



**UNIVERSITA' POLITECNICA DELLE MARCHE**

**FACOLTA' DI INGEGNERIA**

---

Corso di Laurea triennale in Ingegneria Informatica e  
dell'Automazione

Implementazione su macchina virtuale della piattaforma  
OpenHAB e valutazione delle prestazioni.

Virtual machine implementation of the OpenHAB platform and  
performance evaluation.

Relatore:

Prof. Gambi Ennio

Correlatore:

De Santis Adelmo

Tesi di Laurea di:

D'Aurelio Cristina

A.A. 2021 / 2022



# INDICE

<b>1. Introduzione: brevi cenni sulla domotica.....</b>	<b>3</b>
<b>2. OpenHab e virtualizzazione.....</b>	<b>5</b>
2.1. Cos'è openHAB, come si ottiene e quali sono le sue potenzialità.....	5
2.2. Virtualizzazione: vantaggi e svantaggi.....	8
2.3. Integrazione di OpenHAB in ambiente virtuale, eventuali criticità.....	10
<b>3. Integrazione dei sensori.....</b>	<b>17</b>
3.1. Breve panoramica dei sensori utilizzati.....	19
3.2. MQTT e la sua funzione di binding.....	23
3.3. Risultati ottenuti nella integrazione.....	25
<b>4. Conclusioni e sviluppi futuri.....</b>	<b>29</b>
4.1. Prestazioni ottenute.....	29
4.2. Possibilità di sviluppo.....	31
4.3. Possibilità di monitoraggio e reportistica remota.....	31
<b>5. Bibliografia.....</b>	<b>35</b>

# 1. Introduzione: brevi cenni sulla Domotica

Il termine “DOMOTICA” è sempre più diffuso all’interno delle nostre abitazioni. Questa parola deriva storicamente dal francese *domotique*, a sua volta contrazione della parola latina *domus* (casa) e di *automatique* (automatica, o secondo alcuni *informatique*, informatica).

Per domotica si intende quella disciplina che si occupa dello studio e dell’applicazione di dispositivi elettronici e di sistemi di comunicazione per l’automazione della gestione e del controllo della casa, volte a migliorare la qualità della vita negli edifici in generale.

Il sistema domotico consente la gestione coordinata di tutti i dispositivi, massimizzando l’efficienza della propria abitazione, sfruttando anche i vantaggi dell’Internet of Things (IoT), garantendo sicurezza, confort ed efficienza energetica.

L’integrazione della domotica in ambito abitativo consente l’applicazione di diverse tecnologie e dispositivi che possono essere regolati, attivati e controllati anche da remoto. Tutto ciò può avvenire con semplici azioni, ovvero utilizzando il proprio smartphone, pc o tablet.

Sono diversi i settore/ambiti in cui è possibile sfruttare questa tecnologia, la quale offre potenzialità notevoli, agevolando sempre di più edifici pubblici e privati ma anche ambienti di lavoro come cliniche e strutture specializzate.

Infatti, negli ultimi anni si è assistito ad un’espansione della cosiddetta “Domotica assistenziale”, nata dall’esigenza di dare la possibilità ad anziani e persone con disabilità di compiere azioni essenziali.

Questa branca della domotica va oltre la sicurezza della casa, il confort e il risparmio energetico, in quanto è volta a migliorare la qualità della vita delle persone con capacità ridotte e di chi li assiste, permettendo all’utente di godere di un ambiente più agibile ed autonomo.

Infatti, con la domotica assistenziale, è possibile, ad esempio, azionare il riscaldamento o alzare le tapparelle attraverso un telecomando o un comando vocale, tramite un linguaggio intuitivo e semplice, adatto a chiunque.

## 2. OpenHAB e virtualizzazione

### 2.1 Cos'è *openHAB*, come si ottiene e quali sono le sue potenzialità

Open Home Automation Bus, o meglio conosciuto come “OpenHAB”, è una piattaforma di automazione domestica *open source* sviluppata in Java che permette di combinare insieme diversi dispositivi domotici in un'unica soluzione.

OpenHAB è basata su una architettura modulare, tramite la quale è possibile far interfacciare in modo efficace più di duecento tecnologie e sistemi, per un totale di migliaia di dispositivi domotici presenti sul mercato, basata su *addon*.

Infatti, la peculiarità di questo software consiste nell'essere facilmente interfacciabile a diverse tecnologie e prodotti smart della casa come ad esempio impianti di domotica, impianti di allarme, impianti di illuminazione, TV, il modem, lo stereo ed ecc.

Si può notare che OpenHAB non è collegabile solo a sistemi fisicamente connessi, ma anche a qualsiasi altra fonte di informazione come un servizio web o un sistema di videosorveglianza.

Per comprendere al meglio OpenHAB bisogna definire alcuni concetti cardine che sono alla base del suo funzionamento:

- Bindings: ovvero pacchetti software che consentono l'integrazione di servizi e di dispositivi.

Il loro scopo è quello di rendere disponibili i devices al sistema di domotica che ne ha bisogno, e la loro presenza semplifica il lavoro, in quanto comunicano con il dispositivo fisico e traducono tutti i comandi che OpenHAB impartisce e riceve;

- **Things:** sono le entità che possono essere aggiunte fisicamente al sistema.  
Non devono essere necessariamente dei dispositivi, ma possono rappresentare anche dei servizi web o qualsiasi altra fonte di informazione e funzionalità.  
Sono essenziali per il processo di configurazione ma non anche per l'operatività dei suddetti.
  - **Channels:** sono una emanazione delle things e vanno a rappresentare le sue diverse funzionalità.  
Solo una volta configurati i vari *channels*, il sistema può usufruire delle funzionalità rese disponibili.  
Gli stessi, essendo collegati agli elementi, fungono da collante tra il livello virtuale e quello fisico;
  - **Items:** sono tutti gli “oggetti”, quindi le informazioni, i dispositivi, i sensori ed ecc. che possono essere utilizzati dalle applicazioni.  
Diversamente dalle *things* sono importati per l'operatività e possono avere uno o più stati.  
Gli *items* possono essere di due tipologie:
    - **Points:** sono degli oggetti virtuali che rappresentano le proprietà delle *things*.  
Ad ogni point viene associata una singola *things*.
    - **Equipments:** sono delle scatole virtuali che possono contenere svariati *points*; questi rappresentano i dispositivi fisici e regolano il loro funzionamento.
  - **Rules:** sono le regole e i parametri che vengono utilizzati per automatizzare i processi, in modo da eseguire certe azioni in determinate condizioni.  
Ad esempio, quando ad un ora prestabilita si alzano le serrande di una determinata stanza ed ecc.
  - **Sitemap:** è l'interfaccia utente semplice e intuitiva generata da OpenHAB che permette di scegliere come visualizzare gli elementi e di raggrupparli e organizzarli in sotto-categorie.
- Tutto ciò premesso, per capire il funzionamento di un sistema di domotica gestito dalla piattaforma OpenHAB, è necessario definire due parti distinte ma che lavorano insieme:

-Parte fisica: riguarda le caratteristiche hardware del sistema, quindi tutti i collegamenti e i dispositivi che costituiscono il sistema di domotica;

-Parte funzionale: la quale si occupa di determinare e gestire il modo in cui i vari dispositivi fisici vengono connessi tra di loro.

Inizialmente la prima versione di OpenHAB non disponeva di un'interfaccia grafica e quindi per programmare la domotica si era costretti ad utilizzare solo e soltanto i codici.

In seguito, con la release OpenHAB 3 è stata aggiunta l'interfaccia grafica ed il lavoro è risultato meno complesso, facendo tutto tramite quest'ultima.

OpenHAB, essendo scritto completamente in Java, dipende solo dalla JVM (Java Virtual Machine) e può essere eseguito da diverse versioni di Windows, macOS e diverse distribuzioni Linux (Ubuntu, Rasbian ed ecc..).

Per utilizzare questa piattaforma bisogna essere dotati di una rete WiFi domestica, di un computer che funge da server per il sistema domotico che si vuole implementare. È consigliato installare il software su un MiniPC a bassissimo consumo progettato proprio per questo tipo di impiego, come ad esempio il RaspBerryPI. In questo modo, una volta scaricato il software, può essere lasciato sempre acceso e collegato alla rete al fine di sfruttare al massimo tutte le sue potenzialità.



*Figura 2.1 -Logo OpenHAB*



## *2.2 La virtualizzazione: vantaggi e svantaggi*

La virtualizzazione consiste nell'astrazione di quelle che sono le risorse fisiche dell'IT, si possono astrarre sia le componenti hardware che le componenti software.

Lo scopo è quello di fornire tali risorse a livello virtuale e distribuirle in modo flessibile a seconda delle esigenze dei diversi clienti, ottimizzando l'utilizzo dell'hardware disponibile.

La virtualizzazione utilizza il software per simulare le funzionalità hardware e creare un computer virtuale.

Una componente IT creata in questo modo viene definita come componente virtuale o logica, e potrà essere utilizzata allo stesso modo del suo corrispettivo fisico.

In questo campo spesso si tende a confondere la virtualizzazione con la simulazione ed emulazione. Questi due concetti non solo differiscono tra di loro ma, a loro volta, si differenziano anche dalla virtualizzazione.

Infatti, con la simulazione si vuole indicare la riproduzione completa di un sistema attraverso un software.

Si tratta di una riproduzione completa in quanto non sono imitate solo determinate funzioni di interazione con i sistemi differenti, ma anche le varie componenti del sistema. Quindi, i simulatori vengono spesso utilizzati per compilare programmi sviluppati per altre architetture hardware.

L'Emulazione rappresenta le varie funzioni delle componenti hardware e software e non quella che è la loro logica intrinseca.

Il fine dei simulatori e degli emulatori è quello di permettere il funzionamento di applicazioni su sistemi che non lo supportano.

La virtualizzazione, che non dovrebbe essere emulata e simulata, a differenza dei due concetti sopra spiegati, nasce con l'idea di dover stabilire un grado di astrazione che permette di fornire risorse IT, indipendentemente da quella che è la base fisica.

Il concetto di virtualizzazione venne coniato negli anni 60 per indicare la creazione di macchine virtuali, definite come container software altamente isolati che includono un sistema operativo e applicazioni, completamente indipendenti e una dalle altre.

Mentre nell'odierno scenario IT esistono diverse forme di virtualizzazione riferite all'astrazione di risorse IT: hardware, software, memoria, dati e componenti di rete. Più in generale si può fare una distinzione tra:

**VIRTUALIZZAZIONE HARDWARE:** si riferisce a tecnologie che permettono di fornire componenti hardware attraverso software indipendentemente dalla loro base fisica.

Ci sono diversi tipi di applicazione di virtualizzazione hardware:

-Il primo si riferisce alla **Macchina Virtuale** (VM) che funziona come sistema guest virtuale su uno o più sistemi fisici chiamati host.

Il livello di astrazione tra la base fisica e il sistema virtuale in questo tipo di virtualizzazione viene stabilito da un *hypervisor*, che rappresenta il software che permette il funzionamento di più sistemi guest in un unico host.

Gli *hypervisor* regolano le risorse hardware messe a disposizione dal sistema host, come la CPU, le periferiche, la RAM ed ecc. e le condividono con un numero di sistemi guest che preferiscono;

- il secondo riguarda il consolidamento dei **server** in ambito aziendale, in questo caso la virtualizzazione viene utilizzata con lo scopo di aumentare l'utilizzo dell'hardware dei server.

Infatti, vengono eseguiti diversi server guest separatamente, ma sulla stessa piattaforma hardware.

I vantaggi principali della virtualizzazione hardware riguardano:

1. dal punto di vista energetico, avere un hardware consolidato risulta più efficiente rispetto ad avere computer separati;
2. le VM offrono livelli di sicurezza più elevati attraverso l'isolamento dei *workload*.

Mentre tra gli svantaggi troviamo:

1. La possibilità di *overhead*, dovuta alla riproduzione di un ambiente hardware con il proprio sistema operativo;
2. Le prestazioni di una Virtual Machine possono essere influenzate dalla presenza di altre VM sullo stesso host.

VIRTUALIZZAZIONE SOFTWARE: le tipologie più comuni riguardanti questo tipo di virtualizzazione sono:

- Virtualizzazione delle applicazioni: riguardano l'astrazione delle singole applicazioni rispetto a quello che è il sistema operativo sottostante, risultando molto efficaci per l'implementazione locale;
- Virtualizzazione del desktop: è una tecnologia che consente agli utenti di simulare il carico di una *workstation* per accedere a un **desktop** tramite un dispositivo connesso, ideali in ambito aziendale.
- Virtualizzazione del sistema operativo: utilizzate per indicare un metodo di esecuzione delle applicazioni che isola le risorse hardware e software in uso da ognuno di esse, sempre all'interno di un singolo sistema operativo; hanno la particolarità di riuscire ad arginare l'*overhead* degli *hypervisor*.

### *2.3 Integrazione di OpenHAB in ambiente virtuale*

Prima di installare openHAB bisogna procedere con l'installazione e la creazione della macchina virtuale.

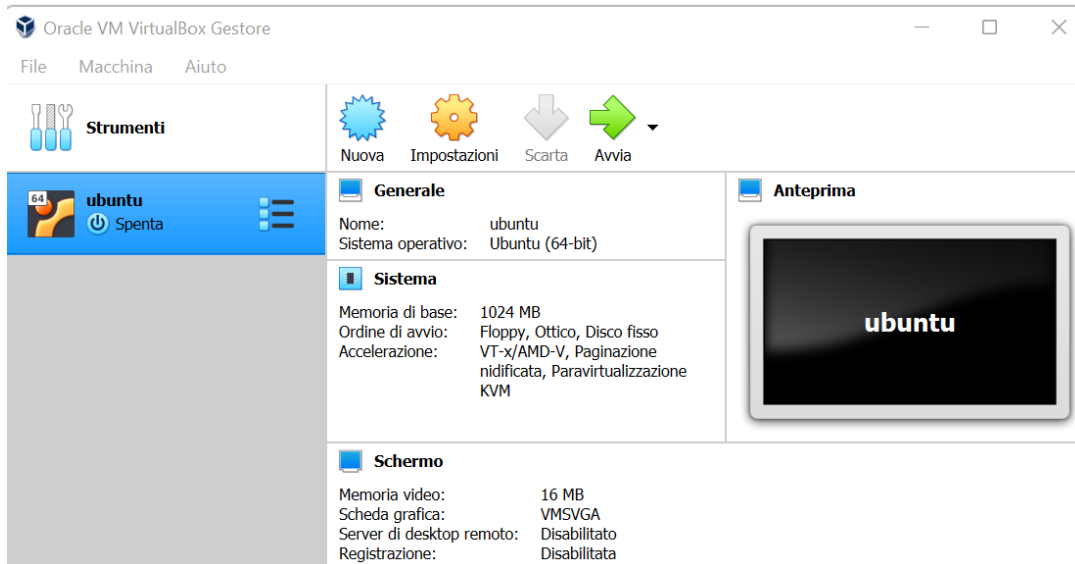
L'installazione di VirtualBox ci permette di avere un sistema operativo differente rispetto a quello che abbiamo nel nostro PC.

Ciò possiamo farlo collegandoci al sito <https://virtualbox.org/>.

Il passaggio successivo consiste nello scegliere la versione del sistema operativo che si preferisce.

In questo caso è stata scelta la versione Ubuntu server 21.10. Terminata anche questa operazione non bisogna far altro che riavviare il computer e procedere

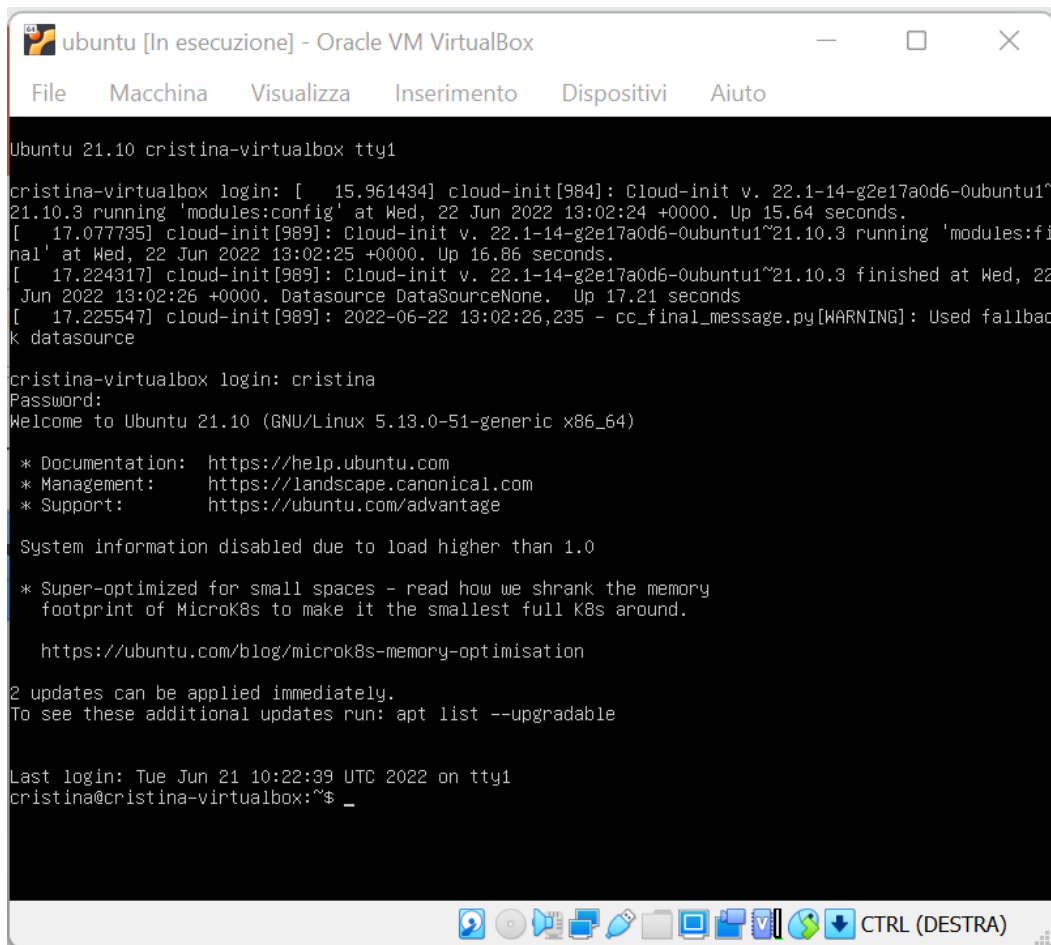
alla configurazione della Ubuntu Virtual machine. Una volta fatte tutte le configurazioni necessarie avremo la nostra macchina virtuale.



*Figura 2.2 - Schermata di avvio della VM*

Premendo sul pulsante “avvia la macchina virtuale” si procederà alla configurazione del sistema operativo, nella quale verranno richieste di effettuare una serie di operazioni.

Infine, è necessario fornire nome, nome pc, nome utente e password.



*Figura 2.3 -Schermata VM*

Completata la Virtual Machine si può procedere all'installazione di OpenHAB.

Innanzitutto, bisogna assicurarsi di aver installato Zulu o Java 11 (in questo caso è stato installato Java 11).

Per installare java11 basta digitare sul terminale i seguenti comandi:

```
$ sudo apt update
```

```
$ sudo apt install openjdk-11-jdk
```

Terminata l'installazione per verificare la versione che è stata installata si può digitare:

```
$ java -version
```

L'output sarà simile a questo:

```
$ Openjdk version "11.0.15" 2022-04-19
```

```
$ OpenJDK Runtime Environment (build 11.0.15+10- Ubuntu-  
Oubuntu.21.10.1)
```

```
$ OpenJDK 64-Bit Server VM (build 11.0.15+10- Ubuntu-Oubuntu.21.10.1,  
mixed mode, sharing)
```

Una volta installato Java11 e verificata la versione si può procedere all'installazione di OpenHAB su Ubuntu Virtual Machine.

L'installazione può essere fatta in 3 modi: attraverso il progetto OpenHABian, con i repository di pacchetti oppure manualmente da file.

Nel caso in esame si è preferita l'installazione con i repository di pacchetti, attraverso i seguenti passi:

1. per prima cosa aggiungere il repository di OpenHAB al proprio gestore di pacchetti, digitando:

```
$ wget -qO - 'https://openhab.jfrog.io/artifactory/api/gpg/key/public' | sudo  
apt-key add
```

2. Scelto "Rilascio stabile" in quanto le build stabili contengono l'ultima versione ufficiale con funzionalità testate, si aggiunge OpenHAB Stable Repository all'elenco apt dei sistemi:

```
$ echo 'deb https://openhab.jfrog.io/artifactory/ openhab-linuxpkg stable  
main' | sudo tee /etc/apt/sources.list.d/openhab.list
```

3. Sincronizzare l'indice del pacchetto:

```
$ sudo apt-get update
```

4. Si installa openHAB con il seguente comando:

```
$ sudo apt-get install openhab
```

5. Per sistemi basati su systemd

```
$ sudo systemctl start openhab.service
```

```
$ sudo systemctl status openhab.service
```

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl enable openhab.service
```

Questi ultimi comandi permettono l'avvio automatico del sistema una volta aver installato OpenHAB.

Terminate le installazioni, l'accesso alla console può essere fatto anche utilizzando SSH, attraverso il seguente comando che permette di connettersi all'interfaccia localhost sulla porta TCP 8101:

```
ssh -p 8101 openhab@localhost
```

Al primo accesso viene richiesto se si vuole proseguire o meno con la connessione, digitare "yes" e procedere con l'inserimento della password "habopen".

Una volta eseguita correttamente la connessione e l'autenticazione, apparirà la console:

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

openHAB
3.2.0 - 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>
```

*Figura 2.4 -OpenHAB da linea di comando*

Per connettersi all'interfaccia grafica bisogna collegarsi all'indirizzo ip:  
`http:// <localhost:8080>`.

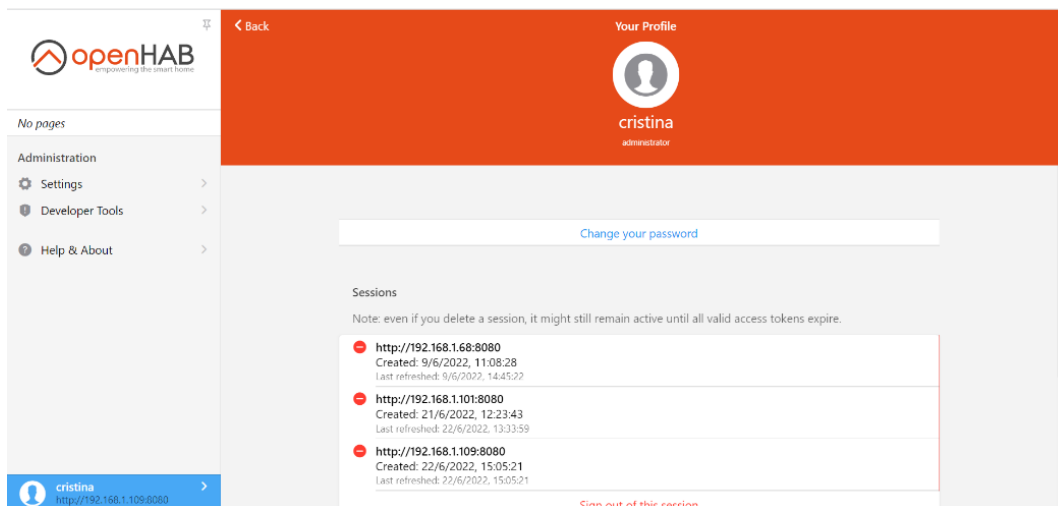
Bisogna registrarsi con un nome utente ed una password per poi effettuare l'accesso.



*Figura 2.5 Interfaccia web OpenHAB*



Una volta fatto l'accesso, inserendo il nome utente e la password si ha la seguente schermata.



*Figura 2.6 -OpenHAB*

### 3. Integrazione dei sensori

Una volta installato e configurato OpenHAB, si potrà procedere con l'integrazione dei sensori.

Tali dispositivi, per prima cosa, verranno collegati alla rete WiFi locale. La configurazione di questi viene spesso effettuata utilizzando le funzionalità Access Point di ogni dispositivo. In questo modo è possibile accedere ad una interfaccia di configurazione che consente di specificare SSID e password della rete WiFi domestica, alla quale si desidera connettersi. L'AP contenuto del dispositivo, si auto esclude al termine della configurazione.

Tramite l'interfaccia del dispositivo, nella sezione setting bisogna selezionare "firmaware updata" per provvedere all'aggiornamento dell'unità; nella stessa sezione si procede con l'abilitazione di CoIoT e di MQTT.

Dopo questi passaggi si procede con l'integrazione dei sensori all'interno di OpenHAB.

All'interno di OpenHAB, verranno installati i *binding* necessari, selezionando dal setting la sezione *binding* e procedere con la ricerca dei *binding* desiderati.

Nel caso corrente, Sono stati installati l'MQTT Binding, lo Shelly Binding e lo Yeelight Binding.

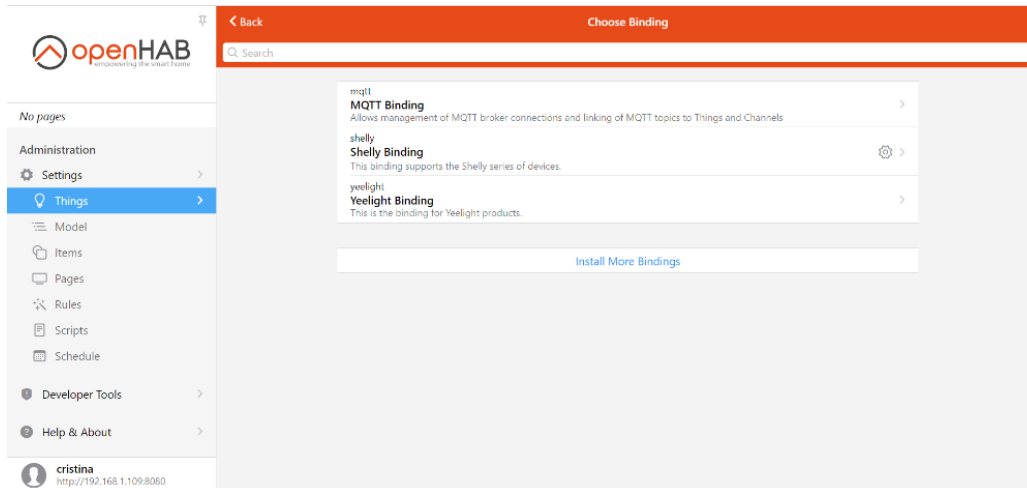


Figura 3.1 Binding

Una volta installati i *binding* necessari, si procede con il pairing dei dispositivi aggiungendo le *things* ad OpenHAB.

Ciò può essere fatto nella sezione *things* cliccando sull'icona in basso a destra per aggiungere un nuovo dispositivo; bisogna poi, selezionare dalla lista dei *binding* già installati quello desiderato e procedere con lo *scan*, che ha lo scopo di rilevare tutti i dispositivi presenti e compatibili con il binding installato; selezionare il dispositivo trovato e assegnargli un nome.

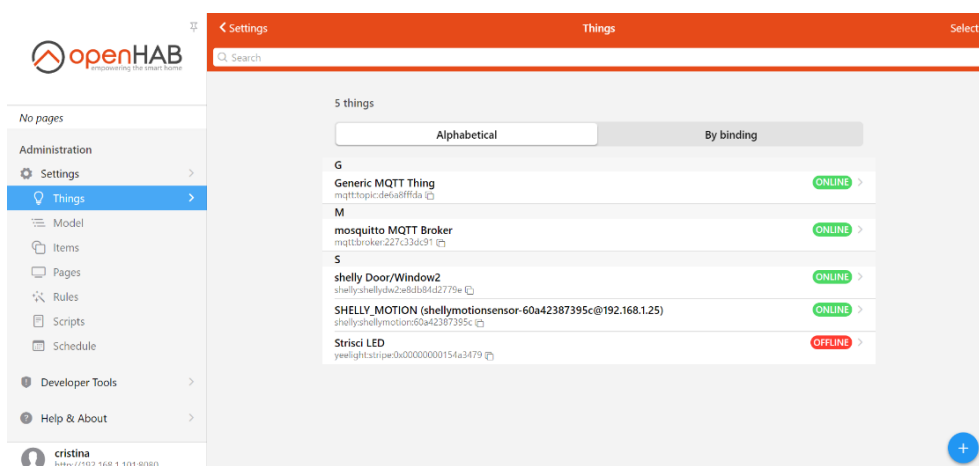


Figura 3.2 -Dispositivi integrati

### *3.1 Breve panoramica dei sensori utilizzati*

Sono stati selezionati alcuni dispositivi dell'azienda bulgara Shelly, che ha rivoluzionato il mondo della domotica. Ha creato una gamma di prodotti con grande versatilità, tale da permettere la loro integrazione all'interno di sistemi domotici e di automazione come, appunto, OpenHAB.

I dispositivi dell'azienda shelly che sono stati integrati sono lo “Shelly Motion Sensor” e lo “Shelly Door/Window2”.

Lo **Shelly Motion Sensor** è un sensore di movimento di ultima generazione, dotato di Wi-Fi IoT di Silicon Labs, connesso ad internet 24 ore su 24.

Per quanto riguarda la connettività, questo tipo di dispositivo è fornito di un Wi-Fi 802.11 b/g/n a basso consumo energetico, solo su frequenza di 2,4 Ghz. Tale dispositivo è, inoltre, dotato di una batteria ricaricabile da 6500 mAh con la durata di oltre un anno e ha un tempo di risposta pari a 200 ms con l'attivazione immediata di un altro dispositivo (se collegato).



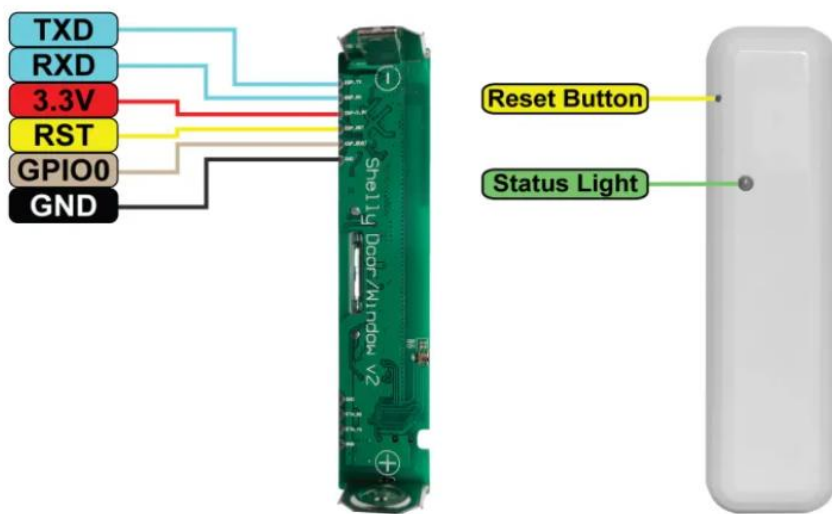
*Figura 3.3 – Parte interna Shelly Motion Sensor*

Lo Shelly Motion Sensor possiede, al suo interno, un sensore di tipo a infrarossi passivi (PIR), il quale utilizza il calore degli oggetti per rilevarne la loro presenza. Mentre esternamente presenta un LED frontale che segnala le modalità di funzionamento del sensore, come ad esempio, la luce blu lampeggiante sta a rappresentare la modalità paring, mentre la luce rossa lampeggiante rappresenta il movimento rilevato e segnalato.



*Figura 3.4 - Shelly Motion Sensor*

Lo **Shelly Door/Window2**, invece, è un dispositivo Wi-fi per porta/finestra. Basato su un'unità principale costituito da un piccolo sensore reed, e una secondaria contenente un piccolo magnete. Esternamente presenta un piccolo LED di stato ed un piccolo tasto di reset raggiungibile dall'esterno.



*Figura 3.5 – Parte interna Shelly Door/Window2*

Alimentato da due batterie da 3V formato CR123A, nonostante ciò, anche questo tipo di sensore supporta il protocollo di trasmissione Wi-Fi 802.11 b/g/n solo su frequenza 2,4 Ghz. Tutto ciò è reso possibile con l'idea di far "dormire" (ovvero abilitare una modalità a bassissimo consumo, che prevede lo spegnimento delle periferiche della CPU interna) il sensore tra una rilevazione e l'altra, quando il sensore rileva un'apertura o una chiusura si collega al wi-fi e fornisce il dato al cloud Shelly o alla rete locale via MQTT (come fatto nel caso in analisi).



*Figura 3.6- Shelly Door/Window2*

È stata integrata ad OpenHAB anche una striscia LED, la **Yeelight lightstrip plus**.

Una striscia lunga 2 metri, contenete 48 LED e 4 punti di taglio che permettono di allungare o accorciare la sua lunghezza. Possiede un pulsante di accensione e spegnimento collegato a due fili, uno che porta alla striscia led e l'altro all'alimentatore.



*Figura 3.7 -Striscia LED*

Anch'essa dotata di un AP interno che permette il collegamento della striscia alla rete locale.

### *3.2 MQTT e la sua funzione di binding*

MQTT (Message Queue Telemetry Transport) è un protocollo ISO standard di messaggistica, adatto per la trasmissione dei dati prodotti dai dispositivi IoT, ideato con l'obiettivo di far comunicare le macchine tra di loro in maniera efficiente.

Una delle caratteristiche di MQTT è quella di essere un protocollo semplice e leggero per lo scambio di messaggi, tale per cui minimizza il traffico sulla rete.

Essendo, quindi, leggero, efficace e versatile lo rendono uno degli standard più apprezzati in domotica personale, adatto a tutti quegli ambienti in cui le risorse disponibili e la larghezza di banda della rete sono limitate; adatto



anche in quei casi in cui non si ha un'ottima stabilità della connessione, e si hanno svariate interruzioni.

MQTT per lavorare in questo modo segue un modello di pubblicazione e sottoscrizione, definito come “*publish and subscribe*”, per scambiare dati tra client e server.

I server MQTT vengono definiti come broker, mentre i client sono i dispositivi connessi.

L'operazione con la quale un client invia dati al broker viene definita come *publish*, mentre, l'operazione con la quale il client riceve dati dal broker viene definita come *subscribe*.

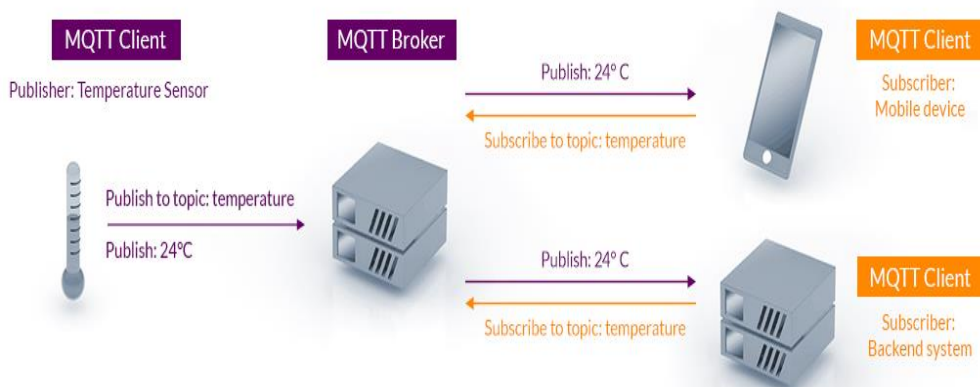


Figura 3.5 -Broker MQTT

OpenHAB non è un broker MQTT e deve connettersi a uno di essi come un normale client. Pertanto, è necessario aver prima configurato un Broker Thing tramite il MQTT Broker Binding.

Una volta configurato un broker, essendo la struttura di MQTT molto generica, consente di aggiungere un numero arbitrario di *generic MQTT*

*thing*, sulla quale, a sua volta, è possibile aggiungere un numero arbitrario di canali.

### 3.3 Risultati ottenuti nell'integrazione

Una volta integrati i sensori all'interno di OpenHAB, sono stati creati gli items, che rappresentano tutte le “cose” che possono essere utilizzate dalle applicazioni.

Sono stati create entrambe le tipologie di items: point ed equipment.

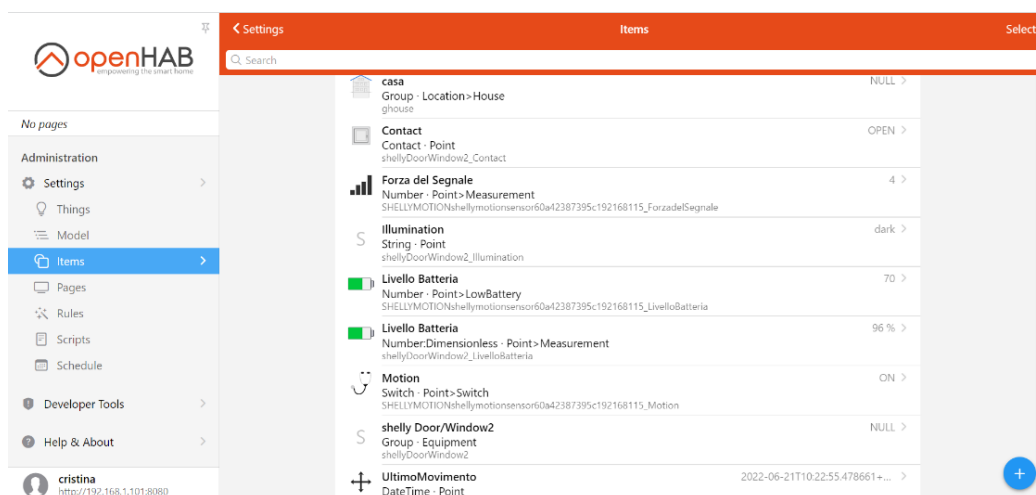


Figura 3.6 -Item

Per quanto riguarda il sensore Shelly Door/Window2 è stato creato per prima cosa l'*equipment* “shelly door/window2” e nel suo interno sono stati aggiunti i vari *point*.

Quest'ultimi permettono di monitorare il comportamento del dispositivo, come ad esempio il livello della batteria del dispositivo.

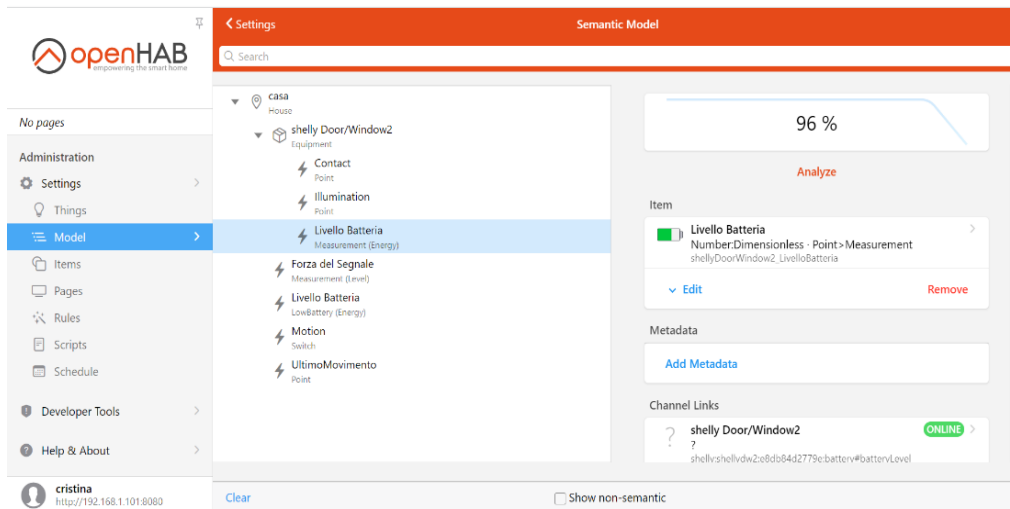


Figura 3.7 -Livello batteria

Il rilevamento dell'apertura e della chiusura di una porta o di una finestra.

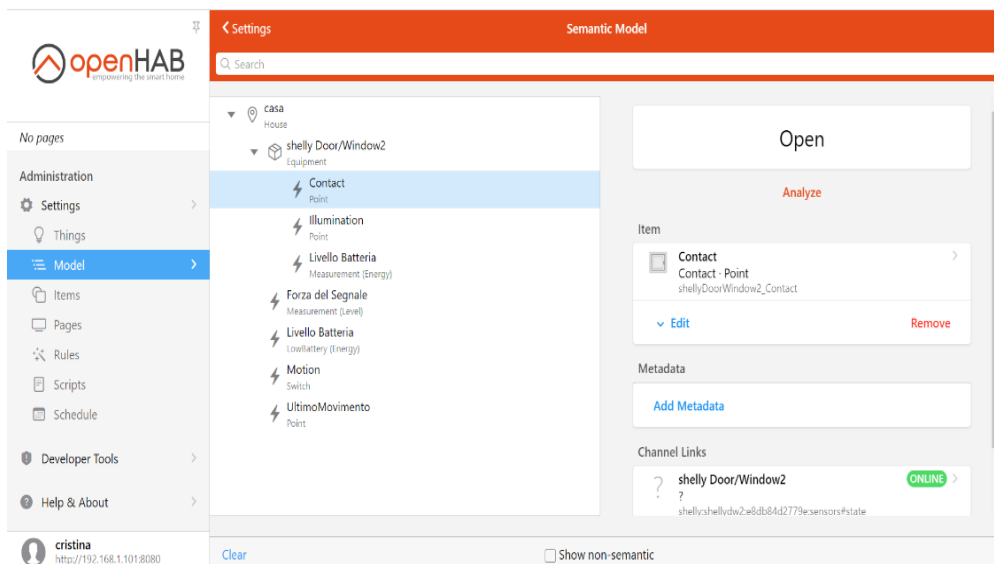


Figura 3.7 -Apertura porta

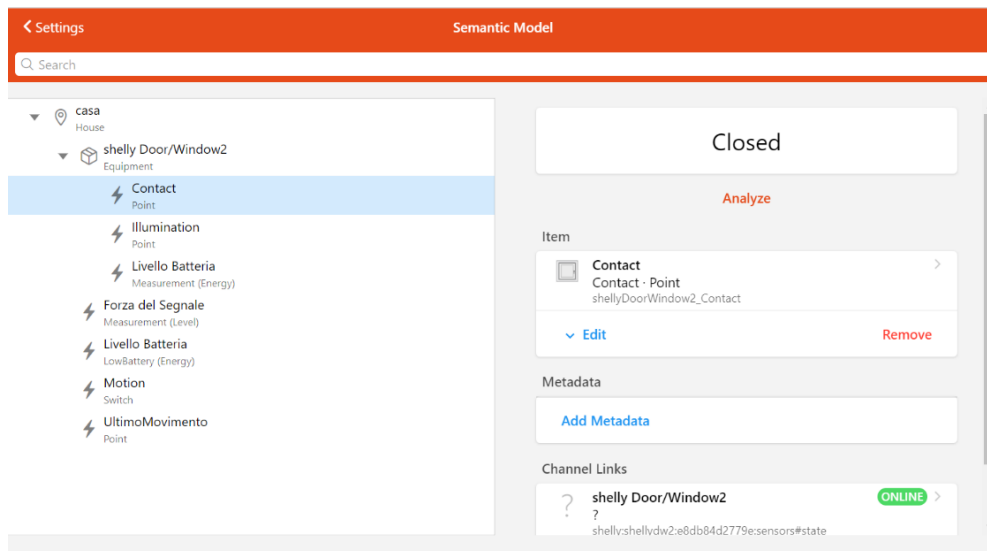


Figura 3.8 -Chiusura porta

Mentre, per lo Shelly motion sensor sono stati creati tutti *items* di tipo *point*, come ad esempio la forza del segnale;

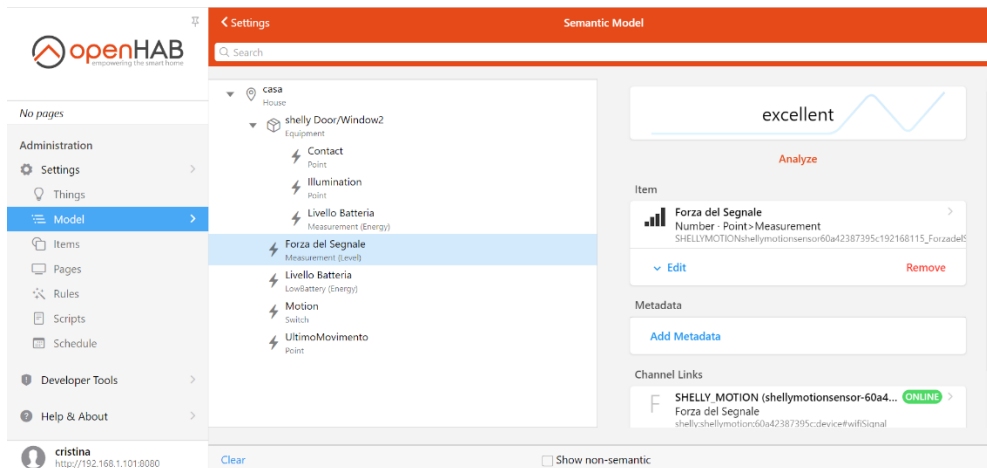


Figura 3.9 -Forza segnale

Il rilevamento dell'ultimo movimento, in cui viene segnalata la data e l'ora del movimento rilevato.

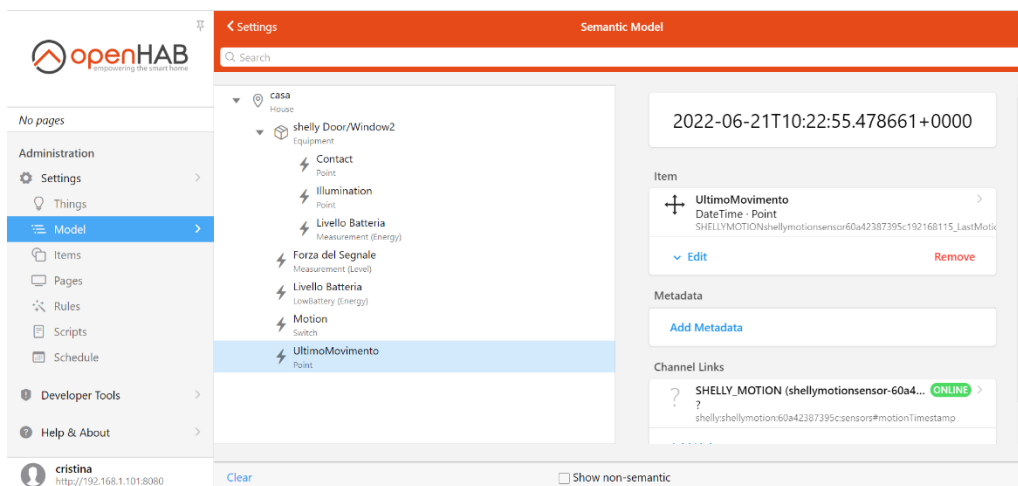


Figura 3.10 -Ultimo rilevamento

Tutto ciò permette di controllare lo stato dei vari dispositivi e capire il loro funzionamento.

## 4. Conclusioni e sviluppi futuri

### 4.1 Prestazioni ottenute

Come abbiamo già visto, con l'integrazione di questi tipi di sensori riusciamo a monitorare i valori acquisiti e le azioni a cui sono sottoposti, con successo. Viene rilevato: l'ultimo movimento, il segnale del dispositivo, l'apertura e la chiusura di una porta/finestra ecc.

Allo stesso tempo sono state riscontrate anche delle imperfezioni nell'introduzione di questi dispositivi.

Sono stati notati dei tempi di risposta lunghi: intercorrono un paio di minuti prima che OpenHAB si aggiorni con lo stato dei dispositivi.

Inoltre, per quanto riguarda i sensori dell'azienda Shelly, spesso accade che se i dispositivi non rilevano nessun tipo di movimento vanno in uno stato di "sleep".

In tal caso OpenHAB interpreta questo stato come un errore nel dispositivo.

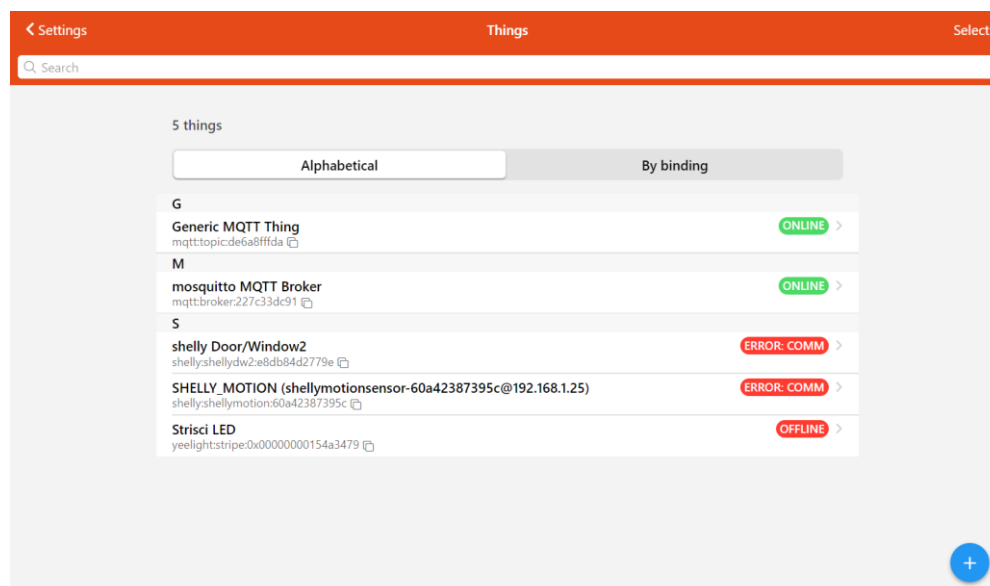
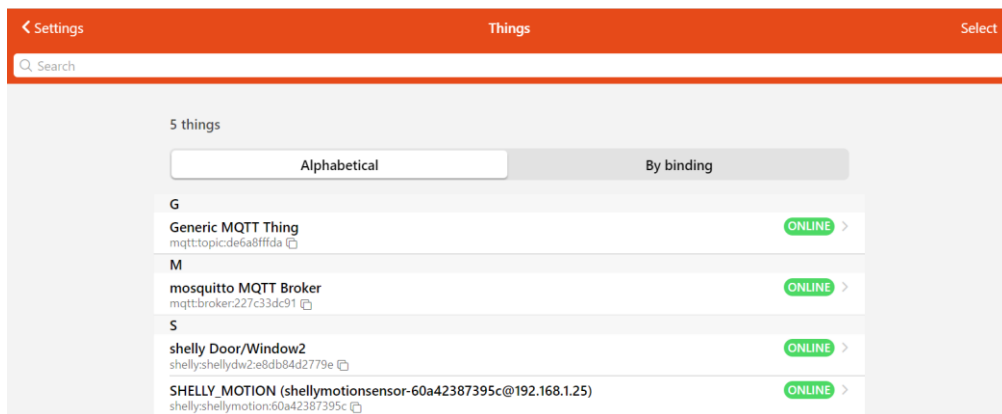


Figura 4.1 -Errori nei dispositivi

In particolare, nello shelly door/window2 solo dopo aver pigiato il pulsante all'interno del dispositivo, OpenHAB torna a visualizzare il device e il suo stato.

Per quanto riguarda lo shelly motion sensor si è notato che quando OpenHAB non viene utilizzato la thing del dispositivo mostra un errore, nonostante il sensore è comunque funzionante e rileva il movimento.

Solo dopo aver ricaricato più volte la pagina il dispositivo torna nello stato “online”.



*Figura 4.2 -Tutto funzionante*

Sono stati riscontrati problemi, invece, con la striscia LED.

Una volta averla connessa alla rete locale ed aver attivato il controllo LAN, OpenHAB ha rilevato la striscia e quindi di conseguenza integrata.

Dopo poco tempo, però, è passata dallo stato “online” allo stato “offline” nonostante la striscia continuasse a funzionare correttamente.

*Figura 4.3 -Striscia LED offline*

## *4.2 Possibilità di sviluppo*

Con le tipologie di sensori utilizzati, potrebbero essere integrate ad OpenHAB delle funzionalità che gli permetterebbero una possibile miglioria al funzionamento del programma, riguardo la possibilità di risvegliare i sensori dallo stato di sleep o quella di potere disattivare un dispositivo per preservarne la durata della batteria e di controllare del tutto tali dispositivi.

Tutto ciò, però, dovrebbe avvenire senza dover configurare nuovamente il dispositivo con la propria rete ogni volta che si vorrà disattivarli.

Ultimo ma non meno importante questo processo deve avvenire senza dover riattivare manualmente il dispositivo.

Bisognerebbe creare un sistema che, dal momento in cui vengono chiuse le porte di casa, un cancello oppure viene inserito il sistema di allarme, tali sensori vengono attivati direttamente attraverso la piattaforma, in modo da ottimizzare le prestazioni dei sensori e del software.

## *4.3 Possibilità di monitoraggio e reportistica remota*

OpenHAB offre anche la possibilità di un accesso remoto alle sue istanze locali senza dover esporre le porte ad internet e senza utilizzare una complessa configurazione VPN.

Ciò può essere fatto grazie ad OpenHAB Cloud che andrebbe a rappresentare un back-and per questo software.



Fornisce agli utenti un accesso remoto sicuro, permette di monitorare, controllare e gestire da remoto le proprie abitazioni e di raccogliere informazioni statistiche sui dispositivi integrati.

Una delle caratteristiche principali di OpenHAB Cloud è un frontend per la gestione degli utenti, accesso remoto sicuro, registro e gestione dei dispositivi e l'accesso al proxy remoto.

OpenHAB Cloud può essere installato sulla ubuntu virtual machine attraverso alcuni passi semplici e veloci:

1. Assicurarsi che l'elenco dei pacchetti e delle dipendenze da tutti i repository sia aggiornato:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. installare redis, mongoDB, Nginx e Python:

```
sudo apt-get install build-essential redis-server mongodb nginx
```

```
python
```

3. installare git e clonare il repository openHAB Cloud nella tua directory con i seguenti comandi:

```
apt-get install git
```

```
cd /home/YOUR-DIR/
```

```
git clone https://github.com/openhab/openhab-cloud.git
```

4. clona il repository openhab-cloud e dopo il checkout completato dovresti avere la directory nella cartella scelta:

```
ls -al
```

```
total 32
```

```
drwxr-xr-x 5 ubuntu ubuntu 4096 Jun  4 17:06 .
```

```
drwxr-xr-x 3 root  root  4096 Jun  4 12:34 ..
```

```
-rw-r--r-- 1 ubuntu ubuntu 220 Apr  9 2014 .bash_logout
```

```
rw-r--r-- 1 ubuntu ubuntu 3637 Apr  9 2014 .bashrc
```

```
drwx----- 2 ubuntu ubuntu 4096 Jun  4 16:30 .cache
```

```
drwxr-xr-x 13 root  root  4096 Jun  4 17:06 openhab-cloud
```

```
-rw-r--r-- 1 ubuntu ubuntu 675 Apr 9 2014 .profile
drwx----- 2 ubuntu ubuntu 4096 Jun 4 12:34 .ssh
```

5. Ora bisogna passare alla directory `openhAB-cloud` e verificare se il nodo è installato:

```
node --version
```

6. Per eseguire `openhAB Cloud` è necessario installare i bundle/stack software richiesti:

```
cd openhab-cloud
```

```
npm install
```

`OpenHAB Cloud` utilizza `redis` per la gestione delle risorse, quindi bisogna verificare se questo è stato correttamente installato, per poi andare avanti con l'installazione.

7. In questo passaggio bisogna rinominare il file di configurazione del sistema

```
config-production.json -> config.json
```

vanno regolati in base alla propria configurazione

8. Ora è possibile eseguire `OpenHAB Cloud`:

```
sudo node app.js
```

Arrivati a questo punto basta accedere con il browser web a: <http://localhost:3000>.

In questo modo `OpenHAB Cloud` è stato installato e si può procedere con l'utilizzo remoto di `OpenHAB`.

Utenti registrati, effettuare il login.

E-mail

Parola d'ordine

[Hai dimenticato la password?](#)

**Registrazione**

Se sei un nuovo utente, registrati.

Indirizzo e-mail

Parola d'ordine

openHAB UUID

openHAB Segreto

Ho letto e accettato [le Condizioni d'uso](#) e [l'Informativa sulla privacy](#).

**Registrati**

*Figura 4.3 -Interfaccia web OpenHAB Cloud*

## SITOGRAFIA

- Scordato, Ivan, “Guida completa OpenHAB: Piattaforma per domotica”, <<https://farelettronica.it/guida-completa-openhab-piattaforma-per-domotica-diy/#:~:text=Moduli%20di%20OpenHAB&text=Items%20%E2%80%93%20sono%20tutti%20gli%20%E2%80%93%20Coggetti,e%20usare%20nel%20proprio%20sistema>>
- OpenHAB, “OpenHAB, introduzione”, <<https://www.openhab.org/docs/>>
- Dalle Feste, Matteo, “OpenHAB, cos’è e come funziona”, <<https://ciaobit.com/openhab/openhab-cose-e-come-funziona/>>
- Digital guide ionos, “Un’introduzione al concetto di virtualizzazione”, <<https://www.ionos.it/digitalguide/server/configurazione/virtualizzazione/>>
- Santonnini, Pierluigi, “Virtualizzazione: cos’è, a cosa serve e i vantaggi della macchina virtuale”, <[https://sceglifornitore.it/blog/virtualizzazione-cose-a-cosa-serve-i-vantaggi/#Virtualizzazione\\_cose\\_e\\_quali\\_vantaggi\\_offre](https://sceglifornitore.it/blog/virtualizzazione-cose-a-cosa-serve-i-vantaggi/#Virtualizzazione_cose_e_quali_vantaggi_offre)>
- Manuali+ , “D’uso del Shelly Motion Detector”, <<https://it.manuals.plus/shelly/motion-detector-manual#axzz7Wpkc087z>,>
- Lo staff inDomus, “Shelly Door/Window 2 – sensore di apertura varchi Wi-Fi”, <<https://indomus.it/componenti/shelly-door-window-2-sensore-di-apertura-varchi-wi-fi/>>
- Torchiani, Gianluigi, “MQTT: cos’è il protocollo e come riesce a far comunicare i dispositivi IoT ”, <<https://www.internet4things.it/iot-library/mqtt-cose-e-come-funziona-il-protocollo-alla-base-delliot/>>
- OpenHAB, “Rilegatura MQTT”, <<https://www.openhab.org/addons/bindings/mqtt/>>

- OpenHAB, “Connettore cloud OpenHAB”,  
<<https://www.openhab.org/addons/integrations/openhabcloud/>>