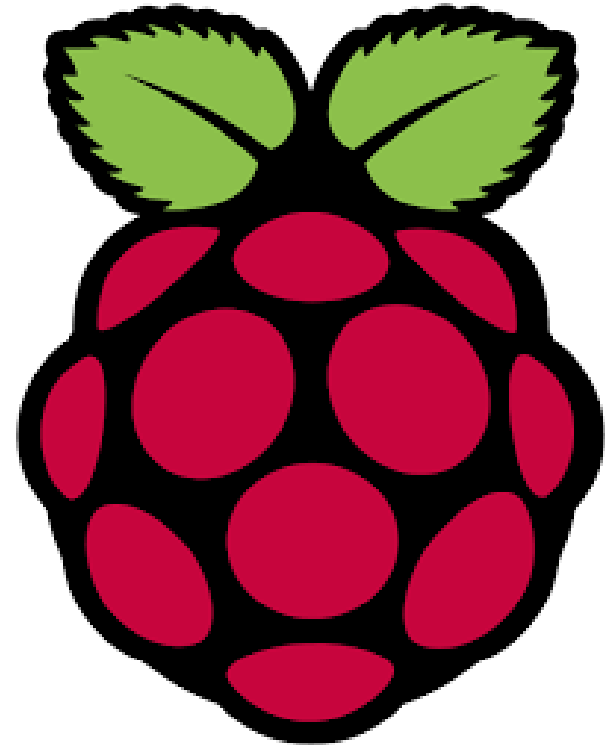
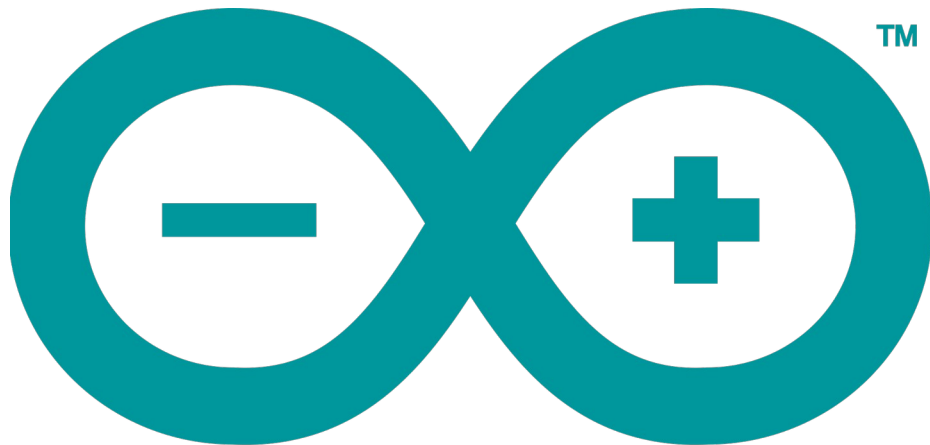


# Aplicacións de Arduino e Raspberry pi en domótica

Setembro 2020

Manuel Guimarey Vila  
Rocío Martínez Martínez





Manuel Guimarey Vila  
Rocio Martínez Martínez



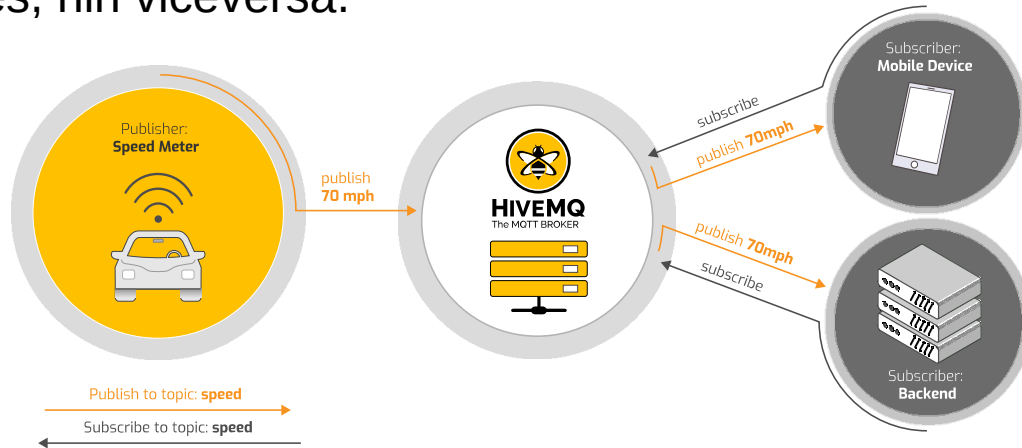
# MQTT

- Message Queuing Telemetry Transport
- Protocolo de mensaxería asíncrona
- Baseado na publicación e subscrición
- Deseñado para conexións con localizacións remotas.
- Ten como característica unha "pegada de código pequena".
- Pódese usar cando o ancho de banda da rede é limitado.
- Este modelo publicación/subscrición require un intermediario (Broker).



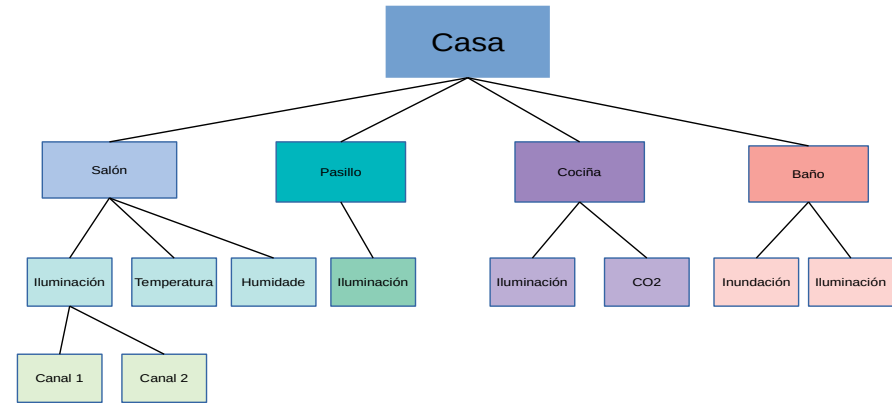
# Sistema MQTT

- Os clientes se conectan cun servidor (intermediario).
- O cliente pode ser publicador, subscritor ou ambos.
- O cliente cando ten datos para enviar os envía o intermediario.
- O intermediario redistribúe eses datos os subscritores, si os hai.
- Os subscritores non precisan saber ningún dato sobre o número, nin localización dos publicadores, nin viceversa.



# Temas MQTT

- A información organizase en temas (topics).
- Estes temas poden estar organizados por niveis, subniveis, etc.
- O separador de niveis e o símbolo /
- Por exemplo:  
Casa/Salón/Iluminación/Canal1



# Calidade servizo MQTT

- MQTT é un protocolo fiable.
- Incorpora un sistema de calidade de servizo QoS.
- Este servizo determina como se entrega os subscritores
- O QoS especificase para cada envío.
- Existen tres grados de calidade:
  - QoS 0: como máximo unha vez (pode que non se entregue).
  - QoS 1: o menos unha vez (pode duplicar mensaxes).
  - QoS 2: exactamente unha vez (garantírase que chegará unha vez a mensaxe).

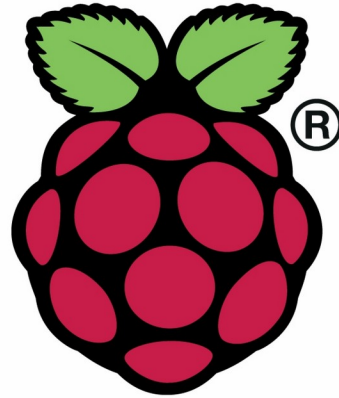


# Servidor MQTT Mosquitto

- Mosquitto é un mediador de mensaxes MQTT.
- É código aberto.
- Debe instalarse nun servidor (sempre acendido).
- Utilizaremos como servidor a Raspberry Pi.



+



# Instalación Mosquitto

- Facemos unha actualización da base de datos do software do repositorio:  
`sudo apt update`
- Instalamos o Broker Mosquitto:  
`sudo apt install mosquitto`
- Instalamos o cliente Mosquitto en calquera outro equipo:  
`sudo apt install mosquitto-clients`





# Instalación Mosquitto

```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
Reading package lists... Done
root@raspberrypi:~# apt-get install mosquitto
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apparmor
The following NEW packages will be installed:
  mosquitto
0 upgraded, 1 newly installed, 0 to remove and 267 not upgraded.
Need to get 139 kB of archives.
After this operation, 339 kB of additional disk space will be used.
Get:1 http://repo.mosquitto.org/debian/ wheezy/main mosquitto armhf 1.4.14-0mosq
uito1~nows1 [139 kB]
Fetched 139 kB in 0s (299 kB/s)
Selecting previously unselected package mosquitto.
(Reading database ... 122495 files and directories currently installed.)
Preparing to unpack ../mosquitto_1.4.14-0mosquitto1~nows1_armhf.deb ...
Unpacking mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for systemd (215-17+deb8u5) ...
Setting up mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for systemd (215-17+deb8u5) ...
root@raspberrypi:~#
```



# Instalación Mosquitto

```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
After this operation, 339 kB of additional disk space will be used.
Get:1 http://repo.mosquitto.org/debian/ wheezy/main mosquitto armhf 1.4.14-0mosq
uito1~nows1 [139 kB]
Fetched 139 kB in 0s (299 kB/s)
Selecting previously unselected package mosquitto.
(Reading database ... 122495 files and directories currently installed.)
Preparing to unpack .../mosquitto_1.4.14-0mosquitto1~nows1_armhf.deb ...
Unpacking mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for systemd (215-17+deb8u5) ...
Setting up mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for systemd (215-17+deb8u5) ...
root@raspberrypi:~# apt-get install mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmosquitto1
The following NEW packages will be installed:
  libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 267 not upgraded.
Need to get 110 kB of archives.
After this operation, 276 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```



# Subscripción a un tema (topic)

Para subscribirse a un tema e visualizalo nunha consola hai que utilizar o seguinte comando:

```
mosquitto_sub -h BROKER -t TOPIC
```

Onde:

BROKER e a dirección IP do intermediario

TOPIC é o tema o que queremos subscribirnos

Por exemplo:

```
usuario@equipo:~$ mosquitto_sub -h 192.168.1.100 -t /casa/salon/temperatura
```



# Publicación de mensajes

Para poder publicar mensaxes dende a consola se utiliza o seguinte comando:

```
mosquitto_pub -h BROKER -t TOPIC -m MENSAXE
```

Onde:

BROKER é a dirección do intermediario

TOPIC é o tema o queremos dirixir a mensaxe

MENSAXE e a mensaxe que queremos transmitir

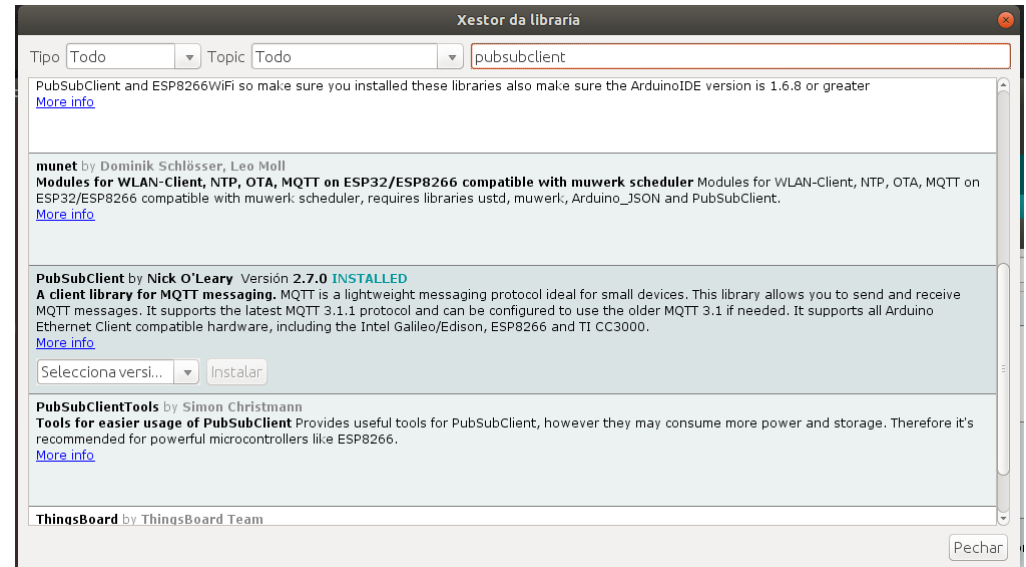
Por exemplo:

```
usuario@equipo:~$ mosquitto_pub -h 192.168.1.100 -t /casa/pasillo/iluminación/0/set 1
```



# MQTT e Arduino

- Usaremos a librería PubSubClient no IDE Arduino.
- Compatible para Arduino e ESP32.
- Permite a publicación e subscripción a mensaxes MQTT.
- Buscar a librería en: Ferramentas>Xestionar bibliotecas.
- Clic en instalar.



# Uso librería PubSubClient

- O constructor e:  
PubSubClient (Cliente)
- Indicamos o servidor MQTT (Broker) coa dirección e o porto:  
PubSubClient setServer (IP\_servidor, porto)
- Conectamos co servidor co método:  
boolean connect (clientID)
- Podemos saber si está conectado co método:  
int connected ()
- Para enviar unha mensaxe relacionado con un tema:  
int publish (tema, mensaxe)



# Uso librería PubSubClient

- Para a subscripción a un tema:  
boolean subscribe (topic, [qos])
- Para a recepción das mensaxes que lle chegan o Broker hai que utilizar un método que xestiona ese evento.
- Cando chega unha mensaxe se executa unha función de código chamada callback.
- O método para definir cal é esa función a executar é:  
PubSubClient setCallback (funcionCallback)
- A referencia da librería podédela encontrar aquí:  
<https://pubsubclient.knolleary.net/api.html>



# MQTT Espurna

Actividades Navegador web Firefox Ven 23 de Ago, 19:39 ESPURNA-8C79CC - ESPURNA 1.13.5 - Mozilla Firefox

ESPURNA-8C79CC - ESPURNA 1.13.5

## MQTT

Configure an MQTT broker in your network and you will be able to change the switch status via an MQTT message.

Enable MQTT  YES

MQTT Broker

MQTT Port

MQTT User

MQTT Password

MQTT Client ID

MQTT QoS

MQTT Retain  YES

MQTT Keep Alive

MQTT Root Topic

Use JSON payload  YES

© 2016-2019 José Pérez Álvarez http://rockerman.cat ESPURNA is GPLv3 licensed





# MQTT Espurna

- Enable MQTT: Permite a activación por MQTT
- MQTT Broker: Dirección IP do mediador MQTT
- MQTT Port: Porto do servizo MQTT
- MQTT User: Configura o usuario, si é necesario
- MQTT Password: Especifica o contrasinal cando o servidor o precisa.
- MQTT Client ID: Identificado do dispositivo dentro da rede MQTT

Enable MQTT	<input type="radio"/> NO <input checked="" type="radio"/> YES
MQTT Broker	<input type="text" value="192.168.1.100"/>
MQTT Port	<input type="text" value="1883"/>
MQTT User	<input type="text" value="Leave blank if no user"/>
<small>You can use the following placeholders: {hostname}</small>	
MQTT Password	<input type="text" value="Leave blank if no password"/>
MQTT Client ID	<input type="text"/>



# MQTT Espurna

- MQTT QoS: Permite especificar a calidade de servizo MQTT.
- MQTT Retain: Indica o mediador si ten que gardar a mensaxe ata que o poida entregar en destino.
- MQTT Keep Alive: Configura cada cantos segundos envía un mensaxe o mediador para saber si a conexión está aberta.
- MQTT Root Topic: é a raíz do tema (topic) das mensaxes que publica Espurna.

MQTT QoS

0: At most once

MQTT Retain

NO

YES

MQTT Keep Alive

300

MQTT Root Topic

casa/pasillo/lampara



# MQTT Tasmota

- Ir a menú: Configuración -> Configuración extra
- Seleccionar Habilitar MQTT
- Gardar os cambios
- Ir a menú: Configuración -> Configurar MQTT.



Módulo Generic

Tasmota\_rele\_lcTech

Otros parámetros

Plantilla

```
{"NAME":"Rele LC Tech","GPIO":[255,255,255,255]}
```

Activar

Clave Administrador Web

....

Habilitar MQTT

Nombre de Dispositivo (Tasmota\_rele\_lcTech)

Tasmota\_rele\_lcTech

Nombre Amigable 1 (Tasmota)

Tasmota\_rele\_lcTech

Emulación

Ninguno

Belkin WeMo dispositivo simple

Hue Bridge dispositivo múltiple

Grabar



# MQTT Tasmota

- Host: Dirección IP do mediador MQTT
- Puerto: Porto do servizo MQTT
- Cliente: Identificador único do dispositivo.
- Usuario: Configura o usuario, si é necesario
- Clave: Especifica o contrasinal cando o servidor o precisa.
- Topic: Palabra identificativa del tema.
- Full Topic: Definición completa del tema. Recomendable estruturar correctamente para temas xerárquicos (multinivel)

Módulo Generic

Tasmota

Parámetros MQTT

Host ( )  
10.0.0.101

Puerto (1883)  
1883

Cliente (DVES\_1FB403)  
rele\_lcTech1

Usuario (DVES\_USER)  
DVES\_USER

Clave ■  
....

Topic = %topic% (tasmota\_1FB403)  
rele\_lcTech

Full Topic (%prefix%/ %topic%/)  
casa/%topic%/ %prefix%/

Grabar

Configuración





Manuel Guimarey Vila  
Rocio Martínez Martínez

