

Relatório Parte 2 – MQTT, Dashboard e Inteligência Operacional

Visão Geral

A segunda fase do projeto CardiolA mostra a integração entre Edge, Fog e Cloud Computing, formando um ecossistema completo de monitoramento cardíaco. A partir do firmware desenvolvido na Parte 1, foram implementados a transmissão via MQTT, a construção de dashboards no Node-RED e opções de análise histórica com Grafana (github.com).

Arquitetura de Comunicação

Camadas

- **Dispositivo Edge (ESP32)** – Continua capturando sinais vitais e lida com resiliência offline usando SPIFFS. Publica dados com QoS 1 no tópico `cardioia/v1/pacientes/<id>/vitals` e consome alertas no tópico `cardioia/v1/pacientes/<id>/alerts` (github.com).
- **Broker MQTT (Fog/Cloud)** – Utiliza HiveMQ Cloud com TLS ativado. Atua como camada intermediária desacoplando produtores (wearables) e consumidores (dashboards e serviços de alerta). A política QoS 1 garante confirmação do recebimento (github.com).
- **Node-RED (Fog/Edge Server)** – Trata dados em tempo real, gera alertas, alimenta interfaces e reencaminha notificações. Pode residir em gateway hospitalar ou servidor na nuvem (github.com).
- **Grafana + Banco de Séries Temporais (Cloud)** – Armazena histórico para análises de tendências e relatórios clínicos. O painel importável assume InfluxDB como backend (github.com).

Topologia de Tópicos

- `cardioia/v1/pacientes/<id>/vitals` – stream primária de sinais vitais (github.com).
- `cardioia/v1/pacientes/<id>/alerts` – canal para alertas emitidos pelo Node-RED, permitindo feedback ao dispositivo (github.com).
- Possível expansão para `cardioia/v1/pacientes/<id>/diagnostics` (resultados de IA) e `cardioia/v1/ops/logs` (observabilidade) (github.com).

Mensagem Padrão

A mensagem JSON publicada inclui campos como `deviceId`, `patientId`, `timestamp`, `temperature`, `humidity`, `heartRate`, `movement` e `battery` (github.com), compatível com serialização JSON Lines no Edge e pronta para ingestão por múltiplos consumidores.

Firmware com MQTT

O firmware `wokwi/src/main.cpp` foi estendido para suportar conectividade segura. As credenciais são isoladas em `secrets.h` e ausentes do controle de versão. O dispositivo só tenta conectar-se quando a chave deslizante indica disponibilidade. A conexão TLS usa `WiFiClientSecure` com `setInsecure()` (ambiente acadêmico), podendo ser substituída por CA real. O cliente MQTT implementa reconexão automática, publica cada linha do buffer e assina os alertas. Logs prefixados (INFO, WARN, ERR, EDGE, SYNC) facilitam troubleshooting e auditoria clínica.

Node-RED Dashboard

Estrutura do Flow

O arquivo `node-red/flow-cardioia.json` pode ser importado diretamente no Node-RED. O fluxo compreende:

- **MQTT In (Vital Signs)** – recebe dados do broker; permite configurar host, porta, usuário e senha (github.com).
- **JSON Parser** – converte payload textual em objeto JSON (github.com).
- **Função “Avaliar Alerta”** – aplica limites (120 bpm e 38 °C) e carimba o estado de alerta (github.com).
- **Dashboard Widgets** – `ui_chart` (batimentos por minuto), `ui_gauge` (temperatura), `ui_led` (indicador visual) e `ui_table` (log consolidado) (github.com).
- **MQTT Out (Publicar Alerta)** – envia eventos críticos ao tópico de alertas com retenção habilitada para que novos clientes recebam o último estado (github.com).

Deploy e Customizações

É necessário instalar Node-RED ($\geq 3.x$) e o pacote `node-red-dashboard`. Importe o fluxo (Menu → Import → Clipboard), edite os nós MQTT para apontar ao broker do projeto (TLS, usuário e senha) e publique o fluxo. O dashboard pode ser acessado via `http://<host>:1880/ui` (github.com). Os limites de alerta podem ser ajustados conforme protocolos médicos (github.com).

Estratégias de Alerta

- **Semáforo** – uso de LED verde/vermelho em destaque (github.com).
- **Retenção** – mensagens retidas garantem que alarmes persistam até revisão manual (github.com).

- **Integração futura** – o fluxo pode acionar webhooks, enviar e-mails ou integrar com sistemas de chamados hospitalares (github.com).

Grafana e Histórico de Dados

Pipeline Sugerido

Adicione um nó `influxdb out` no Node-RED para persistir cada mensagem em InfluxDB. Em seguida, importe `grafana/dashboard-cardioia.json` e mapeie a fonte de dados (DS_CARDIOIA). Os painéis incluem: série temporal de batimentos agrupados por paciente, gauge de temperatura com thresholds clínicos e indicador de bateria (github.com).

Benefícios Operacionais

A solução permite visualização 24h/7d das tendências, identificação de pacientes com variações anormais e suporte a auditorias e estudos clínicos (github.com).

Segurança & LGPD

- **Criptografia em trânsito** – TLS obrigatório no HiveMQ Cloud e recomendado para qualquer broker em produção (github.com).
- **Autenticação e autorização** – cada dispositivo possui credenciais exclusivas e ACLs restritivas (github.com).
- **Segregação de ambientes** – separar tópicos de teste e produção e implementar quotas para evitar DDoS acidental (github.com).
- **Monitoramento de logs** – registrar conexões e falhas para auditoria (github.com).
- **Conformidade LGPD** – dados são sintéticos, mas a arquitetura adota boas práticas de minimização, pseudonimização e rastreabilidade (github.com).

Validação e Testes

Os seguintes cenários foram exercitados (github.com):

- **Operação offline prolongada** – chave Wi-Fi desligada por 30 min com ~360 amostras; ao reativar, o buffer é sincronizado e limpo.
- **Limite de armazenamento** – injeção de > 5 000 registros para validar a rotação do buffer e preservação de dados recentes.
- **Alertas** – simulação de febre (39 °C) resultou em LED vermelho local e publicação de alerta no Node-RED.
- **Broker indisponível** – mensagens permanecem no SPIFFS e são reenviadas após reconexão.
- **Ferramentas de apoio** – o script `scripts/replay_mqtt.py` envia dados gravados para o broker; monitor serial e MQTT Explorer auxiliam na inspeção dos tópicos (github.com).

Próximos Passos

- **Edge Analytics** – aplicar filtros FIR/Kalman e detecção de arritmia local para reduzir volume de dados e gerar insights no vestível (github.com).
- **Integração com IA** – encaminhar dados para modelos preditivos e emitir alertas preditivos (github.com).
- **Eventos Bidirecionais** – permitir que médicos ajustem limites em tempo real via tópico de comandos (github.com).
- **Compliance Avançado** – integrar logs imutáveis (blockchain privado) para rastreabilidade de alertas críticos (github.com).
- **Escalabilidade** – estudar arquiteturas com Kubernetes + HiveMQ Cluster ou AWS IoT Core para elasticidade a milhares de pacientes (github.com).

Conclusão

A Parte 2 conclui a entrega desta fase do CardiolA demonstrando um pipeline completo de captura, processamento local, transmissão MQTT, visualização e alertas. A solução alinha práticas de engenharia de software, IoT médica e governança de dados, mantendo foco em confiabilidade e observabilidade. Esta base prepara o projeto para evoluir para módulos de previsão com IA, diagnóstico assistido e orquestração de respostas clínicas em larga escala (github.com).

Evidências Visuais





