Instalación y configuración de OpenHab en un Raspberry PI 3

Contents

[Introducción 3](#_Toc487027113)

[Instalación de OpenHABian 3](#_Toc487027114)

[Descargar la última versión de la imagen de OpenHabian 3](#_Toc487027115)

[Formatear la SD con SDFormatter 3](#_Toc487027116)

[Grabar la imagen en una SD de 8Gb 4](#_Toc487027117)

[Configurar Wifi 4](#_Toc487027118)

[Todo listo 5](#_Toc487027119)

[Instalación de adicionales 6](#_Toc487027120)

[Servidor MQTT (Eclipse Mosquitto) 6](#_Toc487027121)

[Instalación 6](#_Toc487027122)

[Configuración 6](#_Toc487027123)

[Testing 6](#_Toc487027124)

[Referencias 6](#_Toc487027125)

[Configuración de OpenHab 7](#_Toc487027126)

[Primer acceso 7](#_Toc487027127)

[Habilitar los add-ons 1.x 7](#_Toc487027128)

[Instalar MAP Transform 7](#_Toc487027129)

[Instalar MQTT Bind 7](#_Toc487027130)

[Instalar el Bind 7](#_Toc487027131)

[Configurar el Bind MQTT 8](#_Toc487027132)

[Configurar un MAP 8](#_Toc487027133)

[Agregar ítems MQTT 8](#_Toc487027134)

[Referencias 9](#_Toc487027135)

[Instalar MQTT Action 9](#_Toc487027136)

[Instalar Notify my Android Action 9](#_Toc487027137)

[Instalar HABmin 9](#_Toc487027138)

[Automacion por relgas 9](#_Toc487027139)

[Introducción 9](#_Toc487027140)

[Crear una regla 9](#_Toc487027141)

[Ejemplo de regla 10](#_Toc487027142)

[Referencias 10](#_Toc487027143)

[Consola OpenHab 11](#_Toc487027144)

# Introducción

El motivo inicial de este proyecto fue poder construir la automatización de un hogar. Luego de investigar las opciones se decidió utilizar un sistema abierto (open source) que pudiera integrar la automatización deseada así como los artefactos inteligentes que puedan existir en la misma locación.

Con esto en mente se decidió utilizar **OpenHab** corriendo en un **Raspberry Pi** con un servidor **MQTT** y dispositivos **ESP** necesarios, donde el **Raspberry** brinda la infraestructura necesaria, **OpenHab** funciona como unidad (sistema) de integración y los módulos ESP controlan los artefactos que se deseen automatizar, monitorear o controlar remotamente.

Como SO del **Raspberry** **Pi** se va a utilizar **OpenHABian** que tiene las siguientes bondades frente a otras distro:

* Ya cuenta con **OpenHab** preinstalado
* Tiene una buena tool de personalización con la que se pueden instalar varios adicionales tales como:
  + Mosquitto (broker MQTT)
  + Grafana
  + Log Viewer
* Viene con Samba pre instalado
* Tiene una opción para hacer backups en USB

La única contra es que no tiene interfaz gráfica preinstalada si es que queremos acceder por esa vía al sistema.

# Instalación de OpenHABian

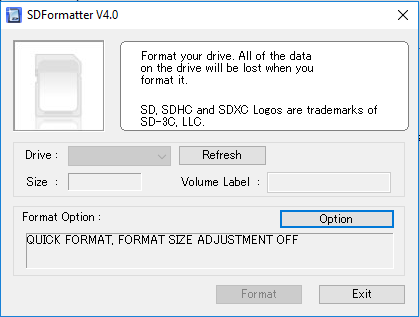
## Descargar la última versión de la imagen de OpenHabian

<https://github.com/openhab/openhabian/releases>

Si el archivo tiene extensión XZ, descomprimirlo con **7Zip** (<http://www.7-zip.org/>)

## Formatear la SD con SDFormatter

<https://www.sdcard.org/downloads/formatter_4/>

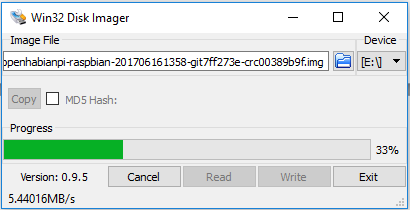


## Grabar la imagen en una SD de 8Gb

Es recomendable este tamaño porque con SD de mayor capacidad tuve problemas de compatibilidad.

Usar **Win32DiskImager**

<https://sourceforge.net/projects/win32diskimager/>



## Configurar Wifi

Explorar la SD y editar el archivo **openhabian.conf** configurando el **wifi\_ssid** y el **wifi\_psk**

## Todo listo

Colocar la SD en el Raspberry y encenderlo.

El primer boot demora varios minutos (dependiendo de la velocidad de conexión entre 5 y 60).

# Instalación de adicionales

## Servidor MQTT (Eclipse Mosquitto)

### Instalación

1. Ingresar sudo openhabian-config
2. Seleccionar la opción **Optional Components**.
3. Seleccionar **Mosquitto**
4. Configurar la password de **Mosquitto** (opcional). Si se cancela este paso, **Mosquitto** opera sin seguridad lo cual facilita la configuración del add-on que se instale en **OpenHab**. Se puede configurar una password más adelante.

### Configuración

Para modificar la configuración, editar **mosquitto.conf**: sudo nano /etc/mosquitto/mosquitto.conf

No es necesario modificar nada para que el servidor funcione.

Por default, mosquitto escucha en: HOST\_IP:1883 (host\_ip es la IP que se asignó al raspeberry)

### Testing

Para probarlo, en dos consolas diferentes escribir estos comandos

1. mosquitto\_sub -v -t 'test/topic'
2. mosquitto\_pub -t 'test/topic' -m 'helloWorld'

En la consola (A) debería aparecer el mensaje 'helloWord'

### Referencias

<https://mosquitto.org/man/mosquitto-conf-5.html>

# Configuración de OpenHab

## Primer acceso

Una vez finalizada la instalación de **OpenHABian** y adicionales acceder a **OpenHab** a través de la url.

<http://openhabianpi:8080>

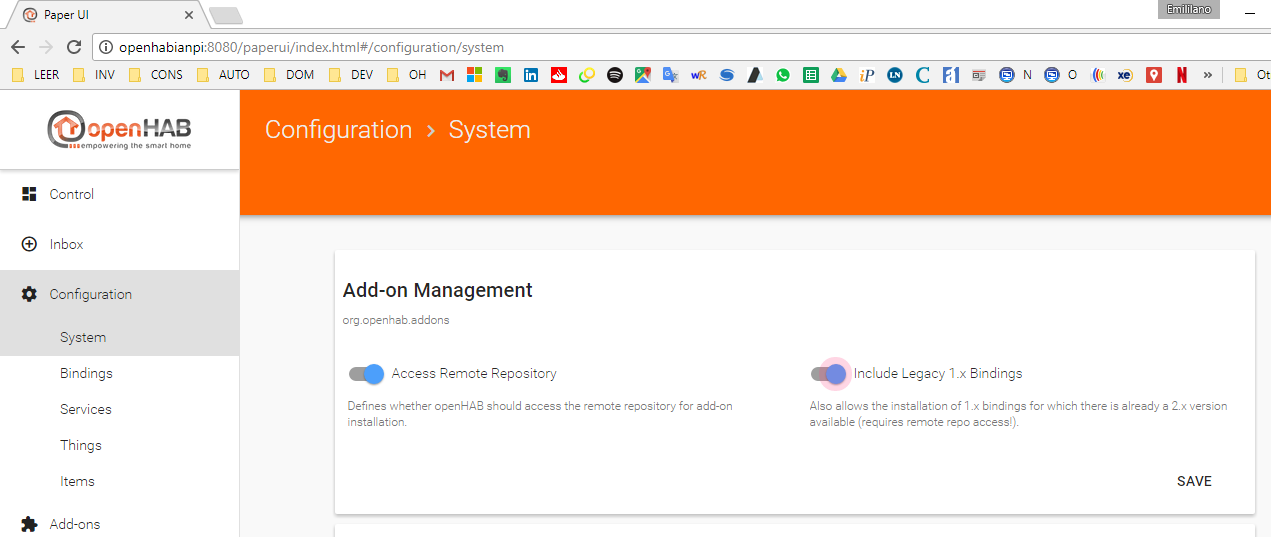
Una vez ahí seleccionar **Standard Setup**. Esto solo se realiza en el primer acceso al sistema y es donde se instalan algunas de las interfaces de acceso.

Luego a medida que se vaya tomando conocimiento del sistema, se pueden instalar mediante otros medios, diferentes interfaces.

## Habilitar los add-ons 1.x

Algunos add-ons deseables (como el de MQTT) fueron desarrollados para la versión 1.x de **OpenHab**, por lo tanto es necesario activar esta propiedad para poder instalarlos.

Desde Paper UI ([http://openhabianpi:8080/paperui/index.html#/configuration/system](http://openhabianpi:8080/paperui/index.html" \l "/configuration/system)) habilitar los add-ons 1.x



## Instalar MAP Transform

Instalar desde [http://openhabianpi:8080/paperui/index.html#/extensions](http://openhabianpi:8080/paperui/index.html" \l "/extensions)

## Instalar MQTT Bind

Para que este add-on pueda instalarse, es necesario el paso de habilitar los add-ons 1.x

### Instalar el Bind

Instalarlo desde [http://openhabianpi:8080/paperui/index.html#/extensions](http://openhabianpi:8080/paperui/index.html" \l "/extensions)

### Configurar el Bind MQTT

Es necesario configurar el broker MQTT al cual se va a apuntar. Pueden configurarse algunas cosas adicionales como:

* Calidad de servicio
* Seguridad
* Id de cliente, etc.

Para modificar la configuración:

1. Ingresar nano /etc/openhab2/services/mqtt.conf
2. Descomentar y completar el <broker>.url: mosquitto.url=tcp://localhost:1883  
   Ingresamos localhost porque el server mosquitto está corriendo en el mismo host que **OpenHab** y **1883** como puerto porque es el default. Si se hubiera configurado otro puerto en la configuración de MQTT ese mismo debería ir acá también.
3. Descomentar y completar el <broker>.clientId: mosquitto.clientId=openhab.mqtt (opcional)

### Configurar un MAP

Este map va a servir para traducir los estados entre los **ESP** y **OpenHab** vía **MQTT**

1. Ingresar touch /etc/openhab2/transform/onoff.map
2. Editar onoff.map agregando las siguientes dos líneas  
   0=OFF  
   1=ON

### Agregar ítems MQTT

1. Ingresar touch /etc/openhab2/items/mqtt.items
2. Editar mqtt.items  
   Switch ESP01 {mqtt="<[mosquitto:esp/125603/state:state:MAP(onoff.map)],>[mosquitto:esp/125603/command:command:OFF:0],>[mosquitto:esp/125603/command:command:ON:1]", autoupdate="false"}

**Explicación:**

* **Switch** indica que el tipo de ítem configurado es justamente del tipo interruptor.
* **ESP01** indica el nombre del ítem
* **Mqtt** indica en inicio de la configuración de mensajes del protocolo. En el ejemplo hay tres mensajes configurados, dos salientes (inician con **>**) y un entrante (inicia con **<**).
  + **Saliente:** Son dos mensajes:
    - Uno indica que cuando desde alguna interfaz se haga clic en el ítem y el estado del mismo pase a **OFF**, se publique en el tópico **esp/125603/command** un **0** (cero).
    - El otro indica lo mismo para el caso de que el estado pase a **ON**, se deberá enviar un **1**.
  + **Entrante:** El mensaje entrante dice que cuando se reciba un mensaje en el tópico **esp/125603/state** se use el servicio de transformación con la transformación definida en **onoff.map**. En esa transformación se traduce 0 > OFF y 1 > ON, y el estado obtenido de dicha transformación se seteará en el ítem (switch)

### Referencias

Se puede ver la documentación del add-on en

<https://github.com/openhab/openhab1-addons/wiki/mqtt-binding>

## Instalar MQTT Action

Instalar desde **Paper UI > Add-ons > Actions > MQTT Action**

## Instalar Notify my Android Action

Instalar desde **Paper UI > Add-ons > Actions > Notify my Android Action**

Para obtener la key loguearse en <https://www.notifymyandroid.com/account.jsp> y acceder a la solapa **Manage API Keys**.

## Instalar HABmin

Instalar desde **Paper UI > Add-ons > User Interfaces > HABmin**

## Automacion por reglas

### Introducción

Se puede desarrollar automatización usando OpenHab a través de la creación de reglas.

Se pueden crear reglas desde la interfaz HABmin o escribiendolas manualmente.

Si bien desde HABmin parece más sencillo, la funcionalidad no es muy robusta (al menos hasta la version 2.1 de OpenHab) por tal motive es recommendable hacerlo manualmente aunque requiera manejo de sintaxis.

### Crear una regla

Para crear una regla manualmente:

1. Crear un archivo <nombre\_regla>.rules en /etc/openhab2/rules  
    nano /etc/openhab2/rules/myRule.rules
2. Editar el archivo y escribir en el la regla deseada

### Ejemplo de regla

rule "riego"

when

Time cron "0 0 21 \* \* ?"

then

notifyMyAndroid("5bc56db507eabfe3fb26b4e97ebddbcc75eaa46d370127e2", "Riego", $

publish("mosquitto","irrigation/valve01/cmd","1");

end

### Referencias

**Documentación**

<https://github.com/openhab/openhab1-addons/wiki/rules>

**Ejemplos**

<https://github.com/openhab/openhab1-addons/wiki/Samples-Rules>

# Consola OpenHab

1. Ingresar ssh -p 8101 openhab@localhost (puede dar timeout, volver a intentar)
2. Ingresar “yes” para aceptar el certificado
3. Password: habopen