

Laboratorio 6:

MQTT Server

MQTT Protocolo

- Protocolo de tipo Publicar/Subscribir.
- Métodos: PUBLISH, SUBSCRIBE y UNSUBSCRIBE
- Requiere la necesidad de un solo Corredor (Broker)
- Clientes se subscriben o publican a recursos.

Corredores Públicos de Servidores MQTT



- Mosquitto:
 - Direccion: test.mosquitto.org
 - Puerto: 1883
- HiveMQ:
 - Direccion broker.hivemq.com
 - Puerto: 8883



- Otros: [public_brokers](#) · [mqtt/mqtt.org Wiki](#) · [GitHub](#)

MQTT Broker con Mosquitto

- Cliente en Mosquitto: Suscripción
 - -h "nombre de él anfitrión"
 - -p "puerto" (1883)
 - -t "nombre del tema"
 - -q "QoS"
 - -d "muestra información de conexión"
 - -v "más información de mi mensaje"
- Ejemplo: `mosquitto_sub -h test.mosquitto.org -p 1883 -t "cursoiot/Lab6/mqtt" -v -d`



MQTT Broker con Mosquitto

- Cliente en Mosquitto: Publicador
 - -h "nombre de él anfitrión"
 - -p "puerto" (1883)
 - -t "nombre del tema"
 - -m "mensaje"
 - -q "QoS"
 - -d "muestra información de conexión"
 - -r "retenedor del último mensaje enviado"
- Ejemplo: `mosquitto_pub -h test.mosquitto.org -p 1883 -t "cuarto8/humedad" -m "48.2" -d -r`



MQTT Temas y Comodines

- Temas:
 - Tema 1: Hotel/piso1/cocina/temperatura
 - Tema 2: Hotel/piso2/cuarto4/iluminacion
 - Tema 3: Hotel/piso2/cuarto3/temperatura
- Comodines
 - Un solo nivel "+": Hotel/piso1/+/temperatura
 - Cliente se suscribe a todos los temas que contengan "temperatura" en ese nivel.
 - Múltiple nivel "#": Hotel/piso2/ #
 - Cliente se suscribe a todos los temas en "piso2".



MQTT Mosquitto Corredor Local



- Crear Servidor Local: `mosquitto -p (puerto) -v (más información)`
`mosquitto -p 1883 -v`
- Estructura subscritor: `mosquitto_sub -h (anfitrión) -p (puerto) -t (tema)`
 - Ejemplo: `mosquitto_sub -h 127.0.0.1 -p 1883 -t "ejemplo"`
- Estructura de publicador: `mosquitto_pub -h (anfitrión) -p (puerto) -t (tema) -m (mensaje)`
 - Ejemplo: `mosquitto_pub -h 127.0.0.1 -p 1883 -t "ejemplo" -m "Hola"`
- Corredor funcione en la máquina virtual:
 - No se puede acceder desde otra máquina.
 - Ideal para desarrollar aplicaciones en servicios de acceso privado

MQTT Estructura de Paquetes

MQTT Connect Message Structure

Connect Clean Session True Client ID =PYTON1

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Meaning	Header	Remaining Length	Length of protocol name	Protocol Name +Version					Connect Flags	Keep Alive	Length										
Hex	0x10	0x13	0x0	0x4	0x4d	0x51	0x54	0x54	0x4	0x2	0x0	0x3c	0x0	0x7	0x70	0x79	0x74	0x68	0x6F	6E	0x31
Ascii		19		4	M	Q	T	T	4			60		7	P	Y	T	H	O	N	1

Notes:

Remaining Length = bytes 3 to 21
 Length of protocol name=4 =MQTT
 length in bytes 13-14 -payload length =7
 Connect Flags show Clean Session =True

Connect Flags

User name flag = bit 7
 Password Flag = bit 6
 Will Retain = bit 5
 Will QOS = bit 5
 Will QOS = bit 4
 Will Flag = bit 2
 Clean Session = bit 1
 Reserved = bit 0



- Wireshark puede ser utilizado para identificar paquetes específicos.
- Encontrar ID de cliente.
- Para más información de filtros: <https://www.wireshark.org/docs/dfref/m/mqtt.html>