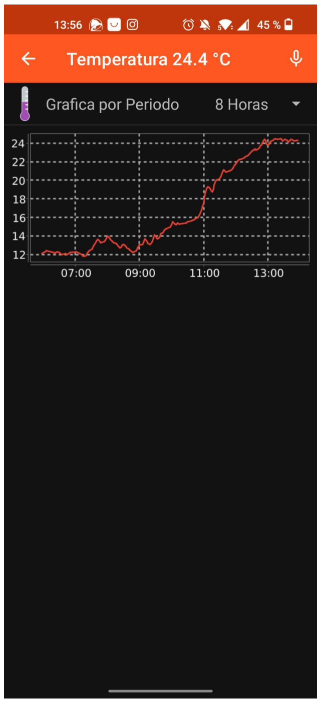
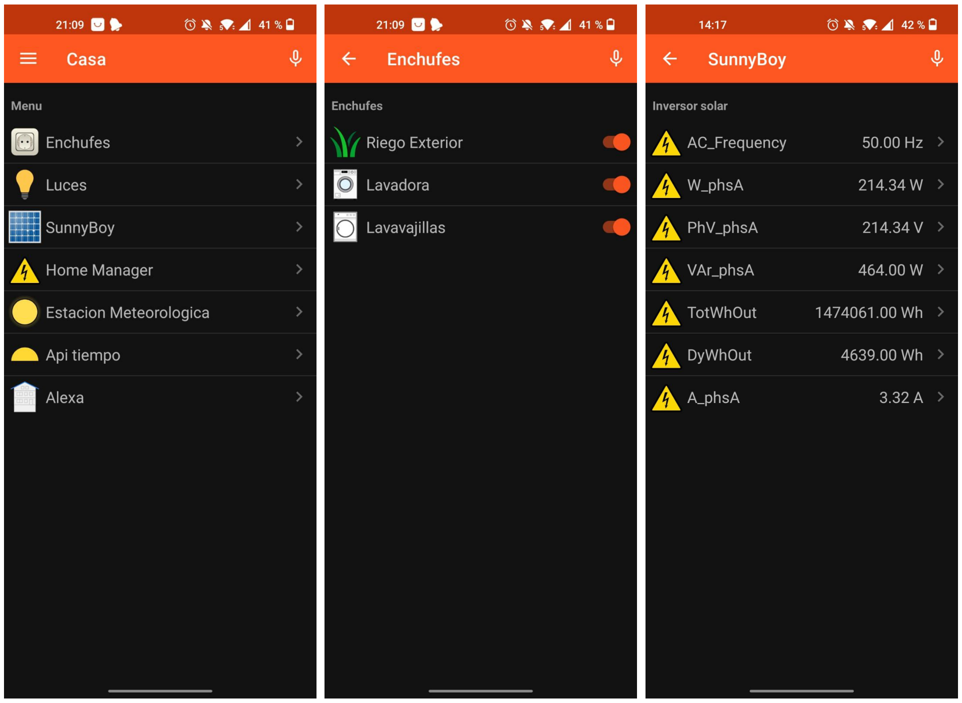


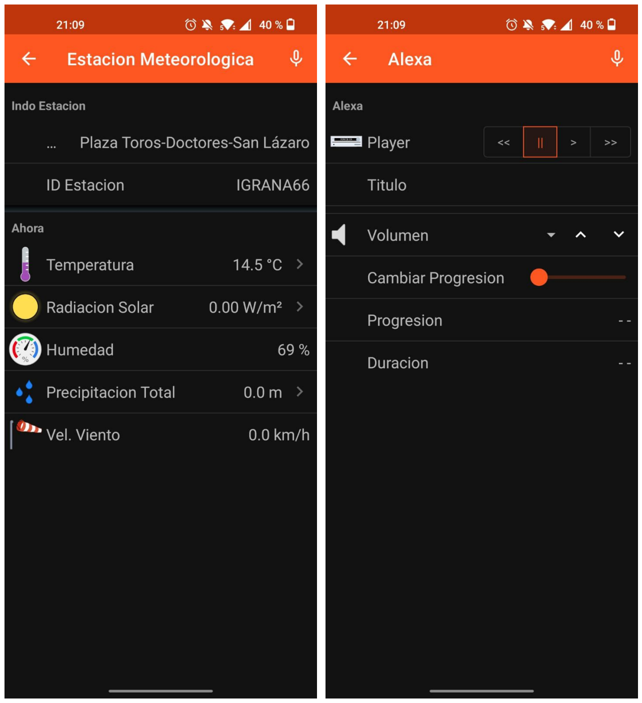
Figure 5.165 – *Charts in mobile app*

In the following images you can see how some of the screens of the application look like.



AC_Frequency	50.00 Hz
W_phsA	214.34 W
PhV_phsA	214.34 V
VAr_phsA	464.00 W
TotWhOut	1474061.00 Wh
DyWhOut	4639.00 Wh
A_phsA	3.32 A

(a) *Menu* (b) *Plugs* (c) *Sunny Boy*
Figure 5.166 – *Sitemap in mobile app part 1*



21:09 40 %

Estacion Meteorologica

Indo Estacion

... Plaza Toros-Doctores-San Lázaro

ID Estacion IGRANA66

Ahora

Temperatura	14.5 °C
Radiacion Solar	0.00 W/m²
Humedad	69 %
Precipitacion Total	0.0 m
Vel. Viento	0.0 km/h

21:09 40 %

Alexa

Alexa

Player << II > >>

Titulo

Volumen

Cambiar Progresion

Progresion - -

Duracion - -

(a) *Weather station* **(b)** *Alexa*

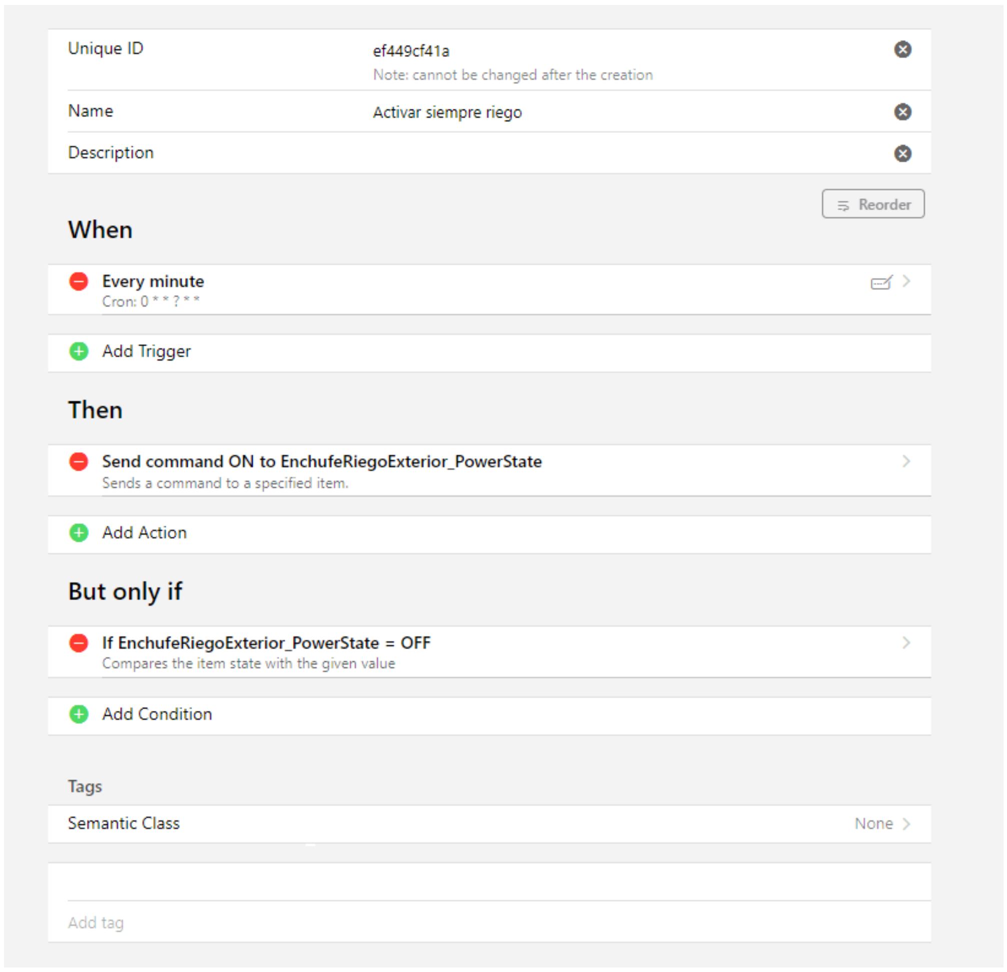
Figure 5.167 – *Sitemap in mobile app part 2*

5.12 Extras

In this section we will explain extra things, which have not been requested by the customer, like a rule to always keep the plug active and extra items like the image of the song that is playing on the echo dot speaker.

5.12.1 Rule Amazon plug

This rule will be done through the **MainUI** interface, it will be set to check every minute (with a cron expression) if the Amazon plug, in this case called "Riego Exterior", is turned off and if it is the case turn it on.



Unique ID	ef449cf41a Note: cannot be changed after the creation
Name	Activar siempre riego
Description	

When

- Every minute
  Cron: 0 * * ? * *
- Add Trigger

Then

- Send command ON to EnchufeRiegoExterior_PowerState
  Sends a command to a specified item.
- Add Action

But only if

- If EnchufeRiegoExterior_PowerState = OFF
  Compares the item state with the given value
- Add Condition

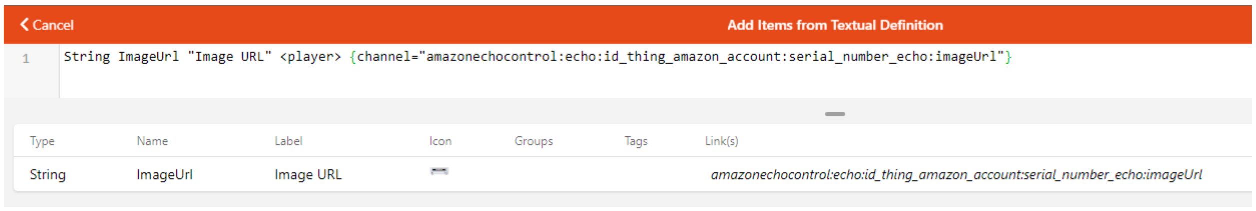
Tags

Semantic Class	None

Figure 5.168 – *Rule for plug in MainUI*

5.12.2 Additional item Amazon Echo dot

Looking at the Amazon echo control binding, I saw that you could get the image of the song that was playing by creating an item, so in **MainUI** go to **Settings-> Items -> + icon -> add from textual definition** and put in the following [32]:

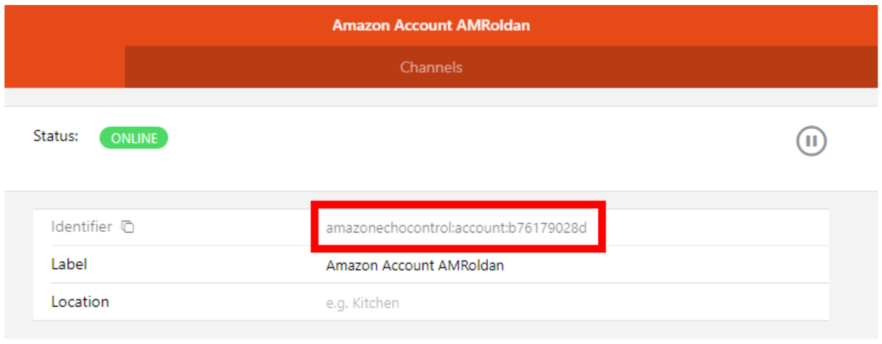


```
String ImageUrl "Image URL" <player> {channel="amazonechocontrol:echo:id_thing_amazon_account:serial_number_echo:imageUrl"}
```

Type	Name	Label	Icon	Groups	Tags	Link(s)
String	ImageUrl	Image URL				amazonechocontrol:echo:id_thing_amazon_account:serial_number_echo:imageUrl

Figure 5.169 – *Add Item extra*

To get the thing id of the amazon account just look at the identifier (the last alphanumeric string) inside the thing.



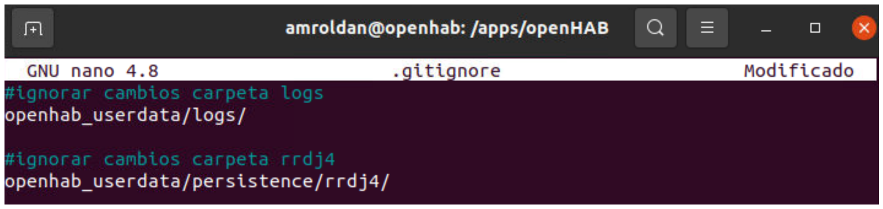
amazonechocontrol:account:b76179028d
Amazon Account AMRoldan

Figure 5.170 – *Identifier of amazon account thing*

5.13 Version control

Once everything is created, we will upload the proyect to a private repository in **GitLab** that Andres has created for our project, to save the changes in the project. We need to have a little knowledge of **git**, especially the basics. To upload the files to the repository we will have to do the following:

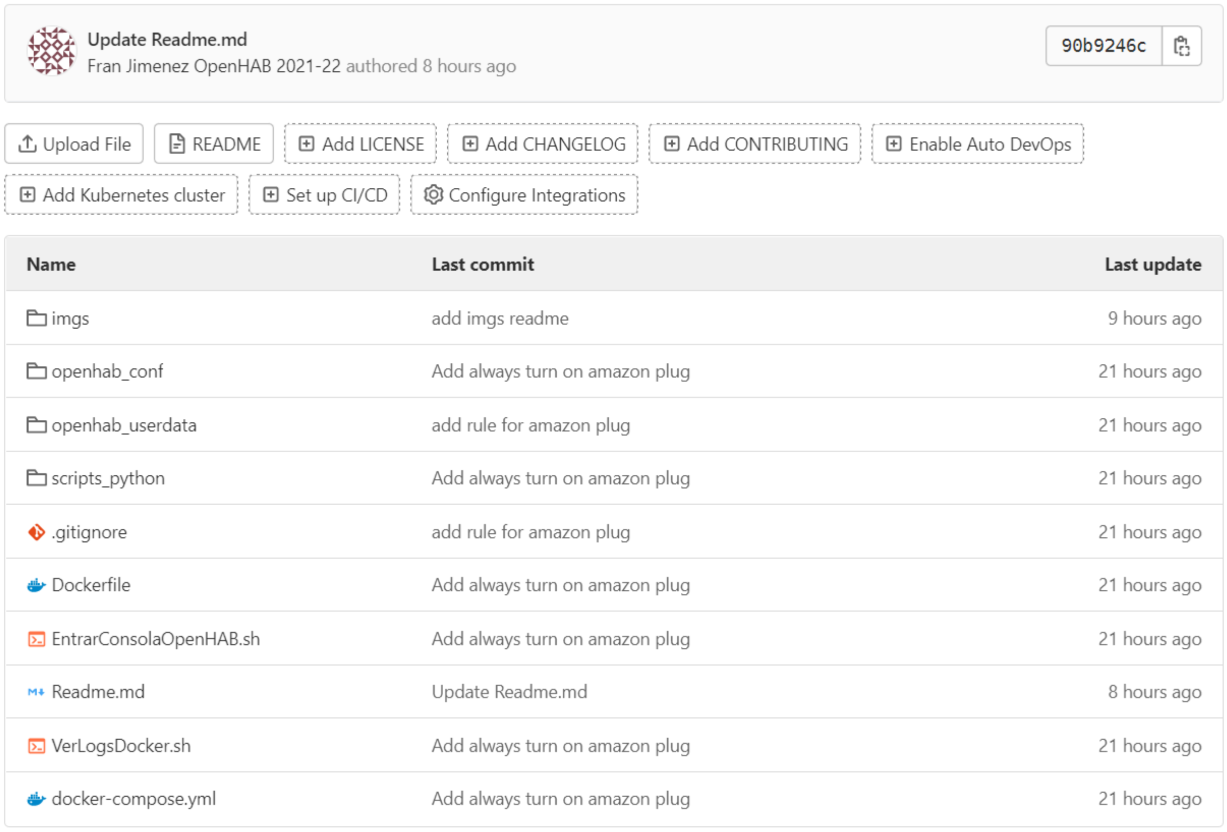
1. Have a repository created.
2. Start the local repository with **git init**.
3. Link our local repository to the remote repository **git remote add origin https://git.granasat.space/openhab/openhab.git**
4. Make the **.gitignore file** to not upload the logs and persistence folder of rrdj4, which is another default persistence service of OpenHAB, as these files are constantly being modified.



```
amroldan@openhab: /apps/openHAB
  GNU nano 4.8                  .gitignore                      Modificado
#ignorar cambios carpeta logs
openhab_userdata/logs/

#ignorar cambios carpeta rrdj4
openhab_userdata/persistence/rrdj4/
```

Figure 5.171 – *Gitignore file*

5. Add the files with **git add .**
6. Configure git user with **git config –global user.email "franlegaza@gmail.com"**.
7. Commit the files added to the local repository with **git commit -m "Description of what is to be uploaded"**.
8. Synchronising the working repositories with **git pull**. But when I did git pull I got a bug that the stories were not related. To fix it I used:
    - **git branch –set-upstream-to=origin/master master**, to link the local master branch to the master branch of the remote repository.
    - **git pull –allow-unrelated-histories origin master**, to fix the unrelated stories error [16]
9. Upload commit from local repository to remote repository with **git push -u origin master**.


Update Readme.md
Fran Jimenez OpenHAB 2021-22 authored 8 hours ago
90b9246c

Name	Last commit	Last update
imgs	add imgs readme	9 hours ago
openhab_conf	Add always turn on amazon plug	21 hours ago
openhab_userdata	add rule for amazon plug	21 hours ago
scripts_python	Add always turn on amazon plug	21 hours ago
.gitignore	add rule for amazon plug	21 hours ago
Dockerfile	Add always turn on amazon plug	21 hours ago
EntrarConsolaOpenHAB.sh	Add always turn on amazon plug	21 hours ago
Readme.md	Update Readme.md	8 hours ago
VerLogsDocker.sh	Add always turn on amazon plug	21 hours ago
docker-compose.yml	Add always turn on amazon plug	21 hours ago

Figure 5.172 – *Repository*

Chapter 6

System verification and testing

This section will present the tests carried out to check the operation of the devices together with the assembled OpenHAB system and an evaluation of the requirements outlined in chapter two will be carried out

6.1 Alexa player test

In this test, it was verified that once Amazon Alexa was implemented in the system, when Alexa was asked to play a song, display the name of the song, the volume and change the progression of the song, it could be operated from both OpenHAB and the Amazon Alexa application.

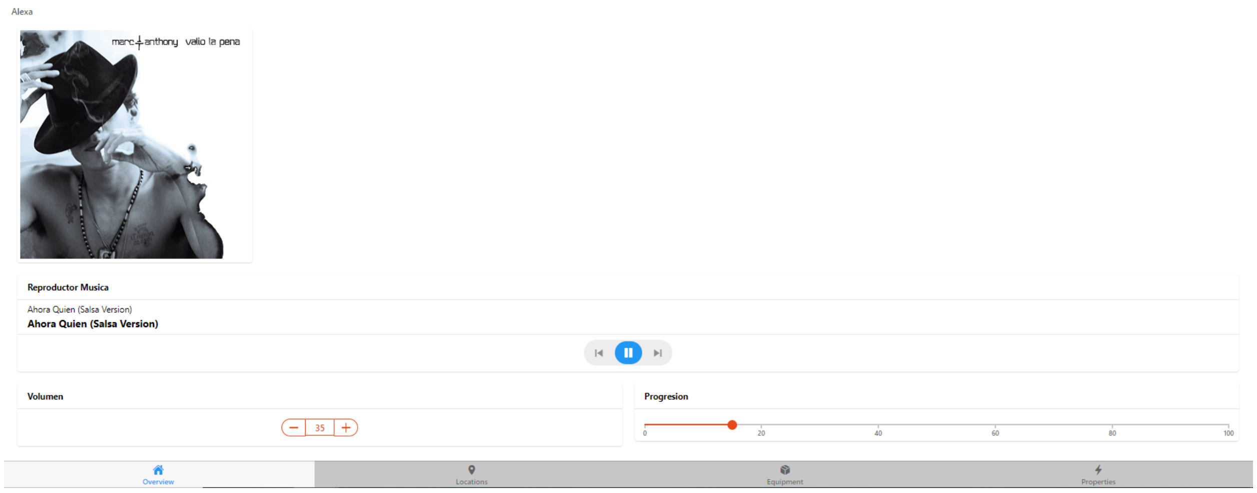


Figure 6.1 – *Alexa Player*

One thing that was discovered in the volume is that if you put a slider widget in the user interfaces, it gets stuck, because when we are sliding the slider, it is sending requests to change the volume, so it is advisable to put a setPoint or go changing in the slider with touches not sliding.

6.2 Voice control test

In this test we checked that when we ask Alexa to turn on a plug, it changes both in the Alexa app and in OpenHAB. To do this, we ask Alexa to turn on the outdoor watering plug and see if the value changes in the Amazon Alexa app and in OpenHAB.

We noticed that it takes more or less a minute or so before the item is updated in OpenHAB and vice versa.

6.3 Control test

This test is like the previous one in section 6.2, but from the application of each device. To carry it out, we open the application of a device, for example Amazon Alexa, and we turn on or off a device to see if it changes in OpenHAB.

We came to the same conclusion that it takes more or less a minute or so to update the item in OpenHAB and vice versa.

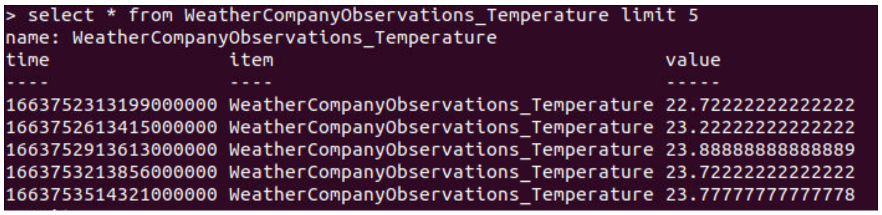
6.4 Results test

In this test we checked that the items linked to measurements such as the weather API, weather station and solar energy were close to reality. We had to look at the data we were getting in OpenHAB and see if these values varied a lot from reality, for example looking at the solar energy data and seeing what the manufacturer offers us in their platforms.

We can conclude that the data we obtain in OpenHAB does not vary much from reality.

6.5 Test data is stored in the database.

This test will check that the values of the items placed in the persist file in the /openhab_conf/persist folder are being saved in the database.

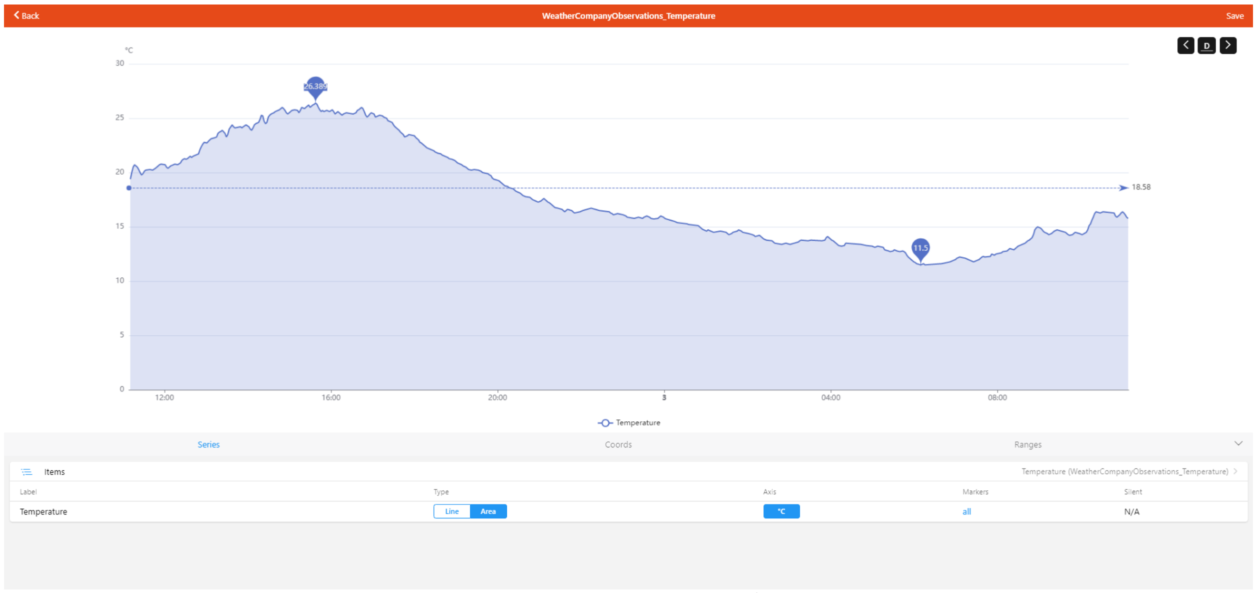


```
> select * from WeatherCompanyObservations_Temperature limit 5
name: WeatherCompanyObservations_Temperature
time                item                                  value
----                ----                                  -----
1663752313199000000 WeatherCompanyObservations_Temperature 22.72222222222222
1663752613415000000 WeatherCompanyObservations_Temperature 23.22222222222222
1663752913613000000 WeatherCompanyObservations_Temperature 23.88888888888889
1663753213856000000 WeatherCompanyObservations_Temperature 23.72222222222222
1663753514321000000 WeatherCompanyObservations_Temperature 23.77777777777778
```

Figure 6.2 – *Show data in data base*

6.6 Test charts show the data in the database.

In this test we check that OpenHAB in the interfaces that we have created in the section 5.11 the graphs that we obtain, agree with the values that we have saved in the InfluxDB database. To check this, we go into the graph of an item and into the table corresponding to the item in the InfluxDB database.



Label	Type	Axis	Markers	Silent
Temperature	Line [✓ Area]	°C	all	N/A

Figure 6.3 – *item graph*

6.7 Test rules update solar energy items

In this test we will check that the items that we have created for the values of the variables of the scripts, are updating well with the execution value of the script and according to the cron expression set in the rule. To do this we will use the log functions to debug the rule, showing in the log the execution of the script and seeing if the values that the script has taken out with those who are putting in the item.

6.8 Test shutdown docker server

This test will check that if the docker container or the computer is turned off, everything is persistent without losing data and still works when the docker container or the computer is turned on again. It is possible that some changes are not persistent. The mobile app and OpenHAB Cloud will not be usable when the docker container is turned off, whatever happens on the devices or services while the docker container or computer is turned off cannot be saved in the database, e.g. temperature changes.

6.9 Interaction test with user interfaces

In this test it was checked whether when clicking on the different widgets of the user interface page or elements of the user interface sitemap, the system responded by performing the action that was linked to the corresponding item, e.g. turn on or off the light or the plugs.

6.10 Evaluation of satisfaction of requirements

Next, the functional requirements of the project will be analysed. To do so, we study whether the resulting product can satisfy each of the functional requirements of the project:

Requirement	FR.1- The system shall be able to allow only correctly identified and verified administrator users to modify system configurations
Section	Installation
Analysis	Satisfies the requirement with system authentication by only allowing the configuration to be changed if the user is logged in as an administrator
Evaluation	Validated

Table 6.1 – *FR 1. Satisfaction Evaluation*

Requirement	FR.2- The system shall store the data obtained from the different devices in a database to guarantee their persistence.
Section	Data persistence
Analysis	Requirement satisfied in which the statuses of the different items of the devices and services, as well as the values of the solar energy scripts are stored in a specific database.
Evaluation	Validated

Table 6.2 – *FR 2. Satisfaction Evaluation*

Requirement	FR.3- The system shall represent the data obtained by means of graphs for a better visibility of the data
Section	Pages and Sitemaps
Analysis	Requirement satisfied with the incorporation of graphs in the different user interfaces, which take the stored data from the database.
Evaluation	Validated

Table 6.3 – *FR 3. Satisfaction Evaluation*

Requirement	FR.4- The user shall be able to operate the system by means of a device (computer/mobile) outside the local network
Section	OpenHAB Cloud and OpenHAB Mobile App
Analysis	Requirement satisfied with the incorporation of OpenHAB Cloud being able to access the system via a device outside the local network
Evaluation	Validated

Table 6.4 – *FR 4. Satisfaction Evaluation*

Requirement	FR.5- The user will be able to make use of the different devices installed, activating and deactivating them at will
Section	Smart Plugs, Sonoff Mini Zigbee Relays and Amazon Alexa
Analysis	Requirement satisfied with the incorporation of different physical devices in the system and being able to interact with them through the different user interfaces created.
Evaluation	Validated

Table 6.5 – *FR 5. Satisfaction Evaluation*

Requirement	FR.6- The user will be able to control by voice the devices by means of a virtual assistant and that everything is compatible with this one
Section	Amazon Alexa
Analysis	Requirement satisfied with the incorporation of Amazon Alexa into the system and thanks to its skills it was able to connect with other devices of other brands.
Evaluation	Validated

Table 6.6 – *FR 6. Satisfaction Evaluation*

Requirement	FR.7- The system shall obtain information from the solar energy devices, Sunny Boy and Home Manager, for display in the system
Section	Solar inverter and Home Manager
Analysis	Requirement satisfied thanks to the scripts developed by the tutor Andres, which have been included in the system to obtain the data of the variables of the devices related to solar energy.
Evaluation	Validated

Table 6.7 – *FR 7. Satisfaction Evaluation*

Requirement	FR.8- The system shall obtain information from a weather station for display in the system
Section	Weather station
Analysis	Requirement satisfied with the incorporation and configuration of a weather station in the system.
Evaluation	Validated

Table 6.8 – *FR 8. Satisfaction Evaluation*

Requirement	FR.9- The system shall obtain information from a meteorological API for display on the system
Section	Weather API and Weather station
Analysis	Requirement satisfied with the integration and configuration of a weather API in the system
Evaluation	Validated

Table 6.9 – *FR 9. Satisfaction Evaluation*

Requirement	FR.10- The system shall be able to obtain metrics of the weather forecast of the future weather for its visualisation in the system
Section	Weather API and Weather station
Analysis	Requirement satisfied with the integration and configuration of a meteorological API and a weather station in the system
Evaluation	Validated

Table 6.10 – *FR 10. Satisfaction Evaluation*

Requirement	FR.11- The system shall group the devices according to their functions
Section	Semantic Model and Pages
Analysis	Requirement satisfied with the configuration of the semantic model, so that the user interface pages show grouped according to the unit of measurement, the different functions of the devices added to the system
Evaluation	Validated

Table 6.11 – *FR 11. Satisfaction Evaluation*

Requirement	FR.12- The user shall have the possibility to modify the time intervals in the graphs of the data obtained from the different devices
Section	Pages and Sitemaps
Analysis	Requirement satisfied thanks to OpenHAB's native charts, in which you can modify the time interval in which you want to visualize the data in the graph.
Evaluation	Validated

Table 6.12 – *FR 12. Satisfaction Evaluation*

Requirement	FR.13- The system shall include interfaces for the user to control the system devices
Section	Pages and Sitemaps
Analysis	Requirement satisfied with the creation and configuration of the different user interfaces to interact with the items added to the system..
Evaluation	Validated

Table 6.13 – *FR 13. Satisfaction Evaluation*

Requirement	FR.14- The user shall be able to control and play music from the system
Section	Amazon Alexa
Analysis	Requirement satisfied with the implementation and configuration of the smart speaker with Amazon Alexa virtual assistant.
Evaluation	Validated

Table 6.14 – *FR 14. Satisfaction Evaluation*

Chapter 7

Conclusions, future work and lessons learned

7.1 Conclusions

In this document we have tried to explain the installation and configuration of a centralized home automation system based on open source software called OpenHAB, with this software we managed to group intelligent home automation devices from different brands under the same application.

The beginning of the project was hard, because I didn't know very well how things worked, besides I couldn't dedicate much time to it due to the company practices, so it was hard for me to learn and carry out the project.

During the development, I encountered a lot of problems, as I had to use technologies that were new to me, so I knew almost nothing about them. For example Docker, I had previously worked with Docker but not to the point of modifying an already created image. Also, I highlight the task that was undoubtedly the most difficult was getting the data from the solar energy scripts, due to the large amount of time I spent trying to find out why they were not running. Also, the different solutions I found were useless because they were old and not adapted to the new devices. Also, another inconvenience could be that the server was not operational 24 hours a day due to power outages at the client's house, which further delayed the development of the project. So, there were times when I thought I would not finish the project, but little by little I was making significant progress, creating in myself a motivation to continue.

To date, I can conclude that the results obtained have been satisfactory and that the system is 100% functional and operative, with an easy use, fulfilling the objectives and requirements of the project successfully. In this project I have demonstrated not only the knowledge acquired during the degree in Computer Engineering but I have also learnt new technologies or areas such as domotics.

As a final comment, I would like to point out that although this project has been a hard and exhausting journey, in which I have had very low and frustrating points where I have felt lost and defeated, I have managed to finish in a more than satisfactory way, and feeling very proud of my effort and the new skills I have learnt, which I am sure I will use in future work and projects.

7.2 Future work

In this project we have achieved all the objectives that the client has proposed, but as it was said at the beginning of the project, this project is the base to make a much bigger and customize system, here are some ideas:

- Home:
  - Improve the home WiFi network architecture by unifying the WiFi networks so that there is only a single sub-network, for better communication between devices.
- Devices:
  - Add sensors, to create automations within the system, e.g. motion sensors, so that when motion is detected, the light in the room is switched on.
  - A very interesting device is an IR control, to control electronic devices that work with an IR remote control.
  - Include the presence of the user according to the mobile phone, to activate routines when arriving home.
  - Include the possibility for the system to send messages via WhatsApp or Telegram to our phone.
  - Matter is a new standard for home automation, in which many companies that develop home automation devices are going to include in their devices. This new standard is a communication protocol so that any device can work with any assistant, making it easier to find new devices without worrying about whether it is compatible with our assistant or ecosystem [10].
- User interfaces:
  - Improve the sitemaps with dynamic icons, depending on the status of the item.
  - Add custom widgets to the pages, to show the information in a more fun way.

7.3 Lessons learned

Being part of a project of this size, in which I knew absolutely nothing, but found the idea interesting, led me to the situation of pushing myself to my limits and knowledge to try to do my best. That is why I wanted to conclude this project with a list of the lessons I have learned during the development of this thesis:

- Learn about domotics, how it works, how it communicates and how the different domotics devices are used and incorporated.
- Learn how to analyse and justify which home automation device is the most suitable according to the need we want to cover.
- Learning more about Docker, I had worked with Docker, but not to the point of building and knowing how to modify an image and add what we needed to meet a customer need.
- Study the different methods for implementation and choose the one that I think is the best for the incorporation of the device in the system.

Bibliography

[1] Domoticz. https://www.domoticz.com/.

[2] Home assistant. https://www.home-assistant.io/.

[3] Jeedom. https://www.jeedom.com/fr/.

[4] La vanguardia, smart plugs comparative 2. https://www.lavanguardia.com/comprar/comparativas/comparativa-mejores-enchufes-intelitgentes/.

[5] Openhab. https://www.openhab.org/.

[6] Adslzone. Upnp. https://www.adslzone.net/reportajes/internet/upnp-router/.

[7] Amazon, A. What is docker. https://aws.amazon.com/es/docker/#:~:text=Docker%20es%20una%20plataforma%20de,probar%20e%20implementar%20aplicaciones%20r%C3%A1pidamente&text=a%20usar%20Docker-,Docker%20es%20una%20plataforma%20de%20software%20que%20le%20permite,probar%20e%20implementar%20aplicaciones%20r%C3%A1pidamente.

[8] Cambioenergetico. Solar inverters comparative. https://www.cambioenergetico.com/blog/comparativa-de-inversores-fotovoltaicos-para-autoconsumo-en-vivienda-actualizado-2019/.

[9] Community, O. Relation between system concepts. https://community.openhab.org/t/lets-talk-about-oh-2-drawings/13096/8.

[10] ComputerHoy. What is matter. https://computerhoy.com/noticias/tecnologia/matter-ya-oficial-te-afectara-nuevo-estandar-amazon-apple-google-cambia-dispositivos-inteligentes-

[11] Dev.to. Api info. https://dev.to/develawyer/que-son-api-s-para-dummies-2adj#:~:text=La%20interfaz%20de%20programaci%C3%B3n%20de,software%20como%20una%20capa%20de.

[12] Docker. Docker-compose. https://docs.docker.com/compose/.

[13] Docker. Install openhab in docker. https://hub.docker.com/r/openhab/openhab/.

[14] Domoticadomestica. Iftttt. http://www.domoticadomestica.com/tag/ifttt/.

[15] Domoticasa. Virtual assistant comparative. https://domoticasa.net/alexa-vs-google-home-vs-siri/comparacionasistentes.

[16] EDTeam. Error historias git. https://ed.team/comunidad/error-al-hacer-git-pull-origin-master.

[17] Estacionesmeteorologicas. Weather station information. https://www.estacionesmeteorologicas.top/blog/que-es-y-para-que-sirve-una-estacion-meteorologica/.

[18] Grafana. Granafana documentation. https://grafana.com/docs/grafana/latest/.

[19] Hartman, J. What is java. https://www.guru99.com/java-platform.html.

[20] Hogarsolarenergia. Solar inverter info. https://hogarsolarenergia.es/mejor-inversor-solar/.

[21] Hostgator. Virtual assistant definition. https://www.hostgator.mx/blog/alexa-siri-asistente-virtual-inteligente/definicionasistentevirtual.

[22] Hostinger. Ssh. https://www.hostinger.es/tutoriales/que-es-ssh#:~:text=SSH%20o%20Secure%20Shell%2C%20es,de%20un%20mecanismo%20de%20autenticaci%C3%B3n.

[23] Keepcoding. Dockerfile. https://keepcoding.io/blog/que-es-dockerfile/.

[24] Meteoclimatic. Froggit wh3000 se names. https://forum.meteoclimatic.net/index.php?topic=2559.0.

[25] Mipodo. Monofasica vs trifasica. https://www.mipodo.com/blog/informacion/una-instalacion-monofasica-trifasica/.

[26] Mozilla, D. Html. https://developer.mozilla.org/es/docs/Web/HTML.

[27] narrative, K. What is unix time. https://kb.narrative.io/what-is-unix-time.

[28] Nerdomus. Virtual assistant definition 2. https://nerdomus.com/googlehome-vs-alexa/paraquesirvenlosasistentesvirtuales.

[29] OpenHAB. Definition states things. https://www.openhab.org/docs/concepts/things.html#thing-status.

[30] OpenHAB. Info console. https://www.openhab.org/docs/administration/console.html#the-console.

[31] OpenHAB. Info text files vs mainui. https://www.openhab.org/docs/configuration/#versatility.

[32] OpenHAB. Items ejemplo binding amazon. https://www.openhab.org/addons/bindings/amazonechocontrol/#example.

[33] OpenHAB. Items types. https://www.openhab.org/docs/concepts/items.html#items.

[34] OpenHAB. Openhab cloud. https://www.openhab.org/addons/integrations/openhabcloud/#openhab-cloud-connector.

[35] OpenHAB. Openhab ui. https://www.openhab.org/docs/ui/.

[36] OpenHAB. Rules introduction. https://www.openhab.org/docs/tutorial/rules_introduction.html.

[37] OpenHAB. Sitemap info. https://www.openhab.org/docs/ui/sitemaps.html.

[38] OpenHAB. State diagram. https://www.openhab.org/docs/concepts/things.html#status-transitions.

[39] OpenHAB. What is openhab cloud. https://github.com/openhab/openhab-cloud/blob/master/README.md.

[40] Oracle. Internet of things. https://www.oracle.com/in/internet-of-things/what-is-iot/.

[41] Redhat. Api rest info. https://www.redhat.com/es/topics/api/what-is-a-rest-api.

[42] Redhat. What-is-a-rest-api. https://www.redhat.com/en/topics/api/what-is-a-rest-api.

[43] Rincondomotica. Openhab. https://rincondomotica.com/openhab-la-mejor-plataforma-domotica-open-source.

[44] SMA. Energy meter info. https://www.sma.de/es/productos/monitorizacion-y-control/sma-energy-meter.

[45] SMA. Sunny boy info. https://www.sma.de/es/productos/inversor-fotovoltaico/sunny-boy-30-36-40-50-60.

[46] Solarplak. What is mppt. https://solarplak.es/energia/que-es-un-regulador-mppt/.

[47] Solutioon. Comparative open source home automation system 2. https://www.solutioon.com/que-sistema-domotico-libre-usar-en-nuestro-hogar/.

[48] Sotysolar. Solar inverter info 2. https://sotysolar.es/blog/que-son-los-inversores-fotovoltaico.

[49] TeamViewer. What is teamviewer. https://www.teamviewer.com/es/productos/teamviewer/.

[50] Techopedia. What is c. https://www.techopedia.com/definition/24068/c-programming-language-c.

[51] Tecmint. Disable suspend and hibernation in linux. https://www.tecmint.com/disable-suspend-and-hibernation-in-linux/.

[52] teradata. What-is-python. https://www.teradata.com/Glossary/What-is-Python.

[53] Uva, E. What is c++. https://www2.eii.uva.es/fund_inf/cpp/temas/1_introduccion/introduccion.html.

[54] Wikipedia. Api definition. https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones.

[55] Wikipedia. Bluetooth. https://en.wikipedia.org/wiki/Bluetooth.

[56] Wikipedia. Http. https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol.

[57] Wikipedia. Https. https://es.wikipedia.org/wiki/Protocolo_seguro_de_transferencia_de_hipertexto.

[58] Wikipedia. Solar inverter info. https://es.wikipedia.org/wiki/Inversor_fotovoltaico.

[59] Wikipedia. What is json. https://en.wikipedia.org/wiki/JSON.

[60] Wikipedia. What is modbus. https://en.wikipedia.org/wiki/Modbus.

[61] Wikipedia. What is mqtt. https://en.wikipedia.org/wiki/MQTT.

[62] Wikipedia. What is raspberry pi. https://es.wikipedia.org/wiki/Raspberry_Pi.

[63] Wikipedia. What is xml. https://en.wikipedia.org/wiki/XML.

[64] Wikipedia. Wifi. https://en.wikipedia.org/wiki/Wi-Fi.

[65] Wikipedia. Z-wave. https://en.wikipedia.org/wiki/Z-Wave.

[66] Wikipedia. Zigbee. https://en.wikipedia.org/wiki/Zigbee.

[67] Xataka. Comparative open source home automation system 2. https://domoticfy.com/openhab-vs-home-assistant-vs-domoticz/.

[68] XATAKA. Smart plugs comparative 1. https://www.xataka.com/seleccion/que-enchufes-inteligentes-posibilidades-modelos-destacados.

[69] XATAKA. What is a nas. https://www.xataka.com/basics/servidores-nas-que-como-funcionan-que-puedes-hacer-uno.