

CardiolA – Parte 2 (Fog/Cloud): MQTT + Node-RED Dashboard

FIAP – Fase 3 | ESP32 (Wokwi) → MQTT (HiveMQ) → Node-RED

Resumo

Implementamos um pipeline completo de monitoramento em saúde simulada com ESP32 (Wokwi), publicação MQTT em nuvem (**HiveMQ**) e visualização em tempo real no **Node-RED** (gráfico de BPM, gauge de temperatura e status/alertas). O firmware usa JSON, publica com *retain* e possui resiliência offline (SPIFFS) com reenvio de backlog.

Arquitetura

- **Dispositivo:** ESP32 (Wokwi) lê DHT22 (temp/umid) e simula BPM via potenciômetro; LED RGB indica faixas de risco.
- **Conectividade:** Wi-Fi (Wokwi-GUEST), MQTT `broker.hivemq.com:1883`, tópico `cardioia/paciente/dados`.
- **Edge/Resiliência:** cada leitura é salva no SPIFFS (JSON por linha) e reenviada ao reconectar.
- **Visualização:** Node-RED assina o tópico e alimenta widgets e regras de alerta.

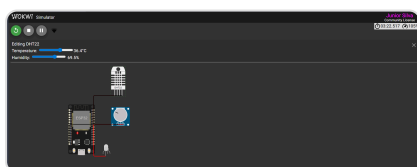


Figura – Simulação Wokwi (ESP32 + DHT22 + Potenciômetro)

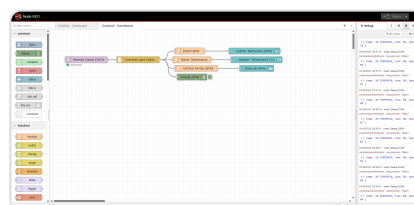


Figura – Fluxo Node-RED (MQTT → JSON → Widgets + Regras)

Protocolo e Tópico

- **Broker:** `broker.hivemq.com` (1883)
- **Tópico:** `cardioia/paciente/dados`

```
{ "temp": 36.7, "hum": 55.0, "bpm": 88 }
```

Mensagens são **retidas** (retain=true) para novos assinantes verem o último valor.

Firmware (src/sketch.ino)

- Bibliotecas: WiFi.h, PubSubClient.h, ArduinoJson.h, DHT.h, SPIFFS.
- Reconexão MQTT não bloqueante (~5s) e `mqttClient.setBufferSize(256)`.
- Fallback do DHT (usa último valor quando NaN) para não interromper publicações em simulação.
- SPIFFS: backlog em JSON e reenvio ao reconectar.

Dashboard (Node-RED)

- Flow: `node-red/cardioia_dashboard.flow.json`
- mqtt in → json → (Extrair BPM → ui_chart) e (Extrair Temperatura → ui_gauge)
- Regras de alerta (BPM): Bradicardia <60 | Normal 60–100 | Taquicardia > 100 → ui_text

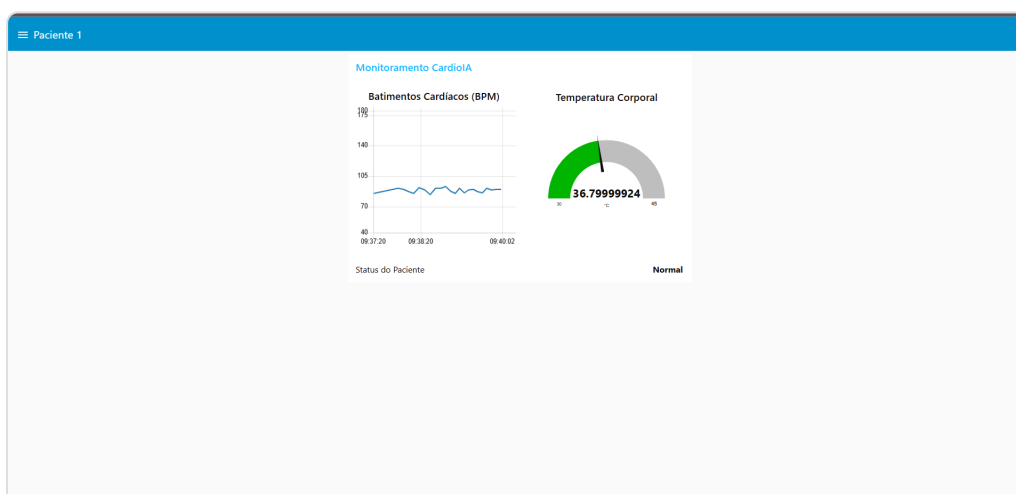


Figura – Dashboard (BPM, Temperatura e Status)

Como executar

1. Windows: execute `start_services.bat` (compila firmware e abre o Node-RED).
2. Wokwi (VS Code): clique **Play** para carregar `build/tmp_sketch.ino.elf`.
3. Node-RED: Import > selecione `node-red/cardioia_dashboard.flow.json` > Deploy.
4. Acesse `http://127.0.0.1:1880/ui` ; gire o potenciômetro para ver o BPM variar.

Critérios atendidos

- MQTT em nuvem (HiveMQ) com JSON: **OK**

- Dashboard com gráfico (BPM), gauge (Temp) e alerta (status): **OK**
- Fog/Edge: SPIFFS + reenvio de backlog: **OK**

Gerado automaticamente a partir do projeto em `cap1_cardioia_fase3` .