SISTEMA IOT PARA MONITORAMENTO DE FREQUÊNCIA CARDÍACA COM ESP32, AD8232 E SUPABASE

João Pedro dos Santos Martins, João Victor de Campos, Hélio Lourenço Esperidião Ferreira.

Universidade do Vale do Paraíba/Curso de Engenharia da Computação, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil, johnmartines.sm@gmail.com, helioesperidiao@gmail.com.

Resumo

Este trabalho apresenta o desenvolvimento de um sistema de Internet das Coisas para aquisição, processamento, armazenamento e visualização de frequência cardíaca. O protótipo combina o sensor AD8232, utilizado no condicionamento do sinal eletrocardiográfico, com um ESP32 responsável pela aquisição, pelo processamento e pela comunicação Wi-Fi. O firmware foi implementado em C com ESP-IDF e FreeRTOS e utiliza filtragem por mediana, estimativa da linha de base, limiar adaptativo, validação temporal dos pulsos e cálculo da frequência cardíaca a partir dos intervalos RR. Os valores são enviados por HTTPS ao Supabase, que realiza a autenticação dos usuários e o armazenamento em PostgreSQL, e posteriormente são apresentados em uma aplicação React. Os testes funcionais confirmaram a leitura contínua do sensor, a indicação do estado dos eletrodos, a comunicação com o servidor e a atualização do banco. Em oito observações estáveis, a diferença absoluta média entre o valor bruto e o suavizado do ADC foi de 12,6 unidades. O conjunto mostrou-se adequado como protótipo acadêmico de integração entre hardware e software, mas ainda depende de comparação com equipamento de referência antes que seja possível avaliar a precisão da frequência cardíaca estimada.
Palavras-chave: Internet das Coisas. Frequência cardíaca. ESP32. AD8232. Supabase.
Área do Conhecimento: Engenharias IV - Engenharia Elétrica e Engenharia da Computação.

Introdução

Tecnologias digitais voltadas à saúde vêm ganhando espaço à medida que cresce a preocupação com prevenção, acompanhamento e qualidade de vida. A Organização Mundial da Saúde reconhece a saúde digital como uma área com potencial para apoiar sistemas de saúde e ampliar o acesso à informação, desde que a tecnologia seja utilizada com objetivos claros e responsabilidade (World Health Organization, 2025). Esse cenário também acompanha a expansão da Internet das Coisas: a IoT Analytics estimou 21,1 bilhões de dispositivos IoT conectados ao final de 2025, evidenciando a presença crescente de sensores e sistemas embarcados conectados em diferentes setores (Sinha, 2025).

Nesse contexto, um sistema simples de monitoramento pode ser útil como plataforma de estudo por reunir, em um único projeto, aquisição de sinais, processamento embarcado, comunicação, banco de dados e interface web. Na área de saúde, porém, conectar o dispositivo à internet não é o principal desafio. A qualidade do sinal, a privacidade, a segurança e a interpretação correta dos dados continuam sendo pontos essenciais. Por isso, protótipos acadêmicos de baixo custo devem ser tratados como ferramentas de engenharia e aprendizagem, e não como dispositivos médicos validados (Abdulmalek et al., 2022).

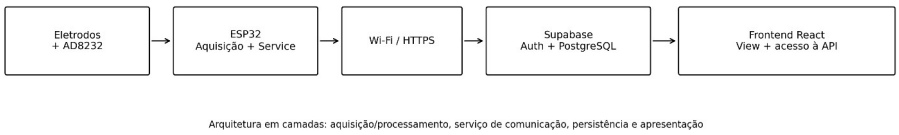
A frequência cardíaca pode ser estimada a partir dos intervalos entre complexos QRS de um sinal eletrocardiográfico. A identificação automática desses complexos é afetada por ruído muscular, movimentação dos eletrodos, interferência da rede elétrica e oscilações da linha de base. Pan e Tompkins (1985) demonstraram que a combinação de filtragem digital, análise de amplitude e critérios temporais aumenta a confiabilidade da detecção, princípios que continuam relevantes em implementações embarcadas com recursos computacionais limitados.

Diante desse cenário, o objetivo deste trabalho foi desenvolver um protótipo acadêmico de monitoramento de frequência cardíaca integrando AD8232, ESP32, comunicação em rede, banco de dados em nuvem e uma interface web responsiva. A proposta foi integrar essas partes em um único sistema e, ao mesmo tempo, manter as responsabilidades suficientemente separadas para facilitar testes, manutenção e futuras melhorias.

Metodologia

O sistema foi organizado em quatro responsabilidades principais: aquisição e processamento embarcado, comunicação, persistência e apresentação. O ESP32 concentra a leitura do sensor e o processamento do sinal; a comunicação HTTPS faz a ligação com o serviço remoto; o Supabase reúne autenticação, API e persistência em PostgreSQL; e o frontend React apresenta as informações ao usuário. Embora não corresponda a um MVC clássico executado em uma única aplicação, a separação segue o mesmo princípio de evitar que interface, processamento e acesso aos dados fiquem misturados. A Figura 1 apresenta esse fluxo.

Figura 1 - Arquitetura em camadas do sistema de monitoramento de BPM.



Arquitetura em camadas: aquisição/processamento, serviço de comunicação, persistência e apresentação

Fonte: elaborado pelos autores (2026).

O AD8232 foi alimentado em 3,3 V e sua saída analógica foi conectada ao GPIO34 do ESP32, correspondente ao ADC1_CHANNEL_6. As saídas LO+ e LO- foram ligadas aos GPIO19 e GPIO18 para indicar a condição dos eletrodos. Antes de detalhar o firmware, a Figura 2 mostra essas conexões de forma simplificada e ajuda a relacionar o circuito físico com os pinos efetivamente utilizados pelo programa.

Figura 2 - Diagrama elétrico simplificado entre AD8232 e ESP32.

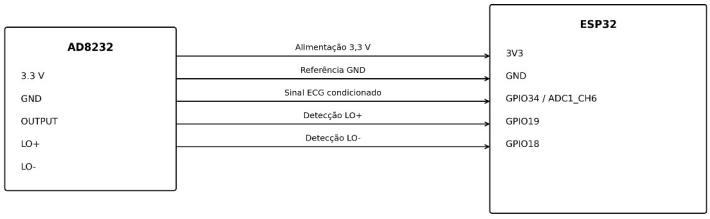


Diagrama elétrico simplificado das conexões explicitamente descritas no firmware e no artigo

Fonte: elaborado pelos autores (2026).

O firmware foi escrito em C no ambiente PlatformIO com o framework ESP-IDF. A amostragem foi executada periodicamente em uma tarefa do FreeRTOS, enquanto o envio HTTPS foi colocado em outra tarefa, interligada por uma fila. Essa separação evitou que o tempo de conexão com o serviço remoto interrompesse a leitura do sensor. O programa também foi configurado com uma lista de redes conhecidas e realiza tentativa de reconexão em caso de queda.

O processamento foi estruturado em etapas para reduzir falsos picos sem tornar o algoritmo excessivamente pesado para o microcontrolador. São utilizadas a mediana de amostras consecutivas, uma suavização rápida, uma estimativa mais lenta da linha de base e a amplitude relativa do sinal. O limiar é atualizado dinamicamente, e um evento somente é aceito como pulso quando atende aos critérios de amplitude, duração e intervalo temporal. Depois de dois pulsos válidos consecutivos, o intervalo RR é utilizado para estimar a frequência cardíaca pela Equação 1.

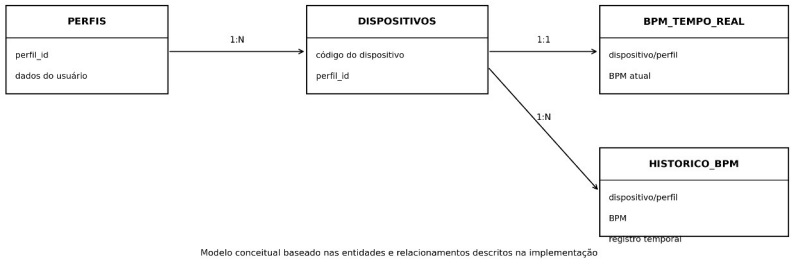
$$\mathbf{BPM = 60.000 / RR} \quad (1)$$

na qual RR é o intervalo entre batimentos consecutivos em milissegundos. Os intervalos válidos são armazenados em vetor circular e sua mediana é utilizada para reduzir a influência de valores isolados. Alterações abruptas também são limitadas para diminuir oscilações causadas por falsos picos.

No backend, o Supabase foi utilizado para autenticação, API e persistência. As entidades principais são perfis, dispositivos, bpm_tempo_real e historico_bpm. A função RPC registrar_bpm_dispositivo

recebe a identificação do dispositivo, o BPM calculado e a indicação de gravação histórica. O valor mais recente é mantido em bpm_tempo_real e os registros selecionados são acrescentados a historico_bpm. As políticas de Row Level Security restringem as consultas aos dados relacionados ao usuário autenticado. A Figura 3 apresenta esse modelo de forma conceitual, permitindo visualizar como perfil, dispositivo, leitura atual e histórico se relacionam.

Figura 3 - Diagrama conceitual das entidades utilizadas no Supabase.



Fonte: elaborado pelos autores a partir da implementação do sistema (2026).

O frontend foi desenvolvido em React com Vite e supabase-js. A aplicação possui cadastro, login, controle de sessão, perfil, Dashboard e histórico. O Dashboard apresenta o BPM atual, a média das leituras do dia, a quantidade de registros, um gráfico de variação e medições recentes. O filtro diário utiliza o horário local do navegador, enquanto registros anteriores permanecem disponíveis no histórico.

Tabela 1 - Componentes e funções do protótipo.

Componente	Tecnologia	Função
Aquisição	AD8232	Condicionamento do sinal eletrocardiográfico.
Processamento	ESP32 / ESP-IDF	Leitura do ADC, filtragem e cálculo do BPM.
Comunicação	Wi-Fi / HTTPS	Envio dos valores para a função RPC.
Backend	Supabase / PostgreSQL	Autenticação, tempo real e histórico.
Interface	React / Vite	Dashboard, gráficos, histórico e perfil.

Fonte: elaborado pelos autores (2026).

A avaliação foi conduzida como ensaio funcional de engenharia. Foram observados os valores brutos e suavizados do ADC, os indicadores LO+ e LO-, os registros do terminal, as respostas HTTP e a atualização das tabelas e da interface. Não foi realizada avaliação clínica, diagnóstico, comparação estatística com equipamento médico ou coleta estruturada de dados de participantes. Uma futura validação envolvendo seres humanos deverá ser submetida previamente ao Comitê de Ética em Pesquisa.

Resultados

O teste isolado do sensor confirmou a aquisição contínua pelo GPIO34. Em uma sequência estável de oito observações, o ADC bruto variou entre 1783 e 2167, com média de 1962,3 unidades. A série suavizada variou entre 1790 e 2170, com média de 1964,9 unidades. A diferença absoluta média entre o valor bruto e o suavizado foi de 12,6 unidades. Em todas as observações, LO+ e LO- permaneceram em nível lógico zero, indicando eletrodos conectados.

Tabela 2 - Resultados preliminares da aquisição do AD8232 (n = 8).

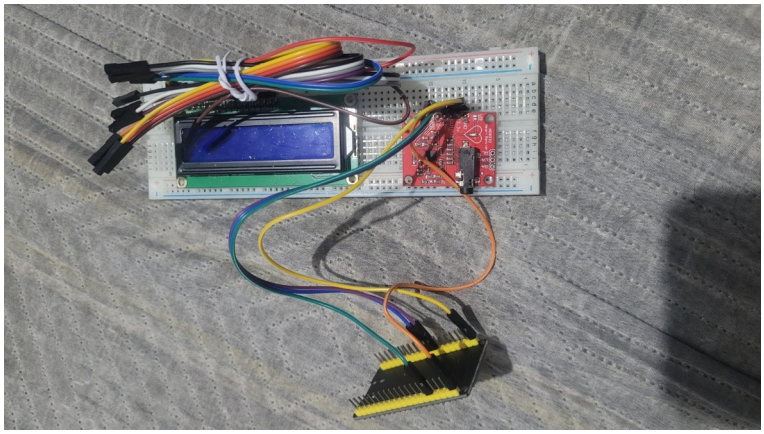
Métrica	Mínimo	Média	Máximo
ADC bruto (unidades)	1783	1962,3	2167
ADC suavizado (unidades)	1790	1964,9	2170
Diferença absoluta bruto-suavizado	1	12,6	31

Fonte: registros experimentais do protótipo (2026).

Nos testes de comunicação, o ESP32 conectou-se às redes cadastradas e permaneceu na primeira rede disponível. A função RPC retornou respostas HTTP de sucesso, com status 204, e os valores foram atualizados em bpm_tempo_real. Quando habilitada a opção de histórico, os registros também foram adicionados em historico_bpm. A execução do envio em tarefa separada permitiu manter a rotina de aquisição ativa durante a comunicação com o servidor.

A Figura 4 apresenta a montagem física utilizada durante o desenvolvimento. O protótipo foi montado em protoboard e reúne o módulo AD8232, a unidade de processamento e as conexões empregadas nos testes. A fotografia permite relacionar o diagrama elétrico e a arquitetura descritos anteriormente com a montagem real de bancada. Nesta etapa, o objetivo foi verificar a integração funcional entre aquisição, processamento e comunicação, e não produzir uma versão definitiva do hardware.

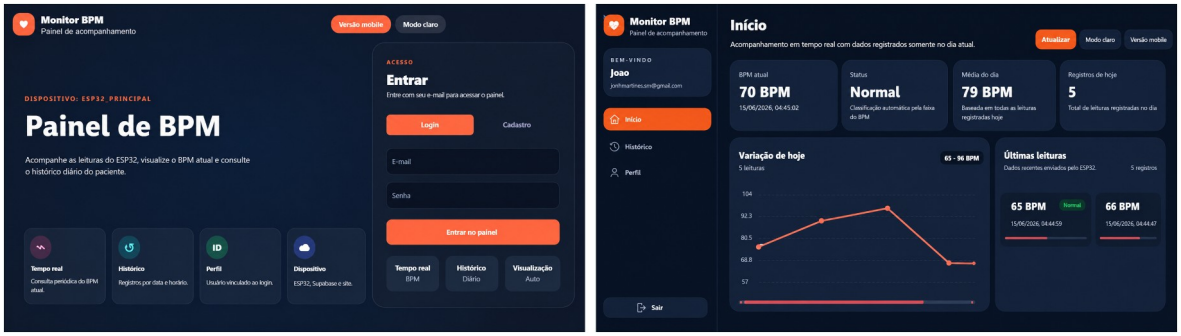
Figura 4 - Protótipo físico utilizado durante o desenvolvimento e os testes funcionais.



Fonte: acervo dos autores (2026).

A aplicação web apresentou autenticação por e-mail, carregamento do perfil, atualização periódica do BPM e navegação entre Dashboard, histórico e perfil. O filtro por data restringiu a média e o gráfico ao dia corrente, enquanto a página de histórico manteve acesso aos registros anteriores. A interface foi testada nos modos desktop e mobile, com adaptação dos gráficos ao espaço disponível. A Figura 5 apresenta a tela de autenticação e o painel principal durante a execução do sistema. A indicação de faixa de BPM exibida na interface é apenas um recurso visual do protótipo e não possui finalidade diagnóstica.

Figura 5 - Tela de autenticação (à esquerda) e Dashboard de acompanhamento (à direita).



Fonte: acervo dos autores (2026).

Nas primeiras versões, o detector classificava alguns picos como novos batimentos, fazendo o BPM variar mais do que o esperado. O algoritmo foi ajustado progressivamente com calibração, mediana, limiar adaptativo, validação temporal, rejeição de pulsos secundários e análise dos intervalos RR. Nos testes funcionais, essas mudanças reduziram oscilações extremas e aumentaram a estabilidade da leitura, embora essa melhora ainda precise ser quantificada em comparação com uma referência.

Discussão

Os testes indicaram que a etapa mais sensível do protótipo não é o envio dos dados para a nuvem, mas a identificação dos eventos do sinal analógico que correspondem a batimentos. O AD8232 auxilia no condicionamento de biopotenciais de pequena amplitude, porém o próprio fabricante prevê que filtros e circuito de aplicação precisam ser ajustados conforme o uso (Analog Devices, 2020). Assim, uma leitura sem saturação não significa automaticamente um ECG adequado: contato dos eletrodos, movimento e ruído eletromagnético ainda podem alterar a forma observada pelo ESP32.

A pequena diferença entre os valores brutos e suavizados é um sinal positivo de que a mediana não deslocou de forma importante o nível do ADC. Mesmo assim, essa análise é limitada. Ela não demonstra que todos os pulsos contados correspondem a complexos QRS reais. Como discutido por Pan e Tompkins (1985), a detecção precisa combinar características de amplitude e tempo; por isso, o limiar adaptativo e o período refratário são úteis, mas ainda precisam ser avaliados com um sinal de referência.

A divisão em camadas também se mostrou útil para a continuidade do projeto. A fila entre aquisição e HTTP evita que atrasos de rede interrompam a leitura do sensor, enquanto o backend mantém autenticação, persistência e regras de acesso fora da interface. A separação entre leitura atual e histórico deixa o banco preparado para funções posteriores, como exportação, alertas e associação de mais dispositivos. Em sistemas digitais de saúde, esse crescimento precisa continuar acompanhado por cuidados de segurança, privacidade e qualidade dos dados (World Health Organization, 2025).

As principais limitações ainda são técnicas: o protótipo trabalha com um único canal, não foi calibrado contra instrumento certificado, depende da qualidade do contato dos eletrodos e foi avaliado apenas por ensaios funcionais. Por esse motivo, o sistema deve continuar sendo tratado como plataforma acadêmica de Engenharia da Computação. A prioridade para a próxima etapa é validar o que já está sendo medido, registrando simultaneamente o sinal bruto e uma referência para calcular erro absoluto e relativo, sensibilidade de detecção e ocorrência de falsos positivos.

Depois dessa validação, o projeto pode avançar em duas frentes. No hardware, uma versão mais organizada pode substituir a montagem de protoboard por uma placa dedicada ou PCB, com melhor fixação das conexões e encapsulamento. No software, podem ser adicionados indicador de qualidade do sinal, armazenamento temporário quando a rede estiver indisponível, exportação do histórico e ferramentas de análise do sinal bruto. Caso futuras avaliações envolvam participantes de forma estruturada, o protocolo deverá ser submetido previamente ao Comitê de Ética em Pesquisa.

Tabela 3 - Etapas propostas para continuidade do sistema IoT.

Etapa	Ação principal	Objetivo
1. Validação	Registrar ECG bruto e referência simultaneamente	Quantificar erro e desempenho da detecção
2. Robustez do sinal	Testar contato, movimento e diferentes condições de aquisição	Definir limites e melhorar a rejeição de artefatos
3. Hardware	Migrar a protoboard para montagem dedicada	Aumentar estabilidade mecânica e reprodutibilidade
4. Software	Adicionar qualidade do sinal, buffer offline e exportação	Tornar o protótipo mais útil para estudo e análise

Fonte: elaborado pelos autores (2026).

Conclusão

Foi desenvolvido um protótipo IoT capaz de adquirir o sinal condicionado pelo AD8232, processá-lo no ESP32, calcular uma estimativa de frequência cardíaca, enviar os dados ao Supabase e apresentá-los em uma aplicação web responsiva. Os testes confirmaram o funcionamento da leitura do ADC, da detecção do estado dos eletrodos, da conexão Wi-Fi, do envio HTTPS, da atualização do banco de dados e da exibição do histórico.

A filtragem digital e a validação dos intervalos RR aumentaram a estabilidade em relação às primeiras versões, mas os resultados ainda não permitem atribuir precisão clínica ao sistema. O trabalho cumpriu seu objetivo como integração acadêmica entre eletrônica, sistemas embarcados, redes, banco de dados e desenvolvimento web. A continuidade deverá começar pela validação contra equipamento de referência e pelo registro do sinal bruto; somente depois dessa etapa será adequado consolidar o hardware e ampliar os recursos de interface, histórico e análise.

Referências

ABDULMALEK, S.; NASIR, A.; JABBAR, W. A.; ALMUHAYA, M. A. M.; BAIRAGI, A. K.; KHAN, M. A.-M.; KEE, S.-H. IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. Healthcare, v. 10, n. 10, art. 1993, 2022. Disponível em: https://www.mdpi.com/2227-9032/10/10/1993. Acesso em: 8 jun. 2026.

ANALOG DEVICES. AD8232: Single-Lead, Heart Rate Monitor Front End. Rev. D. Norwood: Analog Devices, 2020. Disponível em: https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf. Acesso em: 8 jun. 2026.

ESPRESSIF SYSTEMS. Analog to Digital Converter (ADC) - ESP32. ESP-IDF Programming Guide, 2026. Disponível em: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/adc.html. Acesso em: 8 jun. 2026.

PAN, J.; TOMPKINS, W. J. A real-time QRS detection algorithm. IEEE Transactions on Biomedical Engineering, v. BME-32, n. 3, p. 230-236, 1985. DOI: 10.1109/TBME.1985.325532.

REACT. React documentation: Quick Start. 2026. Disponível em: https://react.dev/learn. Acesso em: 8 jun. 2026.

SINHA, S. State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally. IoT Analytics, 28 out. 2025. Disponível em: https://iot-analytics.com/number-connected-iot-devices/. Acesso em: 24 ago. 2026.

SUPABASE. Auth and Row Level Security documentation. 2026. Disponível em: https://supabase.com/docs/guides/auth e https://supabase.com/docs/guides/database/postgres/row-level-security. Acesso em: 8 jun. 2026.

WORLD HEALTH ORGANIZATION. Global strategy on digital health 2020-2027. Geneva: WHO, 2025. Disponível em: https://www.who.int/publications/i/item/9789240116870. Acesso em: 24 ago. 2026.