SISTEMA IOT PARA DETECÇÃO DE QUEDAS E IMOBILIDADE COM ESP32 E MONITORAMENTO WEB EM TEMPO REAL

Rafael Ryan Ramos de Souza, Hélio Lourenço Esperidião Ferreira

Universidade do Vale do Paraíba, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil, souzaramos2001@gmail.com, helioesperidiao@gmail.com

Resumo

Este trabalho apresenta o desenvolvimento e a validação de bancada de um sistema de Internet das Coisas (IoT) para monitoramento de quedas, imobilidade e telemetria de movimento. A solução integra um ESP32, uma unidade de medição inercial de seis eixos da família MPU, firmware embarcado em C++/Arduino, comunicação MQTT, backend em Node.js/Express, banco de dados MySQL, Socket.IO e interface web em React. A detecção é executada localmente por uma máquina de estados que combina impacto, mudança de orientação e imobilidade, enquanto o backend registra eventos e associa evidências de telemetria. A validação documentou um dispositivo real conectado, gráfico de telemetria com 120 amostras, portal local operacional, registro de queda confirmada com imobilidade e aprovação das verificações automatizadas do software. Os resultados demonstram a integração ponta a ponta do protótipo acadêmico, sem constituir validação clínica da detecção de quedas.

Palavras-chave: Internet das Coisas. Detecção de quedas. ESP32. Sensores inerciais. Telemetria.

Área do Conhecimento: Engenharias / Engenharia de Computação.

Introdução

As quedas constituem um problema relevante de saúde pública. A Organização Mundial da Saúde estima 684 mil mortes anuais por quedas no mundo e aponta que adultos com mais de 60 anos concentram o maior número de quedas fatais (World Health Organization, 2021). Nesse contexto, soluções de monitoramento automático podem apoiar a identificação rápida de situações potencialmente críticas, desde que suas limitações sejam explicitadas e sua operação seja validada de forma compatível com o uso pretendido.

Sensores inerciais são recorrentes em estudos de detecção e avaliação de risco de quedas. Em revisão sistemática de 29 estudos, Bet, Castro e Ponti (2019) identificaram o acelerômetro como o sensor mais empregado, frequentemente em taxas de 50 ou 100 Hz, além do uso complementar de giroscópios e outros sensores. Os autores também apontam heterogeneidade metodológica e ausência de consenso em aspectos de aquisição e validação, o que reforça a necessidade de descrever claramente o protocolo, os parâmetros e as limitações de cada sistema.

A combinação de diferentes informações inerciais pode reduzir ambiguidades presentes em decisões baseadas em um único limiar. Wang et al. (2018) avaliaram uma abordagem baseada em acelerômetro e giroscópio e destacaram a utilidade de múltiplos parâmetros para diferenciar quedas de atividades de vida diária. No presente projeto, esse princípio foi traduzido em uma máquina de estados determinística que exige, em sequência, impacto, mudança de orientação e imobilidade antes da confirmação local da queda.

O objetivo deste trabalho foi desenvolver e verificar, em bancada, a integração ponta a ponta de um sistema IoT para detecção de quedas e imobilidade, contemplando aquisição inercial no ESP32, decisão local, transmissão por MQTT, persistência dos dados e atualização de uma interface web em tempo real.

Metodologia

O trabalho foi conduzido em bancada, sem ensaios com seres humanos ou animais. O procedimento de demonstração restringiu os testes físicos da caixa/sensor ao uso sobre superfície macia.

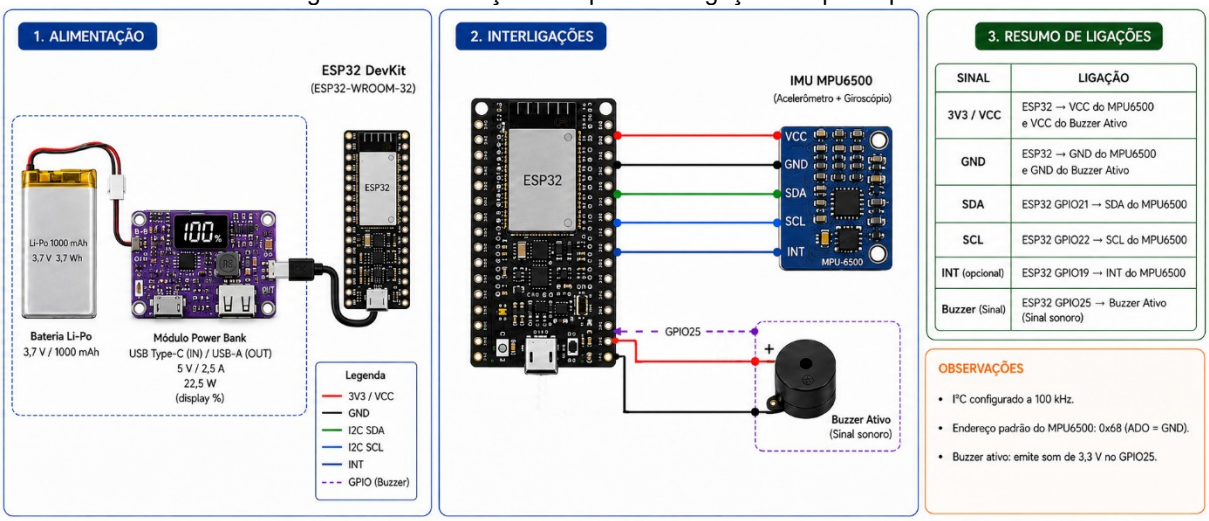
A camada embarcada utilizou uma placa ESP32 programada pelo PlatformIO com framework Arduino. O ESP32 oferece conectividade Wi-Fi e interfaces digitais, incluindo I2C, adequadas à

integração de nós sensores (Espressif Systems, [s.d.]). O firmware foi preparado para reconhecer unidades inerciais da família MPU; no protótipo ensaiado, o módulo foi identificado como MPU6500, uma unidade de seis eixos com acelerômetro e giroscópio (TDK InvenSense, 2020).

A comunicação entre ESP32 e sensor foi configurada em I2C a 100 kHz. O firmware solicita faixa de ±8 g para o acelerômetro e ±500 °/s para o giroscópio e utiliza a configuração efetivamente aceita na conversão dos dados. O protótipo foi alimentado por um módulo power bank USB-C com display digital; a porcentagem exibida podia ser consultada localmente, mas não era lida eletronicamente pelo firmware. Status, telemetria e eventos são transmitidos por Wi-Fi utilizando MQTT.

A Figura 1 apresenta a alimentação por USB-C, a interligação I2C entre ESP32 e IMU e o sinal de acionamento do buzzer pelo GPIO 25.

Figura 1 - Alimentação e esquema de ligações do protótipo.



Fonte: elaborado pelos autores (2026).

O detector de queda foi implementado como máquina de estados no firmware. No estado de monitoramento, um candidato somente é aberto quando a aceleração resultante e a magnitude do giroscópio ultrapassam simultaneamente seus limiares. Em seguida, a orientação atual é comparada com uma referência de postura aprendida durante períodos estáveis. Se a variação angular requerida ocorrer dentro da janela prevista, o algoritmo passa a verificar imobilidade. A queda é confirmada apenas quando a aceleração permanece próxima de 1 g, com tolerância de ±0,15 g, e a magnitude do giroscópio permanece em até 15 °/s pelo tempo mínimo configurado. Ausência de mudança de orientação ou de imobilidade dentro das respectivas janelas faz o candidato ser descartado.

Tabela 1 - Parâmetros dos perfis Normal e Demo utilizados nos ensaios.

Parâmetro	Normal	Demo
Intervalo de leitura do sensor	50 ms	25 ms
Intervalo de telemetria MQTT	2000 ms	500 ms
Limiar de aceleração de impacto	2,2 g	1,7 g
Limiar de giro de impacto	120 °/s	100 °/s
Mudança mínima de orientação	45°	30°
Janela para mudança de orientação	1500 ms	2000 ms
Imobilidade requerida	2000 ms	1000 ms
Janela máxima de imobilidade	4000 ms	5000 ms

Fonte: elaborado pelos autores (2026), a partir dos parâmetros do firmware utilizado nos ensaios.

Foram definidos dois perfis de operação. O perfil Demo foi mantido para apresentação acadêmica controlada, enquanto o perfil Normal utiliza limiares mais conservadores. A decisão principal permaneceu determinística; a extração experimental de características no domínio do tempo estava habilitada, porém a FFT permaneceu experimental e desabilitada como mecanismo de decisão.

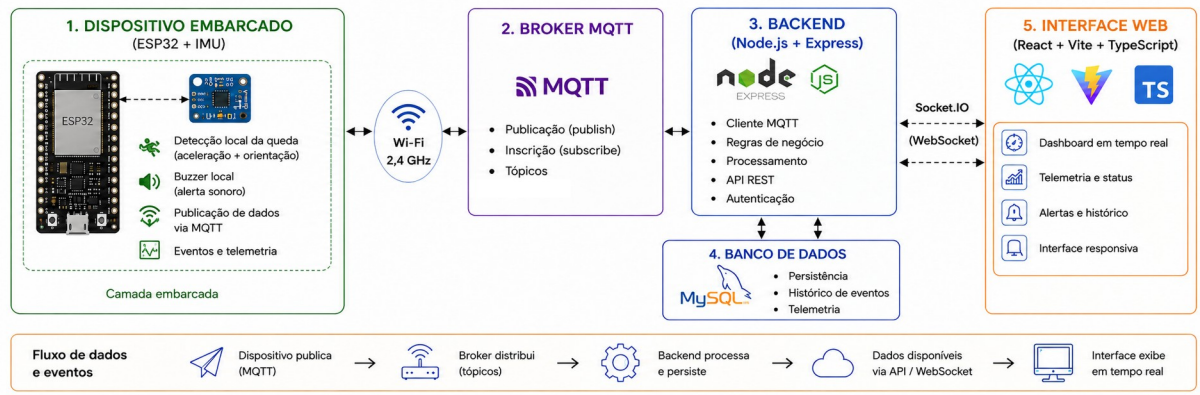
A integração utiliza o padrão publish/subscribe do MQTT. A especificação MQTT 3.1.1 define clientes que publicam e recebem mensagens por intermédio de um servidor, com baixo overhead de transporte (OASIS, 2014). No projeto, o ESP32 publica status, telemetria e eventos. Eventos críticos possuem identificador próprio e mecanismo de reenvio após indisponibilidade de comunicação, enquanto o backend evita registros duplicados.

O backend foi implementado em Node.js com Express, MySQL e integração MQTT. A interface web foi implementada em React com Vite e TypeScript, recebendo atualizações do servidor por Socket.IO. O acesso web utiliza autenticação JWT e escopo por organização, enquanto a identidade técnica do dispositivo é tratada separadamente no fluxo MQTT. Para quedas detectadas, o backend associa ao evento a telemetria temporalmente relacionada antes de registrar o alerta interno. O protótipo avaliado não realizava envio externo de notificações.

A validação final reuniu evidências de operação com ESP32 real, MQTT, backend, banco de dados, Socket.IO e frontend em conjunto. Foram registradas capturas do dispositivo online, do portal local de manutenção, da telemetria e de um evento de queda controlada, além da execução de verificações automatizadas de backend, integração, fluxo MQTT, lint e build do frontend. Como não foi executado um protocolo com conjunto amplo de quedas e atividades cotidianas, não foram calculadas sensibilidade, especificidade, acurácia ou taxa de falsos positivos.

A Figura 2 apresenta a arquitetura geral do sistema e o fluxo de dados entre a camada embarcada e a aplicação web. O ESP32 adquire os dados da IMU por I2C, executa localmente a lógica de detecção e publica status, telemetria e eventos por Wi-Fi utilizando MQTT. O broker encaminha as mensagens ao backend, responsável pelas regras da aplicação e pela persistência em MySQL; a interface web consome a API e recebe atualizações em tempo real por Socket.IO.

Figura 2 - Arquitetura geral do sistema IoT e fluxo de dados.



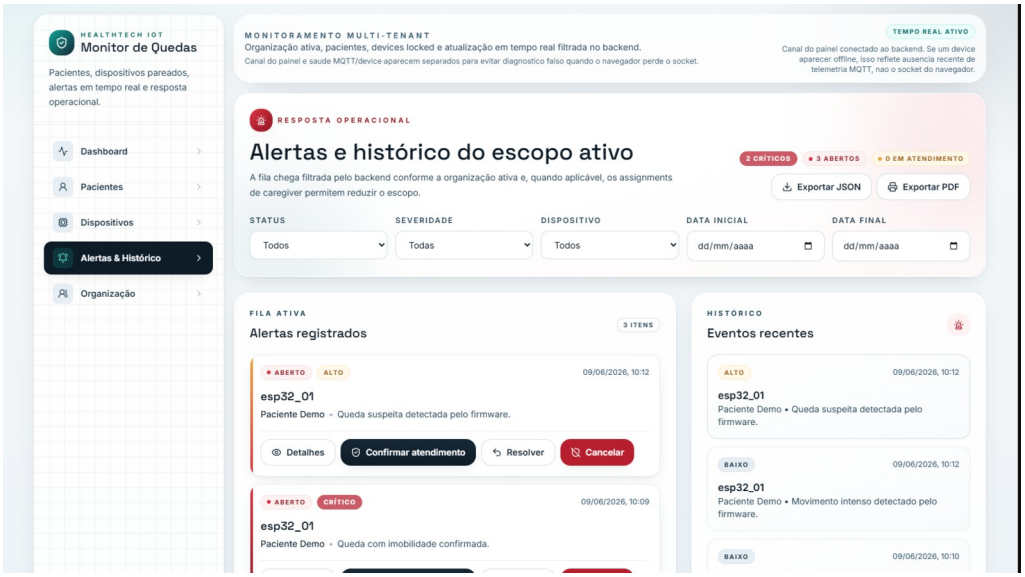
Fonte: elaborado pelos autores (2026), com base na implementação do protótipo.

Resultados

A validação consolidou a comunicação ponta a ponta do protótipo. O ESP32 operou conectado ao broker MQTT, publicou telemetria e status e manteve o portal local de manutenção disponível. No modo Demo foi registrada uma visualização real de 120 amostras de telemetria no frontend. O gráfico apresenta a aceleração resultante e disponibiliza, no detalhamento técnico, a magnitude do giroscópio e os eixos do sensor. Essas amostras são persistidas no banco e atualizadas em tempo real no navegador por Socket.IO.

No ensaio controlado documentado para a versão final, a sequência de impacto, mudança de orientação e imobilidade gerou uma queda confirmada pelo firmware. O evento chegou ao backend por MQTT, foi persistido e apareceu na tela de alertas com a confirmação de imobilidade, como mostra a Figura 3. O fluxo preserva a decisão local do dispositivo e adiciona, no servidor, a associação de telemetria que funciona como evidência técnica do evento. Esse resultado demonstra a integração entre aquisição, decisão embarcada, mensageria, persistência e apresentação, mas não representa uma medida de desempenho clínico do detector.

Figura 3 - Registro de queda com imobilidade confirmada na interface web.



Fonte: acervo do projeto (2026).

Tabela 2 - Evidências e verificações da validação de bancada.

Verificação	Resultado documentado
Dispositivo real conectado	ESP32 online com comunicação MQTT operacional
Telemetria no frontend	Gráfico com 120 amostras atualizado em tempo real
Evento controlado	Queda confirmada com imobilidade registrada no painel
Portal local	Interface de manutenção do ESP32 operacional
Verificações de software	Suítes de backend, integração e MQTT, além de lint e build do frontend, aprovadas

Fonte: elaborado pelos autores (2026), a partir das evidências de validação do protótipo.

Em um ensaio de bancada de aproximadamente 75 s, o sistema manteve aquisição válida do sensor e conexão MQTT, com aceleração resultante de 1,00 g em repouso e sem erros I2C ou recuperações no intervalo. Esse registro foi utilizado como evidência de estabilidade do caminho de aquisição naquele cenário específico, sem ser tratado como ensaio estatístico ou garantia de ausência de falhas em uso prolongado.

A montagem possuía um módulo power bank com display digital para consulta local da carga, conforme a montagem física apresentada na Figura 4. A versão final do firmware não realizava medição elétrica automática dessa porcentagem; o acompanhamento permanecia apenas visual no próprio módulo.

Figura 4 - Montagem física do protótipo utilizada nos ensaios de bancada.



Fonte: acervo do projeto (2026).

Discussão

Os resultados obtidos sustentam a viabilidade técnica da arquitetura como protótipo acadêmico integrado. A decisão de queda é executada no próprio ESP32, o que evita depender do backend para classificar o evento no instante da ocorrência. O servidor assume funções diferentes: persistir, relacionar evidências, controlar o acesso e atualizar o painel. Essa separação é coerente com a arquitetura em camadas adotada durante a disciplina e evita transformar o frontend em autoridade sobre uma regra crítica do dispositivo.

A lógica baseada em múltiplas etapas também é compatível com a motivação observada na literatura. Wang et al. (2018) mostram que a combinação de informações do acelerômetro e do giroscópio pode melhorar a discriminação em relação a métodos baseados em parâmetro único. No protótipo desenvolvido, aceleração e giro são exigidos simultaneamente no impacto e a confirmação ainda depende de mudança de orientação e imobilidade. A contribuição do trabalho, entretanto, está na implementação e integração dessa estratégia no sistema IoT, e não na demonstração de superioridade estatística do algoritmo.

Há uma diferença importante entre o protocolo atual e configurações frequentes encontradas na literatura. Bet, Castro e Ponti (2019) observaram 50 e 100 Hz como taxas recorrentes nos estudos revisados, enquanto o perfil Normal deste projeto realiza leitura a cada 50 ms, equivalente a 20 Hz, e o perfil Demo a cada 25 ms, equivalente a 40 Hz. Como não foi produzido um conjunto de dados suficientemente amplo para comparar taxas de amostragem, não é possível afirmar se esses valores são ótimos para detecção de quedas. Eles devem ser entendidos como parâmetros da implementação atual, sujeitos a validação experimental futura.

Outra limitação é a natureza controlada dos ensaios. O registro de uma queda confirmada, a telemetria de 120 amostras e a estabilidade observada em bancada demonstram que os componentes conseguem operar em conjunto, mas não permitem estimar sensibilidade, especificidade ou comportamento frente à diversidade de movimentos cotidianos de usuários reais. A própria revisão de Bet, Castro e Ponti (2019) ressalta a heterogeneidade de amostras e protocolos na área. Portanto, a interpretação adequada deste trabalho é de validação de integração e funcionamento do protótipo, e não de dispositivo médico validado.

Também permanecem limites operacionais claramente identificados: a notificação é interna ao dashboard, a bateria não é medida eletricamente, a decisão por FFT não está ativa e a comunicação segura por MQTT/TLS é configurável, mas não constitui a configuração padrão do protótipo. Essas limitações foram mantidas explícitas para evitar que funções previstas ou experimentais fossem apresentadas como recursos concluídos. Como continuidade do projeto, seriam necessários ensaios sistemáticos com maior variedade de movimentos, repetições documentadas, medição objetiva da bateria e validação das configurações de segurança antes de qualquer uso além do ambiente acadêmico.

Conclusão

Foi desenvolvido e integrado um sistema IoT acadêmico para detecção de quedas e imobilidade composto por ESP32, sensor inercial de seis eixos, decisão local por máquina de estados, comunicação MQTT, backend Node.js/Express, banco MySQL e interface web React com atualização por Socket.IO. O funcionamento ponta a ponta foi documentado com dispositivo real online, telemetria no painel, portal local operacional e registro de uma queda controlada confirmada com imobilidade. As verificações automatizadas do backend, da integração, do MQTT e do frontend também foram concluídas com sucesso.

Esses resultados permitem concluir que a arquitetura implementada atende ao objetivo de demonstrar a integração técnica entre sensor, firmware, comunicação, persistência e monitoramento web em um protótipo acadêmico. Não é possível, com os dados disponíveis, concluir eficácia clínica ou desempenho estatístico da detecção de quedas. A validação em cenários mais abrangentes e com métricas de desempenho permanece necessária antes de qualquer aplicação real envolvendo segurança de pessoas.

Referências

BET, P.; CASTRO, P. C.; PONTI, M. A. Fall detection and fall risk assessment in older person using wearable sensors: a systematic review. **International Journal of Medical Informatics**, v. 130, 103946, 2019. DOI: 10.1016/j.ijmedinf.2019.08.006. Disponível em: https://doi.org/10.1016/j.ijmedinf.2019.08.006. Acesso em: 24 ago. 2026.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**, v. 5.3. [S. l.], [s. d.]. Disponível em: https://www.espressif.com/documentation/esp32_datasheet_en.pdf. Acesso em: 24 ago. 2026.

OASIS. **MQTT Version 3.1.1**. Edited by Andrew Banks and Rahul Gupta. OASIS Standard, 29 out. 2014. Disponível em: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Acesso em: 24 ago. 2026.

TDK INVENSENSE. **MPU-6500 Product Specification**. Document Number PS-MPU-6500A-01, Revision 1.3, 15 maio 2020. Disponível em: https://invensense.tdk.com/wp-content/uploads/2020/06/PS-MPU-6500A-01-v1.3.pdf. Acesso em: 24 ago. 2026.

WANG, F.-T.; CHAN, H.-L.; HSU, M.-H.; LIN, C.-K.; CHAO, P.-K.; CHANG, Y.-J. Threshold-based fall detection using a hybrid of tri-axial accelerometer and gyroscope. **Physiological Measurement**, v. 39, n. 10, 105002, 2018. DOI: 10.1088/1361-6579/aae0eb. Disponível em: https://doi.org/10.1088/1361-6579/aae0eb. Acesso em: 24 ago. 2026.

WORLD HEALTH ORGANIZATION. **Falls**. Geneva: WHO, 26 abr. 2021. Disponível em: https://www.who.int/news-room/fact-sheets/detail/falls. Acesso em: 24 ago. 2026.