

SEMANA 6



Protocolos en Soluciones IoT: HTTP y MQTT

En el mundo del Internet de las Cosas (IoT), la comunicación entre dispositivos es fundamental. Para que dos o más dispositivos puedan intercambiar información, deben seguir reglas establecidas que permitan un entendimiento común. Estas reglas se llaman **protocolos de comunicación**. Dos de los protocolos más utilizados en aplicaciones IoT son **HTTP** y **MQTT**.



¿Qué es un protocolo de comunicación?

Un **protocolo de comunicación** es un conjunto de reglas y convenciones que definen cómo se transmiten datos entre dos o más dispositivos conectados. Los protocolos garantizan que los mensajes sean interpretados correctamente por todos los participantes.

En el caso del IoT, los protocolos deben ser eficientes, seguros y capaces de funcionar incluso en condiciones de red poco favorables, como baja velocidad de conexión o alta latencia.



Protocolo HTTP en soluciones IoT

¿Qué es HTTP?

HTTP (*HyperText Transfer Protocol*) es uno de los protocolos más antiguos y extendidos en internet. Fue diseñado originalmente para transferir documentos HTML (páginas web), pero hoy en día también se utiliza ampliamente en sistemas IoT.

Características principales:

- **Modelo cliente-servidor:** Un dispositivo (cliente) solicita información a otro (servidor).
- **Sincronización requerida:** El cliente debe esperar una respuesta del servidor antes de continuar.
- **Basado en texto:** Los mensajes son legibles por humanos, lo cual facilita su análisis, pero también aumenta el tamaño de los paquetes de datos.
- **Sin estado (stateless):** Cada solicitud es independiente, no mantiene sesión activa.

Uso en IoT

Aunque HTTP no fue diseñado específicamente para IoT, muchas aplicaciones lo usan porque:

- Es compatible con APIs REST.
- Existe infraestructura y herramientas listas para usarlo.
- Se puede integrar fácilmente con servicios cloud como Firebase, AWS, etc.

Sin embargo, tiene algunas desventajas para IoT:

- Alto consumo de recursos (batería, ancho de banda).
- No es ideal para transmisiones en tiempo real o eventos asincrónicos.

Protocolo MQTT en soluciones IoT

¿Qué es MQTT?

MQTT (*Message Queuing Telemetry Transport*) es un protocolo de mensajería ligero basado en el modelo **publicador-suscriptor**, diseñado específicamente para redes con recursos limitados y conexiones inestables, como las que se encuentran en entornos IoT.

Arquitectura básica

La arquitectura de MQTT consta de tres componentes principales:

- **Cliente:** Puede ser un sensor, dispositivo IoT, aplicación móvil, etc.
- **Broker:** Es el intermediario central que recibe los mensajes y los distribuye a los suscriptores.
- **Tópico (Topic):** Es el canal por donde fluyen los mensajes. Los clientes se suscriben a tópicos para recibir información.

Ejemplo: Un sensor de temperatura publica (publish) datos al broker en el tópico "sensores/temperatura". Una aplicación interesada se suscribe (subscribe) a ese mismo tópico para recibir los datos.

Niveles de Calidad de Servicio (QoS)

MQTT define tres niveles de calidad de servicio para asegurar la entrega correcta de los mensajes, dependiendo de la criticidad de la información:

QoS 0 – “At most once” (Máximo una vez)

- El mensaje se envía una sola vez, sin confirmación.
- Rápido y eficiente, pero no confiable si hay pérdida de paquetes.

✓ QoS 1 – “At least once” (Al menos una vez)

- El emisor espera una confirmación del receptor.
- Posible duplicado de mensajes, pero garantiza que lleguen.

✓ QoS 2 – “Exactly once” (Exactamente una vez)

- Garantiza que el mensaje llegará exactamente una vez, evitando duplicados.
- Requiere más tráfico y procesamiento, pero es el más seguro.

Estos niveles permiten ajustar el comportamiento del sistema según las necesidades de fiabilidad y recursos disponibles.

Implementación y plataformas en casos reales

MQTT es muy popular en proyectos reales de IoT debido a su simplicidad, bajo consumo y compatibilidad. Algunas implementaciones comunes incluyen:

- **Domótica:** Control de luces, sensores de movimiento, termostatos inteligentes.
- **Monitoreo remoto:** Sensores de humedad, temperatura, GPS en zonas rurales o marítimas.
- **Industria 4.0:** Transmisión de datos desde máquinas industriales hacia centros de control.

Plataformas que soportan MQTT:

- **Mosquitto:** Broker MQTT gratuito y de código abierto.
 - **HiveMQ:** Plataforma MQTT comercial escalable.
 - **AWS IoT Core:** Servicio en la nube que permite conectar dispositivos usando MQTT.
 - **Node-RED:** Herramienta visual para programar flujos de datos IoT, incluye soporte nativo para MQTT.
-

Seguridad en MQTT

Como cualquier protocolo de comunicación, MQTT necesita protegerse contra accesos no autorizados y ataques cibernéticos. Algunas medidas de seguridad comunes son:

- **Autenticación:** Usuarios y contraseñas para conectar al broker.
- **Cifrado TLS/SSL:** Protección de los datos durante la transmisión.
- **Certificados X.509:** Identificación segura de dispositivos IoT.
- **Control de acceso:** Permisos por tópico, para evitar que un dispositivo publique o lea donde no debe.

Muchos brokers modernos ofrecen configuraciones prediseñadas para estos aspectos de seguridad.

Ejemplo práctico de uso de MQTT

Imagina este escenario:

1. Tienes un **Arduino** con un sensor de temperatura.
2. El Arduino está conectado a Wi-Fi y tiene instalada una librería MQTT (como PubSubClient).
3. Se conecta a un **broker MQTT** (por ejemplo, Mosquitto en tu red local o un broker público como `test.mosquitto.org`).
4. Publica los valores de temperatura en un tópico llamado `sensor/temperatura`.
5. Tu computadora o smartphone tiene una aplicación suscrita a ese tópico y muestra los datos en tiempo real.

Este tipo de arquitectura es flexible, escalable y eficiente, especialmente útil cuando hay muchos dispositivos comunicándose simultáneamente.

Conclusión

Entender los protocolos de comunicación es clave para construir sistemas IoT robustos y eficientes. Mientras que **HTTP** sigue siendo relevante por su universalidad y facilidad de integración, **MQTT** destaca por su diseño específico para IoT: es ligero, rápido, adaptable y escalable.

Elegir el protocolo adecuado dependerá de tus necesidades:

- Si necesitas integración con la web y APIs: usa **HTTP**.
- Si priorizas eficiencia, bajo consumo y comunicación bidireccional: usa **MQTT**.

Ambos protocolos pueden coexistir en un mismo proyecto, aprovechando lo mejor de cada uno.
