Domótica
openHAB + The Things Network + Alexa

Juan Félix Mateos Barrado
jfmateos@educa.madrid.org
juanfelixmateos@gmail.com

Medialab Prado
Noviembre 2019

Objetivo

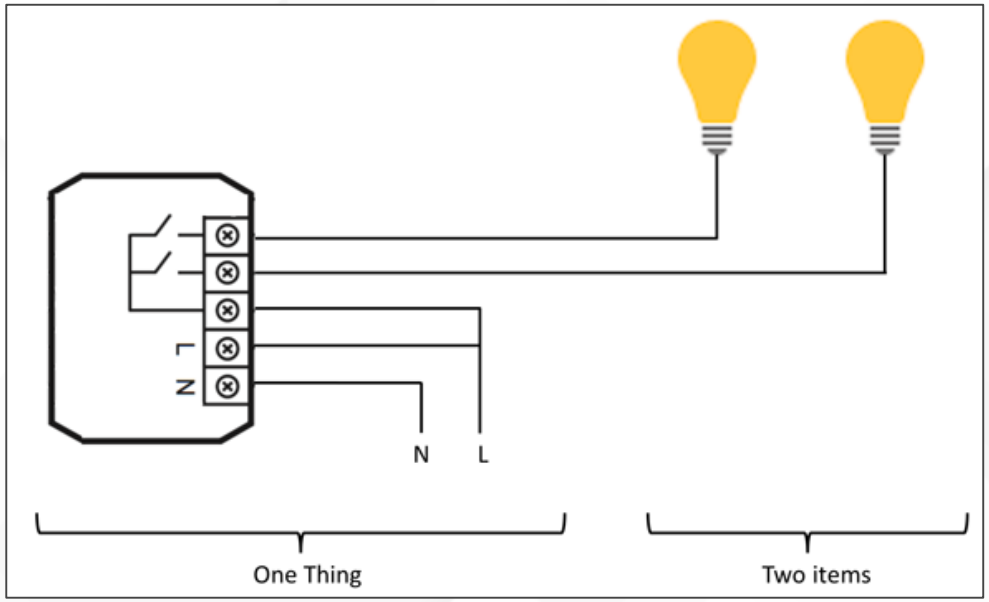
Integrar un nodo de The Things Network en openHAB para poder conocer su temperatura y controlar su LED utilizando comandos de voz a través de Alexa.

openHAB

- Open Home Automation BUS
- Creado por **Kai Kreuzer** (2010) en Java
- Sofware abierto de domótica patrocinado por KNX, Z-Wave…
- Funciona en Windows, Linux, MacOS, Raspberry Pi…
- A partir de openHAB se creó Eclipse SmartHome (2013) para aplicaciones comerciales.

Conceptos básicos

- **Things**, **Items** y **Channels**
  - Un sensor de temperatura, presión y humedad es una thing (cosa física), que ofrece 3 canales (temperatura, presión y humedad), con los que el usuario podrá interactuar a través de un item (elemento virtual)
- **Bindings**
  - Son las extensiones de openHAB que nos permiten usar distintos tipos de things (Z-Wave, Zigbee…)
- **Rules**
- **Sitemaps**
- **Dashboards**



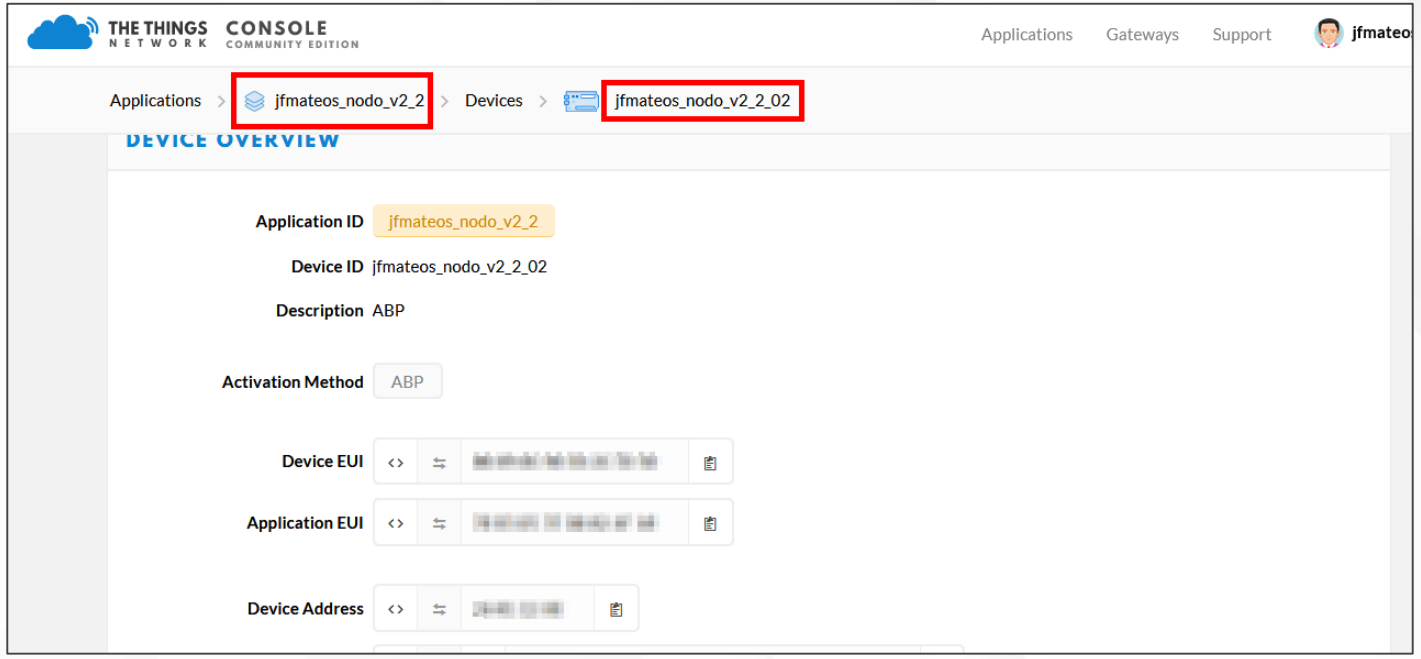
Orden del día

- Crear una aplicación y un dispositivo en The Things Network
- Instalar Java
- Instalar openHAB
- Instalar el binding JSONPath
- Crear una thing de tipo Generic MQTT Thing con 2 canales vinculados a 2 items:
  - Temperatura
  - LED
- Crear una regla que active/desactive el LED en el umbral de temperatura de 25º
- Crear un sitemap que muestre la fecha/hora, y los controles de Temperatura y el LED.
- Instalar el binding openHAB Cloud Connector
- Instalar la app de openHAB para el móvil.
- Instalar el binding Rest documentation
- Añadir metadatos a los items para poder controlarlos con Alexa.
- Instalar Alexa en el móvil.
- Configurar el skill de openHAB

Crear aplicación y dispositivo en TTN

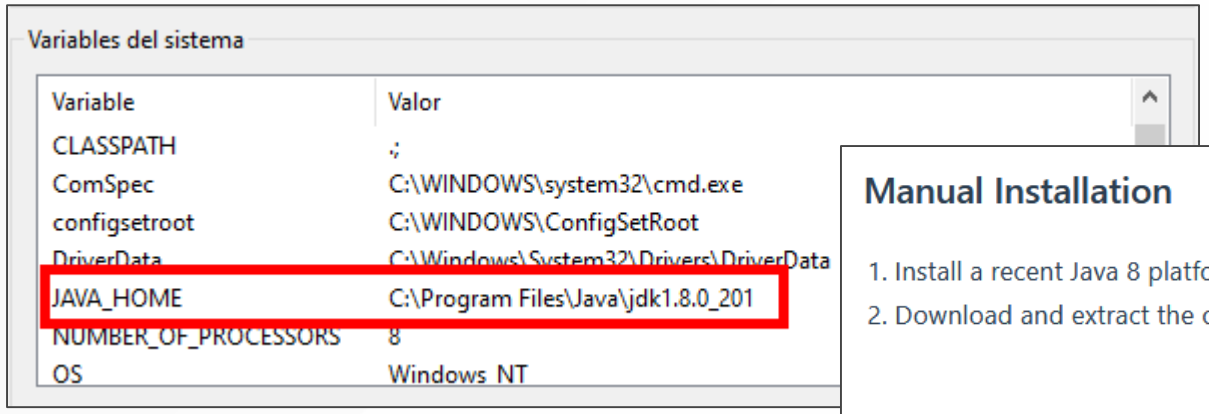
- No es imprescindible un nodo físico. Podemos simularlo con la siguiente carga de pago
  - 01 02 01 2D 02 00 01 **03 67 00 E2** 04 73 24 77 05 68 7D 06 01 01 (T=22.6)
  - 01 02 01 2D 02 00 01 **03 67 01 07** 04 73 24 77 05 68 7D 06 01 01 (T=26.3)

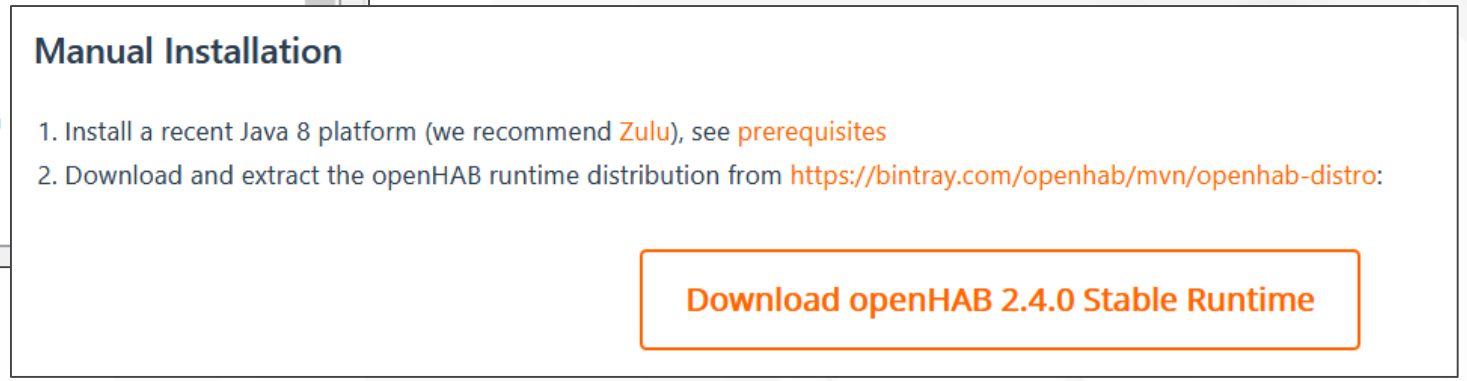

```
{
  "analog_in_1": 3.01,
  "barometric_pressure_4": 933.5,
  "digital_in_2": 1,
  "digital_out_6": 1,
  "relative_humidity_5": 62.5,
  "temperature_3": 22.6
}
```

Instalación en Windows 1/2

- Requiere Java 8 (se recomienda la versión Zulu)
- Asegurarse de que está definida la variable de entorno JAVA_HOME



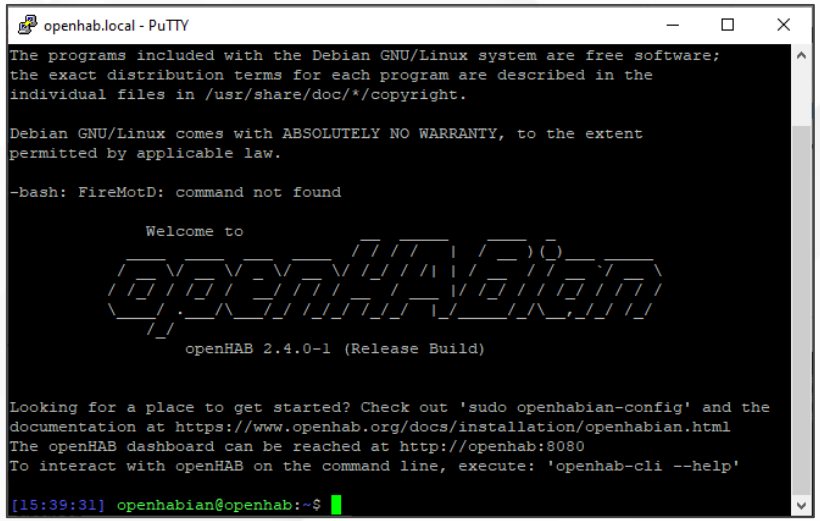


Instalación en Windows 2/2

- Descargar el archivo comprimido de openHAB
- Descomprimirlo en una carpeta de nuestro disco duro; por ejemplo C:/openHAB
- Ejecutar el archivo start.bat para iniciar openHAB
- Acceder con el navegador web a localhost:8080

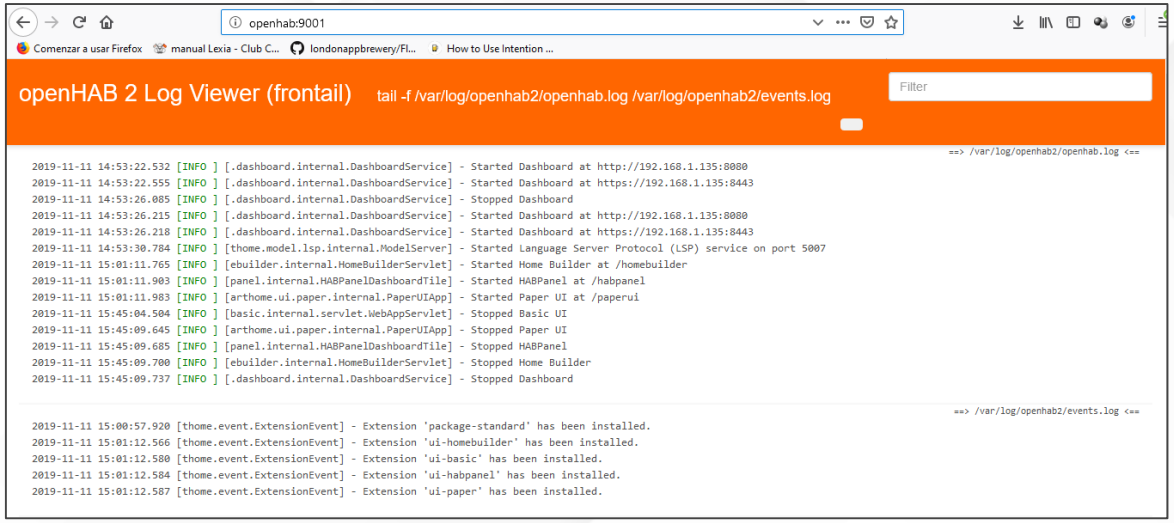
Iniciar y detener openHAB

- A través de la consola:
  - Windows: C:\openHAB\runtime\bin\karaf.bat
    - logout
  - Raspberry pi: Putty/SSH
    - Usuario: openHABian
    - Contraseña: openHABian
    - sudo shutdown –h now



Logs

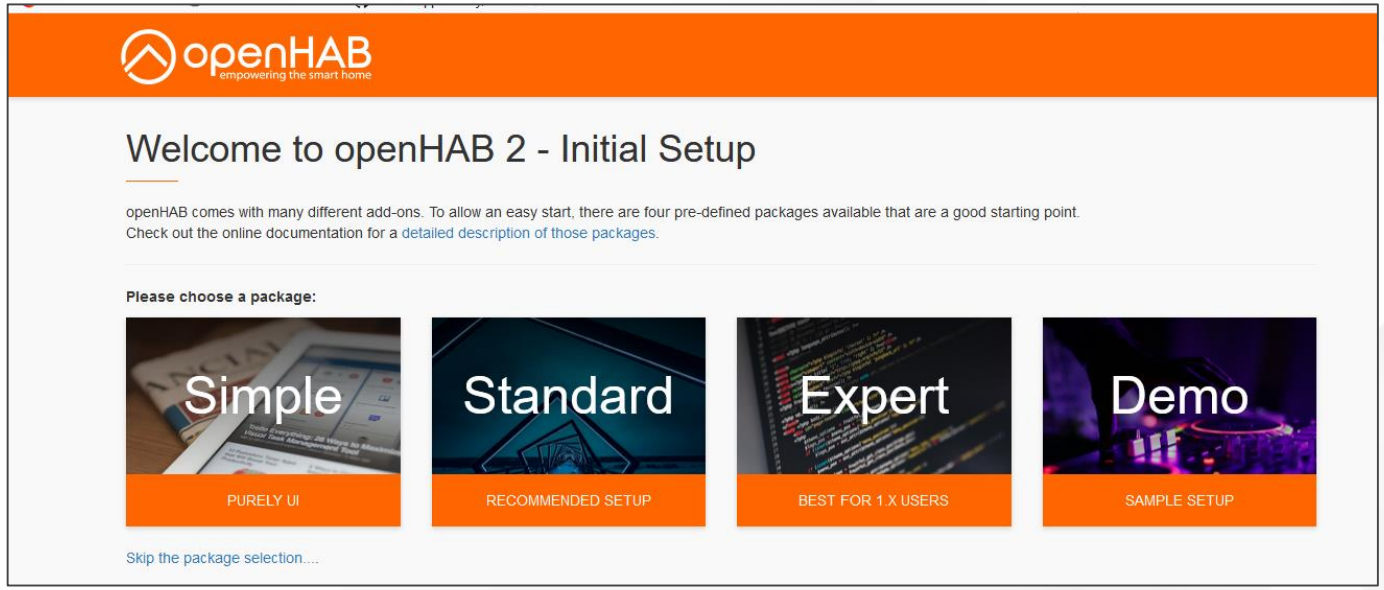
- Hay 2 logs importantes:
  - events.log → Todo los relacionado con las things
  - openHAB.log → Todo lo relacionado con los bindings
- Windows
  - C:\openHAB**userdata**\logs
- Raspberry pi
  - http://openHAB:9001/



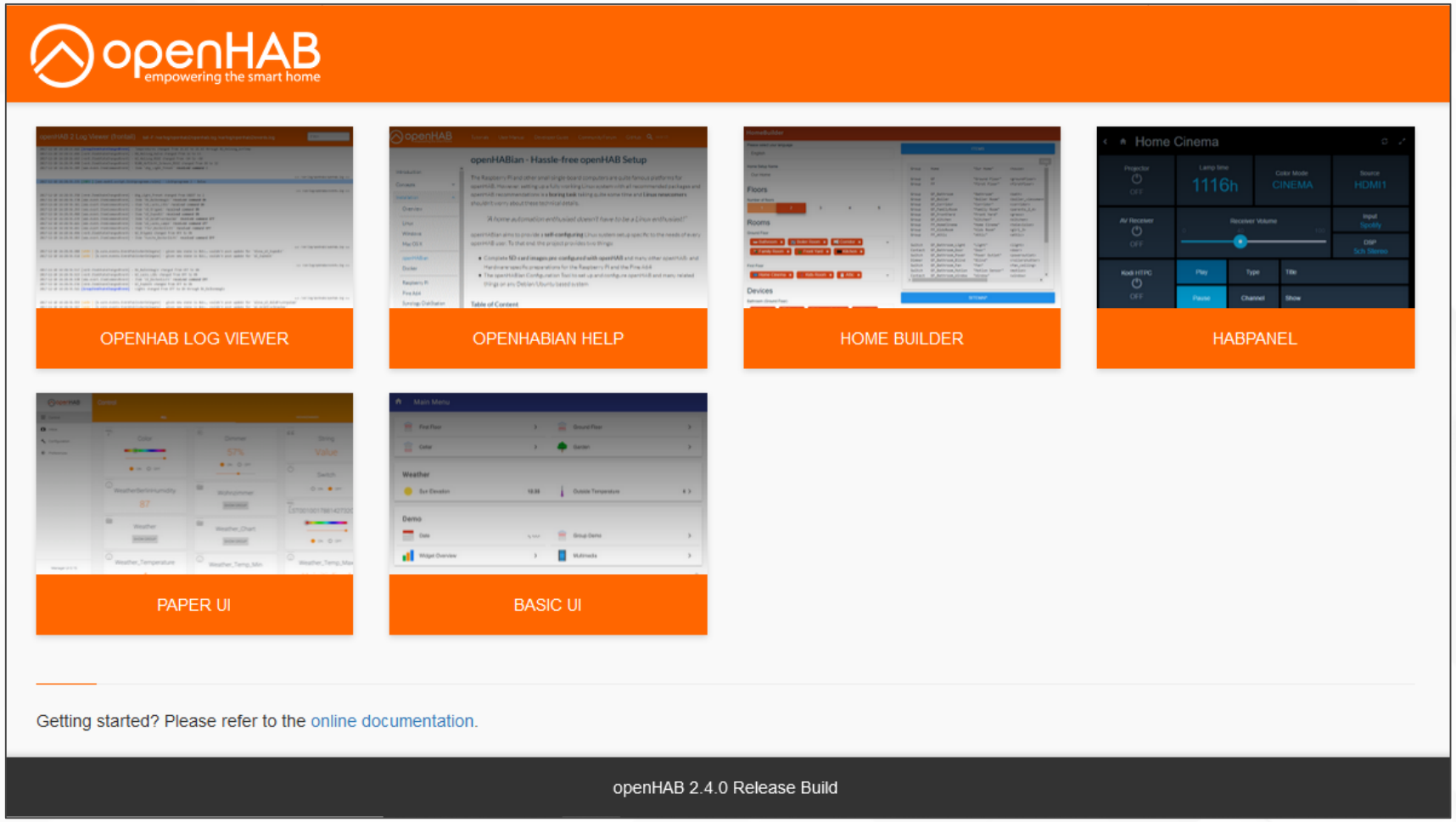
Configuración inicial

Se recomienda utilizar la opción Standard, que instalará las interfaces:

- Paper UI
- Basic UI
- Hab PANEL
- Home Builder



Interfaces (1/2)

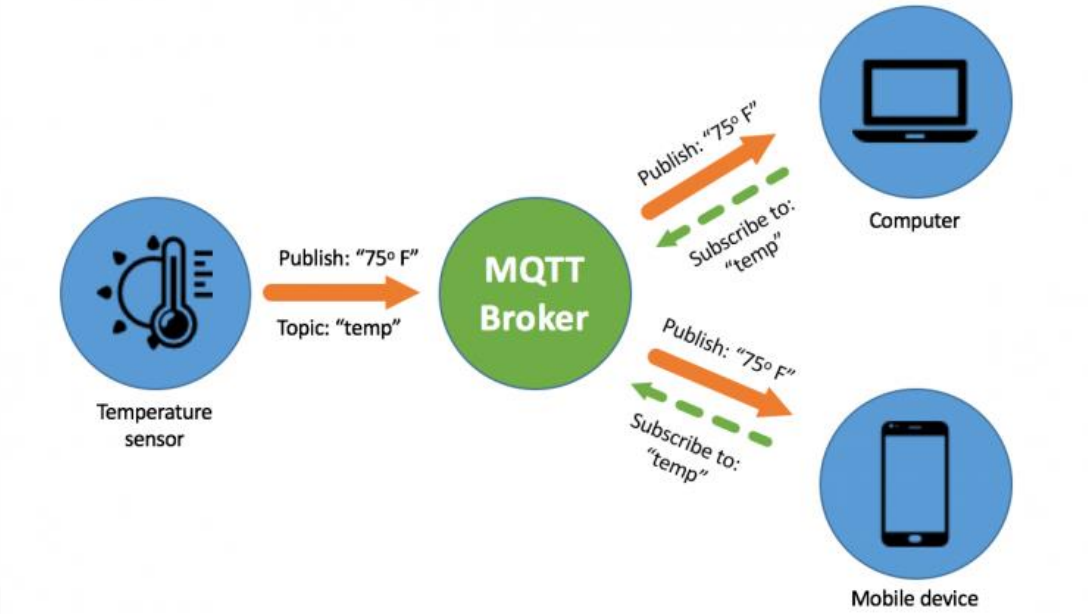


Interfaces (2/2)

- **Paper UI**: Ideal para configurar openHAB.
- **Basic UI**: Ideal para mostrar los Site maps
- **HABPanel**: Sirve para diseñar Dashboards.
- **Home Builder**: Sirve para generar una configuración inicial compuesta por:
  - Items
  - Sitemaps
  - Dashboards

MQTT

- Broker
- Suscripción/Publicación
  - Topic
- Vamos a vincular openHAB con el bróker MQTT de The Things Network

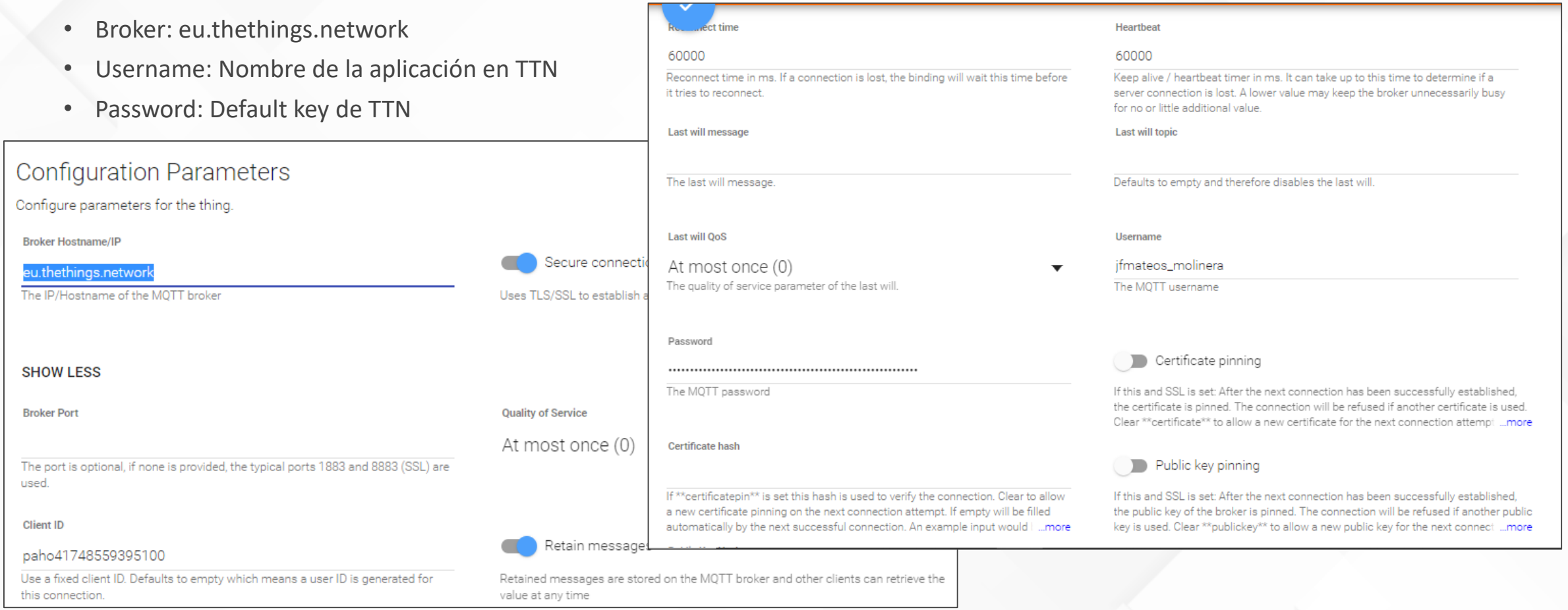


MQTT

- Instalar el binding MQTT
- Crear un thing de tipo MQTT Broker

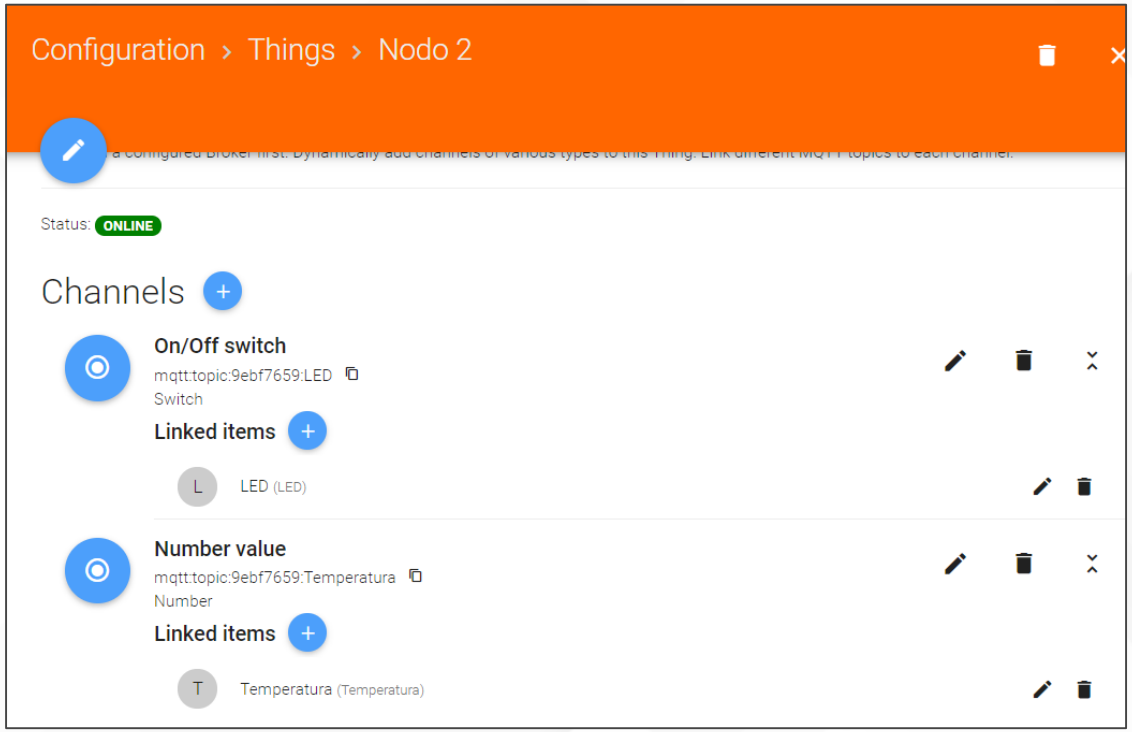
MQTT: Configurar la thing MQTT Broker

- Broker: eu.thethings.network
- Username: Nombre de la aplicación en TTN
- Password: Default key de TTN



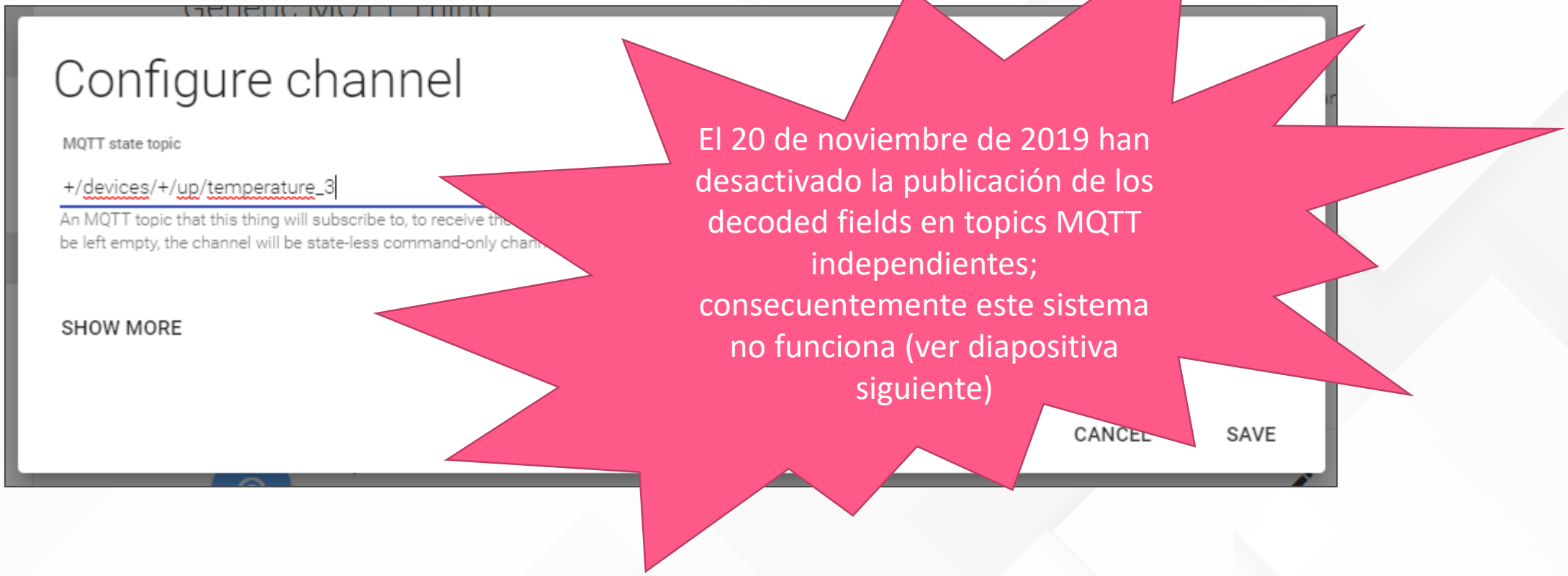
MQTT: Crear una Generic MQTT Thing

- El thing Broker es el soporte general para poder usar MQTT en openHAB, pero adicionalmente tenemos que crear un thing de tipo Generic MQTT Thing para cada elemento que queramos controlar.
- Crear una thing de tipo Generic MQTT Thing con 2 canales vinculados a los siguientes items:
  - Number value: **Temperatura**
  - On/Off Switch: **LED**



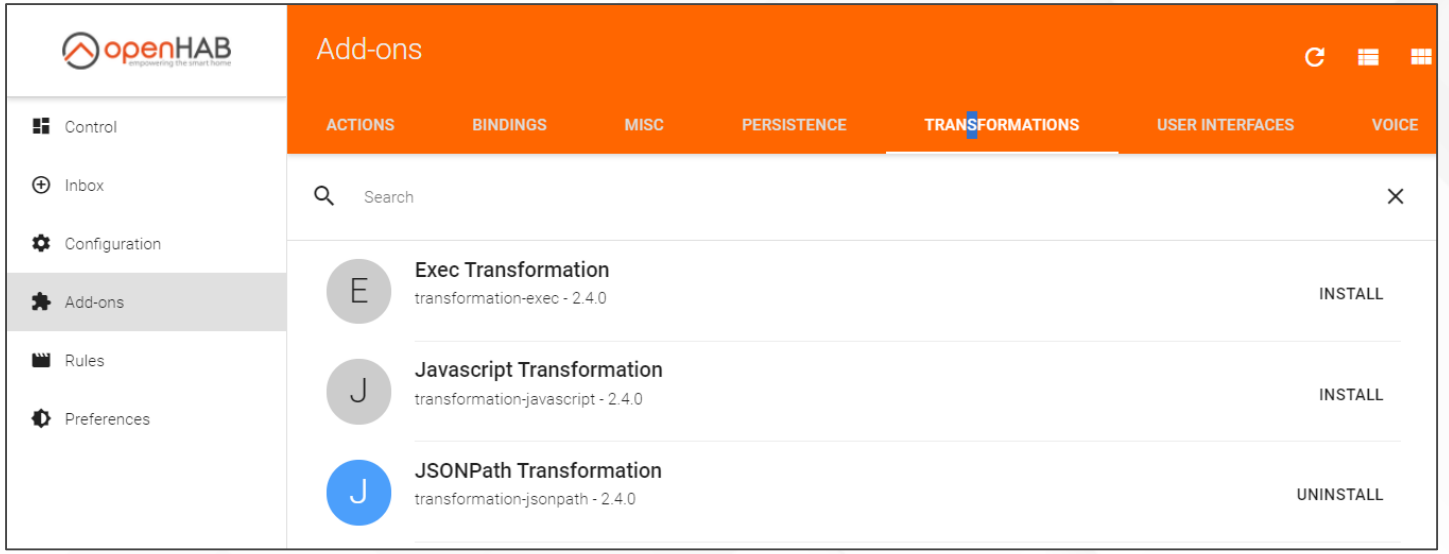
MQTT: Canal para recibir uplink

- Configurar el canal de tipo Number Value del siguiente modo



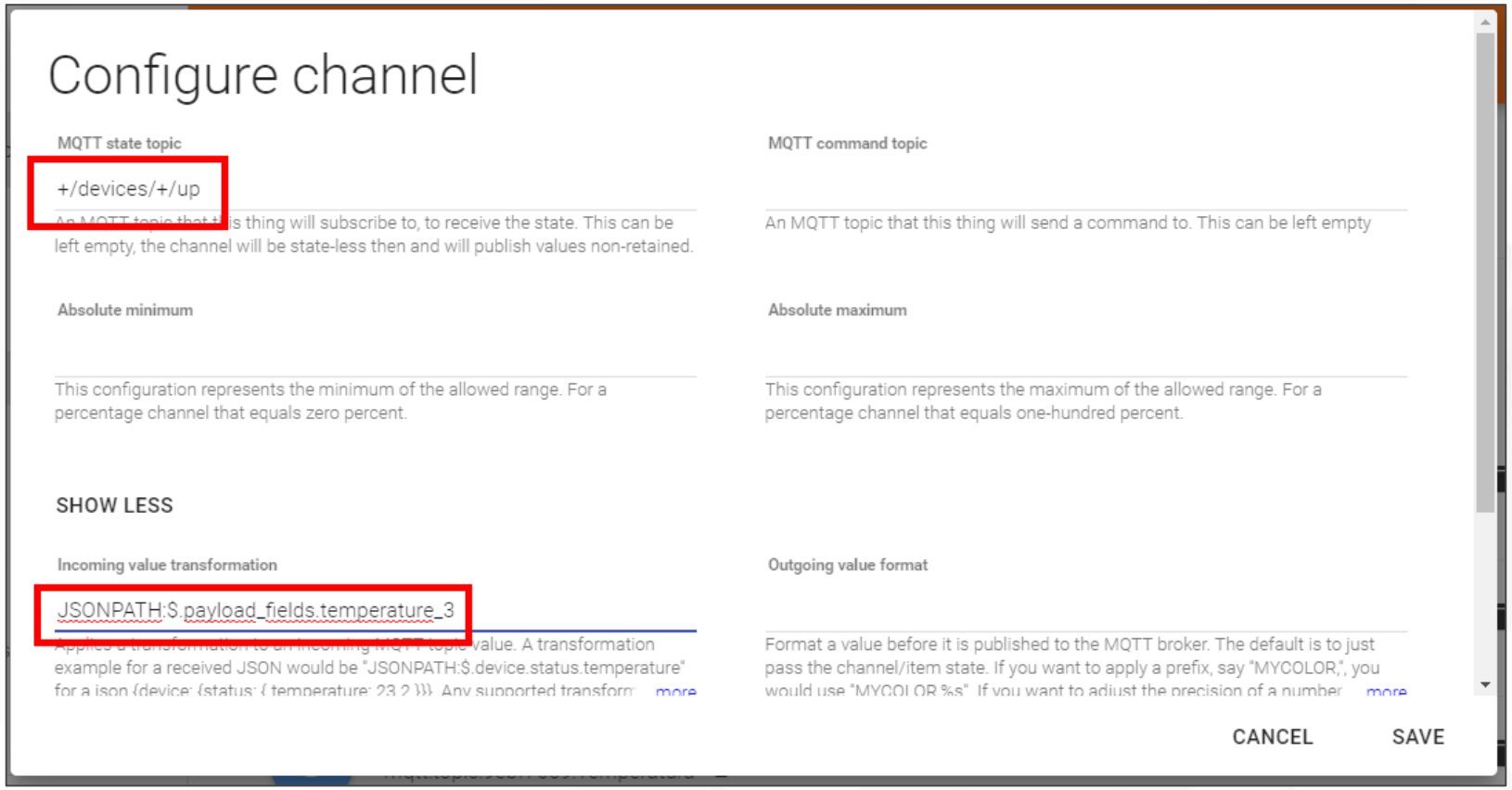
MQTT: Canal para recibir uplink (alternativa)

- En lugar de suscribirnos al topic concreto de la temperatura, nos suscribiremos al topic completo, y extraeremos de él la temperatura mediante una transformación.
  - Instalar el binding JSONPath Transformation



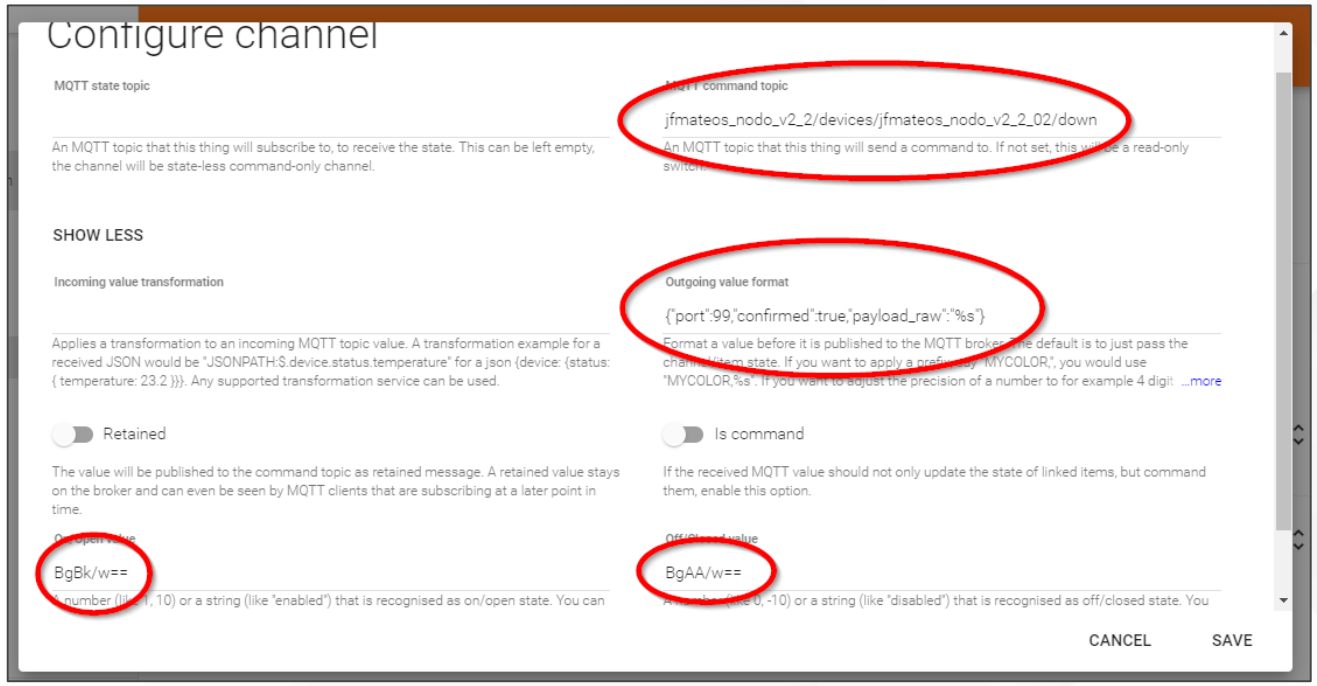
MQTT: Canal para recibir uplink (alternativa)

- Configurar el canal de temperatura del siguiente modo



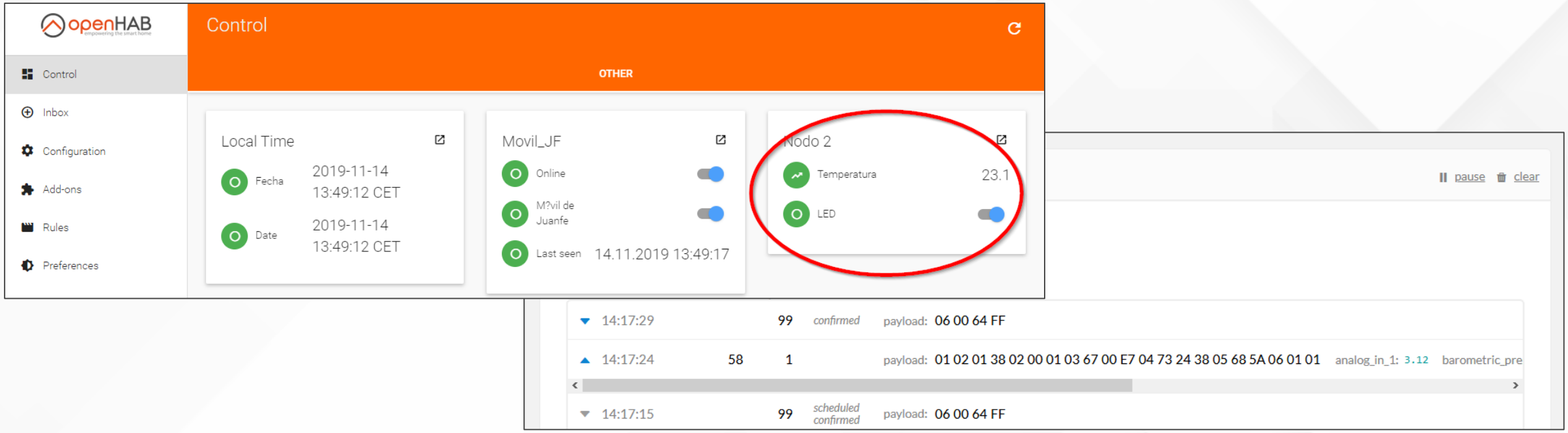
MQTT: Canal para enviar downlink

- Configurar el canal de tipo On/Off Switch del siguiente modo
  - **BgBk/w==** y **BgAA/w==** son la codificación en BASE64 de 06 00 64 FF y 06 00 00 FF, que son la carga de pago que espera recibir el nodo para encender y apagar el led, respectivamente
    - jfmateos_nodo_v2_2/devices/jfmateos_nodo_v2_2_02/down
    - {"port":99,"confirmed":true,"payload_raw":"%s"}



MQTT: Canal para enviar downlink

- En la versión 2.4 de openHAB hay un bug que impide que se inicie automáticamente el cliente MQTT
  - Para resolverlo hay que cerrar openHAB (logout) y volver a iniciarlo.



Reglas

Nos permiten reaccionar ante:

- el cambio de estado de los **ítems**
  - Things y canales (de forma más limitada)
- instantes (fecha y/u hora) concretos
- eventos del sistema (que se inicie o se apague)

con acciones como:

- enviar un comando a un item
- utilizar un servicio (enviar un mail, notificación…)

- La sintaxis está basada en Xbase

Reglas: Encender/apagar el LED en el umbral de 25º

Crear en la carpeta C:\openHAB\conf\rules un archivo llamado default.rules con el siguiente contenido:

```
rule "LED"
when
        Item Temperatura changed
then
        if (Temperatura.state >= 25 && previousState<25) {
                LED.sendCommand(ON)
        } else {
                if(Temperatura.state < 25 && previousState>=25) {
                        LED.sendCommand(OFF)
                }
        }
end
```

Sitemaps

Son representaciones destinadas al usuario para conocer el estado del sistema o interactuar con él.

Están compuestos por elementos de distintos tipos:

- Pulsadores
- Deslizadores
- Selectores de color
- …

Los sitemaps se crean en archivos de texto con la extensión **sitemap** dentro de la carpeta:

- C:\openHAB\conf\sitemaps

En Paper UI disponemos de la sección Control, pero sólo sirve para Things (los sitemaps y dashboards funcionan principalmente a nivel de items).

Sitemaps

Cada sitemap debe empezar con un elemento de tipo sitemap

sitemap **<nombre>** label="<título de la pantalla inicial>" {

    [resto de elementos]

}

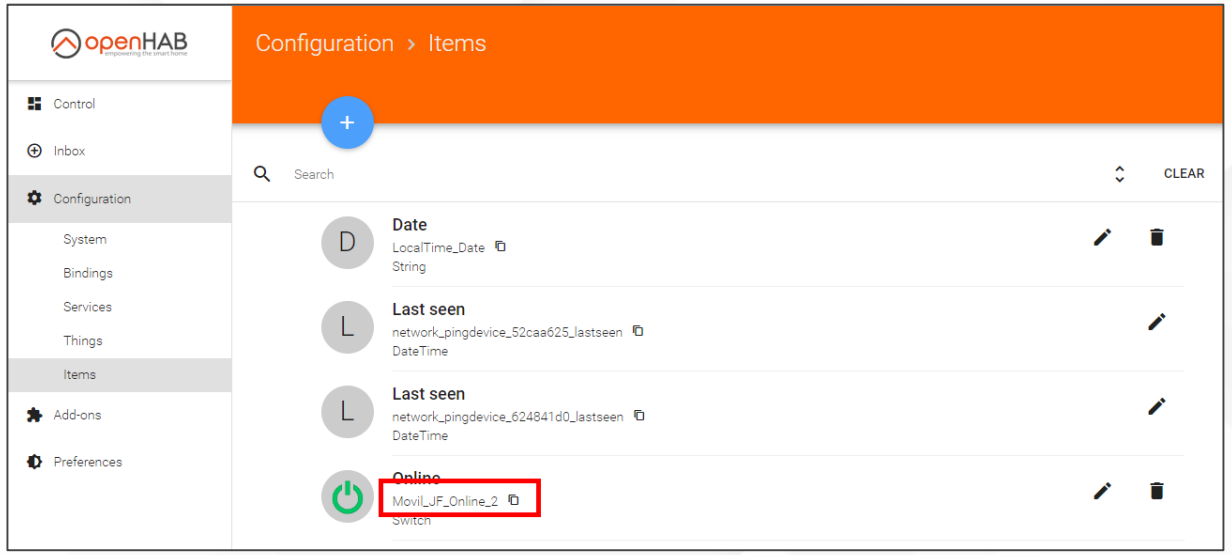
El **<nombre>** debe coincidir con el nombre del archivo; por ejemplo, si el sitemap se llama habitacion_invitados, el archivo deberá llamarse habitacion_invitados.sitemap.

Sitemaps

Cada elemento tendrá la siguiente sintaxis:
tipo item=nombre label="Rotulo"
Donde tipo puede ser:

- Switch
- Slider
- Text
- Image
- Frame (para agrupar)
- …

Y nombre es el nombre del item.

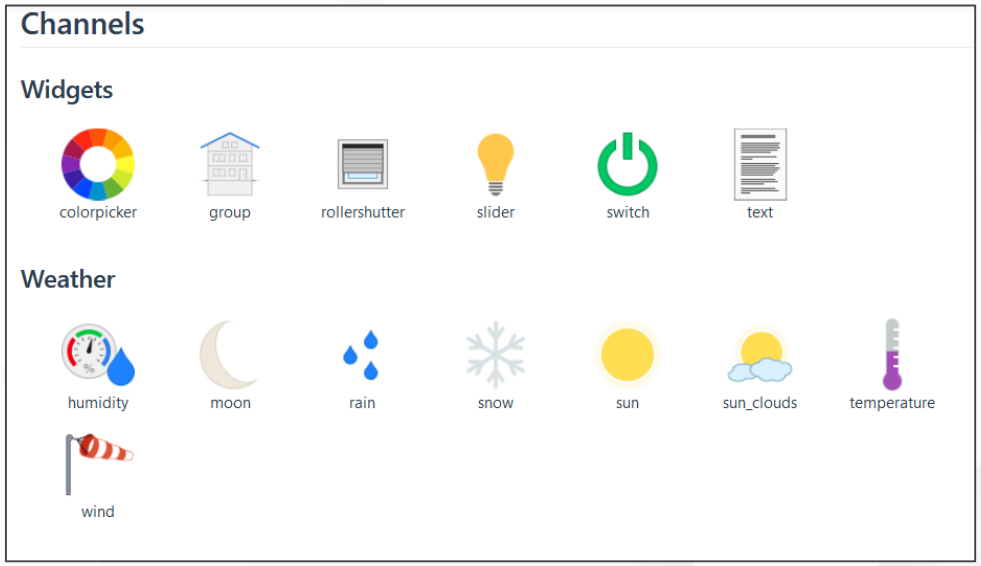


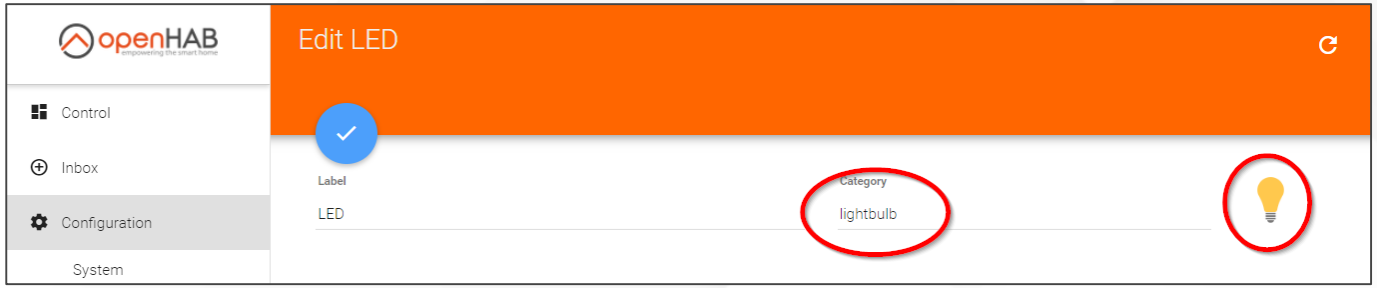
Sitemaps

Si lo deseamos podemos cambiar el icono del **item** en el sitemap eligiendo cualquiera de los disponibles en:

https://www.openHAB.org/docs/configuration/iconsets/classic/

Y escribiendo su nombre en **Category**





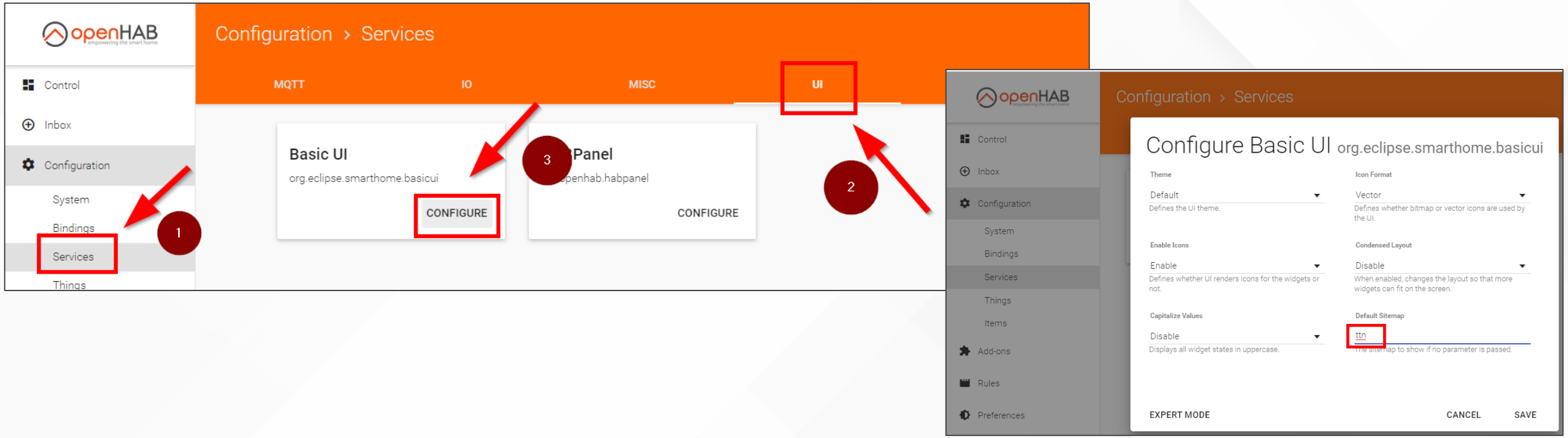
Sitemaps: things

Crear el archivo C:\openHAB\conf\sitemaps**ttn.sitemap** con el siguiente contenido:

```
sitemap ttn label="<The Things Network Madrid>" {
    Switch item=LED label="LED"
    Text item=Temperatura label="Salon [%.1f °C]" icon="temperature"
}
```

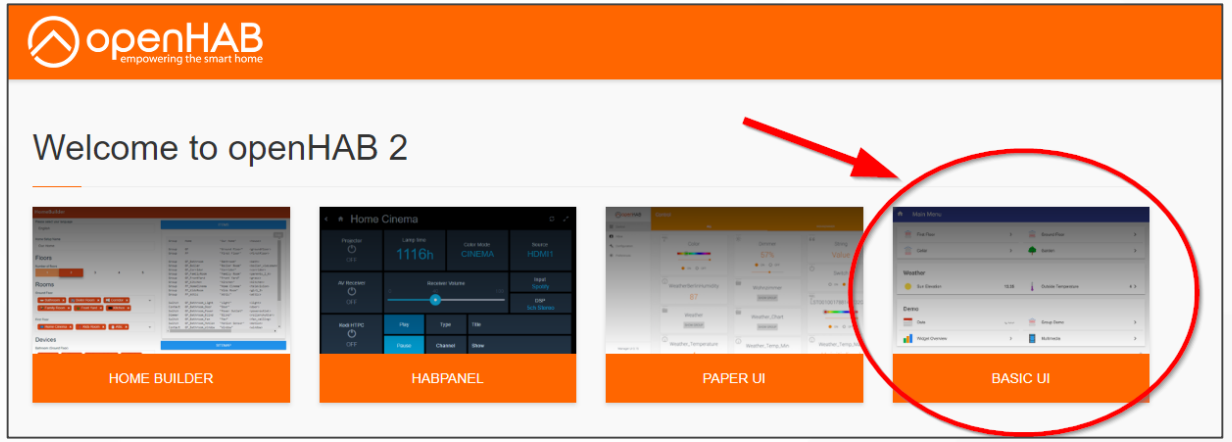
Sitemaps: Basic UI

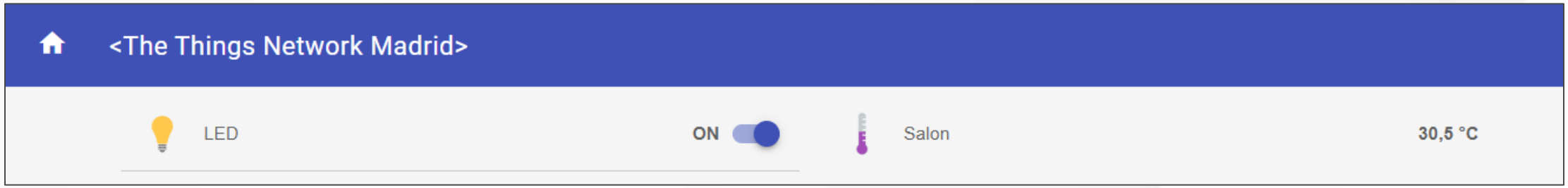
Configura Basic UI para que utilice como sitemap por defecto el que hemos creado anteriormente (habitación_invitados).



Sitemaps: Basic UI

Acceder a Basic UI a través de http://localhost:8080





Sitemaps

Además de crear ítems a través de Paper UI, también podemos hacerlo a través de documentos de texto con la extensión .items en la carpeta

C:\openHAB\conf\items

Por ejemplo, vamos a incluir en el sitemap un item que muestre la fecha:

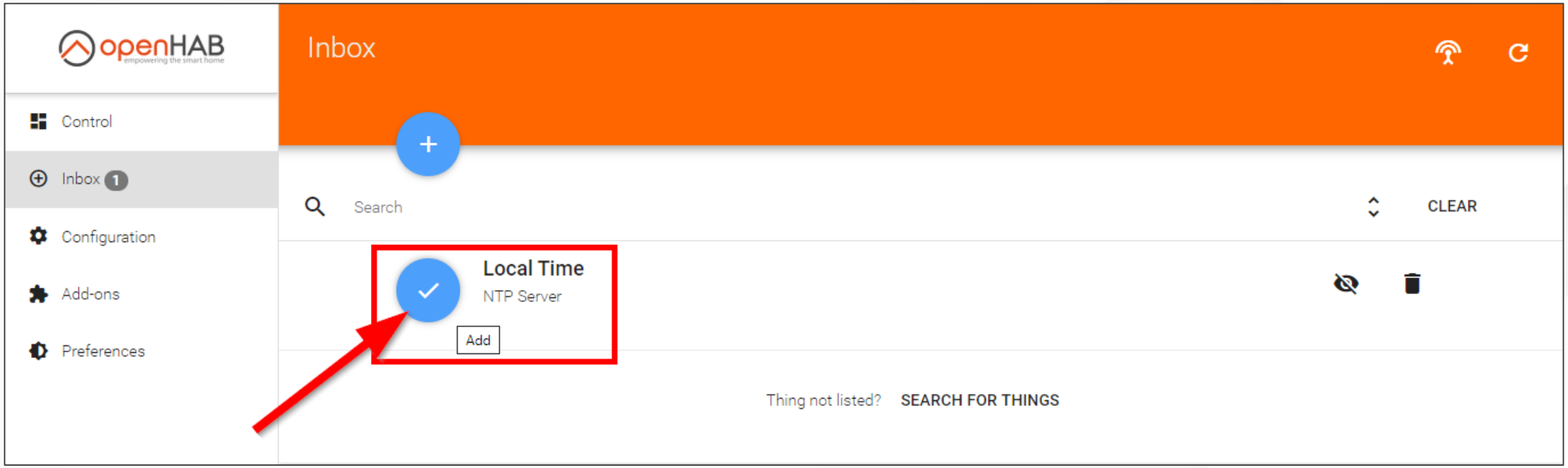
1. Importar el binding NTP.
2. Crear la thing correspondiente a través de Inbox.
3. Configurar el item a través de un archivo .items.

Sitemaps

Importar el binding NTP.

Sitemaps

Crear la thing correspondiente a través de Inbox.



Sitemaps

Cada elemento (item) debe crearse con la siguiente sintaxis

tipoElemento nombreElemento "Descripción" <icono> { canal }

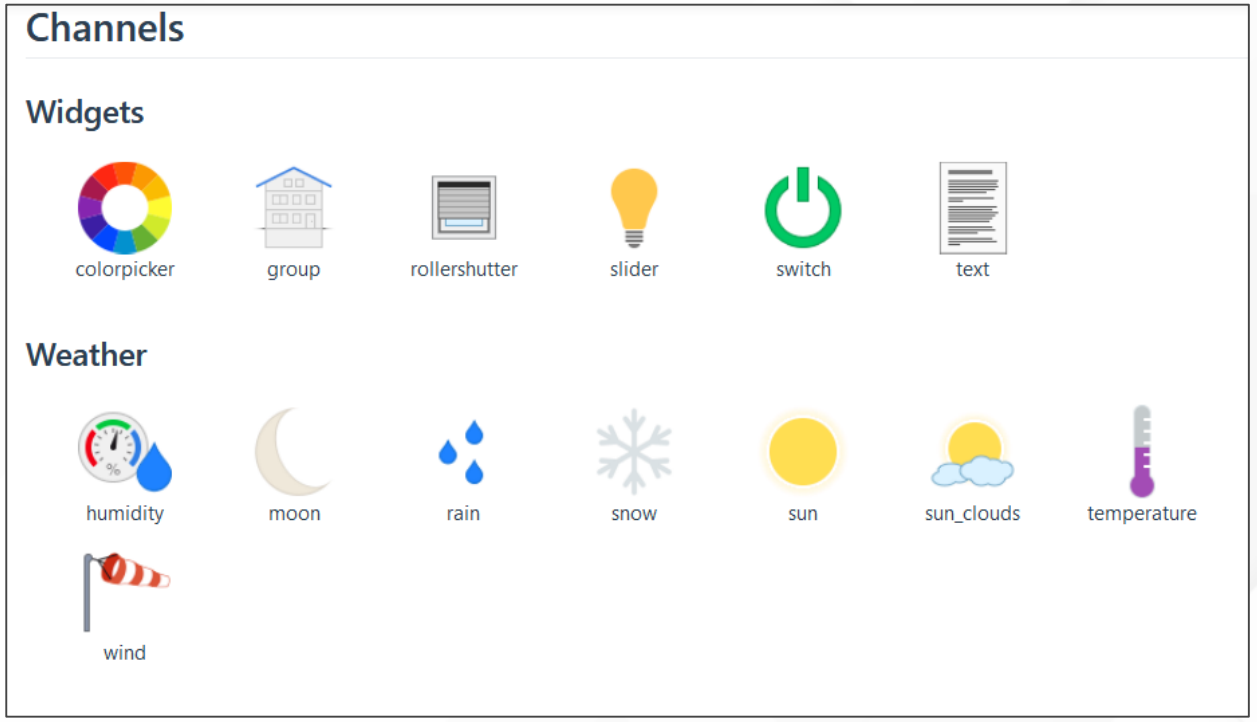
Donde tipoElemento puede ser cualquiera de la tabla.

Item Name	Description	Command Types
Color	Color information (RGB)	OnOff, IncreaseDecrease, Percent, HSB
Contact	Item storing status of e.g. door/window contacts	OpenClosed
DateTime	Stores date and time	-
Dimmer	Item carrying a percentage value for dimmers	OnOff, IncreaseDecrease, Percent
Group	Item to nest other Items / collect them in Groups	-
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:<dimension>	like Number, additional dimension information for unit support	Quantity
Player	Allows to control players (e.g. audio players)	PlayPause, NextPrevious, RewindFastforward
Rollershutter	Typically used for blinds	UpDown, StopMove, Percent
String	Stores texts	String
Switch	Typically used for lights (on/off)	OnOff

Sitemaps

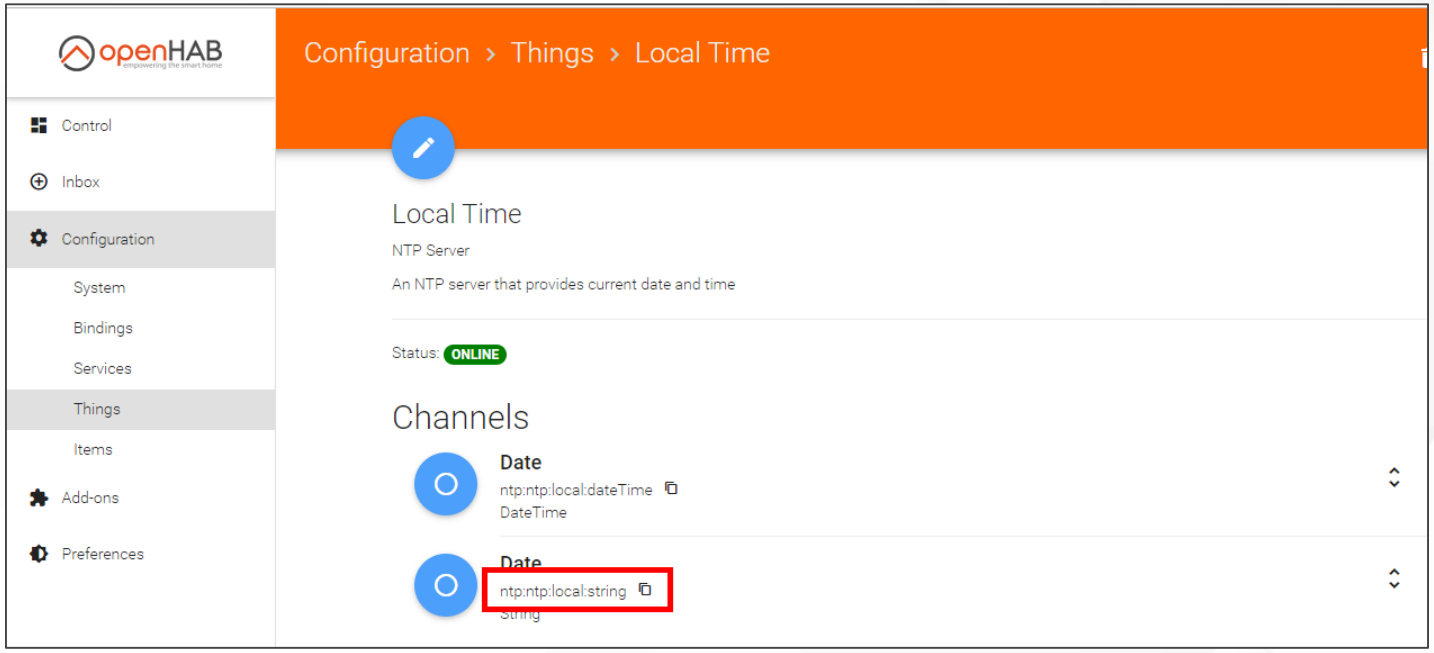
<icono> puede ser cualquiera de esta lista

https://www.openHAB.org/docs/configuration/iconsets/classic/



Sitemaps

Y **canal** es el nombre del canal que vincula la thing con el item, y que podemos obtener en la configuración de la thing



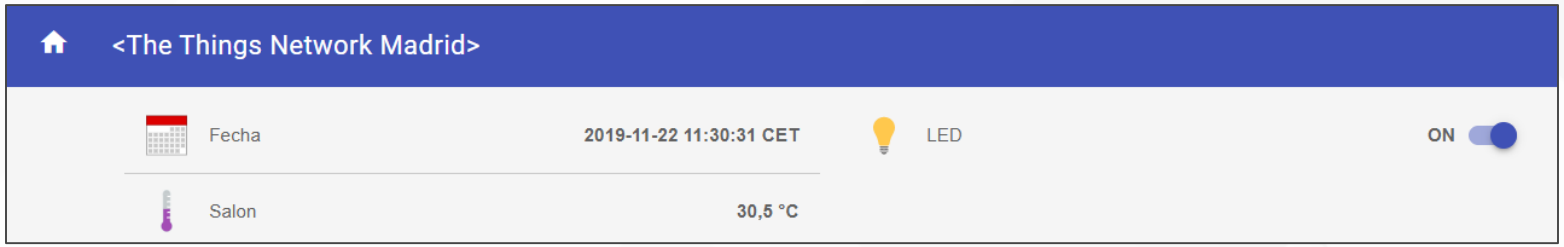
Sitemaps

Crear el archivo C:\openHAB\conf\items\miselementos.items con el siguiente contenido:

String Fecha_NTP "Fecha" <calendar> { channel="ntp:ntp:local:string" }

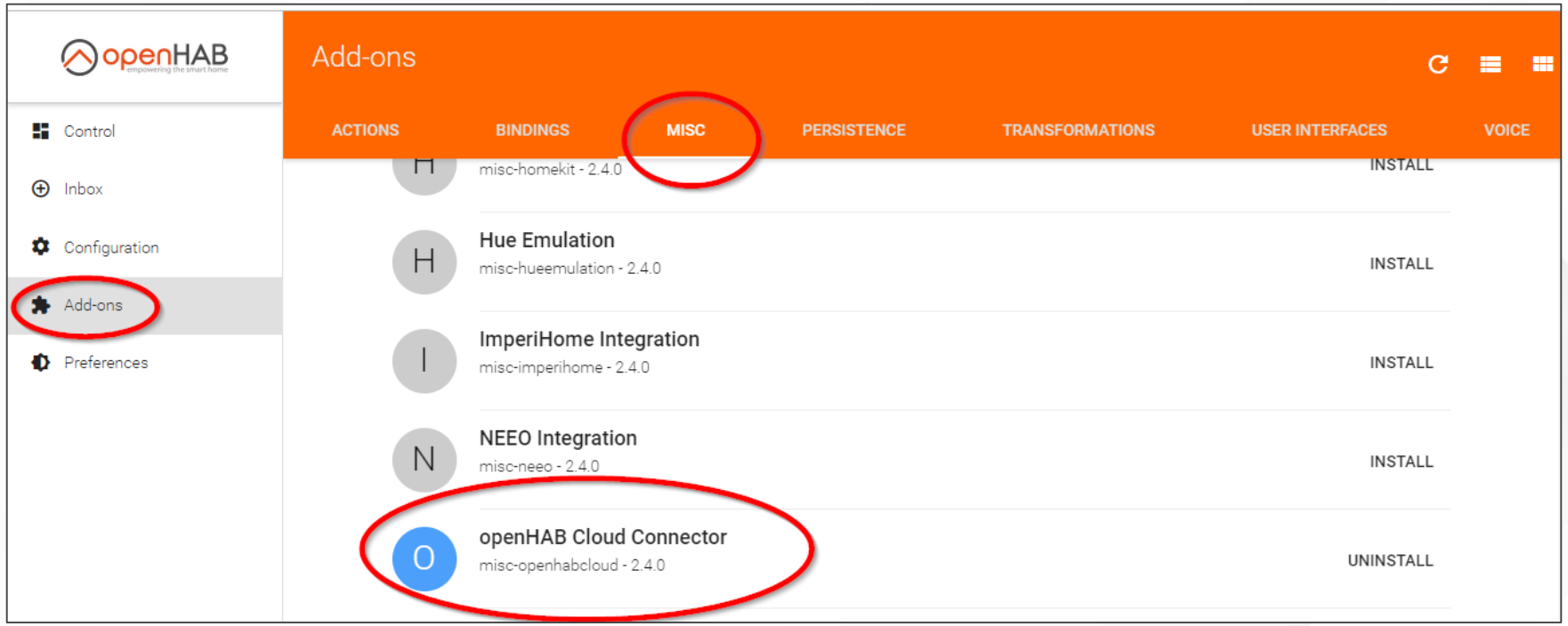
Y modificar el archivo ttn.sitemap:

```
sitemap ttn label="<The Things Network Madrid>" {
        Text item=Fecha_NTP label="Fecha"
        Switch item=LED label="LED"
        Text item=Temperatura label="Salon [%.1f °C]" icon="temperature"
}
```



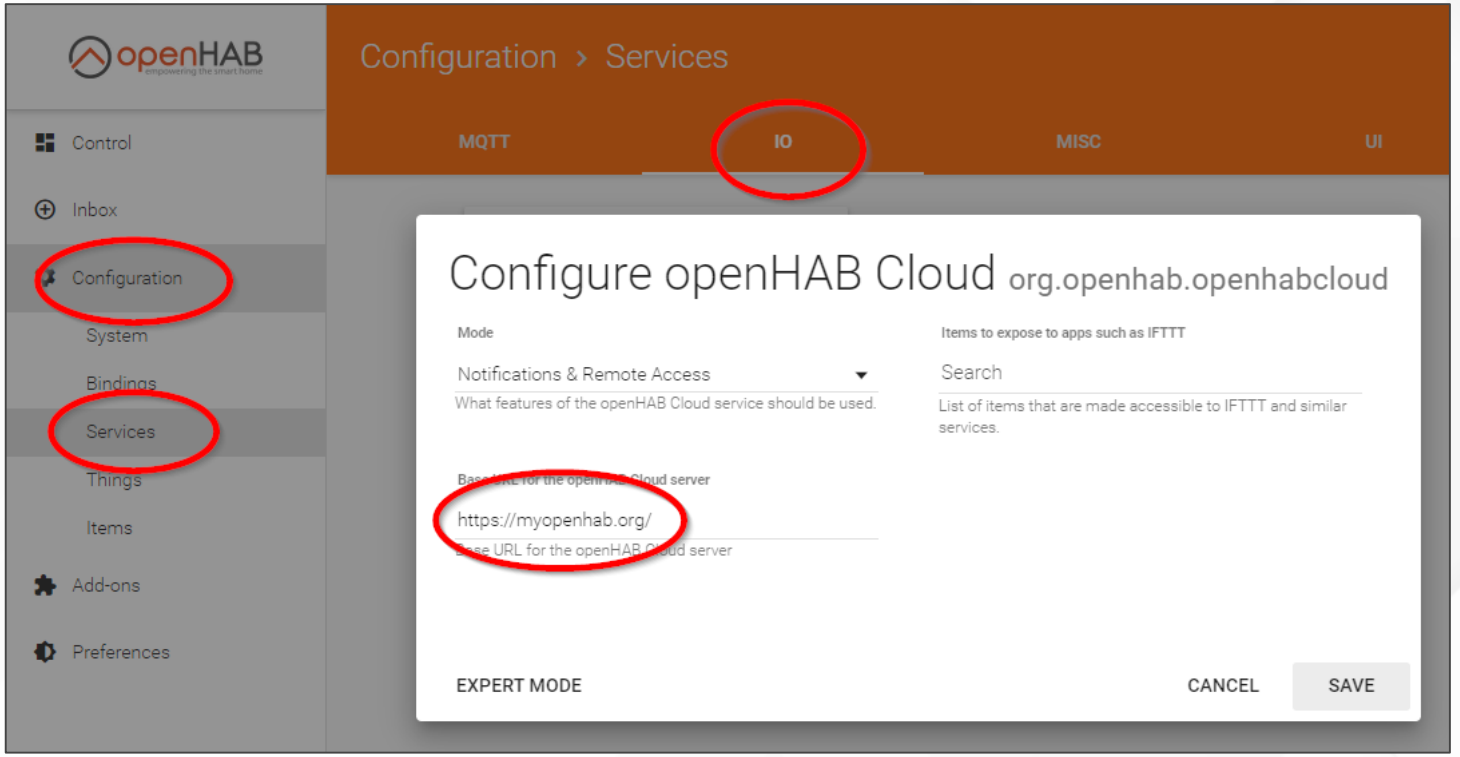
App móvil

Instalar el Add-on openHAB Cloud Connector



App móvil

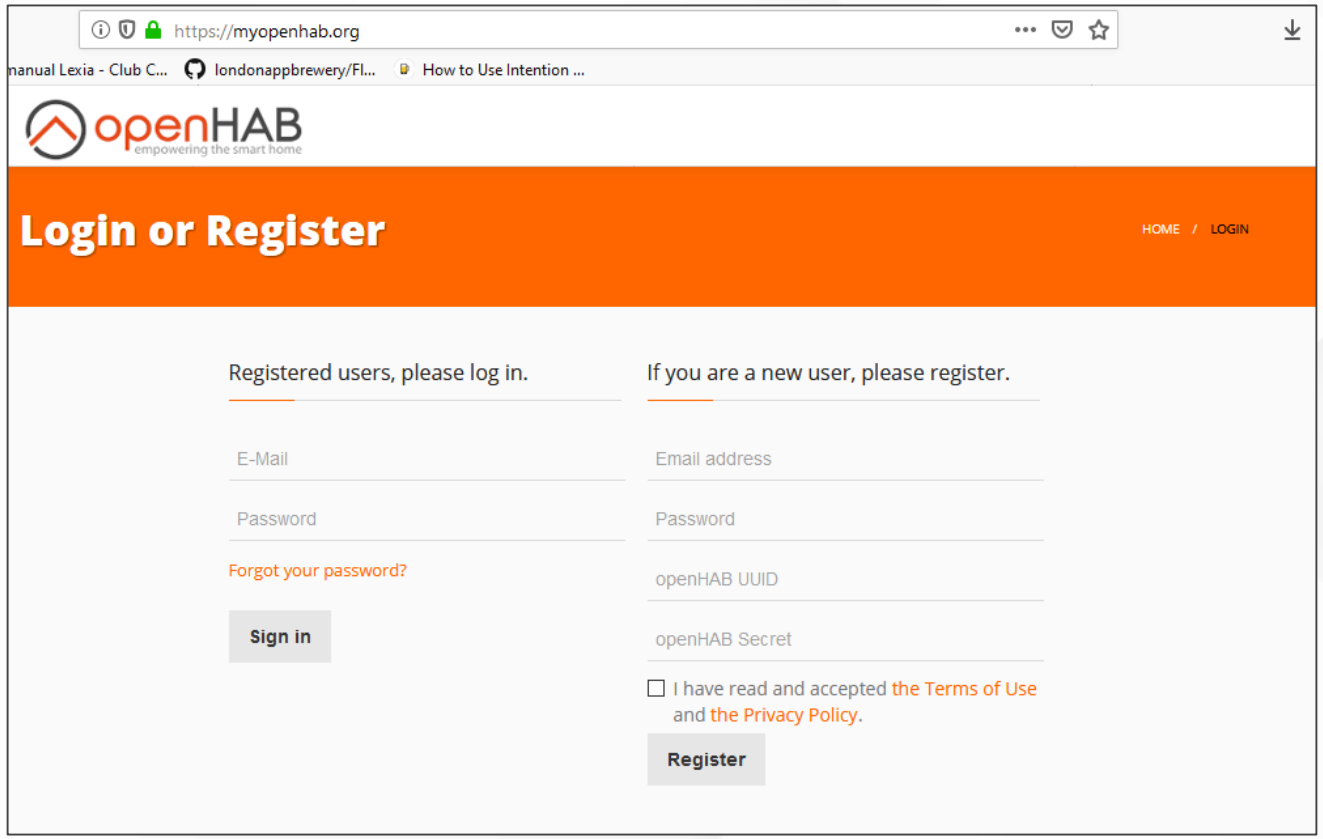
Configurar openHAB Cloud Connector para que utilice el servidor https://myopenhab.org



App móvil

Crear una cuenta en https://myopenhab.org

- UUID: C:\openHAB\userdata\uuid
- Secret: C:\openHAB\userdata\openhabcloud\secret

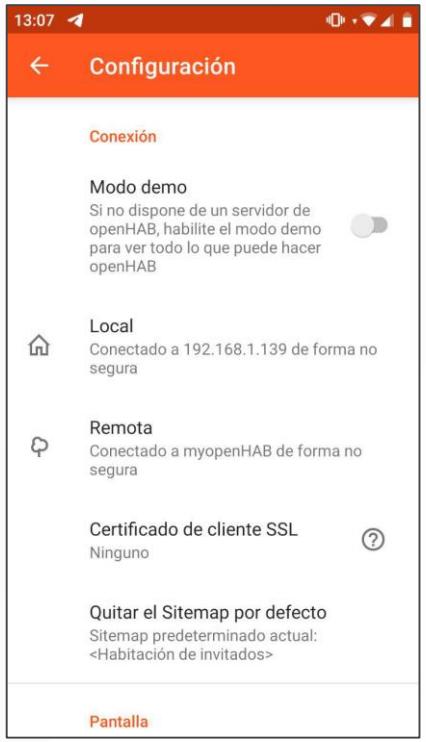


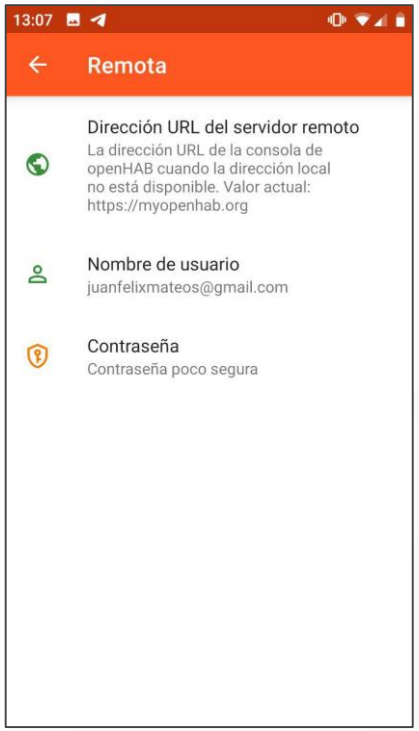
App móvil

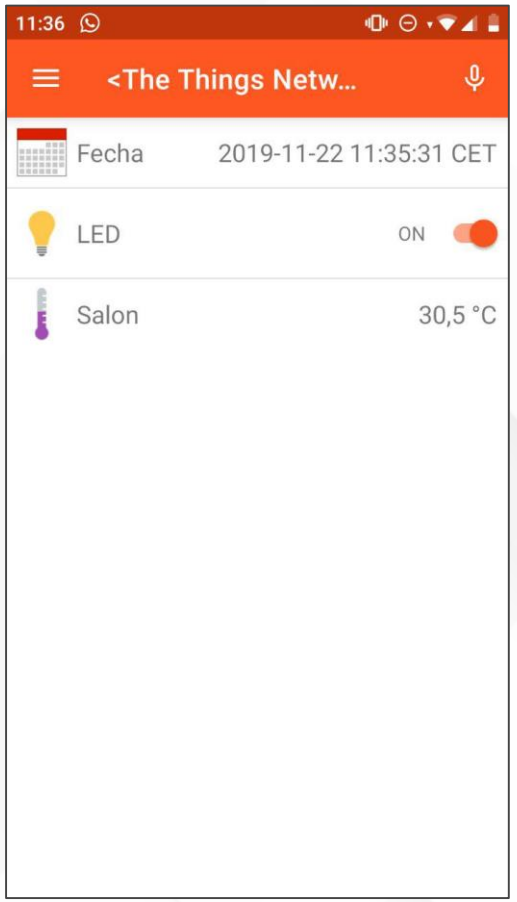
Instalar la aplicación en el móvil

App móvil

Configurar la app para que se conecte a myopenhab.org





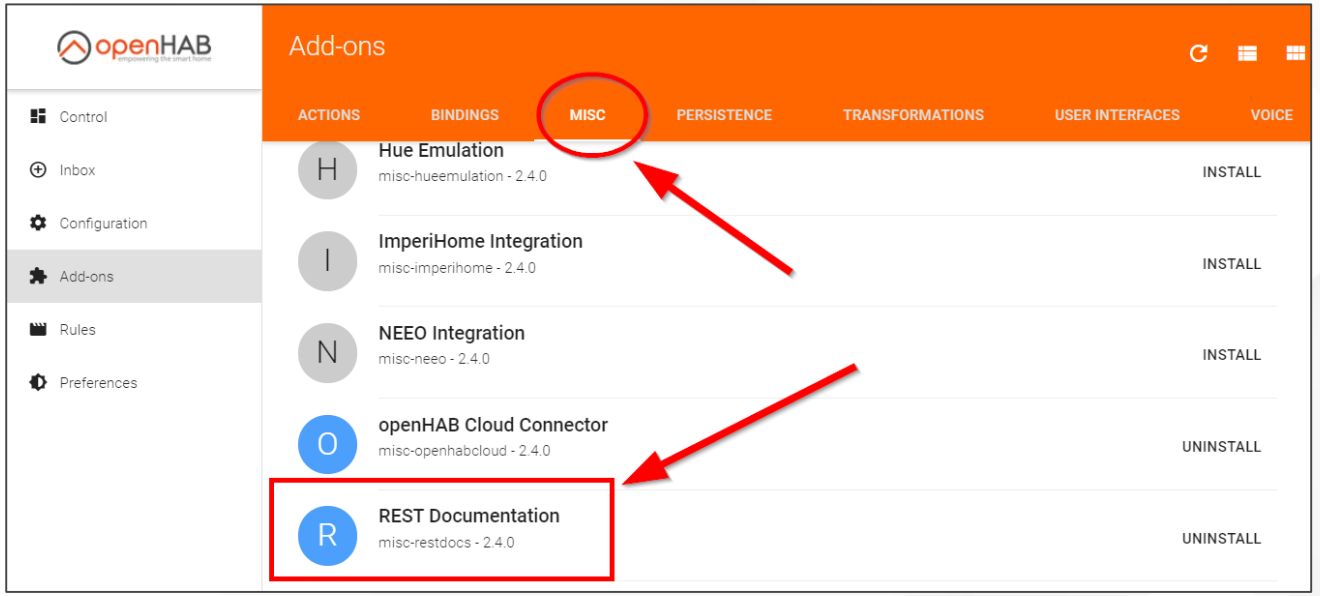


Alexa

Para que Alexa pueda identificar a nuestro nodo necesitamos aportar a los items **metainformación** que indique qué posibilidades ofrecen.

Para aportar esta metainformación vamos a recurrir a la Rest API de openHab que, si instalamos el binding **openHAB Rest Documentation**, está disponible a través de Swagger en:

- http://localhost:8080/doc/index.html



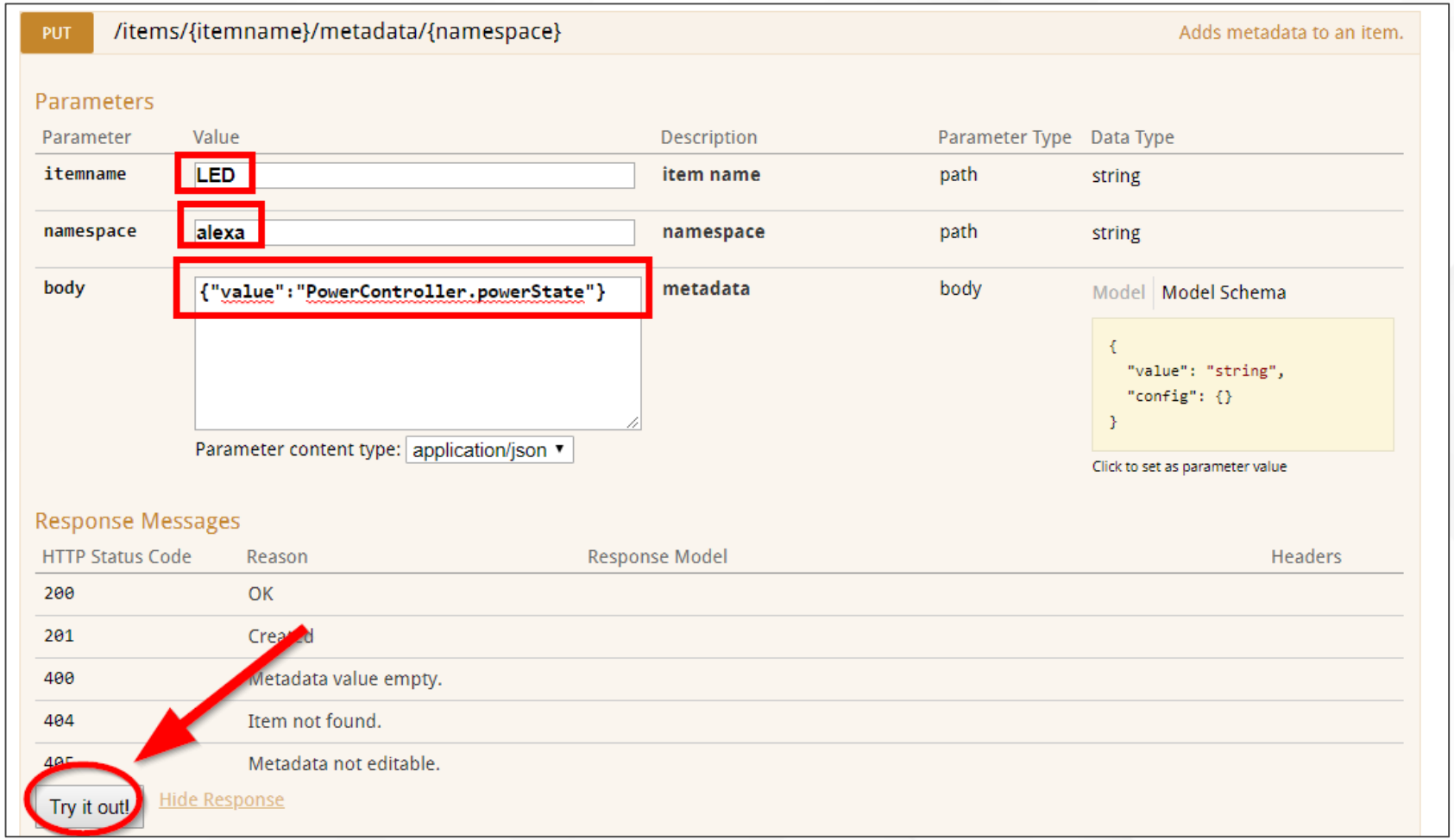
Alexa

El listado de metadatos está disponible en:

- https://www.openhab.org/docs/ecosystem/alexa/#supported-item-metadata
- Por ejemplo:
  - Interruptor: PowerController.powerState
  - Sensor: TemperatureSensor.temperature

Alexa

Metadatos de LED



Alexa

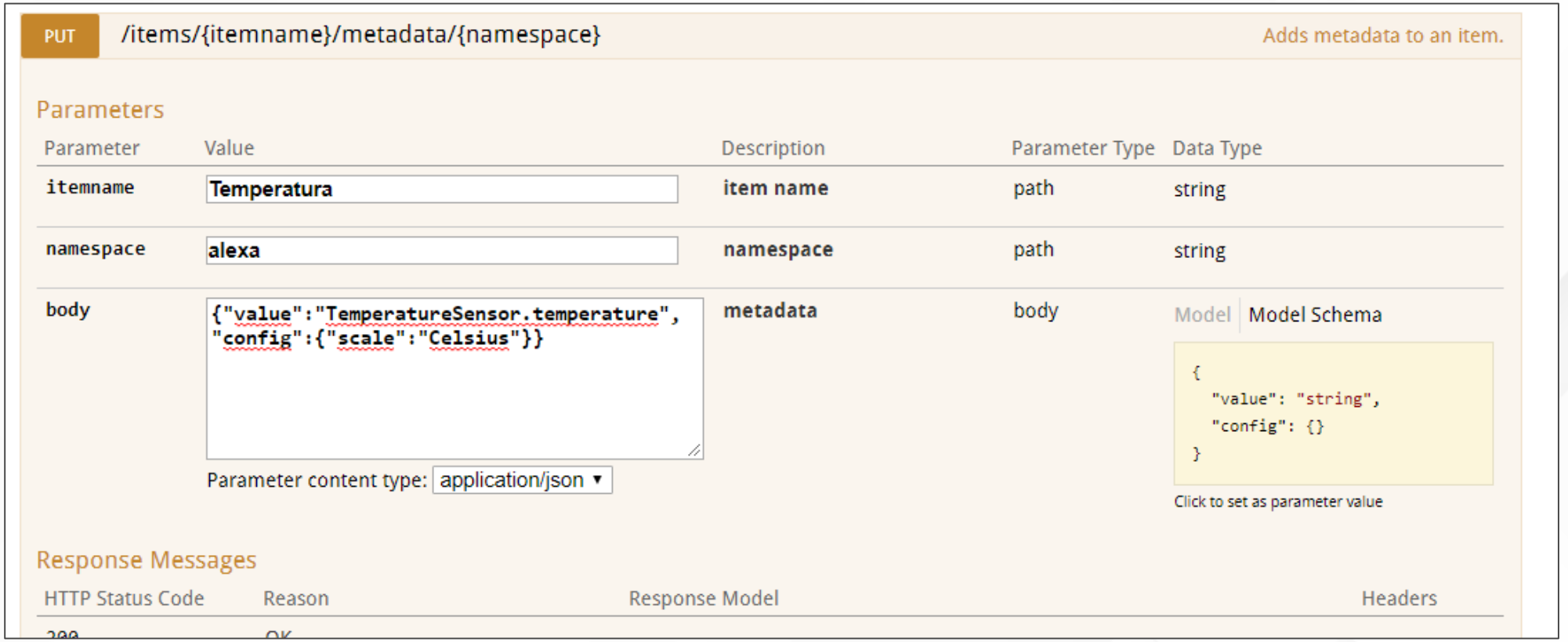
Podemos comprobar que el metadato se ha añadido correctamente revisando el archivo:

- C:\openHAB\userdata\jsondb**org.eclipse.smarthome.core.items.Metadata.json**

```
28   "alexa:LED": {
29     "class": "org.eclipse.smarthome.core.items.Metadata",
30     "value": {
31       "key": {
32         "segments": [
33           "alexa",
34           "LED"
35         ]
36       },
37       "value": "PowerController.powerState",
38       "configuration": {}
39     }
40   }
```

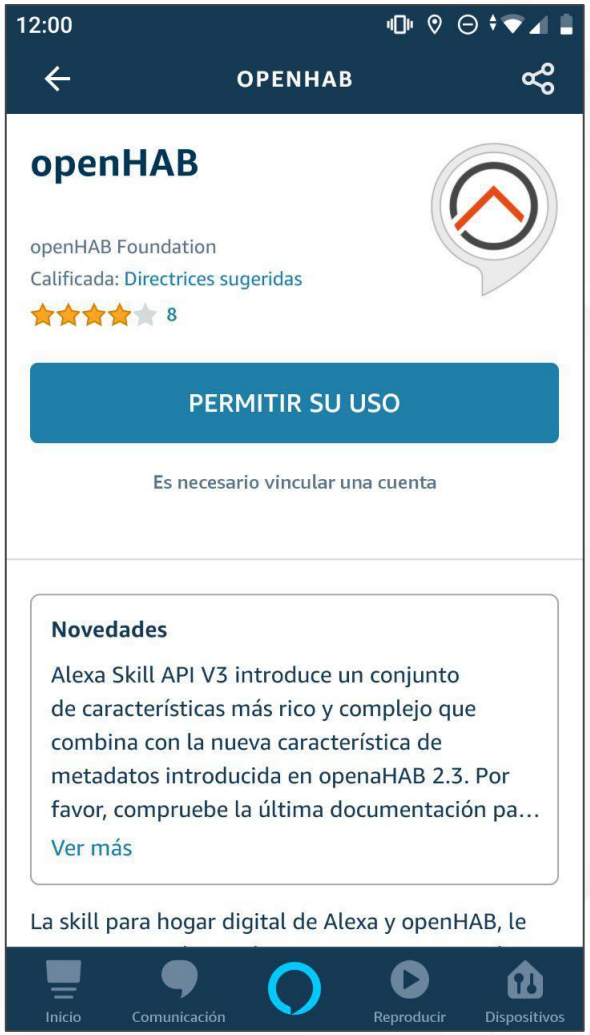
Alexa

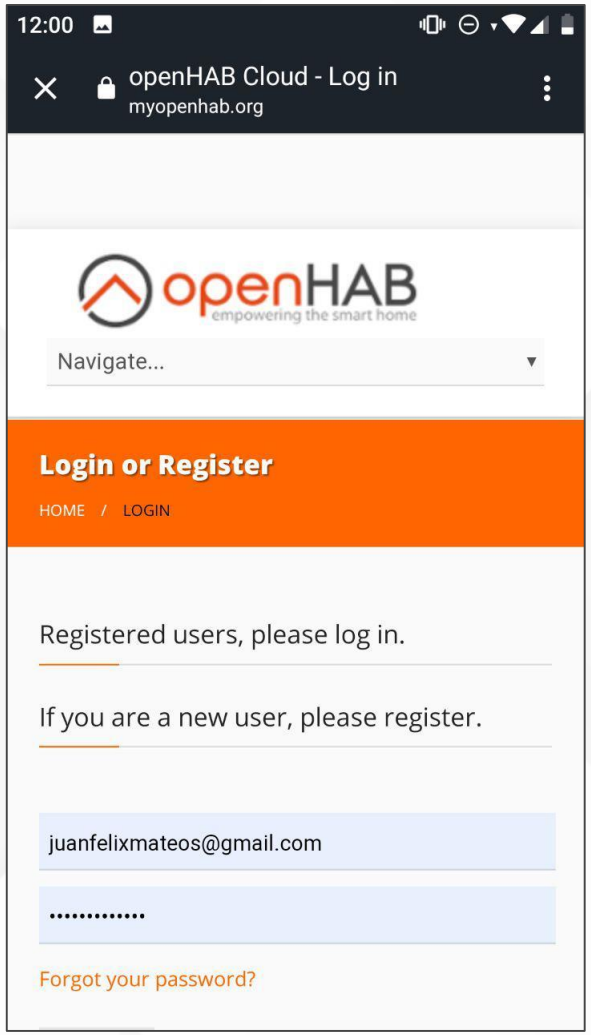
Metadatos de Temperatura



Alexa

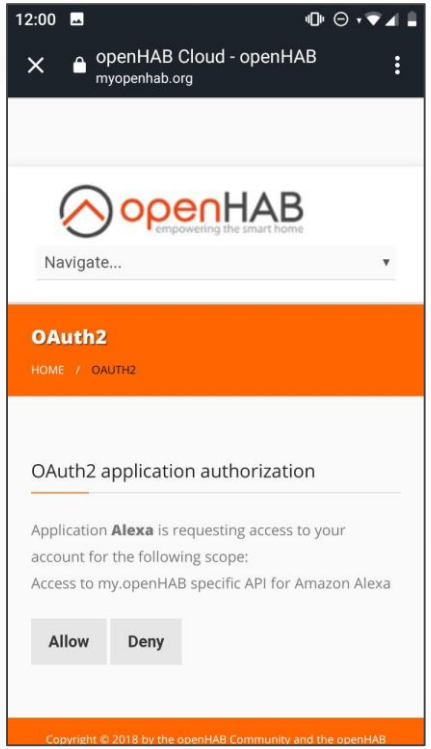
- Instalar la app Alexa en el móvil
- Activar el skill oficial de openHAB
- Acreditarse en myopenhab

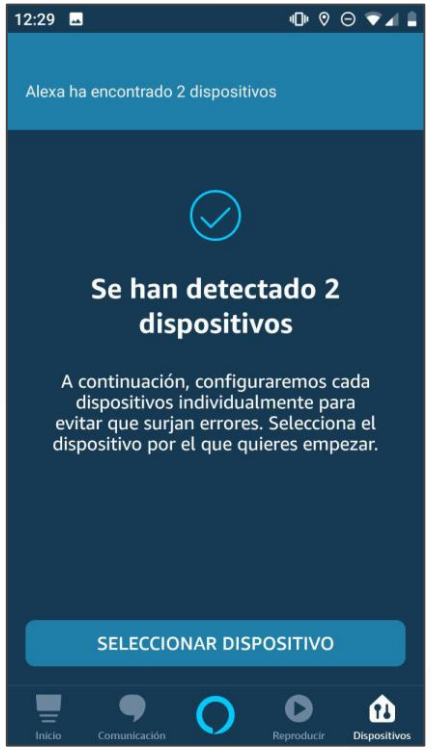




Alexa

- Aceptar la autorización OAuth2
- Permitir que Alexa detecte nuestros 2 dispositivos





Alexa

Interactuar con los siguientes comandos de voz:

- ¿Cuál es la temperatura de Temperatura?
- Activa LED
- Desactiva LED
- ¿Cuál es el estado de LED?
  - Todavía no está disponible ¿o sí?
    - https://github.com/openhab/openhab-alexa/issues/239

jfmateos@educa.madrid.org

juanfelixmateos@gmail.com

GRACIAS