

# MQTT : Message Queue Telemetry Transport

## Modelo MQTT

**Clientes:** Se comunican recibiendo y mandando mensajes por medio de Suscripciones y Publicaciones etiquetadas por **Temas**.

**Servidor:** es el agente centralizado (**Broker**) que comunica a los clientes.

**Temas** es una forma de direccionamiento que permiten a los clientes compartir información. Son estructurados jerárquicamente parecido al sistema de archivos y carpetas. Los temas son:

1. Sensibles a las mayúsculas
2. Codificadas en UTF-8
3. deben tener un carácter para ser válidos

Tema \$SYS es el único que puede ser creado por el broker y es de solo lectura. contiene información propia de la implementación de MQTT, número de clientes, clientes activos. etc...

Los clientes pueden suscribirse a un grupo de temas utilizando comodines *wildcard characters* # (multi nivel) + (nivel simple).

Temas son creados cuando hay al menos un suscriptor. Publica un mensaje con bandera de retención (**retain**) en verdadero.

Los temas se remueven cuando el último cliente asociado a él se desconecta. Cuando un cliente se conecta con la bandera de limpiar sesión en verdadero.

**Sesión:** Identifica la incorporación de un cliente al broker. Es necesario para establecer la comunicación. **Suscripción:** Asocia un cliente a un tema. Pueden ser:

- Pasajeras (Transient): terminan junto con la sesión. Los mensajes no se entregan.
- Durables: Sesión cerrada pero suscripción abierta. Se reciben los mensajes tan pronto y se reconecta el cliente.

El tipo de suscripción lo define una **bandera de sesión**.

**Mensaje:** Unidades de intercambio de datos entre temas de clientes. Es la carga útil.

## Estructura del Formato de mensajes

### Encabezado de dos bytes mínimo

Tipo de mensaje: 4 bit

bandera duplicado DUP: 1 bit

Nivel de Calidad de servicio (QoS): 2 bit

Bandera de retención (Retain) 1 bit

longitud remanente >=8 bit

16 bit = 2 bytes

### longitud remanente.

Contiene el tamaño enseguida del encabezado mínimo que incluye:

- el tamaño de encabezado variable

- carga util.

El byte 0 es el menos significativo.

## Formato de mensajes

1. Connect - Connack. Broker acusa de recibido al cliente que la conexión es exitosa para publicar o suscribir.
2. subscribe - suback
3. publish - puback

Los clientes informan cada 60 segundos que siguen conectados con el mensaje keep alive.

Identificación de clientes MQTT.

Nombres de cliente son únicos. Si un cliente se intenta reconectar con el mismo nombre, tumba al anterior que tratará de recuperarse y generarán un ciclo de interrupciones causadas por la conexión-desconexión.

## Sesiones de clientes

Por defecto los clientes comienzan sesiones nuevas en los brokers, borrando toda información asociada al cliente con anterioridad.

Si la sesión no es nueva, el broker recordará la sesión y entregará mensajes no recibidos.

Se indica con la bandera de sesión.

## Mensajes de último deseo Will message.

Son mensajes especiales incluidos en **connect** que los suscriptores reciben en solo en caso de que el publicador de caiga debido a problemas técnicos de red.

El **will message** es asociado a cada tópico y es retenido por el broker quien se encargará de difundirlo en dado caso.

Si el cliente publicador se desconecta normalmente, el **will message** normalmente no se envía.

Ejemplo proceso:

1. El publicador se conecta al broker y le informa que quiere notificar su falla a todos los suscriptores.
2. Si el broker detecta una ruptura de conexión (puede ser por medio del mensaje KILL (Keep to live) )

## Esquema Publicador - Suscriptor

El esquema no permite dirigir mensajes a otros clientes.

El cliente publicador solo puede publicar un tópico a la vez. La relación es de 1:N

## Calidad de servicio y flujo de mensajes de mensajes publicados.

QoS 0, no se garantiza mensaje de envío. QoS 1, Se garantiza el mensaje con duplicados. QoS2, Se garantiza el mensaje sin duplicados.

## Mensajes publicados y bandera Retain

Retain = false : el broker no guarda el mensaje al ser enviado. Retain = true: el broker guarda el último mensaje del publicador.

Uso: cuando los sensores no se actualizan frecuentemente.

# Seguridad

## Mecanismos de autenticación

Mosquitto verifica el cliente a través de tres modos:

- Id
- Usuarios y contraseñas: se transmiten en texto plano sin encriptar y es usado para restringir acceso a tópicos
- Certificados de clientes.