

# OPEN SOURCE SMART HOME SYSTEM

(Arduino - Raspberry Pi - OpenHAB)

**José Fernando Adrán Otero**  
Grado de Tecnologías de las Telecomunicaciones  
Arduino

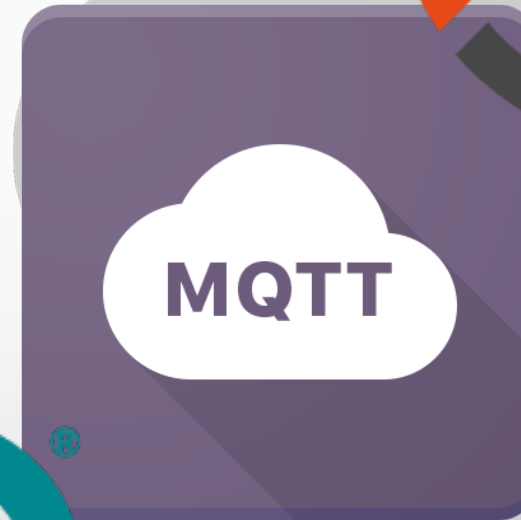
**Antoni Morell Pérez**  
**Pere Tuset Peiró**

10/06/2018



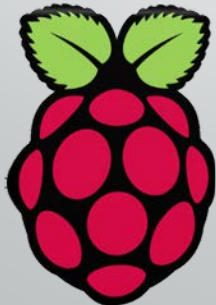
# INDICE

- Introducción
- Objetivos
- Estructura del sistema
  - Ejemplo Básico
  - Arquitectura
- Elementos del sistema
  - Raspberry Pi
  - MQTT
  - OpenHAB
  - Arduino
  - Sensores and actuadores
- Implementación
  - Arduino
  - OpenHAB
- Video demostrativo
- Conclusiones



# Introducción

- Los sistemas domóticos actuales son caros y complejos, lo cual no facilita el acceso a los mismos a la mayoría de la población. Con el uso de sistemas Open Source se puede crear un Sistema domótico comparable con los sistemas comerciales, pero con un coste mucho menor.
- Para la realización de este proyecto se usará únicamente Hardware y Software Open Source:
  - Raspberry Pi, para el nodo principal
  - Arduino Mega, para los nodos remotos (1 por estancia)
  - Sensores y actuadores compatibles con Arduino y/o Openhab
  - Protocolo MQTT, para la comunicación entre nodos (M2M: Machine to Machine)
  - OpenHAB, como interfaz de usuario, el cual proporciona un entorno visual moderno y amigable
  - InfluxdB y Grafana, como BBDD y sistema de representación de datos

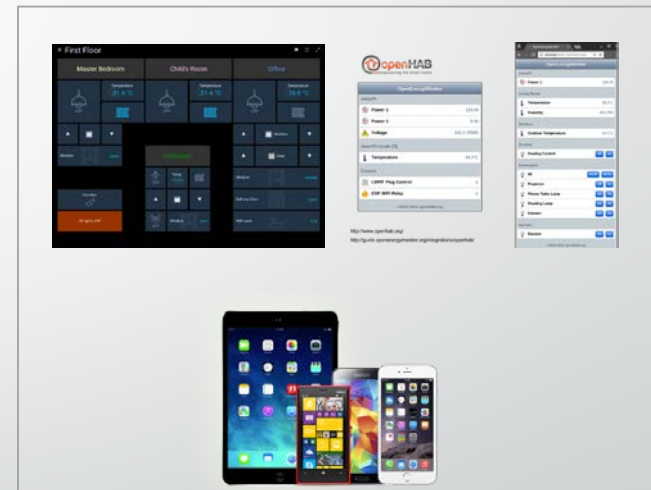
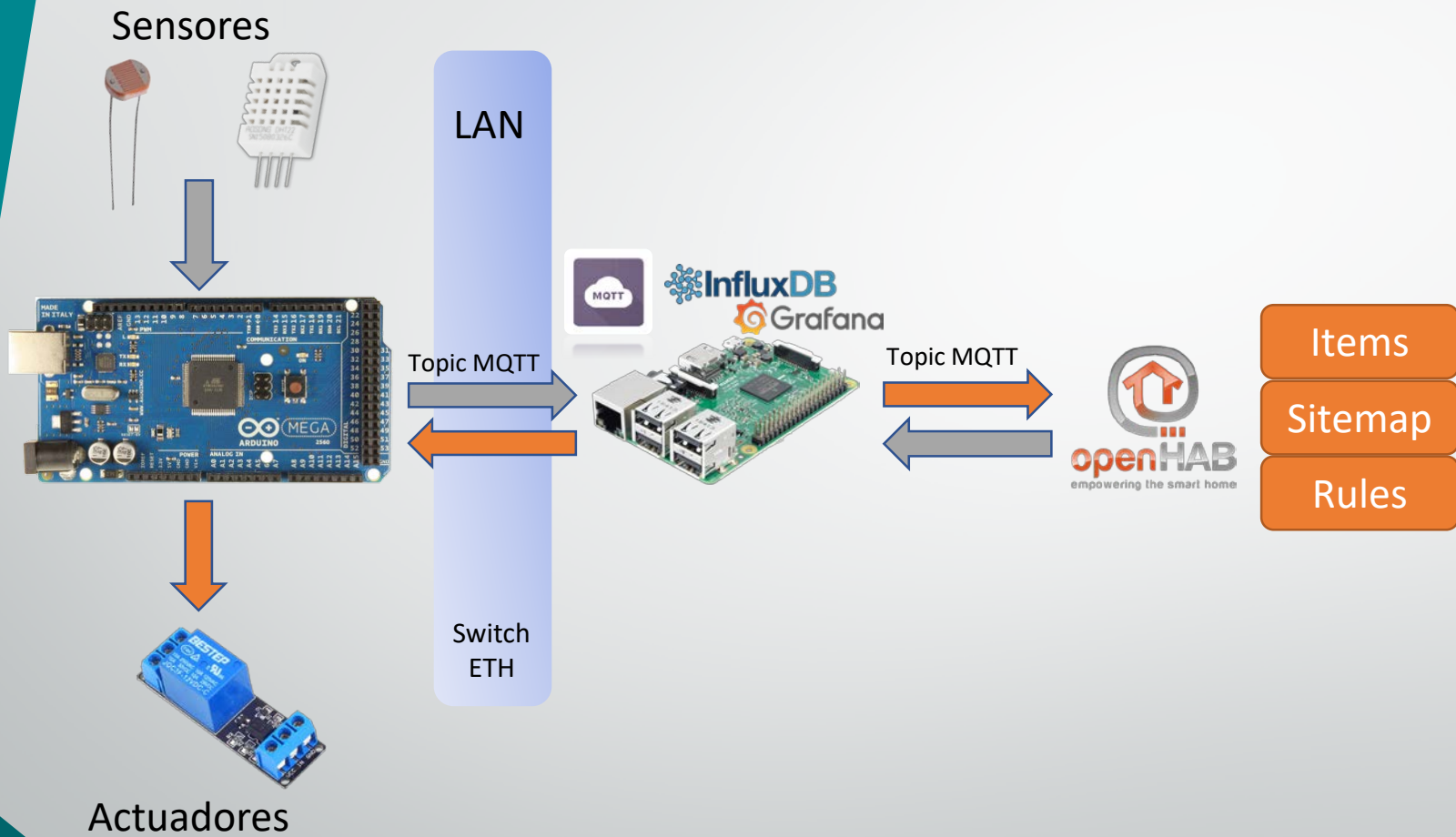


# Objetivos

- Conectividad entre OpenHab y Arduino por medio de protocolo MQTT
  - Se instalará un bróker MQTT (Message Queue Telemetry Transport)
  - Dispondrá del software OpenHAB y de BBDD para interactuar con los usuarios
- Competir con un sistema comercial en términos de calidad, características e interfaz visual.
- Uso de sensores y actuadores para medir diferentes estados del hogar, así como ser usados para mostrar alarmas o notificaciones
- Activación de calefacción automáticamente



# Estructura del sistema - Arquitectura

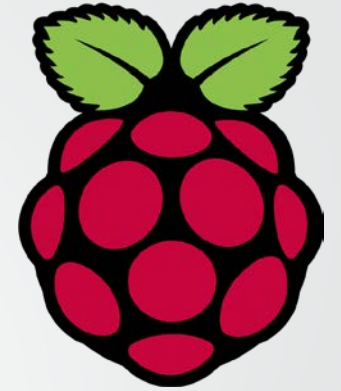


# ELEMENTOS DEL SISTEMA



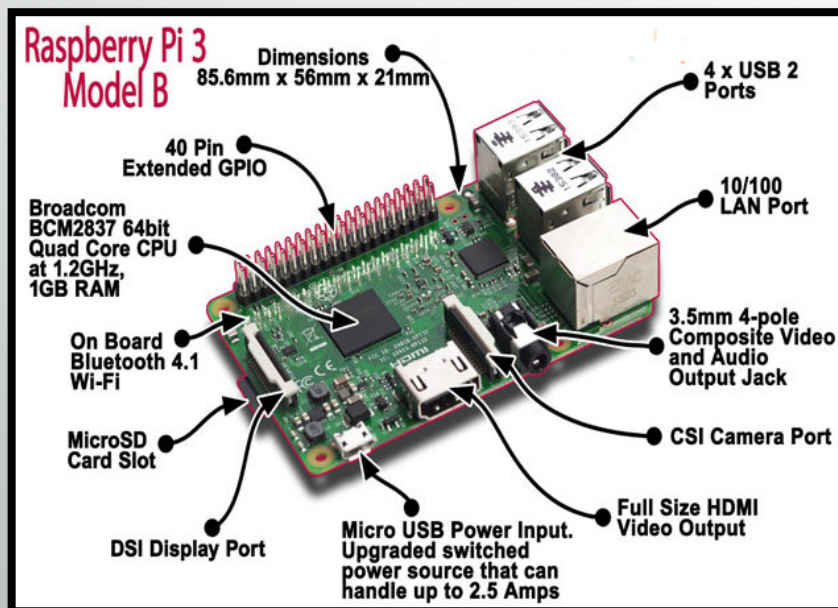


# Raspberry Pi



- Raspberry Pi será el núcleo del Sistema domótico, sobre el cual se instalarán los diferentes servicios de control (Bróker MQTT, OpenHab, Grafana, InfluxBD...)
- Se requiere una alimentación de al menos 2.5 A para su correcto funcionamiento

## Hardware Layout



### CPU, GPU, RAM

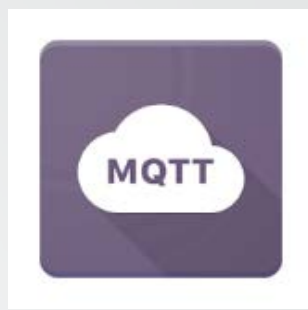
- Chipset Broadcom BCM2387
- 1,2 GHz de Quad Core ARM Cortex-A53
- GPU: Dual Core Video Core IVRAM
- RAM: 1GB LPDDR2

### Conectividad

- Ethernet socket 10/100 BaseT
- 802.11 b / g / n WLAN & Bluetooth 4.1 (Classic Bluetooth and LE)
- HDMI rev 1.3 y 1.4
- 3,5 mm Jack and HDMI
- USB 4 x Conector USB 2.0

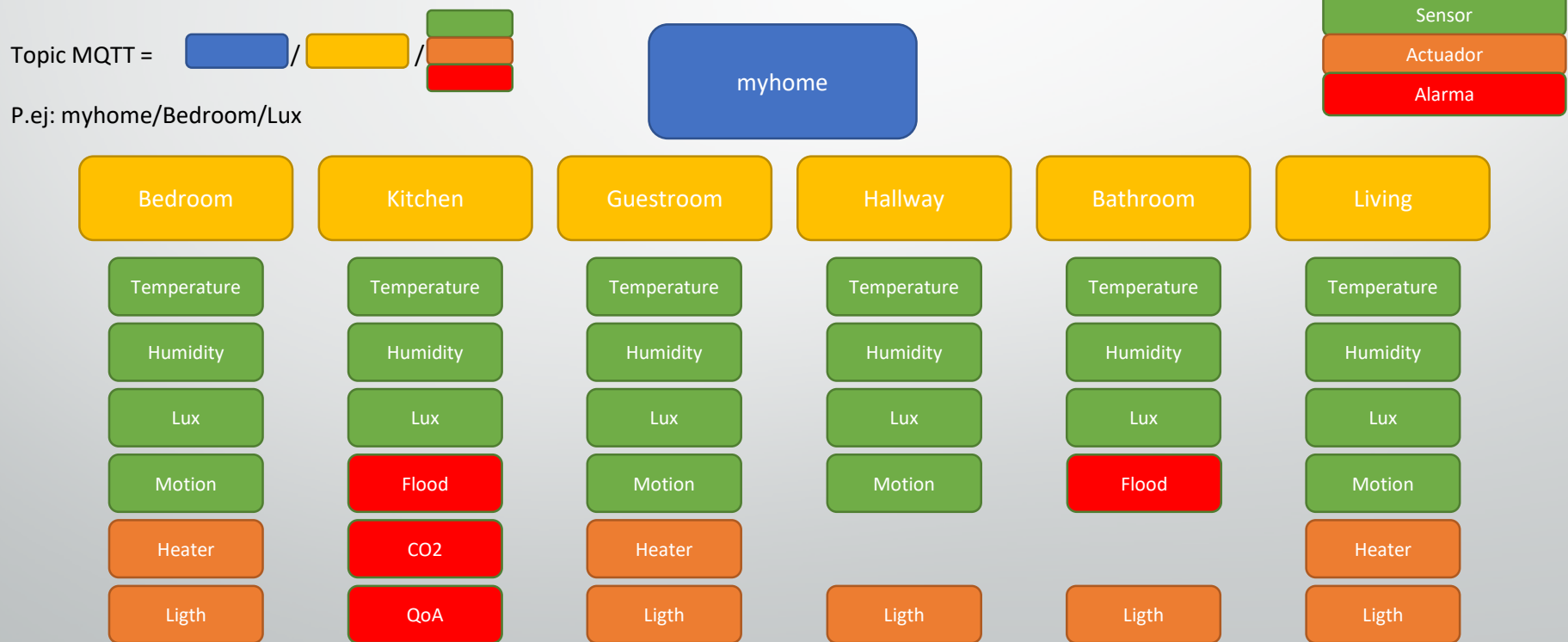


# MQTT



MQTT es un protocolo machine-to-machine (M2M), adecuado para los servicios del Internet de las cosas (IoT - Internet of Things). Está diseñado como un publicador y suscriptor de mensajes muy ligero

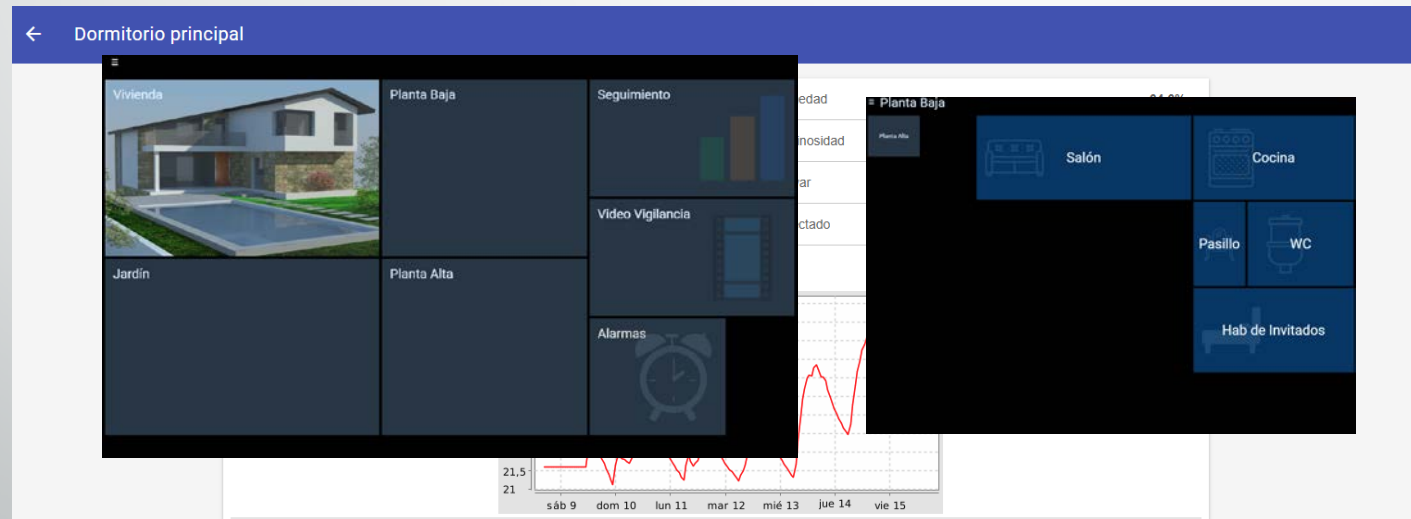
La comunicación está basada en "topics", sobre los cuales el cliente emisor publica el mensaje y los clientes receptores que deseen recibirlo deben suscribirse a él





# OpenHAB

- Software libre
- Funciona sobre Windows, OSX, UNIX, IOS o Android, o por interfaz web.
- Arquitectura modular, sistema escalable: es muy sencillo añadir nuevos elementos
- Dispone de multitud de Bindings (Interfaces), que aportan flexibilidad y conectividad con diferentes sistemas y protocolos libres o propietarios



OpenHAB proporciona un interfaz moderno y amigable, apto para competir con diferentes sistemas propietarios

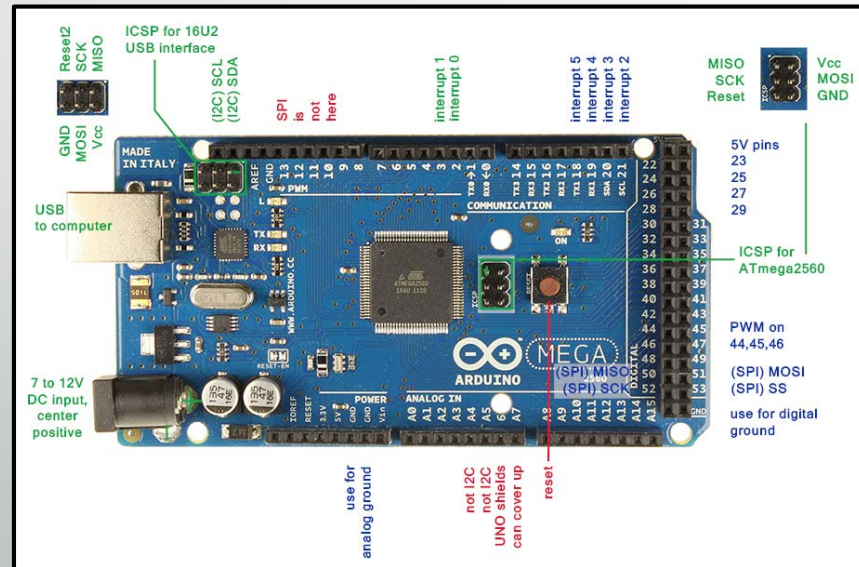


# Arduino

El Sistema usará Arduino Mega como controladores remotos  
Los Arduino ejecutarán las ordenes recibidas por OpenHab, así como las suyas propias programadas en su Sketch

Necesita una Fuente de alimentación de 5V 2.5A como mínimo, sobre todo en el caso de utilizar su propia Fuente para la activación de relés

## Hardware Layout



Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g



# Sensores y actuadores

## SENSORES



LDR



DHT22



Sensor de Inundación



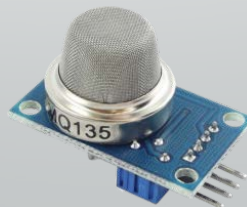
PIR



Sensor de llama



Sensor de Gas



M  
Q  
T  
T  
-  
A  
R  
D  
U  
I  
N  
O  
M  
E  
G  
A



LED



Buzzer



Relé



## ACTUADORES



# IMPLEMENTACIÓN



# Código Arduino

## INICIO

Librerías

Variables

Inicialización de Red

## FUNCIONES

Callback / Reconnect

Lectura de sensores

MQTTPublish

## Setup()

Inician valores de conexión y estado de los pines (IN/OUT)

## Loop()

Se lanzan las funciones definidas en el programa

readDHT22()

Luz()

getVPP()

Pir()

Leer\_RFID()

printHex()

printDec()

Alarmas()

Agua()



# OpenHAB



## ITEMS

```
Switch GF_GuestRoom_Light "Luz" <light> (GF_GuestRoom, gLight) {mqtt=">[broker:myhome/hab2/luz:command:ON:1],>[broker:myhome/hab2/luz:com
String GF_GuestRoom_Heating "Calefaccion[%s]" <heating> (GF_GuestRoom, gHeating) {mqtt=">[broker:myhome/hab2/heat:command:ON:1],>[broker:myhome/hab2/heat:c
Number GF_GuestRoom_Temperature "Temperatura[%1f °C]" <temperature> (GF_GuestRoom, gTemperature) {mqtt="<[broker:myhome/cocina/temperature:state:default]"
Number GF_GuestRoom_Humidity "Humedad[%1f%%]" <humidity> (GF_GuestRoom, gHumidity) {mqtt="<[broker:myhome/cocina/humidity:state:default]"
Number GF_GuestRoom_Luminosidad "Luminosidad[%1.0f %% de Luz]" <lightbulb> (GF_GuestRoom, gLuminosidad) {mqtt="<[broker:myhome/cocina/luz:state:default]"
Contact GF_GuestRoom_Motion "Sensor de movimiento [MAP(pir.map):%s]" <motion> (GF_GuestRoom, gMotion) {mqtt="<[broker:myhome/bedroom/PIR:state:default]"
DateTime FF_MasterBedroom_Motion_Detected "Detectado [%1$d/%1$tM %1$tH:%1$tM]" (FF_MasterBedroom, gMotion)
Number GF_GuestRoom_Thermostat "Termostato [%d °C]"
Number GF_GuestRoom_Temperature_Period "Chart period"
```

## SITEMAP

```
Frame label="Planta Baja" icon="groundfloor" {
  Group item=GF_Hallway{
    Default item=GF_Hallway_Temperature label="Temperatura"
    Default item=GF_Hallway_Humidity label="Humedad"
    Default item=GF_Hallway_Light label="Luz"
    Default item=GF_Hallway_Luminosidad label="Luminosidad"
    Default item=GF_Hallway_Motion label="Sensor de movimiento"
    Default item=GF_Hallway_Motion_Detected label="Detectado"
```

	Temperatura	25,4 °C
	Luz	<input type="checkbox"/>
	Calefaccion	OFF
	Sensor de movimiento	OFF





# OpenHAB



## RULES

```
rule "Hab Principal"
when
  Item FF_MasterBedroom_Temperature received update or
  Item FF_MasterBedroom_Thermostat received update
then
  if (FF_MasterBedroom_Temperature.state < FF_MasterBedroom_Thermostat.state) {
    FF_MasterBedroom_Heating.sendCommand(ON)
  }
  if (FF_MasterBedroom_Temperature.state > FF_MasterBedroom_Thermostat.state) {
    FF_MasterBedroom_Heating.sendCommand(OFF)
  }
}
end
```

## PERSISTENCE

```
Strategies {
  everyHour : "0 0 * * * ?"
  everyDay : "0 0 0 * * ?"
  everyMinute : "0 * * * * ?"

  default = everyMinute
}

Items {
  * : strategy = everyMinute, everyChange, everyDay, restoreOnStartup
}
```



# VIDEO DEMOSTRATIVO





<https://1drv.ms/f/s!AiNiqMgIUdaRhvpz4Y-Rsz5PhG6zOA>



# CONCLUSIONES

# Conclusiones

- Se han logrado los objetivos del trabajo, tanto en planificación como en hitos
  - Conectividad E2E
  - Completamente Open Source
  - Diseño amigable
- La Raspberry Pi soporta perfectamente el sistema completo, con una carga de CPU no superior al 25%
- Gran satisfacción con el desarrollo del TFG
- Se ha echado en falta más tiempo disponible para la presentación de este documento y de la memoria para entrar más en detalle



GRACIAS







**José Fernando Adrán Otero**  
Grado de Tecnologías de las Telecomunicaciones  
Arduino

**Antoni Morell Pérez**  
**Pere Tuset Peiró**

10/06/2018