

Práctica con *Home Assistant*

Autor: Alberto López del Amo Gorgojo

Profesor de prácticas MAES

albertolag@correo.ugr.es

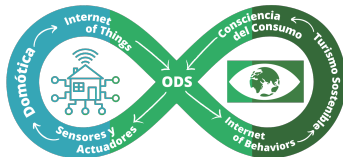


Table of Contents

- 1 *Home Assistant*
- 2 *ESP32 y ESPHome*
- 3 Sensores y Actuadores
- 4 Automatizaciones
- 5 *Node-RED y Telegram*
- 6 Procedimiento a seguir
- 7 Conclusión



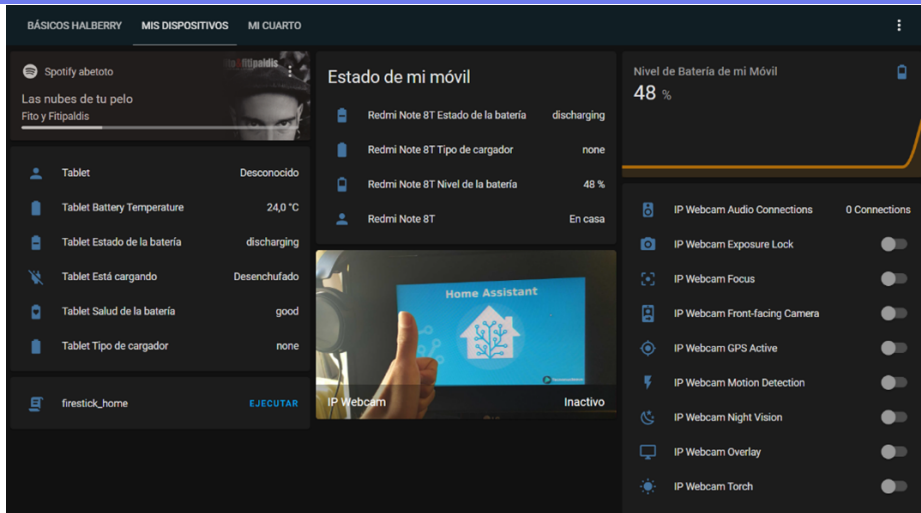
Home Assistant



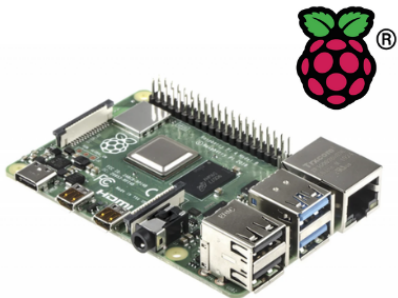
- Software gratuito y de código abierto.
- Se utiliza para la automatización del hogar.
- La instalación se hace centralizada mediante un único *hub* a través de WiFi.



Ejemplo de interfaz en *Home Assistant*



Raspberry Pi 4



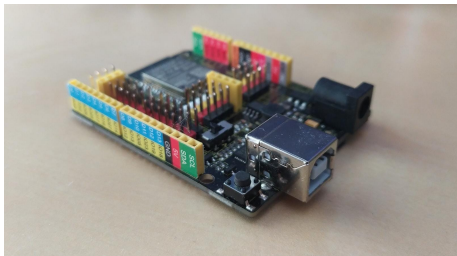
- Para centralizar la instalación domótica, se va a utilizar una **Raspberry Pi 4** como *hub*. Estas placas se denominan *single-board computer*: “Ordenadores de una sola placa”. Tienen gran potencia e infinidad de aplicaciones.



Placa ESP32

- Funcionalidad similar al Arduino UNO, con compatibilidad con la mayoría de las funciones.
- Es más potente y versátil con conectividad WiFi y Bluetooth.
- Para más de sus características:

Características ESP32 STEAMaker



ESPHome

- Es un sistema de control para ESP8266 y **ESP32** que, con archivos de **configuración sencillos**, pueden ser usados para crear **automatizaciones** con programas como *Home Assistant*.
- Tiene preparadas las configuraciones de la **mayoría de sensores y actuadores** que se pueden utilizar.
- Es perfecto para **domotizar una vivienda** si se tienen ciertos conocimientos de **domótica y programación**.











ESPHome

```
# Flame sensor
binary_sensor:
  - platform: gpio
    pin:
      number: GPIO14
      inverted: true
    name: Flame
```










Sensores a utilizar en la práctica

Sensor	Tipo	Características	Imagen
DHT11	Temperatura Humedad	Pines GND, V, señal digital. Medidas relativas.	
LDR	Luz ADC*	Pines GND, V, señal analógica. Se debe escalar la salida.	
Botón capacitivo	Binario (0 1)	Pines GND, V, señal digital. Interruptor muy sensible.	
PIR	Presencia IR	Pines GND, V, señal digital. Detecta <80° horiz. y <55° vertical.	

Flame	Binario (0 1)	Pines GND, V, señal digital. Detecta la llama hasta a 1 metro.	
Vapor	Binario (0 1)	Pines GND, V, señal digital. Detecta el vapor de agua.	
Foto interruptor	Binario (0 1)	Pines GND, V, señal digital. Detecta cuando la señal IR se interrumpe.	
Vibración	Binario (0 1)	Pines GND, V, señal digital. Detecta vibración	

Actuadores a utilizar en la práctica

Actuador	Tipo	Características	Imagen
LED	Luz LED	Pines GND, V, señal digital.	
LED RGB	Luz LED modificable.	Pines V, Red, Blue, Green. Se programan como 3 separados.	
Relé	Interruptor "Switch"	Entradas: Señal digital, GND, V Salida conmutada.	
Ventilador	Motor	Pines V, GND y null. Requiere >4,5V.	

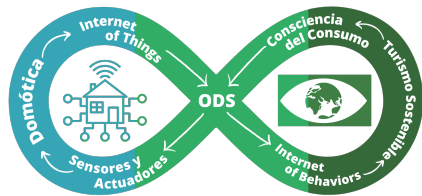
Servomotor	Servo	Pines V, GND y señal digital. Requiere >4,5V.	
Motor	Motor	Pines V y GND. Requiere >4,5V.	
Buzzer activo	Buzzer	Pines V, GND y señal digital.	



Automatizaciones

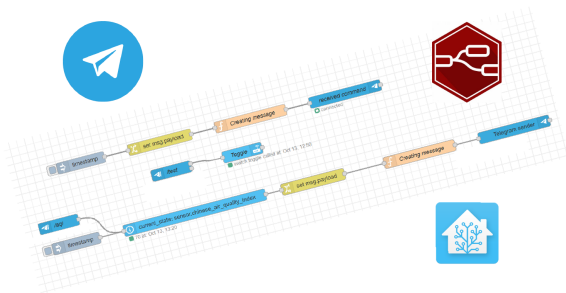
¿Qué podemos hacer?

- Deberéis crear, al menos, una automatización coherente con el equipo de trabajo, premiando a las que ayuden a **reducir el consumo**.
- Podréis juntaros con **otros grupos** para obtener diversas combinaciones.



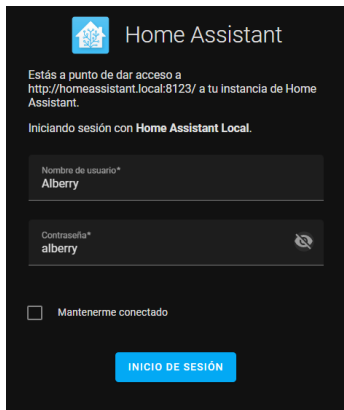
Node-RED y Telegram

¿Quién se atreve?



- Para realizar una automatización con **Telegram**, se necesita:
 - Instalación y configuración de **Node-RED**.
 - Bot de **Telegram**: **chat ID -704553654**.

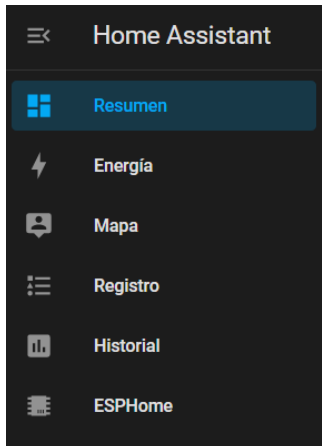
Inicio sesión



- Nos conectamos al WiFi preparado: **"Albertolag": "holacaracola"**
- Tenemos que introducir en el navegador el siguiente enlace para conectarnos a la *Raspberry Pi*:
<http://homeassistant.local:8123/>
- Ponemos el usuario y contraseña: "Alberry" → "alberry" como se observa en la imagen.



Menú de *Home Assistant*



- Al iniciar sesión, entramos remotamente en la *Raspberry Pi* con *Home Assistant* previamente instalado.
- En el menú situado a la izquierda encontramos las diferentes aplicaciones y opciones con las que trabajar.
- Aquí trabajaremos con las pestañas de **Resumen**, **ESPHome**, **Node-RED** y **Ajustes**.



Identificación de la placa

Pestaña de *ESPHome*

The screenshot displays the ESPHome web interface. At the top, there's a header with the ESPHome logo and a house icon. To the right of the header are two buttons: "UPDATE ALL" and "SECRETS". Below the header, there are six device cards arranged in a 2x3 grid. Each card represents a configured device and includes the following information:

- Device Name:** BlackWidow, Ironman, Profesor, ScarlettWitch, Superman, and Thor.
- Status:** ONLINE (indicated in the top right corner of each card).
- Configuration File:** blackwidow.yaml, ironman.yaml, profesor.yaml, scarlettwitch.yaml, superman.yaml, and thor.yaml (displayed in a light blue box).
- Actions:** EDIT and LOGS buttons are located at the bottom left of each card.
- More Options:** A vertical ellipsis (three dots) is located at the bottom right of each card, indicating additional actions.

Configuración del .yaml

¡Programamos los dispositivos!

```
30 captive_portal:
31
32 #Flame
33 binary_sensor:
34   - platform: gpio
35     pin:
36       number: GPIO14
37       inverted: true
38     name: Flame
```

- En la web de ESPHome, buscamos el sensor/actuador que queremos programar y adaptamos el código:
<https://esphome.io/>
- Editamos el *yaml* correspondiente a la placa asignada, debajo de *Captive Portal*.



Instalación por WiFi

How do you want to install profesor.yaml on your device?

Wirelessly

Requires the device to be online



Plug into this computer

For devices connected via USB to this computer



Plug into the computer running ESPHome Dashboard

For devices connected via USB to the server



Manual download

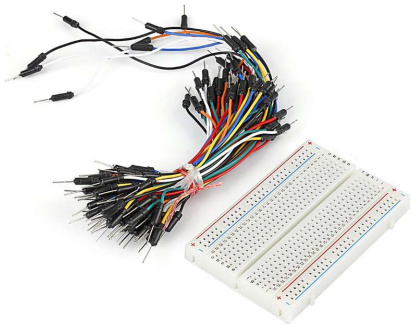
Install it yourself using ESPHome Web or other tools



CANCEL

- Arriba a la derecha, guardamos y le damos a instalar.
- De las diferentes opciones, ya que se ha hecho una configuración previa, elegimos la opción **Wirelessly** y esperamos a que realice la instalación.
- Si falla, se debe volver a intentar. Si aún así no se consigue, avisamos al profesor.
- Una vez hecho esto, se deberá crear una etiqueta en la pestaña de resumen para su uso.

Conexión de los dispositivos a la *ESP32*




- Se deberá conectar el dispositivo a la *ESP32* según lo programado con el uso de cables y una *protoboard*.
- Es muy importante saber a qué voltaje trabaja el sensor/actuador y en qué pin lo hemos definido.
- ¿Necesita algún requisito especial?



Automatizaciones

Acciones

^  **Notificaciones: Send a notification with homeassistantgroup_bot**

Servicio
Notificaciones: Send a notification with homeassistantgroup_bot

Sends a notification message using the homeassistantgroup_bot service.

Message
Message body of the notification.

☒ **Title**
Title for your notification.

Target
☐ An array of targets to send the notification to. Optional depending on the platform.

- En la pestaña de ajustes, hacemos *click* en automatizaciones.
- Aquí deberemos crear una automatización que se compondrá de:
 - Desencadenantes.
 - Condiciones.
 - Acciones



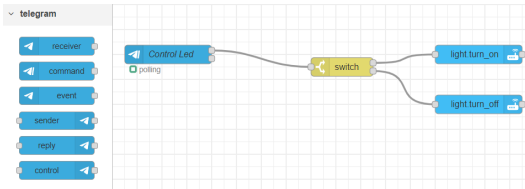
Telegram

Entramos en el grupo creado con el *bot*

`https://t.me/+pRTyOM3ZIwI0NwY8`



Node-RED



- Una vez que estamos dentro del grupo de *Telegram* y sabemos que el **chat ID: -704553654**, nos vamos a la pestaña de *Node-RED* e intentamos crear un nuevo comando que controle el dispositivo programado desde *Telegram*.
- Una opción interesante es que el comando "llame" a la automatización previamente configurada.



Cuestionario final

<https://forms.gle/amZXgiGHYTZVuV1z8>



Para finalizar, un poco de debate.

