



Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Taller Integrador

Definición del Protocolo de Comunicación

Definición técnica y diseño conceptual

Estudiantes:

Milagro Rojas
Kimberley Morales
Angie Hernández

Profesor: Guillermo Castro Badilla

Fecha: Febrero 2026

1. Introducción

El proyecto se centra en el desarrollo de una Estación Electrónica de Diagnóstico Médico General, cuyo objetivo principal es integrar distintos módulos de medición biomédica en una plataforma centralizada capaz de recolectar, procesar y visualizar información clínica básica.

El sistema contempla los siguientes módulos: medidor de presión arterial y frecuencia cardíaca, electrocardiograma, medidor de temperatura corporal y saturación de oxígeno, y medidor de glucosa, cada uno de estos módulos genera datos que deben ser transmitidos hacia un sistema de integración, donde posteriormente serán almacenados y presentados al usuario mediante interfaces digitales.

Debido a la naturaleza distribuida del sistema, se vuelve fundamental establecer un mecanismo de comunicación confiable que permita la interacción entre los dispositivos de adquisición y las plataformas de visualización, garantizando eficiencia, escalabilidad y facilidad de implementación.

2. Alcance del equipo de integración

Dentro del marco general del proyecto, al equipo de integración se le asigna la responsabilidad de unificar los diferentes módulos del sistema en una infraestructura común, esto implica la definición de un protocolo estándar de comunicación, el desarrollo del servidor de integración, la implementación de una base de datos, así como la creación de una plataforma web y una aplicación Android para la visualización de los datos médicos.

Como primer paso del proceso de integración, resulta indispensable seleccionar el protocolo de comunicación que servirá como eje central del sistema, esta decisión impacta directamente en el rendimiento, la escalabilidad y la complejidad de la arquitectura final. Por esta razón, se realizó una investigación preliminar de distintos protocolos que pudieran adaptarse a los requerimientos del proyecto.

3. Protocolos de comunicación analizados

Con el fin de identificar la mejor alternativa, se analizaron cuatro protocolos de comunicación ampliamente utilizados en aplicaciones de Internet de las Cosas (IoT) y sistemas distribuidos: HTTP/REST, MQTT, WebSocket y CoAP.

Cada uno de estos protocolos presenta características particulares en cuanto a modelo de comunicación, consumo de recursos, facilidad de implementación y soporte para transmisión en tiempo real. Durante la etapa de investigación se evaluaron sus ventajas y desventajas considerando aspectos como la compatibilidad con plataformas como ESP32 y Raspberry Pi, la capacidad de manejar múltiples dispositivos, el consumo de ancho de banda y la posibilidad de integración con servicios web y aplicaciones móviles.

A partir de este análisis, se establecieron criterios de comparación enfocados en la eficiencia del intercambio de datos, la escalabilidad del sistema y la simplicidad de desarrollo, los cuales sirvieron como base para la selección del protocolo final.

4. Comparación de protocolos

Los protocolos estudiados fueron comparados tomando en cuenta factores técnicos relevantes para el proyecto, tales como el modelo de comunicación, el consumo de recursos, la facilidad de integración con servidores y bases de datos, y su adecuación para aplicaciones en tiempo real. Este análisis permitió identificar las fortalezas y limitaciones de cada alternativa dentro del contexto específico de la estación de diagnóstico médico.

Cuadro 1: Comparación de protocolos de comunicación

Protocolo	Modelo de comunicación	Transporte	Overhead	Escalabilidad	Consumo	Caso de uso recomendado
HTTP/REST	Cliente-Servidor (Request/Response)	TCP	Alto	Media	Alto	APIs web, autenticación y configuración remota
MQTT	Publicación-Suscripción (broker central)	TCP	Bajo	Alta	Bajo	Telemetría IoT y monitoreo en tiempo real con múltiples sensores
CoAP	Cliente-Servidor ligero tipo REST	UDP	Muy bajo	Media	Muy bajo	Redes de sensores con recursos limitados
WebSocket	Comunicación bidireccional persistente	TCP	Medio	Media	Medio	Dashboards en tiempo real y notificaciones

5. Selección del protocolo

Con base en los resultados obtenidos, se determinó que MQTT es el protocolo que mejor se adapta a las necesidades del sistema. Su arquitectura basada en publicador/suscriptor facilita la comunicación entre múltiples módulos y un servidor central, además de presentar un bajo consumo de recursos y una alta escalabilidad, características especialmente importantes en aplicaciones IoT.

La adopción de MQTT permite establecer una comunicación eficiente entre los dispositivos de medición y las plataformas digitales, proporcionando una base sólida para el desarrollo del servidor de integración, la base de datos y las interfaces web y móvil.

6. Diseño Conceptual

6.1. Visión general del sistema

El sistema se divide en tres capas: **adquisición**, donde cada módulo biomédico opera en su propia Raspberry Pi 4 y publica datos vía MQTT; **integración**, donde una Raspberry Pi 4 central ejecuta el broker Mosquitto, el backend en Python/Flask y la base de datos MySQL; y **presentación**, donde una interfaz web y una aplicación Android consumen los datos mediante REST y WebSocket. Todas las unidades se comunican dentro de la misma red local WiFi.

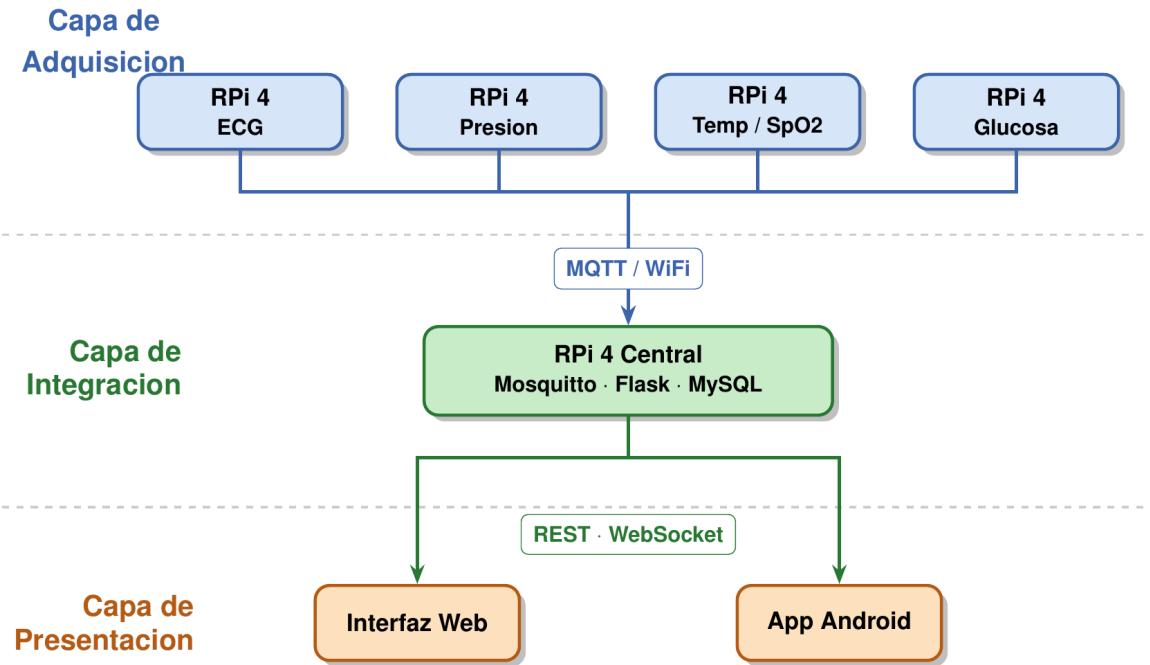


Figura 1: Arquitectura general del sistema en tres capas.

6.2. Flujo de datos

Cada Raspberry Pi de módulo lee su sensor, filtra la señal y publica un mensaje JSON en el broker MQTT. El backend, suscrito a esos tópicos, valida el dato, lo almacena en MySQL y lo reenvía en tiempo real a la interfaz web mediante WebSocket. La aplicación Android consulta el historial a través de la API REST.

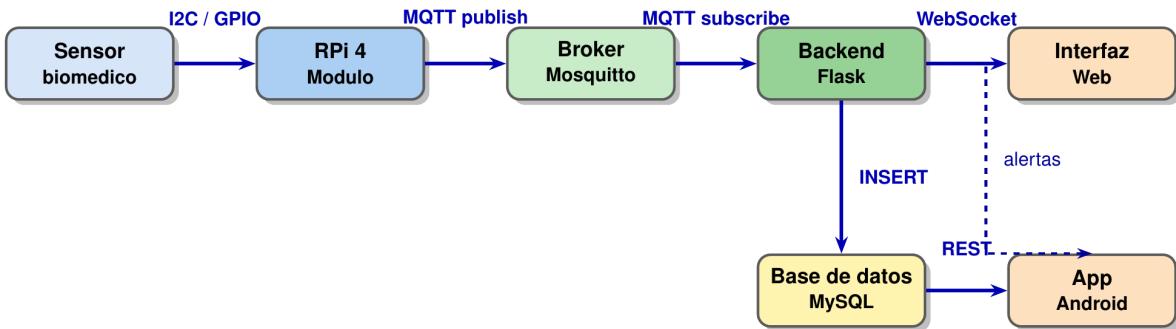


Figura 2: Flujo de datos desde el sensor hasta las interfaces de usuario.

6.3. Tópicos MQTT y formato de mensaje

Los módulos publican bajo la jerarquía `estacion/{modulo}/datos`. El payload es un objeto JSON con el identificador de usuario, marca de tiempo y valores de la medición. Ejemplo para presión arterial:

```
{  
    "id_usuario": 5,  
    "timestamp": "2026-02-21T10:35:00",  
    "sistolica": 120,  
    "diastolica": 80,  
    "frecuencia": 72  
}
```

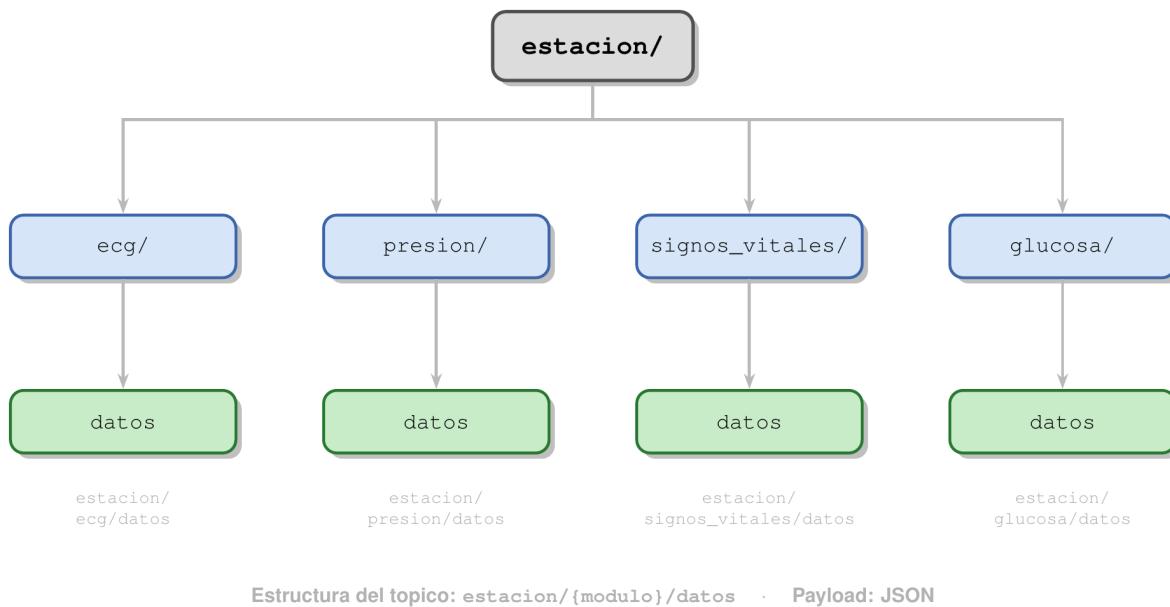


Figura 3: Jerarquía de tópicos MQTT del sistema.

6.4. Datos Requeridos en los modulos y en el registro de paciente

6.4.1. Tabla de usuarios (General)

Esta tabla es compartida por todos los módulos del sistema. Almacena los datos de registro de cada paciente y permite que la app y la interfaz web verifiquen su identidad mediante usuario y contraseña. El campo `id_usuario` es el vínculo entre esta tabla y todas las tablas de mediciones.

Cuadro 2: Tabla general de usuarios — Registro e inicio de sesión

Campo	Descripción	Ejemplo
<code>id_usuario</code>	Identificador único del usuario (asignado automáticamente)	5
<code>nombre</code>	Nombre completo del paciente	Juan Pérez
<code>username</code>	Nombre de usuario para iniciar sesión (único en el sistema)	jperez
<code>password_hash</code>	Contraseña convertida a hash por el sistema (nunca en texto plano)	<i>generada por bcrypt</i>
<code>created_at</code>	Fecha y hora en que se registró el usuario	2026-02-21 09:00:00

6.4.2. Electrocardiograma

El módulo de ECG debe enviar el identificador del paciente, el arreglo de muestras captadas por el sensor y la fecha de la medición. Las muestras se envían como un arreglo JSON porque son muchos valores en el tiempo, no un único número.

Cuadro 3: Campos obligatorios del JSON — Módulo ECG

Campo	Descripción	Ejemplo
<code>id_usuario</code>	ID del paciente registrado en el sistema	5
<code>datos</code>	Arreglo de muestras de la señal ECG	[0.1, 0.4, 0.9, 1.2, ...]
<code>fecha</code>	Fecha y hora de la medición	2026-02-21 10:35:00

6.4.3. Presión arterial y frecuencia cardíaca

El módulo de presión arterial debe enviar los valores sistólico y diastólico medidos en mmHg, junto con la frecuencia cardíaca en latidos por minuto. Los tres valores son numéricos enteros y se deben incluir en cada medición.

Cuadro 4: Campos obligatorios del JSON — Módulo Presión y Frecuencia Cardíaca

Campo	Descripción	Ejemplo
<code>id_usuario</code>	ID del paciente registrado en el sistema	5
<code>sistolica</code>	Presión sistólica en mmHg	120
<code>diastolica</code>	Presión diastólica en mmHg	80
<code>frecuencia</code>	Frecuencia cardíaca en lpm	72
<code>fecha</code>	Fecha y hora de la medición	2026-02-21 10:35:00

6.4.4. Temperatura corporal y saturación de oxígeno

El módulo de signos vitales envía dos variables en el mismo mensaje: la temperatura corporal con dos decimales (por ejemplo 36.58 °C) y la saturación de oxígeno como un porcentaje entero entre 0 y 100. Ambos campos pueden enviarse juntos o por separado según la disponibilidad del sensor.

Cuadro 5: Campos obligatorios del JSON — Módulo Temperatura y Saturación de Oxígeno

Campo	Descripción	Ejemplo
<code>id_usuario</code>	ID del paciente registrado en el sistema	5
<code>temperatura</code>	Temperatura corporal en °C	36.58
<code>saturacion_oxigeno</code>	Saturación de oxígeno en % (rango: 0–100)	98
<code>fecha</code>	Fecha y hora de la medición	2026-02-21 10:35:00

6.4.5. Glucosa

El módulo de glucosa es el más simple: envía un único valor numérico que representa el nivel de glucosa en sangre medido en mg/dL. Es importante incluir siempre la fecha para poder distinguir mediciones tomadas en diferentes momentos del día.

Cuadro 6: Campos obligatorios del JSON — Módulo Glucosa

Campo	Descripción	Ejemplo
<code>id_usuario</code>	ID del paciente registrado en el sistema	5
<code>glucosa_mg_dl</code>	Nivel de glucosa en mg/dL	95
<code>fecha</code>	Fecha y hora de la medición	2026-02-21 10:35:00

7. Conclusiones

La definición de un protocolo de comunicación es fundamental para garantizar la interoperabilidad entre los distintos módulos del sistema, permitiendo que todos transmitan información bajo un formato común y estructurado, además, el uso de protocolos estandarizados facilita la escalabilidad del sistema, mejora la seguridad mediante mecanismos de autenticación y cifrado, y reduce la complejidad del mantenimiento. Sin un protocolo definido, los módulos funcionarían de forma aislada, dificultando la integración con el servidor central, la base de datos y las interfaces web y móvil.

La investigación y comparación de los diferentes protocolos de comunicación permitió seleccionar una solución adecuada para la integración de los módulos biomédicos del proyecto, la elección de MQTT contribuye a una arquitectura más eficiente y escalable, asegurando una transmisión confiable de los datos médicos y facilitando el crecimiento futuro del sistema.

Referencias

- [1] MitB For All, “API REST y el protocolo HTTP,” Medium, 2020. Disponible: <https://medium.com/mitb-for-all/rest-apis-and-the-http-protocol-fde03a77a970>.
- [2] Amazon Web Services, “¿Qué es MQTT”. Disponible: <http://aws.amazon.com/what-is/mqtt/>.
- [3] EMQX, “Protocolo CoAP: características, casos de uso, ventajas y desventajas para IoT,” 2021. Disponible: <https://www.emqx.com/en/blog/coap-protocol>.
- [4] IBM, “WebSocket,” documentación oficial. Disponible: <https://www.ibm.com/docs/es/was/9.0.5?topic=applications-websocket>.
- [5] Expanice, “IoT Communication Protocols Comparison”. Disponible: <https://expanice.com/article/iot-communication-protocols-comparison>.