

Módulo 3.2: Protocolos de comunicación IoT

MSc. Carlos Cedeño

Al finalizar esta sesión, entenderás por qué no podemos usar las herramientas web tradicionales para todo en IoT. Conocerás los protocolos clave como MQTT y CoAP y te convencerás de la necesidad de dominarlos para diseñar y construir sistemas de IoT eficientes, escalables y profesionales.

1. El Ancestro Universal: HTTP

Todos conocemos **HTTP (HyperText Transfer Protocol)**. Es el motor de la web. En IoT, lo vemos principalmente en las **APIs RESTful**, que son fantásticas para comunicar servicios en la nube o para que una aplicación móvil obtenga datos de nuestra plataforma.

Modelo: Cliente-Servidor (Petición-Respuesta).

- **Cliente:** "¡Oye, Servidor! ¿Cuál es la temperatura?" (GET /temperatura)
- **Servidor:** "La temperatura es 25°C." (Respuesta 200 OK)

¿Por qué NO es ideal para nuestros sensores?

- **Es "Pesado" y "Verborrágico":** Las cabeceras de HTTP son texto plano y muy grandes. Es como enviar una carta de una sola palabra dentro de una caja de cartón gigante y pesada. Mucho desperdicio de energía y datos.
- **Modelo Ineficiente para Notificaciones:** El servidor no puede iniciar una conversación. Si un sensor necesita reportar una alerta, el servidor no se entera hasta que la aplicación le pregunte. Para tener datos "en tiempo real", la app tendría que preguntar constantemente (`polling`), ¡destrozando la batería del sensor y la red!
- **Alto Consumo de Recursos:** Cada conexión TCP y transacción HTTP consume una cantidad significativa de memoria y energía, recursos que un microcontrolador básico no tiene de sobra.

Conclusión sobre HTTP: Excelente para la capa de servicios y aplicaciones (la cima de la pirámide IoT), pero muy ineficiente para la comunicación de dispositivos en campo (la base de la pirámide).

2. MQTT: El Estándar de la Mensajería para IoT

MQTT (Message Queuing Telemetry Transport) fue diseñado desde cero para superar las limitaciones de HTTP en entornos de telemetría. Es, por mucho, el protocolo más popular en IoT.

Modelo: Publicación-Suscripción (Pub/Sub).

Este es el cambio de paradigma clave. Los dispositivos no se hablan directamente, sino a través de un intermediario llamado **Broker**.

- **Broker:** Es el centro de mensajería, como una oficina de correos inteligente.
- **Publicador (Publisher):** Un dispositivo (ej. un sensor) que envía mensajes a un "tema" (`topic`) específico en el broker. Por ejemplo, publica `25` en el tema `hogar/sala/temperatura` .
- **Suscriptor (Subscriber):** Cualquier dispositivo o aplicación (ej. una app móvil, una base de datos) que le dice al broker: "Estoy interesado en el tema `hogar/sala/temperatura` ".

El broker recibe el 25 y lo reenvía al instante a TODOS los suscriptores interesados. El sensor no necesita saber quién recibe los datos.

¿Por qué MQTT es tan potente?

- **Extremadamente Ligero:** La cabecera de un mensaje MQTT es mínima (solo 2 bytes), reduciendo drásticamente el consumo de datos.
- **Comunicación "Push" Nativa:** El broker empuja los datos a los suscriptores en tiempo real.
- **Eficiencia Energética:** Mantiene una conexión TCP persistente pero inactiva, "despertando" solo para enviar o recibir datos.
- **Flexibilidad y Escalabilidad (1 a Muchos):** Un solo sensor puede publicar un dato y este puede ser recibido por múltiples sistemas (una app, una base de datos, un sistema de alertas) sin ningún cambio en el sensor.
- **Fiabilidad (QoS - Quality of Service):** MQTT te permite elegir cómo de importante es un mensaje QoS.

Conclusión sobre MQTT: Es la elección ideal para la mayoría de las aplicaciones de IoT que requieren comunicación fiable, en tiempo real y eficiente desde los sensores hacia la nube.

3. CoAP: El Protocolo para el Mundo de las Restricciones Extremas

¿Qué pasa si MQTT todavía es "demasiado" para nuestro dispositivo? Pensemos en sensores que operan en redes de muy baja potencia (LPWAN) o que funcionan con la energía recolectada del ambiente. Aquí entra **CoAP (Constrained Application Protocol)**.

Modelo: Cliente-Servidor (como un "micro-HTTP").

CoAP se parece a HTTP en su modelo, pero está ultra-optimizado.

¿Qué lo hace tan especial?

- **Funciona sobre UDP:** A diferencia de HTTP y MQTT que usan TCP, CoAP usa UDP. UDP no establece una conexión persistente ni garantiza la entrega, lo que lo hace mucho más rápido y ligero, ideal para redes con pérdidas de paquetes.
- **Cabeceras Mínimas Binarias:** Al igual que MQTT, sus cabeceras son extremadamente pequeñas.
- **Soporte Asíncrono:** Aunque es Petición-Respuesta, tiene un mecanismo de "Observación". Un cliente puede decir "obsérvame este recurso" y el servidor le notificará automáticamente cada vez que cambie. Es una forma de simular el "push" en un modelo Cliente-Servidor.
- **Descubrimiento de Servicios:** Los dispositivos pueden "preguntar" a otros qué funcionalidades ofrecen, facilitando la autoconfiguración de redes.

Conclusión sobre CoAP: Es el protocolo para los entornos más hostiles y con mayores restricciones de energía y cómputo. Es la máxima expresión de la eficiencia.

4. Otros Protocolos que Debes Conocer

El mundo de IoT es vasto. Aunque MQTT y CoAP son los reyes, podrías encontrarte con otros:

- **Modbus:** Un protocolo muy antiguo (1979) pero todavía omnipresente en la **automatización industrial (OT)**. Se usa para comunicar PLCs, variadores de frecuencia y maquinaria industrial. Si vas a trabajar en una fábrica, necesitas saber Modbus (en sus variantes Serial y TCP).
- **LoRaWAN / Sigfox:** No son protocolos de aplicación como los anteriores, sino **protocolos de red** para comunicación de largo alcance y baja potencia. A menudo, un dispositivo LoRaWAN enviará sus datos a un gateway, y ese gateway usará MQTT para enviar los datos a la nube.