

Hub de sensores y Controladores IoT con repositorio cloud y control desde app móvil

Abel Borrega Alvarez

Abstract – Este proyecto pretende crear un sistema domótico capaz de adaptarse a las necesidades que tenga el usuario. Dependiendo del ámbito de uso en el que se le quiera dar uso (personal, industrial o instalación agrícola remota) este usuario buscará unas especificaciones diferentes. Para satisfacer estas necesidades usaremos sensores que nos permitirán recoger información del entorno. Además, controlaremos dichos sensores a través de un dispositivo móvil y recogeremos los datos en un repositorio cloud para poder controlar el sistema a lo largo del tiempo.

Palabras clave – IoT, Hub, cloud, sensores, raspberry

1. Introducción:

Automatización de escenarios

El uso de tecnología en aspectos cotidianos, que permite controlar y monitorizar lo que sucede en el hogar, fábrica o cualquier otro tipo de instalación, ha aumentado enormemente en los últimos años. Debido a esto, hay empresas que ofrecen un servicio domótico, encargándose tanto de la instalación como del mantenimiento del mismo. Esto provocaba que al tener cada compañía su propio hardware, el usuario no pudiera cambiar de compañía sin modificar toda la tecnología que ya tenía instalada.

Para solucionar esta problemática se busca usar software libre para no depender de un proveedor específico. Con respecto al hardware, usaremos uno específico, que comentaremos más adelante, aunque deberá ser fácilmente sustituible por otro similar de otra marca.

Para este proyecto buscamos poder controlar y tener información de nuestra instalación desde el móvil. Para ello necesitaremos un servidor cloud donde guardar la información de nuestros sensores y poder notificar cualquier cambio a nuestro móvil. La primera aplicación en la que solemos pensar es en aumentar la seguridad, ya que podemos controlar si alguien está entrando en el recinto donde está instalado el sistema o si hay un incendio. Por otro lado, tenemos el beneficio del confort, ya que el usuario podrá manipular el sistema de manera remota. Para acabar, uno de los usos más comunes en estos casos es evitar un consumo excesivo e innecesario en el recinto, por ejemplo, encendiendo la calefacción en una cierta temperatura.

Usos:

Este sistema se puede enfocar de diferentes puntos de vista según el uso que le vayamos a dar al mismo:

1. **Personal:** Un entorno donde buscaremos que el sistema funcione, buscando un equilibrio entre rendimiento y consumo, ya que no necesitamos que ninguno de estos dos factores destaque con respecto al otro.
2. **Industrial:** Un entorno orientado a una fábrica donde buscaremos la máxima efectividad del producto, que se consiga detectar un problema lo antes posible para poder solucionarlo y evitar posibles pérdidas.
3. **Instalación agrícola remota:** Un entorno apartado donde se buscará el consumo mínimo, ya que los sensores estarán en un entorno poco accesible o donde el usuario va poco, debido a su situación geográfica.

2. Objetivos:

De acuerdo con el contexto y las necesidades introducidas en el apartado anterior, se definen los siguientes objetivos principales y específicos.

2.1 Objetivo principal

El objetivo principal es conseguir desarrollar un sistema capaz de automatizar una determinada instalación, sea doméstica, industrial o agraria, aportando soluciones dependiendo de las necesidades del usuario. También se requiere que el sistema sea abierto y de fácil sustituir, configurar o ampliar según las necesidades del usuario. Además, permite la integración de componentes de otros distribuidores.

2.2 Objetivos específicos

Los objetivos específicos serán necesarios para conseguir el mejor sistema para cumplir el objetivo principal.

- Definir los beneficios del proyecto.
- Diseñar el sistema.
- Configurar los ajustes básicos.
- Configurar el sistema para el uso de Openhab
- Configurar el sistema para el uso de Home assistant.
- Comparar resultados de Openhab con los obtenidos con Home assistant.

3. Propuesta de solución:

La propuesta de solución consiste en crear un sistema que sea capaz de recibir información de los eventos que suceden en la instalación a través de los sensores instalados y controlarlo de manera remota.

Para realizar el proyecto necesitaremos seleccionar los componentes hardware más adecuados que encontremos en el mercado.

Como pieza principal del proyecto tendremos un controlador IoT que se encargará de gestionar el proyecto. Para cumplir con estos requisitos tenemos tres opciones: Arduino, RaspberryPi y BeagleBone.

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	\$29.95	\$35	\$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

Tabla 1: Comparativa posibles controladores IoT

Necesitaremos acceso a internet y puertos usb para realizar el proyecto, por tanto podemos descartar la opción de usar arduino.

Usaremos RaspberryPi ya que es la opción más económica, aunque en términos de consumo energético es mayor BeagleBone.

A la hora de elegir el Hub IoT deberemos tener en cuenta el protocolo de comunicación que queremos utilizar. Son muy comunes los protocolos WiFi y Bluetooth, pero si hablamos de domótica debemos tener en cuenta Zigbee y Z-Wave.

Zigbee y Wifi comparten frecuencia, por tanto, pueden generar interferencias entre ellas mientras que Z-Wave funciona en otra frecuencia.

El principal problema de los dispositivos Wi-Fi es que depende de Internet y, por tanto, si pierden dicha conexión dejan de funcionar. No pasa lo mismo con Zigbee o Z-Wave, ya que se gestionan a través de un Hub.

En nuestro caso elegiremos Z-Wave, aunque la opción de usar Zigbee también es perfectamente viable.

Para ello usaremos el siguiente hardware:

1. Raspberry pi: Encargado de gestionar el sistema.
2. Z-Stick Gen5: Crea un hub de "Z wave plus" similar a Wi-fi.
3. Door/Window Sensor 7: Sensor de puerta/ventana abierta.
4. Dual Nano Switch: Controlador de corriente o encendido/apagado.
5. MultiSensor 6: Sensor que detecta luz, vibraciones, temperatura, ultravioleta, humedad, movimiento.

Sistema de gestión software

Aunque existen muchas plataformas de automatización del hogar, nos centraremos en dos que son de código abierto, imprescindible para este proyecto:

1. **OpenHAB:** Esta plataforma es independiente del proveedor y de la tecnología, de código abierto, ideal para este proyecto, ya que queremos poder cambiar de hardware si fuera necesario. Desarrollada en Java.

2. **Home Assistant:** Esta plataforma busca simplificar y conseguir que los productos sean fáciles de usar. Desarrollada en Python. Se utiliza plugins para ampliar su funcionalidad.
3. **Domoticz:** Plataforma muy liviana, pero que ofrece una cantidad suficiente de funciones. La configuración se realiza con una interfaz web y puede aumentar su funcionalidad con complementos.

4. Metodología y planificación

Para llevar a cabo el proyecto usaremos la metodología Kanban. Esta es una metodología ágil que nos permite repartir la tarea principal en diferentes subtareas y organizarlas para llevarlas a cabo en un cierto orden.

- Diseñar el sistema. (19 de Abril)
- Instalar el sistema operativo en la raspberryPi y OpenHAB/Home Assistant. (16 de Abril)
- Conectar dispositivo móvil a través de wifi desde sus respectivas apps. (23 de Abril)
- Conectar dispositivo que genera el hub. (30 de Abril)
- Conectar sensores. (7 de Mayo)
- Recibir información de los sensores y manipularlos (14 de Mayo)
- Crear el entorno cloud(21 de Mayo)
- Conectar dispositivo móvil a través de datos desde sus respectivas apps. (28 de Mayo)
- Manipular los sensores desde el teléfono móvil (fuera de casa). (4 de Junio)
- Comparar HomeAssistant y OpenHAB. (11 de Junio)
- Generar informe final (20 de Junio)
- Crear presentación (27 de Junio)
- Redactar dossier (28 de Junio)
- Realizar póster (4 de Julio)

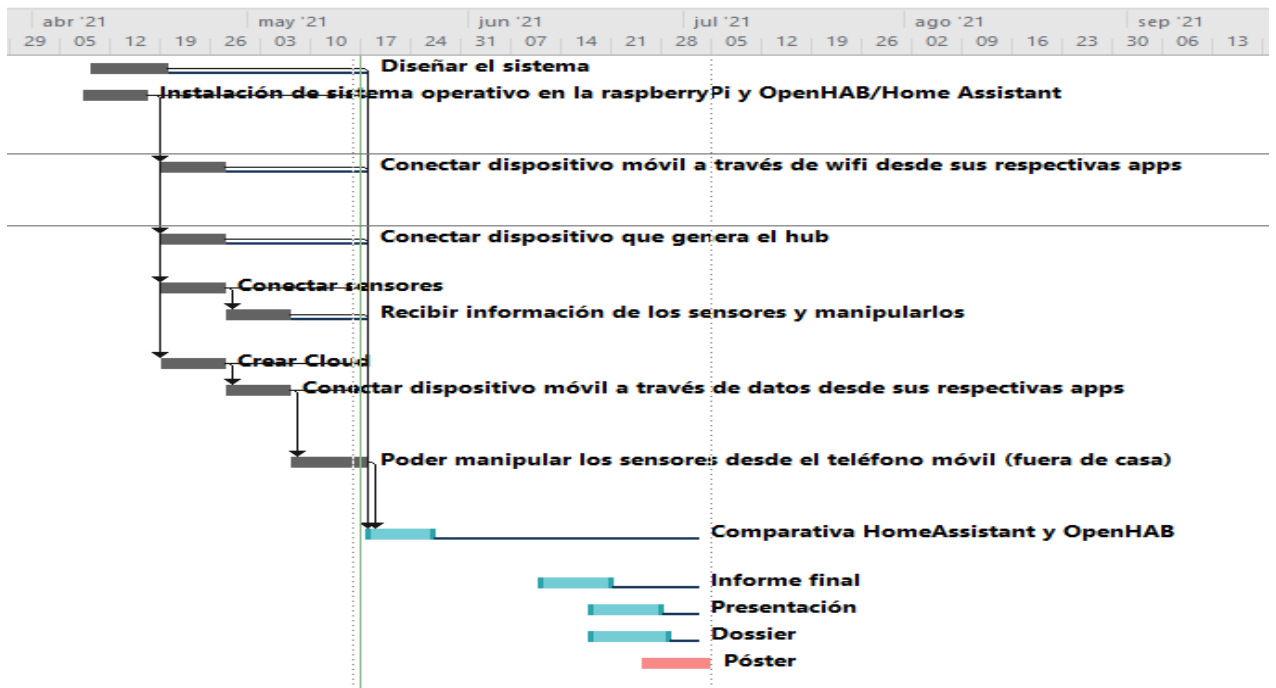


Imagen 1: Diagrama de Gantt

5. Diseño del sistema

Para especificar el diseño del sistema debemos tener en cuenta cómo se conectan las diferentes partes del proyecto. Como eje central del proyecto tenemos una raspberry pi, que tendrá que recibir información de los diferentes sensores para luego informar al usuario a través de su dispositivo móvil. Conectado a la raspberry tenemos el Z-Stick Gen5 que creará un hub necesario para comunicarnos con los diferentes sensores door/window sensor y dual nano switch.

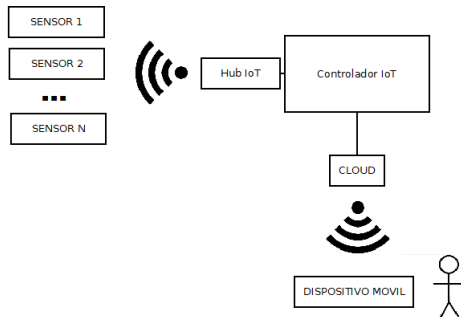


Imagen 2: Arquitectura Hardware

Por parte del software de nuestro sistema tendremos dos modelos, uno para HomeAssistant y otro para OpenHAB, aunque las arquitecturas son similares hay que diferenciarlos.

Home Assistant:

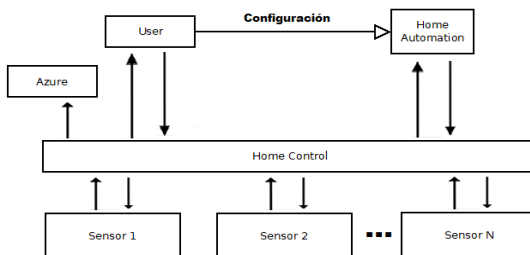


Imagen 3: Arquitectura Software Home Assistant

Home control es la pieza central de Home Assistant, recibe información de los diferentes sensores, esa información es enviada a User (interfaz de usuario), a Home Automation (automatizaciones) y a Azure. Con esta información tanto el usuario como las automatizaciones pueden enviar información al control de home assistant para que modifique algunos sensores, por ejemplo, encender la luz.

OpenHAB:

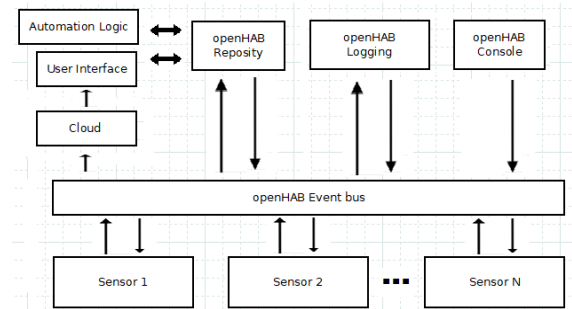


Imagen 4: Arquitectura Software OpenHAB

En OpenHAB los sensores se conectan a openHAB Event Bus que es el encargado de gestionar el sistema, recibe comandos tanto de la consola, logging como repositorio a los cuales ha enviado datos del sistema, también al cloud. Automation Logic(automatizaciones) y User interface(usuario) reciben información de openHAB Repository y a la vez puede enviar instrucciones, como en Home Assistant el ejemplo es encender una luz.

6. Implementación

Dual Nano Switch nos permite controlar dos enchufes de corriente, para ello tendremos que cablearlo:

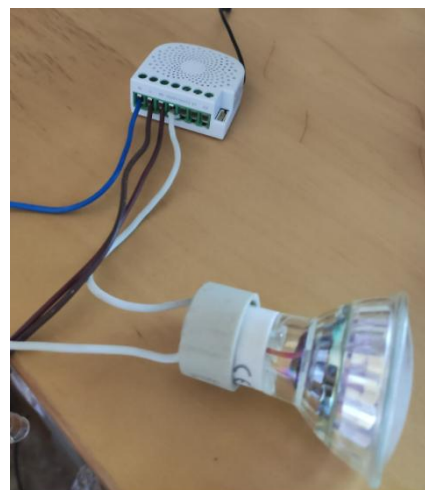


Imagen 5: Connexión Dual Nano Switch

Debemos conectar N y L a neutro y fase en dicho orden, en In conectamos la entrada que también será la fase y Out1 será la salida que va a parar a la bombilla. La otra entrada de la bombilla ira a parar a neutro para cerrar el circuito.



Información de los sensores:

Al añadir los sensores a la red, podemos ver en todo momento la información que nos ofrecen.

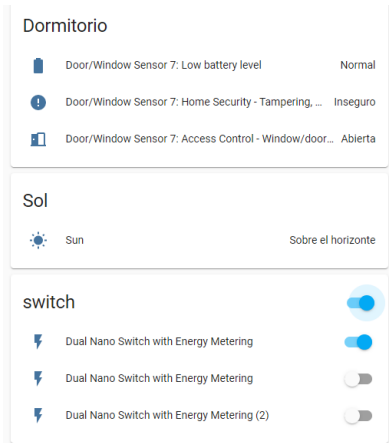


Imagen 6: Visualización de sensores en Home Assistant

En este caso, vemos que el sensor de puerta detecta que la puerta está abierta.

Por otro lado, podemos ver que la bombilla está encendida.

Automatización:

Hemos realizado unas automatizaciones que consisten en encender la bombilla conectada al Dual Nano Switch cada vez que el estado de Door/Window Sensor 7 pase a estar abierto, es decir, que la puerta se abra. Por otro lado, tenemos la automatización de apagar la bombilla cuando detecta que la puerta se ha cerrado.

7. Pruebas y evaluación

Uso de sensores:

Cada sensor tiene diferentes entidades, cada entidad nos proporciona información independiente de las otras entidades ya puede ser el estado de la batería, estado del sensor o cualquier información que nos pueda aportar dicho sensor. En el sensor Door/Window Sensor 7 tenemos la entidad Access Control que nos indica si el sensor de puerta está abierto o cerrado.



Imagen 7: Puerta abierta

Imagen 8: Puerta cerrada

Uso del sistema remotamente:

Una vez configurado el sistema podemos recibir la información directamente a nuestro móvil, siempre y cuando estemos en la misma red local que nuestra raspberry, haciendo posible la gestión del sistema desde dicho dispositivo. Más adelante, cuando se cree el cloud se podrá acceder al sistema fuera de la red local.

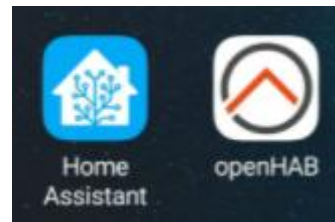


Imagen 9: Apps de móvil

La aplicación de HomeAssistant nos muestra la misma interfaz de usuario que podemos ver desde un ordenador.

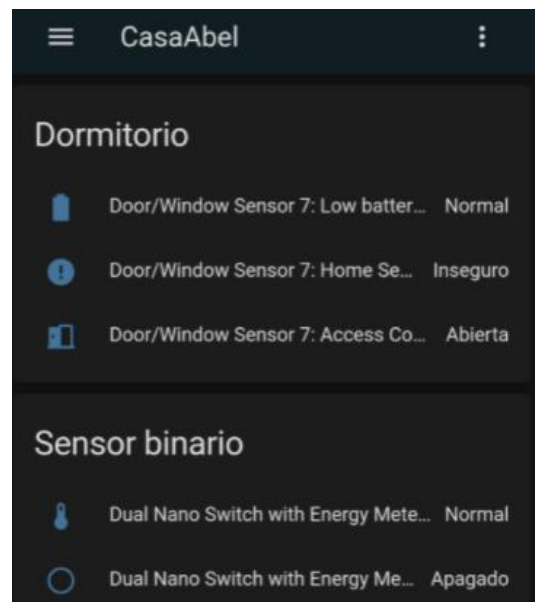


Imagen 10: Sensores visualizados por la app de Home Assistant

OpenHAB es diferente, ya que requiere crear una interfaz personalizada, esto nos permite crear diferentes interfaces dependiendo del usuario.

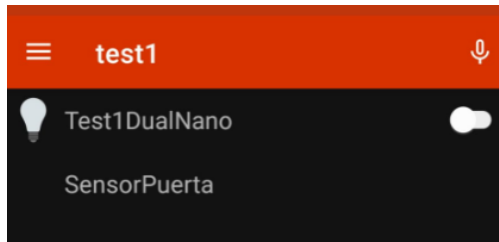


Imagen 11: Ejemplo interfaz para la app de OpenHAB

En esta interfaz podemos ver el interruptor que enciende la bombilla y el sensor de puerta que nos indica si está abierta o no.

estemos en casa tal y como hacíamos desde el móvil de manera local.

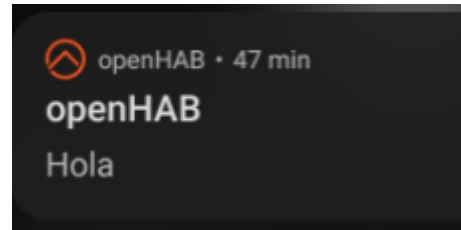


Imagen 12: Notificación recibida en el móvil de OpenHAB

Cloud en OpenHAB:

Para tener acceso a nuestro sistema fuera de casa y mantenernos informados de lo que pasa en ella, necesitaremos configurar un cloud. Cuando lo tengamos configurado podremos recibir notificaciones, configuradas anteriormente en el sistema, por ejemplo cuando se abre la puerta. También podremos encender la bombilla aunque no

Envío de datos a Azure desde HomeAssistant:

Configuramos HomeAssistant para enviar datos hacia Azure. Esto nos proporciona la capacidad de enviar datos proporcionados por los sensores a un servidor externo para poder trabajar con ellos más adelante.

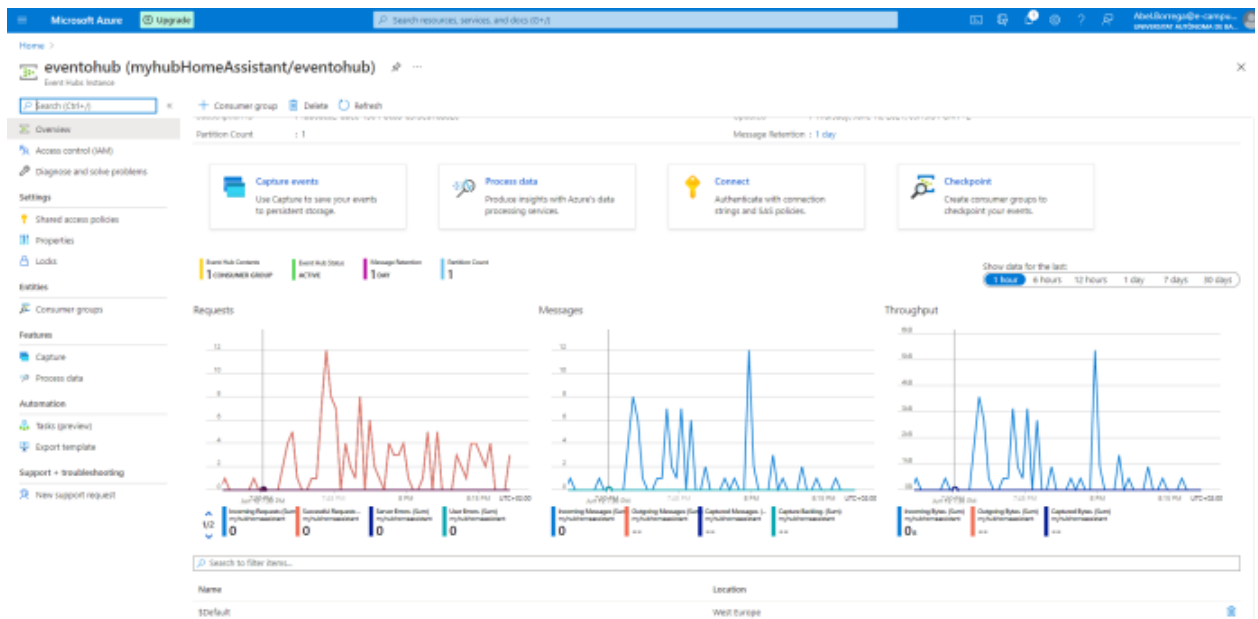


Imagen 13: Recibo de mensajes a Azure

En este ejemplo podemos ver los mensajes que llegan cada minute. Estos mensajes indican un cambio en el sensor de puerta abierta. Cada vez que cambiamos la puerta de abierta a cerrada o viceversa, HomeAssistant enviara un mensaje a Azure. Por tanto, sabiendo el estado inicial del sensor y todos los mensajes que ha recibido Azure, podremos saber el estado en el que se encuentra el sensor en la actualidad.

8. Casos de usos/Aplicación

Este sistema se puede usar en diferentes casos de uso, depende del caso de uso en el que se utilice conseguiremos satisfacer unas necesidades u otras.

Uso doméstico: En el uso doméstico se busca aumentar la comodidad del usuario y la seguridad del domicilio. El sensor dual nano switch y tener el sistema disponible desde el móvil nos proporciona la comodidad de movilidad. Por otro lado, el sensor de puerta y el multisensor nos proporcionan utilidad, el primero avisándonos que se ha abierto la puerta y el segundo nos puede avisar de vibraciones y movimiento.

Uso industrial: En el uso industrial se busca mejorar la seguridad, tanto en caso de robos como con posibles incidentes. Para ello, los sensores que se usaran será el de sensor de puerta para saber si se abre puertas o ventanas y el multisensor que nos informa del movimiento, vibraciones que pueden ser generadas por una máquina, temperatura que puede ser superior a la temperatura ideal y humedad.

Uso Agrícola: En el uso agrícola es importante mantener la seguridad, ya que podría ser una instancia donde no haya nadie mucho tiempo al día y por tanto es más fácil que se produzcan robos. El multisensor nos aportar la seguridad de que los productos o la cosecha de un almacén no se estropeen por la temperatura o por la humedad.

9. Conclusiones

Como conclusiones del trabajo podemos extraer que los sensores se pueden adaptar a diferentes entornos, no dependen de un estándar impuesto por dichos sensores o viceversa. La automatización hace que el usuario no tenga que realizar tanto trabajo y las acciones son instantáneas. Se pueden controlar desde el móvil recibiendo información de los sensores y enviando esa información al cloud. El cloud al que se envía no tiene por qué ser el del estándar, pudiendo buscar otras opciones en el mercado.

Como futuras mejoras se deberían analizar los mensajes que llegan al servidor de Azure que usamos como cloud en HomeAssistant. Por otra parte, también se debería conectar a Azure desde OpenHAB.

10. Bibliografía

- <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- <https://aeotec.com/z-wave-usb-stick/>
- <https://aeotec.com/z-wave-outlet-socket/>
- <https://aeotec.com/z-wave-door-window-sensor/>
- <https://www.openhab.org/>
- <https://www.home-assistant.io/>
- <https://www.openhab.org/docs/installation/rasppi.html>
- <https://www.openhab.org/docs/installation/openhabian.html>
- <https://www.openhab.org/addons/integrations/openhabcloud/>
- <https://www.home-assistant.io/getting-started/>
- <https://www.home-assistant.io/installation>
- <https://www.home-assistant.io/integrations/alexa/>
- <https://www.youtube.com/watch?v=iPisuXWW07A>
- <https://www.wikiversus.com/informatica/z-wave-vs-zigbee-vs-wifi/>
- <http://proyectosincreibles.blogspot.com/p/arduino-vs.html>
- <https://www.smarthomepoint.com/zigbee-zwave-wifi-bluetooth-comparison/>
- <https://www.adslzone.net/reportajes/tecnologia/estandares-conexion-inalambrica/>
- <https://www.smarthomeblog.net/openhab-home-assistant-domoticz/>
- <https://latecnologiaatualcance.com/asistente-de-hogar-vs-openhab/>
- <https://www.actualizatucasa.com/openhab-cloud-connector/>

11. ANEXO

OpenHAB:

Lo primero que debemos hacer es instalar el sistema operativo requerido para el servidor utilizando el programa Etcher.

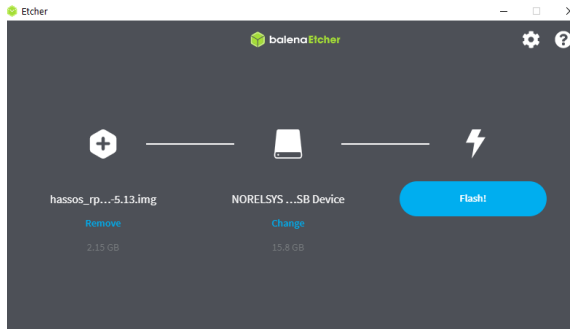


Imagen 14: Programa Etcher

Deberemos conectarnos a la raspberry a través de SSH para configurar el servicio.

```
[2021-09-14 22:42:45] ~
[Abel.Bsuario] ~ ssh pi@192.168.0.24
Warning: Permanently added '192.168.0.24' (RSA) to the list of known hosts.
pi@192.168.0.24's password:
Linux raspberrypi 5.4.83-v7+ #1379 SMP Mon Dec 14 13:08:57 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
/usr/bin/xauth: file /home/pi/.Xauthority does not exist

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

Wi-Fi is currently blocked by rfkill.
Use rasi-config to set the country before use.

pi@raspberrypi:~$
```

Imagen 15: Consola de OpenHAB

A continuación, ya podremos conectarnos a OpenHAB a través de un navegador web.

Deberemos configurar el z-wave que queremos utilizar como hub. Para ello, debemos instalar el add-on Z-Wave Binding

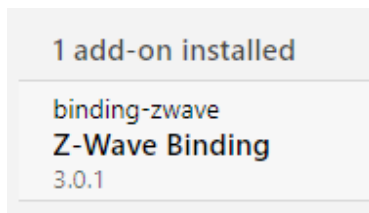


Imagen 16: Z-Wave Binding

Ahora solo es necesario añadir el z-wave serial control en things y especificar el Puerto usb donde está conectado a la raspberry.

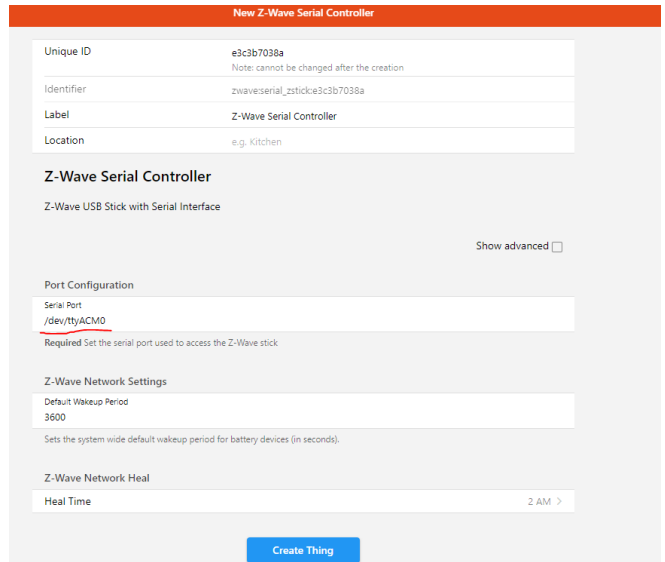


Imagen 17: Configuración del controlador del Z-Wave

A la hora de conectar sensores, se conectaran directamente al controlador y en OpenHAB solo nos saldrá una pequeña pestaña indicándonos que se ha añadido un nuevo elemento.

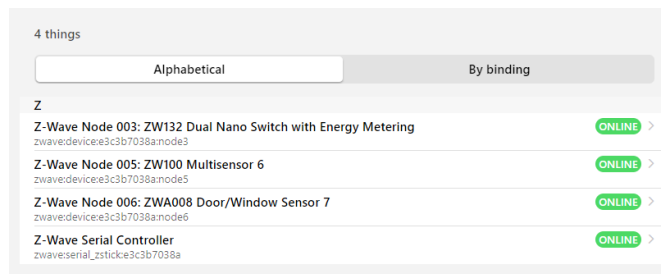
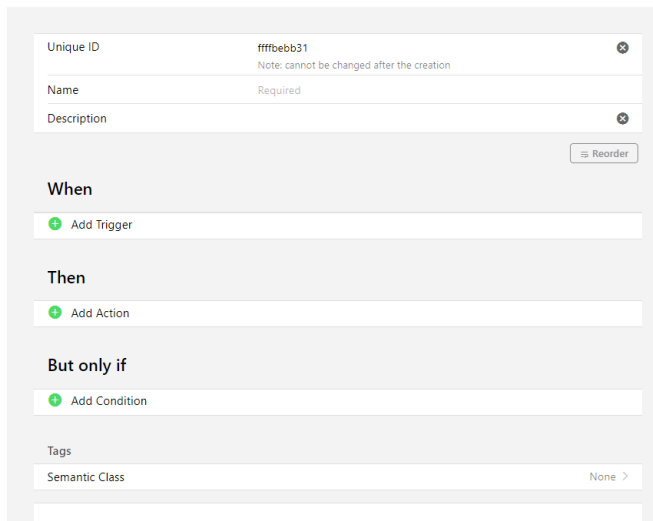


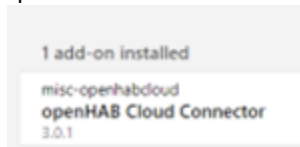
Imagen 18: Sensores conectados a OpenHAB

Si queremos crear una nueva automatización en el sistema solamente hace falta ir a Rules y añadir una.

En When indicamos la acción que va a generar la automatización, en Then que pasara cuando se realice la automatización y en But only if indicaremos si hay una condición necesaria para que esta automatización se realice.



Para crear el cloud de OpenHAB debemos añadir el add-onn openHAB Cloud Connector



Después necesitaremos crearnos una cuenta en myopenhab.org, en ella nos pedirá tanto el uuid como el secret de nuestro cloud. Estos dos valores se encuentran en /var/lib/openhab/uuid y en /var/lib/openhabcloud/secret respectivamente.

Al introducir estos datos solo nos faltará reiniciar el sistema para tener el cloud operativo.

Para finalizar queremos hacer que lleguen notificaciones a nuestro cloud, para ello habrá que añadir un then a la automatización que queramos que nos informe a nuestro cloud. Añadiendo el enviar mensaje broadcast nos llegará la notificación tanto al cloud como a todos los móviles que esten conectados a él.

HomeAssistant:

Para HomeAssistant, a diferencia de OpenHAB, usaremos una imagen específica que cree el servidor de HomeAssistant automáticamente usando el mismo programa (Etcher).

Al tener el sistema conectado solo necesitaremos entrar a la ip asignada al servidor y esperar.

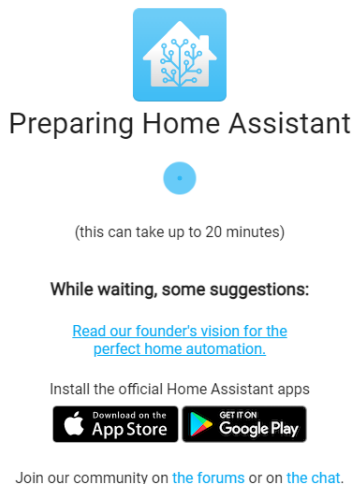


Imagen 19: Preparación de HomeAssistant

Para conectar el controlador Z-Wave solo es necesario añadir la integración de z-wave JS e indicarle en que puerto usb está conectado el z-stick.



Imagen 20: Z-Wave JS

Los sensores se conectarán automáticamente al sistema, pero no se verá que sensor ha sido conectado hasta su primer uso.



Imagen 21: controlador Z-Wave conectado a HomeAssistant

Para crear una automatización simplemente nos dirigimos a configuración, automatizaciones y creamos una nueva. Al crear la nueva automatización solo debemos elegir el desencadena que la provoca y la acción que genera. Además podemos añadir condiciones que si no están activas no se genere dicha automatización.

Configurar Azure: Después de crear la cuenta de azure deberemos ir a nuestro portal y crear un resource group, un event hub namespace y dentro del mismo un event hub instance. Deberemos ver los privilegios de acceso a esa instancia y copiar el campo connection string-primary key.

En HomeAssistant deberemos modificar el archivo configuration.yaml para que se conecte a azure. Añadimos event_hub_connection_string donde pondremos lo copia anteriormente en Azure y además añadiremos filter donde especificaremos lo que queremos enviar a Azure (sensores).

Imagen 22: Automatización HomeAssistant

```

/config/configuration.yaml
1 |
2 | # Configure a default setup of Home Assistant (frontend, api, etc)-
3 | default_config:
4 |
5 | # Text to speech-
6 | tts:
7 |   -- platform: google_translate-
8 |
9 | group: !include groups.yaml
10 | automation: !include automations.yaml
11 | script: !include scripts.yaml
12 | scene: !include scenes.yaml
13 |
14 | # Example configuration.yaml entry-
15 | azure_event_hub:
16 |   -event_hub_namespace: myhubHomeAssistant-
17 |   -event_hub_instance_name: eventohub-
18 |   -event_hub_sas_policy: RootManageSharedAccessKey-
19 |   -event_hub_sas_key: 69EcXlAmChOGdenSWdglLkG4V6+U09ViLSn/G/bgCGk=-
20 |   -event_hub_connection_string: Endpoint=sb://myhubhomeassistant.service-
21 |   -send_interval: 15
22 |   -max_delay: 5-
23 |
24 |
25 |
26 | filter:
27 |   -- include_domains:
28 |     --- homeassistant
29 |     --- light
30 |     --- media_player
31 |   -- include_entities:
32 |     --- binary_sensor.door_window_sensor_7_access_control_window_door_is-
33 |     --- sensor.multisensor_6_air_temperature-
34 |

```

Imagen 23: configuration.yaml