UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MATHEUS CRUZ DA SILVA
OMAR ZAGONEL EL LADEN

SISTEMA INTELIGENTE DE MONITORAMENTO PATRIMONIAL (SIMPAT)

OFICINA DE INTEGRAÇÃO 2 – RELATÓRIO FINAL

CURITIBA

2026

MATHEUS CRUZ DA SILVA
OMAR ZAGONEL EL LADEN

SISTEMA INTELIGENTE DE MONITORAMENTO PATRIMONIAL (SIMPAT)

Relatório Final da disciplina Oficina de Integração 2, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnologica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Dr. César Manuel Vargas Benítez
Prof. Daniel Rossato

CURITIBA

2026

Este trabalho é dedicado aos Professores Daniel Rossato e Benitez, pela orientação, incentivo e contribuição ao longo dessa Disciplina e do desenvolvimento deste projeto.

Dedico também à Vania, do departamento do DAGEE e responsável atual pela auditoria de patrimônio da UTFPR-Rebouças, cuja percepção e incentivo despertaram a ideia inicial que deu origem ao *SIMPAT*, permitindo transformar uma necessidade prática em uma proposta tecnológica aplicada ao monitoramento patrimonial.

AGRADECIMENTOS

Agradeço, primeiramente, aos Professores Daniel Rossato e Benitez, pela orientação, disponibilidade e apoio durante o desenvolvimento deste trabalho. Suas contribuições foram fundamentais para a definição do escopo, para o amadurecimento da proposta e para a condução técnica do projeto.

Agradeço à Vânia, do DAGEE, por ter despertado a ideia inicial que motivou este trabalho, ao identificar uma necessidade prática relacionada ao controle e ao monitoramento de patrimônios. Sua contribuição foi essencial para que o problema observado pudesse ser transformado em uma proposta acadêmica de solução tecnológica.

Agradeço também à Universidade Tecnológica Federal do Paraná, pelo acesso ao saber proporcionado e pelo ambiente acadêmico que favorece a integração entre teoria, prática, inovação e desenvolvimento tecnológico.

Estendo meus agradecimentos ao colega de equipe, pelo empenho, parceria e colaboração nas etapas de planejamento, discussão e desenvolvimento do protótipo. O trabalho conjunto foi importante para a construção de uma solução mais consistente e alinhada aos objetivos propostos.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste projeto, seja por meio de apoio, incentivo, orientação ou troca de conhecimentos ao longo desta trajetória.

“A tecnologia cumpre seu papel quando transforma uma necessidade real em uma solução útil, acessível e capaz de melhorar processos.”

RESUMO

. SISTEMA INTELIGENTE DE MONITORAMENTO PATRIMONIAL (SIMPAT) . 41 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2026.

Este trabalho apresenta o desenvolvimento do *SIMPAT* – Sistema Inteligente de Monitoramento Patrimonial, um protótipo voltado à identificação de movimentações suspeitas em ativos previamente vinculados a uma sala de referência. A solução utiliza um dispositivo embarcado baseado em ESP32, integrado a sensor de movimento, buzzer, alimentação por bateria e comunicação Wi-Fi. O sistema não tem como finalidade realizar geolocalização em tempo real, mas detectar indícios de deslocamento não autorizado por meio da repetição de eventos em uma janela curta de tempo. Em situações críticas, o dispositivo aciona alerta sonoro local e transmite ao servidor o identificador único do patrimônio. O servidor simplificado registra os eventos, exibe o estado operacional do ativo e permite a classificação administrativa das ocorrências. O protótipo busca validar a viabilidade técnica de uma solução ciberfísica de baixo custo, incremental e aplicável ao monitoramento patrimonial em ambientes institucionais.

Palavras-chave: monitoramento patrimonial, monitoramento, ESP32, sensor de vibração, sistema embarcado, Wi-Fi, rede

ABSTRACT

. INTELLIGENT SYSTEM FOR ASSET MONITORING. 41 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2026.

Este trabalho apresenta o *SIMPAT* – Sistema Inteligente de Monitoramento Patrimonial, um protótipo destinado à identificação de movimentações suspeitas em ativos vinculados a uma sala de referência. A solução utiliza um dispositivo embarcado com ESP32, sensor de movimento, buzzer, alimentação por bateria e comunicação Wi-Fi. O sistema não realiza geolocalização em tempo real, mas avalia eventos de movimentação recorrentes em uma janela curta de tempo. Quando uma condição crítica é identificada, o dispositivo aciona alerta sonoro local e envia ao servidor o identificador único do patrimônio. O servidor registra os eventos, apresenta o estado operacional do ativo e permite a classificação administrativa das ocorrências. O protótipo busca validar a viabilidade de uma solução ciberfísica simples, incremental e de baixo custo para apoio ao monitoramento patrimonial institucional.

Keywords: Asset monitoring, Monitoring, ESP32, Vibration sensor, Embedded system, Wi-Fi, Network

LISTA DE FIGURAS

FIGURA 1 – Diagrama de alto nível do sistema. 22
FIGURA 2 – Diagrama geral do sistema. 23
FIGURA 3 – Diagrama de blocos do hardware do protótipo. 24
FIGURA 4 – Diagrama de blocos de baixo nível do dispositivo de monitoramento patrimonial. 25
FIGURA 5 – Fluxograma geral de funcionamento do software. 26
FIGURA 6 – Tabelas do banco de dados. 28
FIGURA 7 – Tabela de alertas. 28

FIGURA 8 – Módulo sensor de vibração KY-002. 29
FIGURA 9 – Módulo sensor de vibração SW-420. 30
FIGURA 10 – Comparação entre duas placas com ESP32. 31
FIGURA 11 – Caixa do protótipo. 31
FIGURA 12 – Caixa fixada em um monitor da universidade. 32
FIGURA 13 – Circuito final. 32
FIGURA 14 – Visão inferior da placa. 33
FIGURA 15 – Página inicial do servidor. 33
FIGURA 16 – Tabela de próximos acordamentos. 33
FIGURA 17 – Tela de cadastro. 34
FIGURA 18 – Tela de atualização de cadastros. 34
FIGURA 19 – Caixas e placas avaliadas. 35

FIGURA 20 – Cronograma proposto. 36
FIGURA 21 – Cronograma final. 37
FIGURA 22 – Custos. 38

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO 13
1.1 MOTIVAÇÃO 13
1.2 OBJETIVOS 14
1.2.1 Objetivo geral 14
1.2.2 Objetivos específicos 14
2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 SISTEMAS EMBARCADOS 16
2.2 MICROCONTROLADOR ESP32-S3 17
2.3 MODOS DE BAIXO CONSUMO E DEEP SLEEP 17
2.4 SENSORES DE MOVIMENTO E DETECÇÃO DE EVENTOS 18
2.5 ALERTA LOCAL POR BUZZER 18
2.6 COMUNICAÇÃO WI-FI E ENVIO DE EVENTOS 19
2.7 SERVIDOR DE EVENTOS E RASTREABILIDADE PATRIMONIAL 19
2.8 INTERNET DAS COISAS APLICADA AO MONITORAMENTO PATRIMONIAL 20
2.9 CONSIDERAÇÕES SOBRE A FUNDAMENTAÇÃO 20
3 METODOLOGIA 22
3.1 VISÃO GERAL 22
3.2 PROJETO MECÂNICO 23
3.3 PROJETO DE *HARDWARE* 24
3.4 PROJETO DE *SOFTWARE* 25
3.5 INTEGRAÇÃO 27
4 EXPERIMENTOS E RESULTADOS 29
5 CRONOGRAMA E CUSTOS DO PROJETO 36
5.1 CRONOGRAMA 36
5.2 CUSTOS 38
6 CONCLUSÕES 39
6.1 CONCLUSÕES 39
6.2 TRABALHOS FUTUROS 39
REFERÊNCIAS 41

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Instituições de ensino superior dependem de uma infraestrutura composta por diversos ativos patrimoniais para manter suas atividades acadêmicas, administrativas e operacionais. Equipamentos como computadores, notebooks, projetores, monitores, televisores, caixas de som e outros recursos de apoio didático são frequentemente distribuídos entre salas, laboratórios e setores. Embora esses bens sejam essenciais para a rotina institucional, nem sempre há mecanismos capazes de acompanhar, de forma contínua, se cada patrimônio permanece no ambiente ao qual está vinculado.

A ausência de rastreabilidade contínua pode gerar problemas recorrentes, como remanejamento não registrado, dificuldade de auditoria, extravio de equipamentos e suspeitas de furto. Em muitos casos, a identificação de uma movimentação indevida ocorre apenas posteriormente, quando o bem já não se encontra em seu local esperado ou quando sua ausência compromete alguma atividade. Essa limitação evidencia a necessidade de soluções que auxiliem no monitoramento preventivo e no registro de eventos relacionados aos ativos institucionais.

Nesse contexto, surge a proposta do *SIMPAT* – Sistema Inteligente de Monitoramento Patrimonial, um protótipo funcional voltado ao acompanhamento de patrimônios por localização esperada. A premissa central do projeto não é realizar geolocalização precisa em tempo real, nem substituir sistemas institucionais completos de controle patrimonial, mas desenvolver uma solução embarcada capaz de identificar indícios de movimentação suspeita, emitir alerta local e registrar eventos em um servidor simplificado.

O sistema proposto utiliza um dispositivo embarcado acoplado ao patrimônio monitorado. Esse dispositivo é composto por um microcontrolador ESP32-S3, sensor de movimento, buzzer e alimentação por bateria. Em operação normal, o dispositivo permanece em modo de baixo consumo energético e realiza verificações periódicas. Quando o sensor detecta movimentação, o firmware avalia se o evento ocorreu de forma isolada ou recorrente em

uma janela curta de tempo. A repetição do movimento em intervalo aproximado de dois a três segundos caracteriza uma condição crítica, resultando no acionamento do buzzer e na tentativa de envio do identificador único do patrimônio ao servidor.

A motivação do projeto também está relacionada à possibilidade de integrar conhecimentos de eletrônica, sistemas embarcados, redes de computadores, desenvolvimento de software, banco de dados e engenharia de sistemas em uma solução ciberfísica. Dessa forma, o protótipo permite validar, em escala reduzida, uma arquitetura composta por dispositivo físico, comunicação sem fio, lógica de decisão embarcada, servidor de eventos e painel administrativo. Essa abordagem favorece a evolução futura do sistema para cenários mais completos, com múltiplos patrimônios, integração institucional, notificações automatizadas, monitoramento de bateria e mecanismos anti-remoção.

Assim, o desenvolvimento do *SIMPAT* justifica-se pela necessidade de uma solução simples, incremental e de baixo custo para apoiar o monitoramento patrimonial em ambientes institucionais, contribuindo para maior rastreabilidade, resposta mais rápida a eventos suspeitos e melhoria no controle operacional dos ativos.

1.2 OBJETIVOS

Esta seção apresenta o objetivo geral e os objetivos específicos do trabalho, delimitando o escopo de desenvolvimento do protótipo *SIMPAT* e as principais funcionalidades previstas para validação da solução proposta.

1.2.1 OBJETIVO GERAL

Desenvolver um protótipo embarcado de monitoramento patrimonial capaz de identificar movimentação suspeita de um patrimônio previamente vinculado a uma sala, emitir alerta sonoro local e registrar eventos remotos a partir de um identificador único do ativo monitorado.

1.2.2 OBJETIVOS ESPECÍFICOS

- Projetar a arquitetura embarcada do dispositivo de monitoramento patrimonial com foco em baixo consumo energético.
- Implementar o núcleo de controle do protótipo utilizando o microcontrolador ESP32-S3.

- Integrar sensor de movimento, buzzer, bateria e circuito de alimentação ao dispositivo embarcado.
- Implementar lógica de detecção de movimentação baseada na repetição de eventos em uma janela curta de confirmação.
- Diferenciar eventos isolados de movimentação e eventos recorrentes, classificando as ocorrências em níveis de severidade, como aviso e crítico.
- Acionar alerta sonoro local por meio de buzzer quando o sistema identificar condição crítica.
- Transmitir ao servidor, por comunicação Wi-Fi em ambiente de teste, o identificador único do patrimônio em eventos relevantes.
- Desenvolver um servidor simplificado capaz de manter cadastro do patrimônio, sala de referência, histórico de eventos e estado operacional do ativo.
- Disponibilizar uma interface administrativa básica para visualização dos eventos e classificação das ocorrências como falso positivo, movimentação autorizada ou ocorrência real.
- Validar o funcionamento integrado do protótipo, contemplando cenários de operação normal, evento isolado, evento recorrente, alerta crítico, envio ao servidor e falha momentânea de comunicação.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho aborda os principais conceitos relacionados ao desenvolvimento do *SIMPAT* – Sistema Inteligente de Monitoramento Patrimonial. O projeto envolve a integração entre sistemas embarcados, sensores, comunicação sem fio, modos de baixo consumo energético e registro de eventos em uma aplicação de software. Esses elementos são essenciais para compreender a proposta de um protótipo capaz de detectar movimentações suspeitas em ativos patrimoniais e registrar ocorrências associadas a um identificador único.

2.1 SISTEMAS EMBARCADOS

Sistemas embarcados são sistemas computacionais projetados para executar funções específicas dentro de um dispositivo ou equipamento. Diferentemente de computadores de uso geral, esses sistemas normalmente possuem recursos limitados de processamento, memória e energia, sendo desenvolvidos para operar de forma dedicada, eficiente e, muitas vezes, autônoma.

No contexto deste projeto, o sistema embarcado é responsável por monitorar fisicamente um patrimônio, processar sinais provenientes de um sensor de movimento, tomar decisões com base em regras previamente definidas e acionar respostas locais ou remotas. Dessa forma, o dispositivo embarcado atua como o núcleo operacional do protótipo, realizando a interface entre o ambiente físico e o sistema de software.

A utilização de sistemas embarcados em aplicações de monitoramento é adequada porque permite a construção de soluções compactas, de baixo custo e com capacidade de operação contínua. No caso do *SIMPAT*, essa característica é fundamental, pois o dispositivo deve permanecer acoplado ao patrimônio monitorado e operar de maneira autônoma, sem depender de infraestrutura física complexa.

2.2 MICROCONTROLADOR ESP32-S3

O ESP32 é um microcontrolador amplamente utilizado em aplicações de prototipagem, automação e Internet das Coisas. Sua adoção em projetos embarcados ocorre devido à integração de recursos relevantes em um único módulo, como processamento, comunicação Wi-Fi, suporte a entradas e saídas digitais, modos de economia de energia e compatibilidade com linguagens e ambientes de desenvolvimento acessíveis.

No protótipo proposto, o ESP32-S3 desempenha a função de unidade central de controle. Ele é responsável por receber os sinais do sensor de movimento, executar a lógica de classificação dos eventos, controlar o acionamento do buzzer e realizar a comunicação com o servidor por meio da rede Wi-Fi. Essa integração reduz a necessidade de componentes externos adicionais e simplifica a arquitetura física do dispositivo.

Além disso, o ESP32-S3 permite o uso de estratégias de baixo consumo energético, aspecto relevante para dispositivos alimentados por bateria. Como o monitoramento patrimonial não exige processamento contínuo em alta frequência, o microcontrolador pode permanecer em estado de repouso durante grande parte do tempo e ser despertado apenas em situações específicas, como eventos de movimento ou rotinas periódicas de verificação.

2.3 MODOS DE BAIXO CONSUMO E DEEP SLEEP

Em dispositivos alimentados por bateria, a eficiência energética é um requisito fundamental. O consumo contínuo de energia pode inviabilizar aplicações embarcadas que precisam permanecer em operação por longos períodos sem manutenção frequente. Por esse motivo, microcontroladores modernos oferecem modos de economia de energia, nos quais parte dos circuitos internos é desligada ou reduzida temporariamente.

O modo *Deep Sleep* do ESP32-S3 permite reduzir significativamente o consumo energético do dispositivo durante períodos de inatividade. Nesse estado, o microcontrolador mantém apenas os recursos essenciais para possibilitar o seu despertar por eventos previamente configurados, como temporizadores ou sinais externos. A documentação técnica da Espressif descreve os modos de economia de energia disponíveis para o ESP32-S3 e apresenta o *Deep Sleep* como uma alternativa adequada para aplicações que exigem baixo consumo e operação intermitente (Espressif,).

No *SIMPAT*, o uso do *Deep Sleep* está associado à necessidade de manter o dispositivo em funcionamento autônomo por meio de bateria. Durante a operação normal, o dispositivo

permanece em repouso, reduzindo o consumo energético. O despertar ocorre em situações relevantes, como a detecção de movimento pelo sensor ou a execução de uma rotina diária de verificação. Essa abordagem contribui para aumentar a autonomia do protótipo e tornar o sistema mais adequado ao monitoramento contínuo de patrimônio.

2.4 SENSORES DE MOVIMENTO E DETECÇÃO DE EVENTOS

Sensores de movimento ou vibração são dispositivos utilizados para identificar perturbações físicas em um objeto ou ambiente. Em aplicações de monitoramento patrimonial, esses sensores podem indicar que determinado bem foi tocado, deslocado ou submetido a uma movimentação inesperada. Entretanto, a simples detecção de um único evento não é suficiente para caracterizar uma ocorrência crítica, pois pequenas vibrações ou interações acidentais podem gerar falsos positivos.

Para reduzir esse problema, o *SIMPAT* adota uma lógica baseada na repetição de eventos em uma janela curta de tempo. Dessa forma, uma movimentação isolada pode ser interpretada como aviso, enquanto eventos recorrentes em intervalo aproximado de dois a três segundos são classificados como condição crítica. Essa estratégia busca diferenciar perturbações ocasionais de movimentações potencialmente suspeitas.

A lógica de detecção proposta não tem o objetivo de determinar a posição exata do patrimônio, mas sim identificar indícios de movimentação incompatíveis com o estado esperado do ativo. Assim, o sistema trabalha com o conceito de localização esperada: cada patrimônio é associado a uma sala de referência, e o dispositivo embarcado atua como mecanismo de alerta quando há sinais de movimentação não autorizada ou anormal.

2.5 ALERTA LOCAL POR BUZZER

O buzzer é um atuador sonoro utilizado para emitir alertas audíveis em sistemas eletrônicos. Em protótipos embarcados, seu uso é comum devido à simplicidade de acionamento, baixo custo e capacidade de fornecer resposta imediata ao usuário ou ao ambiente. No caso do *SIMPAT*, o buzzer funciona como uma camada de alerta local diante de uma condição crítica.

A presença do alerta sonoro é importante porque o sistema não deve depender exclusivamente da comunicação com o servidor. Em situações de indisponibilidade temporária da rede Wi-Fi, o dispositivo ainda pode sinalizar localmente que uma movimentação crítica foi

detectada. Essa redundância operacional aumenta a robustez do protótipo, pois garante algum tipo de resposta mesmo quando a transmissão remota não é concluída.

No fluxo proposto, o buzzer não é acionado em qualquer movimentação. Seu acionamento ocorre apenas quando a lógica embarcada identifica repetição de eventos dentro da janela de confirmação. Essa decisão reduz a possibilidade de alertas indevidos e torna o comportamento do sistema mais adequado ao ambiente institucional.

2.6 COMUNICAÇÃO WI-FI E ENVIO DE EVENTOS

A comunicação sem fio é um componente essencial em sistemas de monitoramento baseados em dispositivos embarcados. Por meio dela, o dispositivo físico consegue transmitir informações para uma aplicação externa, permitindo o registro, a visualização e a análise dos eventos detectados. No *SIMPAT*, a comunicação ocorre por Wi-Fi, utilizando uma rede de teste para validar o funcionamento do protótipo.

O envio de eventos ao servidor é realizado a partir do identificador único do patrimônio. Esse identificador permite associar a ocorrência recebida ao ativo cadastrado no sistema, mantendo a rastreabilidade das movimentações detectadas. Dessa forma, o servidor não depende apenas do evento em si, mas da vinculação entre o dispositivo embarcado, o patrimônio monitorado e sua sala de referência.

A escolha por uma rede de teste, como hotspot de celular, permite validar a comunicação do protótipo sem depender inicialmente da infraestrutura institucional. Essa decisão reduz riscos de implementação e concentra a validação na troca básica de informações entre o dispositivo e o sistema de software. Em versões futuras, a comunicação poderá ser adaptada para redes institucionais, com políticas de autenticação e segurança mais específicas.

2.7 SERVIDOR DE EVENTOS E RASTREABILIDADE PATRIMONIAL

O servidor de eventos representa a camada de software responsável por receber, registrar e disponibilizar as informações enviadas pelo dispositivo embarcado. No escopo do protótipo, essa camada não precisa possuir todos os recursos de um sistema patrimonial completo, mas deve preservar funcionalidades essenciais, como cadastro do patrimônio, vínculo com sala de referência, armazenamento de histórico e exibição do estado operacional do ativo.

A rastreabilidade patrimonial consiste na capacidade de acompanhar informações relevantes sobre determinado bem ao longo do tempo. No contexto do *SIMPAT*, essa

rastreabilidade é construída a partir do registro dos eventos de movimentação e da associação desses eventos ao identificador único do patrimônio. Com isso, torna-se possível consultar ocorrências anteriores, avaliar padrões de movimentação e classificar administrativamente os eventos detectados.

A classificação administrativa das ocorrências permite distinguir situações como falso positivo, movimentação autorizada ou ocorrência real. Esse recurso é importante porque a detecção automática indica apenas a existência de um evento físico relevante, cabendo à análise posterior determinar sua natureza. Dessa forma, o sistema combina automação embarcada com validação administrativa, tornando o processo mais confiável e adequado ao uso institucional.

2.8 INTERNET DAS COISAS APLICADA AO MONITORAMENTO PATRIMONIAL

A Internet das Coisas, ou *Internet of Things* (IoT), refere-se à integração de objetos físicos com sistemas computacionais por meio de sensores, atuadores, processamento embarcado e comunicação em rede. Essa abordagem permite que dispositivos físicos coletem dados do ambiente, processem informações e interajam com aplicações remotas.

O *SIMPAT* enquadra-se nesse contexto por integrar um patrimônio físico a um dispositivo embarcado capaz de detectar eventos e comunicar informações a um servidor. O ativo monitorado passa a possuir uma representação digital no sistema, permitindo acompanhar seu estado operacional e registrar ocorrências relacionadas à sua movimentação.

A aplicação de IoT ao monitoramento patrimonial contribui para transformar um processo tradicionalmente manual e reativo em um processo mais automatizado e preventivo. Embora o protótipo desenvolvido neste trabalho tenha escopo reduzido, sua arquitetura permite evolução para cenários mais amplos, como múltiplos patrimônios monitorados, notificações automatizadas, integração com sistemas institucionais e uso de sensores mais precisos.

2.9 CONSIDERAÇÕES SOBRE A FUNDAMENTAÇÃO

Os conceitos apresentados sustentam a proposta do *SIMPAT* como uma solução ciberfísica de monitoramento patrimonial. A combinação entre ESP32-S3, sensor de movimento, buzzer, modo *Deep Sleep*, comunicação Wi-Fi e servidor de eventos permite construir um protótipo funcional capaz de identificar movimentações suspeitas, emitir alerta local e registrar ocorrências remotas.

A fundamentação evidencia que o projeto não busca solucionar integralmente todos os

desafios de um sistema institucional de controle patrimonial, mas validar uma arquitetura inicial, incremental e tecnicamente viável. Essa abordagem é coerente com a natureza de protótipo acadêmico, pois prioriza a validação do núcleo funcional antes da incorporação de recursos avançados, como caixa física definitiva, anti-remoção da tag, medição de bateria, integração com rede institucional e expansão para múltiplos ativos.

3 METODOLOGIA

3.1 VISÃO GERAL

O desenvolvimento do *SIMPAT* foi conduzido com foco na construção de um protótipo funcional de monitoramento patrimonial por localização esperada. A proposta consiste em acoplar a um patrimônio um dispositivo embarcado capaz de detectar movimentações suspeitas, emitir alerta local e registrar eventos em um servidor simplificado.

O sistema proposto foi organizado em uma arquitetura de alto nível composta pelo patrimônio monitorado, dispositivo embarcado, comunicação Wi-Fi, servidor de eventos e painel administrativo. Essa organização permite separar as responsabilidades do protótipo em camadas, facilitando a validação individual e integrada de cada parte do sistema. 1

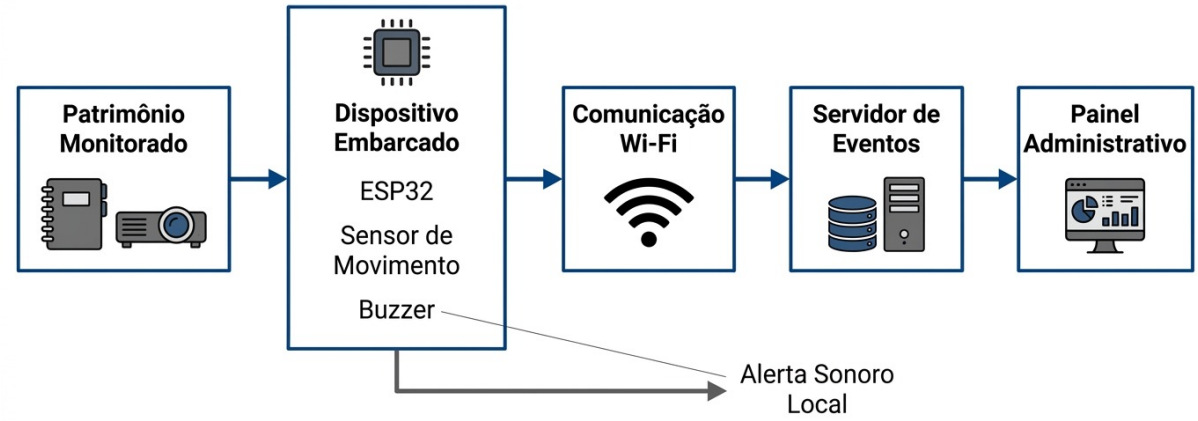


Figura 1: Diagrama de alto nível do sistema.

O sistema é composto por quatro camadas principais: dispositivo físico, firmware embarcado, comunicação sem fio e aplicação de software. O dispositivo realiza a detecção

do movimento, o firmware interpreta os eventos, a comunicação Wi-Fi transmite o identificador do patrimônio e o servidor registra as ocorrências para consulta e classificação administrativa. Uma visão geral pode ser vista na Figura 2.

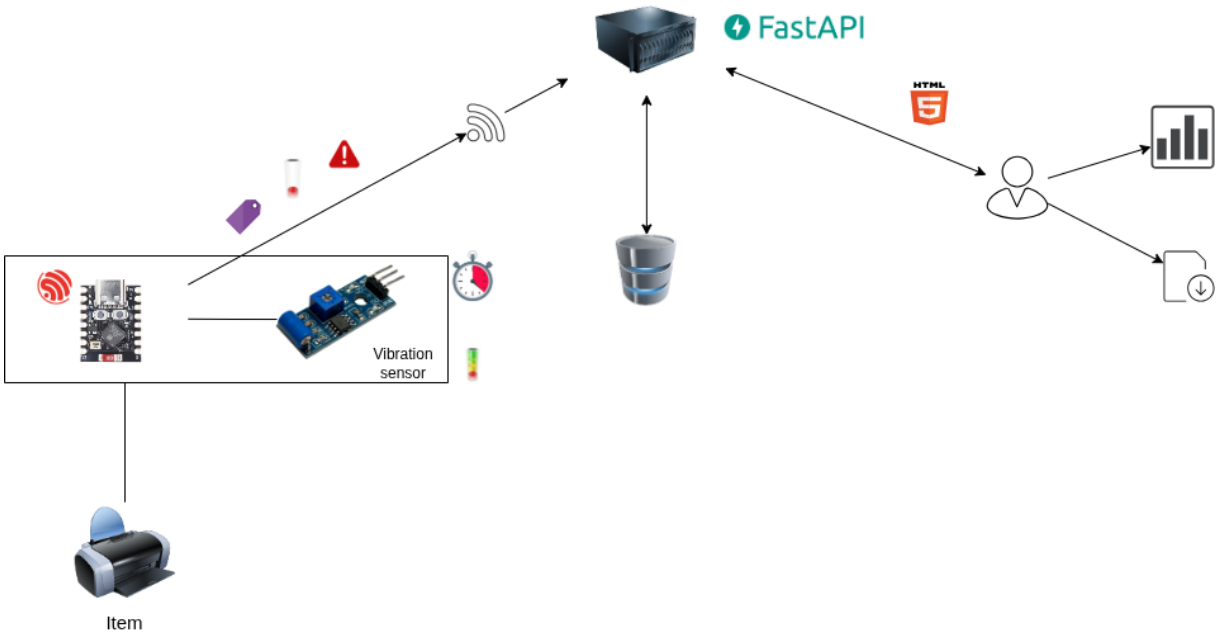


Figura 2: Diagrama geral do sistema.

O método adotado prioriza a validação do núcleo funcional do projeto antes da expansão para recursos futuros, como múltiplos patrimônios, integração com rede institucional, monitoramento de bateria e mecanismo anti-remoção.

3.2 PROJETO MECÂNICO

O projeto mecânico desta etapa foi limitado à montagem funcional do protótipo, sem desenvolvimento de caixa física definitiva. O dispositivo embarcado foi previsto para ser acoplado ao patrimônio monitorado de forma experimental, permitindo ajustes, testes e manutenção durante a fase de validação.

A estrutura física considera a acomodação mínima do ESP32, sensor de movimento, buzzer, bateria e circuito de alimentação. Por se tratar de uma prova de conceito, a prioridade foi garantir acessibilidade aos componentes, facilidade de conexão e possibilidade de alteração rápida durante os ensaios.

Nesta fase, não foram implementados mecanismos anti-remoção, vedação, acabamento industrial ou fixação definitiva. Esses elementos foram tratados como melhorias

futuras, após a validação técnica da arquitetura proposta.

3.3 PROJETO DE *HARDWARE*

O hardware do protótipo foi projetado a partir de uma arquitetura simples e de baixo custo. O ESP32 atua como unidade central de controle, sendo responsável pela leitura do sensor, processamento da lógica de decisão, acionamento do buzzer e comunicação Wi-Fi com o servidor. A Figura 3 mostra o simples esquemático do projeto.

O sensor de movimento ou vibração é utilizado para identificar perturbações físicas no patrimônio. O buzzer funciona como atuador de alerta local em condições críticas. Já a bateria fornece alimentação autônoma ao dispositivo, enquanto o circuito de regulação garante níveis adequados de tensão para os componentes eletrônicos.

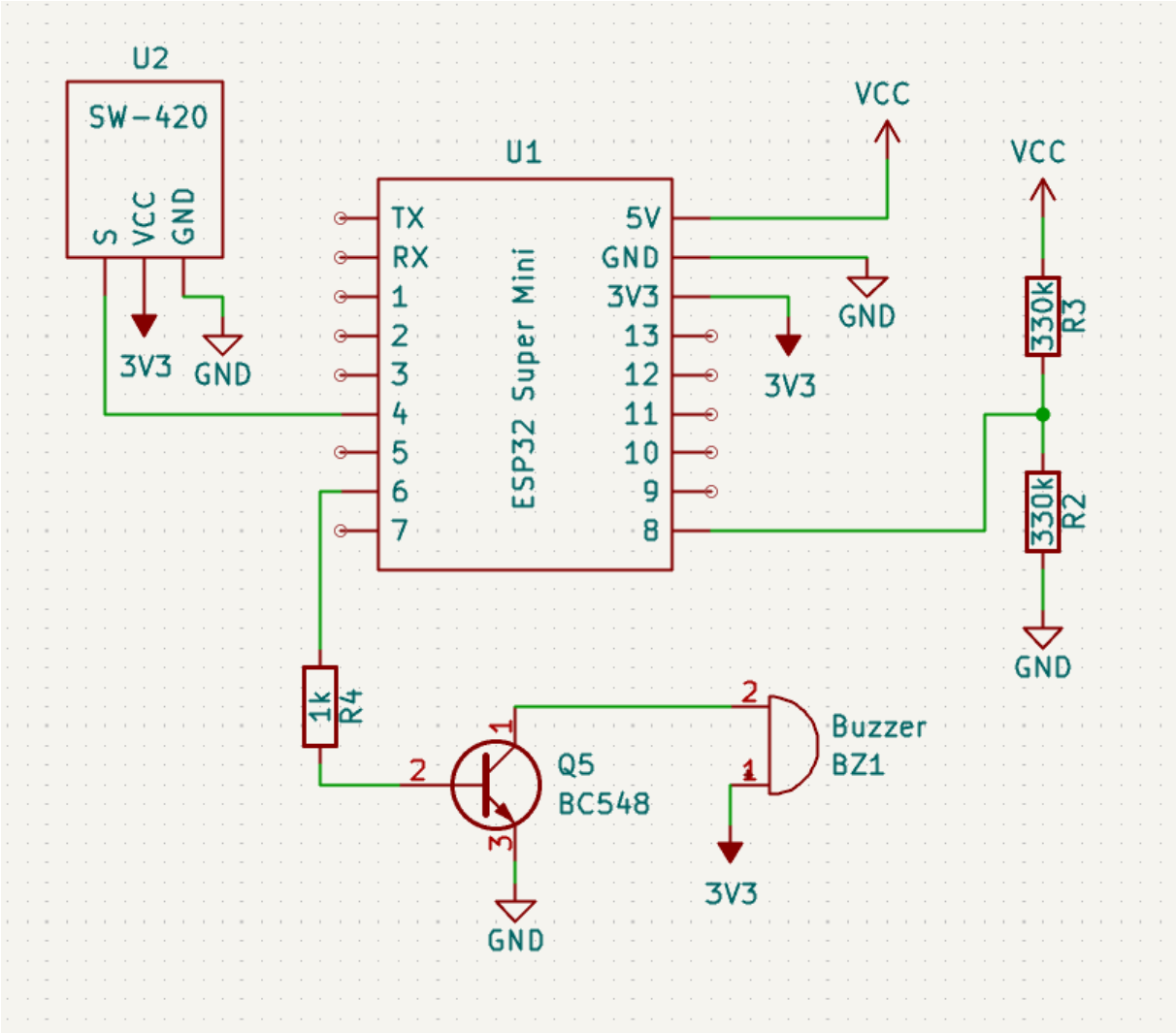


Figura 3: Diagrama de blocos do hardware do protótipo.

A arquitetura de baixo nível do dispositivo detalha os principais componentes físicos utilizados no protótipo. O ESP32 atua como núcleo de controle, recebendo sinais do sensor de movimento, acionando o buzzer em situações críticas, controlando o modo de baixo consumo e realizando a comunicação com o servidor por meio do módulo Wi-Fi.

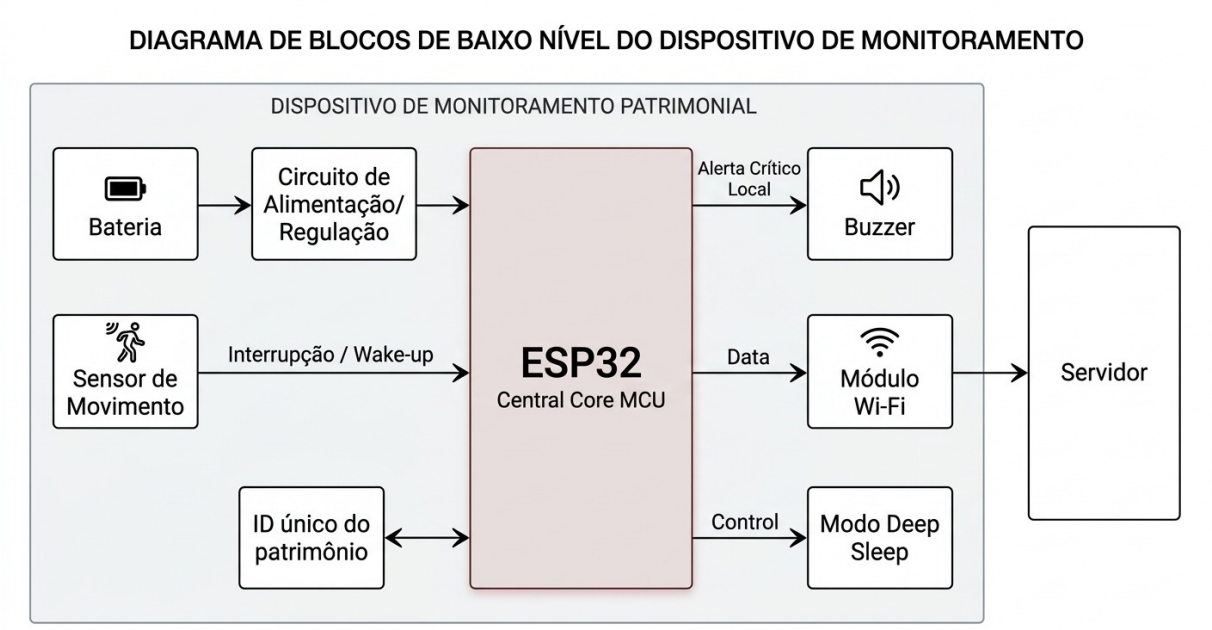


Figura 4: Diagrama de blocos de baixo nível do dispositivo de monitoramento patrimonial.

A arquitetura de hardware foi definida para permitir operação em baixo consumo, utilizando modos de repouso do ESP32 durante períodos de inatividade. Dessa forma, o dispositivo permanece ativo apenas quando ocorre evento de movimento ou rotina periódica de verificação.

3.4 PROJETO DE *SOFTWARE*

O software do *SIMPAT* foi dividido em duas partes: firmware embarcado e sistema servidor. O firmware, executado no ESP32, realiza a inicialização do dispositivo, leitura do sensor, tratamento dos eventos, acionamento do buzzer, comunicação Wi-Fi e envio do identificador único do patrimônio.

Conforme o fluxograma da Figura 5, lógica embarcada diferencia eventos isolados de movimentações recorrentes. Um único evento de vibração é tratado como aviso — apenas acordando o microcontrolador para rotinas básicas — enquanto a repetição de movimentos em uma janela curta de tempo caracteriza uma condição de risco. Nessa situação em que há múltiplos acordamentos em sequência, o buzzer é acionado e o dispositivo tenta transmitir o

evento ao servidor.

Além dos alertas críticos, o microcontrolador também realiza requisições quando é reiniciado e também deve diariamente enviar uma requisição. Esta requisição é planejada para que o servidor possa verificar se há algum item que ficou muito tempo sem se comunicar com ele — o que pode indicar um problema.

Caso não consiga se conectar à rede ou não receba resposta do servidor para os eventos de risco, o ESP32 guarda em sua memória que precisa tentar novamente realizar uma requisição e é programado para acordar após um curto período de tempo.

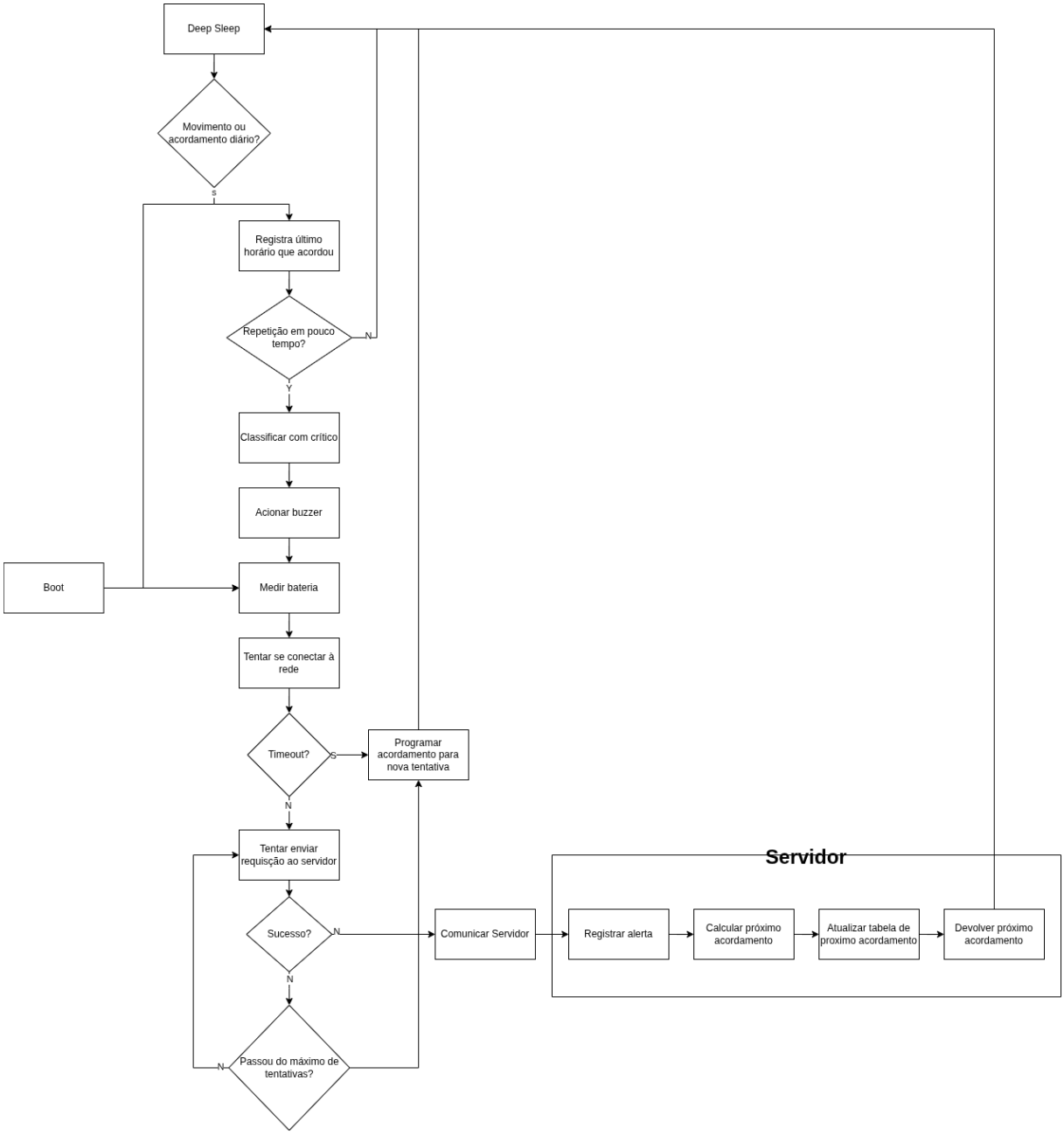


Figura 5: Fluxograma geral de funcionamento do software.

O servidor simplificado é responsável pelo cadastro do patrimônio, associação com a sala de referência, registro dos eventos recebidos, exibição do estado operacional e classificação das ocorrências.

Ele foi desenvolvido utilizando o framework moderno FastAPI de Python (FastAPI,). A escolha foi principalmente pela facilidade de uso e de programação assíncrona, assim como no interesse da equipe de aprender a utilizar esta ferramenta. Cada requisição do microcontrolador é enviada com dados em formato JSON, e o servidor responde com o tempo em segundos até o próximo acordamento. Este parâmetro é sorteado pelo servidor dentro de um intervalo de horas — de forma a ter menos chance de múltiplos itens acordarem ao mesmo tempo. Com isso, o ESP32 se programa para acordar novamente no instante determinado e os acontecimentos são registrados em uma tabela no banco de dados.

O banco de dados utilizado é o SQLite, escolhido por ser o mais simples para o protótipo. A Figura 6 mostra as três tabelas armazenadas. Pode se observar que a tabela de alertas armazena dados úteis como o identificador do item, a classificação do evento e o nível de bateria medido através de um divisor de tensão resistivo alimentado pela bateria. Além disso, outros dados importantes são a contagem total de boot, o número de reptições que geraram um alerta crítico, e o BSSID — que é o endereço MAC do Access Point ao qual o microcontrolador se conectou.

Um exemplo de registro de alertas é mostrado na Figura 7.

3.5 INTEGRAÇÃO

A integração do protótipo foi realizada pela conexão entre as camadas física, embarcada, comunicação e software. O sensor envia sinais ao ESP32, o firmware processa os eventos, o buzzer é acionado em condição crítica e o identificador do patrimônio é transmitido ao servidor por Wi-Fi.

Os testes de integração devem contemplar os principais cenários de funcionamento: operação normal, evento isolado, evento recorrente, acionamento do alerta sonoro, envio de evento ao servidor, registro no histórico e classificação administrativa da ocorrência.

Também deve ser verificado o comportamento do sistema em caso de falha momentânea de comunicação. Nesse cenário, o buzzer permanece como mecanismo de alerta local, garantindo resposta mínima mesmo quando o envio remoto não é concluído.

A validação do protótipo considera como critérios principais a detecção de movimento

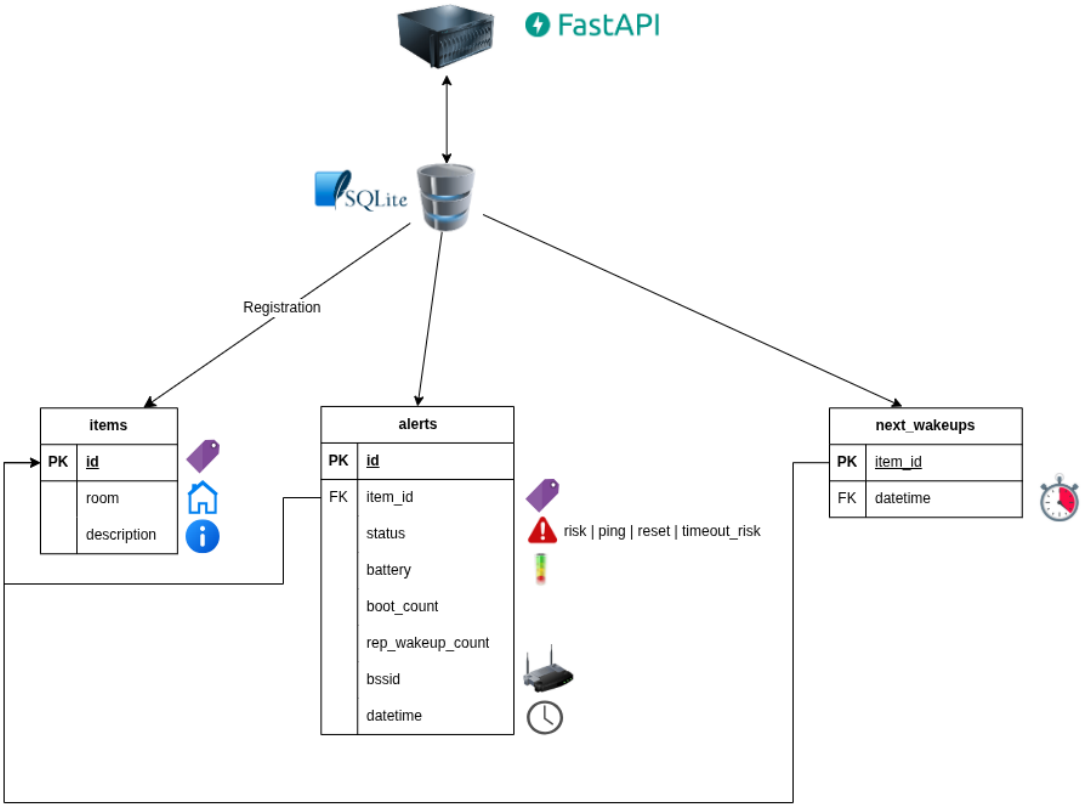


Figura 6: Tabelas do banco de dados.

Alertas

id	item_id	status	battery	boot_count	rep_wakeup_count	bssid	datetime	localtime
1	0	reset	2165	1	0	DA:68:A0:FE:30:F6	2026-06-24 14:29:38	2026-06-24 11:29:38
2	0	reset	2121	1	0	DA:68:A0:FE:30:F6	2026-06-24 14:30:04	2026-06-24 11:30:04
3	0	risk	2127	12	11	DA:68:A0:FE:30:F6	2026-06-24 14:30:43	2026-06-24 11:30:43
4	0	ping	2095	13	11	DA:68:A0:FE:30:F6	2026-06-24 14:31:45	2026-06-24 11:31:45
5	0	risk	2077	25	11	DA:68:A0:FE:30:F6	2026-06-24 14:49:30	2026-06-24 11:49:30
6	0	reset	2163	1	0	DA:68:A0:FE:30:F6	2026-06-24 14:49:33	2026-06-24 11:49:33
7	0	reset	1899	1	0	DA:68:A0:FE:30:F6	2026-06-24 14:49:45	2026-06-24 11:49:45
8	0	reset	2143	1	0	DA:68:A0:FE:30:F6	2026-06-24 14:50:20	2026-06-24 11:50:20

Figura 7: Tabela de alertas.

recorrente, a correta diferenciação entre aviso e crítico, o acionamento adequado do buzzer, o envio do identificador único, o registro do evento no servidor e a exibição do estado operacional do patrimônio.

4 EXPERIMENTOS E RESULTADOS

Esta etapa teve como objetivo validar o funcionamento do protótipo do *SIMPAT*, considerando a detecção de movimentação, o acionamento do alerta local, a comunicação com o servidor e o registro das ocorrências. Os experimentos foram conduzidos de forma incremental, partindo da avaliação dos sensores até a integração entre hardware, firmware e aplicação web.

Inicialmente, foram avaliadas alternativas de sensores para identificação de movimentação do patrimônio. O módulo KY-002 — mostrado na Figura 8 — foi utilizado nos primeiros testes por sua simplicidade de conexão e resposta direta a vibrações, mas ele se mostrou pouco sensível e sem customizações. Também foi considerado o sensor LIS3DH, um acelerômetro mais completo para leitura de movimento. Entretanto, durante os testes, sua integração apresentou dificuldades e instabilidades incompatíveis com o tempo e o escopo do protótipo. Por esse motivo, optou-se pelo uso do sensor SW-420 (Figura 9), que apresentou comportamento mais simples e adequado à validação da proposta, apesar de possuir um circuito que fica sempre ligado.

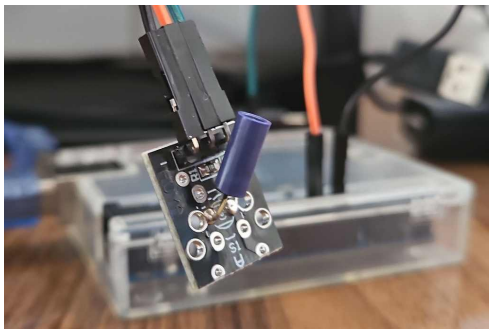


Figura 8: Módulo sensor de vibração KY-002.

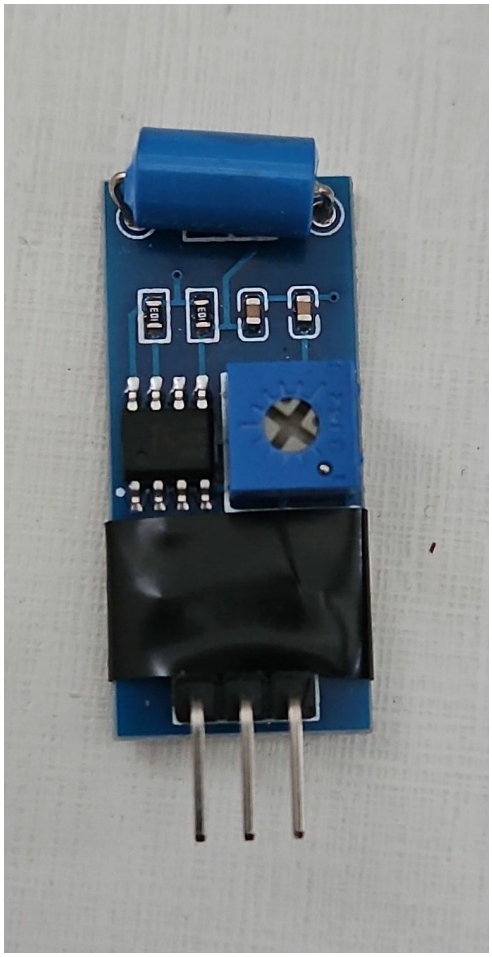


Figura 9: Módulo sensor de vibração SW-420.

Após a definição do sensor, foram realizados testes de bancada com o ESP32, buzzer e módulo de vibração. A montagem inicial em protoboard permitiu verificar a leitura dos sinais digitais do sensor, o tratamento do evento no firmware e a resposta do sistema por meio de mensagens de depuração na porta serial.

Também foram comparadas diferentes placas baseadas em ESP32, com o objetivo de avaliar dimensões físicas, facilidade de montagem e adequação ao protótipo. Essa comparação auxiliou na escolha da placa mais conveniente para a montagem final do dispositivo. As placas de teste pode ser vistas na Figura 10, em que é evidente a vantagem da placa ESP32 Super Mini na questão de tamnho em relação a uma placa de desenvolvimento convencional. Há ainda melhor de consumo de utilizando-se aquela por possuir um chip mais moderno e menos componentes externos a ele.

No lado do servidor, foram implementadas funcionalidades para registrar os alertas recebidos, armazenar os eventos e exibir as informações principais do patrimônio monitorado. A tabela de alertas permitiu validar o recebimento das ocorrências e acompanhar os estados

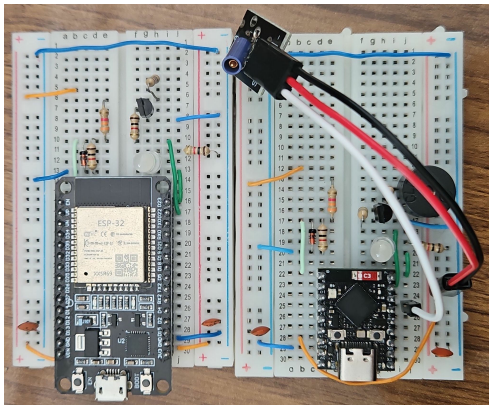


Figura 10: Comparação entre duas placas com ESP32.

gerados pelo dispositivo.

Com a evolução dos testes, foi montado o circuito final do protótipo, reunindo ESP32, sensor SW-420, buzzer e alimentação. A montagem foi organizada para permitir o acoplamento em uma caixa e a fixação experimental em um patrimônio real da universidade.

A caixa do protótipo, mostrada nas Figuras 11 e 12, consistiu em uma caixa de power bank reutilizada, utilizada como solução física inicial para acondicionar os componentes e possibilitar a fixação no patrimônio monitorado. Embora não represente uma versão final de produto, a caixa permitiu validar a viabilidade mecânica básica da proposta. Mesmo assim, pelo tamanho grande da bateria utilizada — uma CR18650 de lítio — a placa teve que ficar bem pequena. Isso dificultou bastante a soldagem, apesar da quantidade reduzida de componentes.



Figura 11: Caixa do protótipo.



Figura 12: Caixa fixada em um monitor da universidade.

Nas Figuras 13 e 14 é possível ver com mais detalhes da placa final desenvolvida.



Figura 13: Circuito final.

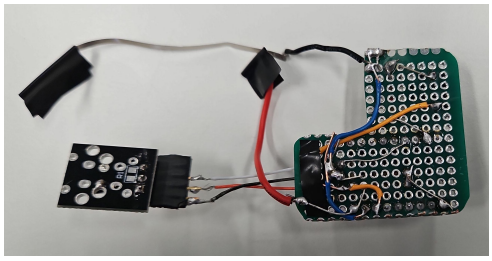


Figura 14: Visão inferior da placa.

A aplicação web foi estruturada com banco de dados para persistência dos cadastros, alertas e informações relacionadas aos dispositivos. A página inicial do servidor permitiu acompanhar o estado geral do sistema e acessar os registros necessários para análise, conforme a Figura 15. Esta e as outras páginas são bastante simples, utilizando apenas HTML para facilitar o desenvolvimento e manutenção do sistema, além de consumir menos recursos.



Figura 15: Página inicial do servidor.

Também foi implementado o controle de próximos acordamentos do dispositivo, relacionado à operação em modo de baixo consumo e às rotinas periódicas de verificação. Esse recurso contribui para o acompanhamento da comunicação entre o dispositivo e o servidor. Um exemplo de tabela com esses dados é mostrado na Figura 16.



Figura 16: Tabela de próximos acordamentos.

As telas de cadastro e atualização — mostradas nas Figuras 17 e 18 — permitem validar as operações administrativas básicas do sistema, incluindo o registro dos patrimônios e

a manutenção dos dados associados ao monitoramento. Não foram implementados mecanismos de exclusão pensando em questões de segurança.

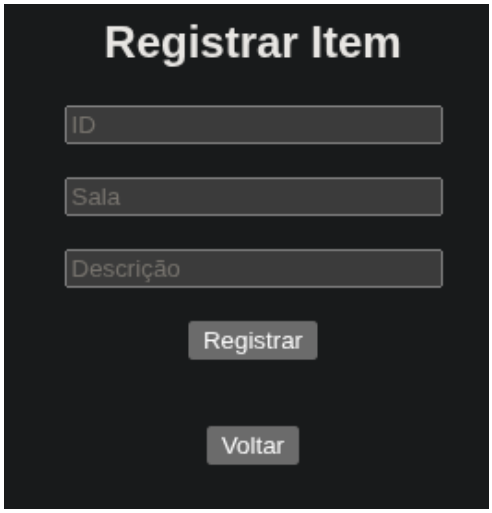


Figura 17: Tela de cadastro.



Figura 18: Tela de atualização de cadastros.

Durante a montagem física, foram avaliadas diferentes caixas e placas para verificar qual configuração apresentava melhor adaptação ao tamanho do circuito e à proposta de fixação no patrimônio. Essa etapa reforçou a necessidade de melhorias futuras no projeto mecânico, especialmente em relação à miniaturização e ao mecanismo anti-remoção. A Figura 19 mostra alguns dos materiais inicialmente avaliados.

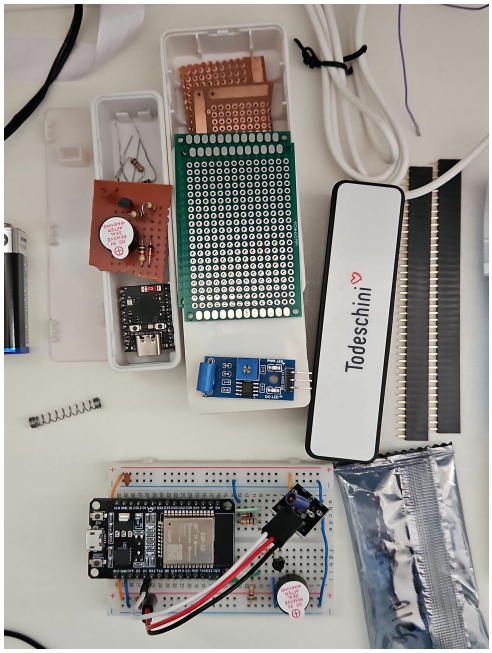


Figura 19: Caixas e placas avaliadas.

Como resultado, o protótipo demonstrou a viabilidade do fluxo principal proposto: detecção de movimentação pelo sensor, processamento do evento no ESP32, emissão de alerta local em condição crítica e registro da ocorrência no servidor. Os testes também evidenciaram limitações importantes, como sensibilidade do sensor, dependência de rede Wi-Fi, ausência de caixa definitiva e necessidade de aprimoramento na autonomia energética. Apesar dessas limitações, o sistema atendeu ao objetivo de validar uma solução inicial de monitoramento patrimonial baseada em dispositivo embarcado e aplicação web.

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

O cronograma do projeto foi organizado em etapas correspondentes ao desenvolvimento incremental do protótipo. A versão proposta contemplava as atividades de testes com sensores, lógica do microcontrolador, desenvolvimento do servidor, montagem da placa, construção da caixa e preparação da demonstração final.

	Completo	Em progresso	Cancelado															
	Completo atrasado	A ser realizado	Atrasado															
	Completo adiantado			E1			E2			E3			E4		E5	DP	Av1	Av2
Atividade		Duração prevista	Tempo levado	08/04	15/04	22/04	29/04	06/05	13/05	20/05	27/05	03/06	10/06	17/06	24/06	01/07	08/07
E1: Sensores e acordamento																		
Definir e implementar conteúdo do site	Todos	3																
Testar leitura do sensor	Todos	6																
Testar deep sleep temporizado do esp	Todos	3																
Testar acordamento pelo sensor	Todos	6																
Atualização do blog	Todos	2																
Total Time		20	0															
E2: Lógica do microcontrolador																		
Criar regras para tempo acordado e timeout de conexão	Todos	12																
Testar conexão do esp à rede local após acordamento	Todos	3																
Criar medidor de bateria	Todos	6																
Testar leitura do medidor	Todos	6																
Atualização do blog	Todos	2																
Total Time		29	0															
E3: Servidor																		
Modelar banco de dados	Todos	6																
Criar scripts de setup	Omar	2																
Criar contêiner para executar o FastAPI no Raspberry	Omar	4																
Criar endpoint para cadastro de item	Omar	1																
Criar endpoint para atualização de dados de item	Omar	1																
Criar endpoint para receber ping diário	Omar	1																
Criar endpoint para receber alertas de movimento	Omar	2																
Criar homepage para acessar dados	Todos	4																
Criar e testar fila de dados a serem escritos no banco	Todos	3																
Testar endpoint de ping com esp	Todos	6																
Testar endpoint com alarme	Todos	6																
Atualização do blog	Todos	2																
Total Time		38	0															
E4: Placa																		
Projetar placa	Todos	8																
Soldar componentes em uma placa	Todos	6																
Testar placa	Todos	4																
Testar e duração de bateria buscando otimizações	Todos	4																
Atualização do blog	Todos	2																
Total Time		24	0															
E5: Caixa																		
Projetar caixa	Todos	8																
Imprimir caixa e testar resultado	Todos	6																
Testar detecção de movimento com a caixa	Todos	6																
Atualização do blog	Todos	2																
Total Time		22	0															
DP: Demonstração do protótipo																		
Gravação do vídeo de demonstração	Todos	6																
Escrita do relatório final	Todos	8																
Atualização do blog	Todos	2																
Total Time		16	0															
	Tempo previsto																	
	149																	

Figura 20: Cronograma proposto.

No cronograma proposto, as atividades foram distribuídas entre as entregas E1 a E5 e

a demonstração do protótipo. A estimativa inicial previa um total de 149 horas, contemplando desde a definição do conteúdo do site e testes iniciais dos sensores até a gravação do vídeo de demonstração e escrita do relatório final.

A execução final apresentou ajustes em relação ao planejamento inicial. Algumas atividades demandaram mais tempo que o previsto, principalmente o desenvolvimento do servidor, os testes de comunicação, a construção da lógica de acordamento do ESP32-S3 e as atividades adicionais relacionadas ao acelerômetro e à atualização do blog. Essas alterações refletem dificuldades técnicas encontradas durante a integração entre hardware, firmware e sistema web.

	Completo	Em progresso	Cancelado															
	Completo atrasado	A ser realizado	Atrasado															
	Completo adiantado			E1			E2			E3		E4		E5	DP	Av1	Av2	
Atividade		Duração prevista	Tempo levado	08/04	15/04	22/04	29/04	06/05	13/05	20/05	27/05	03/06	10/06	17/06	24/06	01/07	08/07	
E1: Sensores e acordamento																		
Definir e implementar conteúdo do site	Todos	3	4															
Testar leitura do sensor KY-002	Todos	6	2															
Testar Deep Sleep temporizado do ESP32	Todos	3	5															
Testar acordamento pelo sensor	Todos	6	6															
Total Time		18	17															
E2: Lógica do microcontrolador																		
Criar regras para tempo acordado e timeout de conexão	Todos	12	12															
Testar conexão do esp à rede local após acordamento	Todos	3	8															
Criar medidor de bateria	Todos	6	10															
Testar leitura do medidor	Todos	6	6															
Total Time		27	36															
E3: Servidor																		
Modelar banco de dados	Todos	6	12															
Criar scripts de setup	Omar	2	4															
Criar contêiner para executar o FastAPI	Omar	4	12															
Criar endpoint para cadastro de item	Omar	1	6															
Criar endpoint para atualização de dados de item	Omar	1	4															
Criar endpoint para receber ping diário	Omar	1	0															
Criar endpoint para receber alertas de movimento	Omar	2	8															
Criar homepage para acessar dados	Todos	4	6															
Criar e testar fila de dados a serem escritos no banco	Todos	3	0															
Testar endpoint de ping com esp	Todos	6	5															
Testar endpoint com alarme	Todos	6	8															
Total Time		36	65															
E4: Placa																		
Projetar placa	Todos	8	8															
Soldar componentes em uma placa	Todos	6	10															
Testar placa	Todos	4	8															
Testar e duração de bateria buscando otimizações	Todos	4	6															
Total Time		22	32															
E5: Caixa																		
Projetar caixa	Todos	8	8															
Obter caixa e testar resultado	Todos	6	8															
Testar detecção de movimento com a caixa	Todos	6	8															
Total Time		20	24															
Outras Atividades																		
Teste de acelerômetro	Matheus		30															
Atualização do blog	Todos		20															
Total Time			50															
DP: Demonstração do protótipo																		
Gravação e edição do vídeo de demonstração	Todos	6	6															
Escrita do relatório final	Todos	8	4															
Total Time		14	10															

Figura 21: Cronograma final.

Comparando os dois cronogramas, observa-se que o escopo geral foi mantido, mas houve redistribuição do esforço ao longo das etapas. A etapa do servidor foi a que apresentou maior aumento de tempo, devido à implementação dos cadastros, endpoints, banco de dados, registros de alertas e funcionalidades de consulta. Também houve acréscimo de atividades não previstas inicialmente, como testes com acelerômetro, o que contribuiu para a ampliação do tempo total do projeto.

5.2 CUSTOS

Os custos do projeto foram concentrados nos componentes necessários para a montagem funcional do protótipo, conforme a Figura 22. A lista inclui o microcontrolador ESP32-S3, sensores de vibração SW-420, buzzer, bateria, regulador de tensão, componentes eletrônicos auxiliares, placa, caixa impressa em 3D e itens de conexão.

Componente	Quantidade	Valor Unitário	Valor Total
ESP32 - S3 Super Mini	1	R$ 30.00	R$ 30.00
Sensor SW-420 (vibration sensor)	2	R$ 6.00	R$ 12.00
Buzzer 5V continuous	1	R$ 3.00	R$ 3.00
Bateria CR18650 Li-ion 3.7V 1200mA	1	R$ 13.50	R$ 13.50
Regulador 3.3V AMS1117	1	R$ 3.90	R$ 3.90
Capacitor ceramico 104	2	R$ 0.30	R$ 0.60
BC548 transistor	1	R$ 0.30	R$ 0.30
Diodo Zener 3.3V	1	R$ 0.20	R$ 0.20
Barra 40 pinos femea 180	1	R$ 1.90	R$ 1.90
PCB	1	R$6.00	R$ 10.00
Caixa impressao 3D	1	R$ 10.00	R$ 10.00
Jumper e outros	1	R$ 11.00	R$ 11.00
Total	**12**		**R$ 86.40**

Figura 22: Custos.

O custo total estimado do protótipo foi de R$ 86,40, considerando 12 itens principais. Esse valor evidencia a viabilidade econômica da proposta em nível de prova de conceito, uma vez que o sistema utiliza componentes de baixo custo e de fácil aquisição. Os itens de maior impacto no orçamento foram o ESP32, a bateria, a caixa impressa em 3D, a placa de circuito e os materiais de conexão. Dessa forma, o protótipo demonstrou ser financeiramente acessível para validação acadêmica e para futuras melhorias incrementais.

6 CONCLUSÕES

6.1 CONCLUSÕES

O desenvolvimento do *SIMPAT* permitiu validar uma arquitetura funcional para monitoramento patrimonial baseado em eventos. O protótipo demonstrou que é possível integrar um ESP32, sensor de vibração, buzzer, comunicação Wi-Fi e servidor web em uma solução de baixo custo, capaz de identificar movimentações suspeitas e registrar ocorrências associadas a um patrimônio monitorado.

A abordagem por eventos mostrou-se adequada ao escopo do projeto, pois evita a necessidade de rastreamento contínuo ou geolocalização em tempo real. A lógica de diferenciação entre movimentações isoladas e recorrentes contribuiu para classificar situações em níveis de severidade, permitindo o acionamento do alerta sonoro apenas em condições críticas.

Os principais problemas observados foram a ocorrência de falsos positivos provocados pela sensibilidade do sensor de vibração e a dependência da rede Wi-Fi para transmissão dos eventos ao servidor. Além disso, o protótipo ainda não contempla mecanismo físico anti-remoção ou anti-furto, o que limita sua aplicação em um cenário real sem melhorias estruturais.

Apesar dessas limitações, o sistema atendeu ao objetivo de validar uma prova de conceito ciberfísica para apoio ao monitoramento patrimonial. O baixo custo de implementação e a simplicidade da arquitetura indicam que a solução pode evoluir para versões mais compactas, robustas e integradas ao ambiente institucional.

6.2 TRABALHOS FUTUROS

Como trabalhos futuros, uma melhoria relevante consiste no desenvolvimento de uma placa de circuito impresso dedicada, reduzindo o volume do circuito, melhorando a organização elétrica e aumentando a confiabilidade da montagem. A adoção de uma bateria menor e mais adequada ao consumo real do dispositivo também pode contribuir para a miniaturização do

protótipo.

Outra evolução importante é o desenvolvimento de uma caixa física definitiva, com melhor fixação ao patrimônio, acabamento mais compacto e mecanismo anti-remoção. Esse recurso permitiria ampliar a proteção física do dispositivo e reduzir a possibilidade de retirada indevida da unidade de monitoramento.

Também se destaca como melhoria futura o uso de um acelerômetro de baixo consumo, como o LIS3DH. Embora sua integração tenha apresentado dificuldades nesta etapa, o sensor oferece maior potencial para leitura de movimento em diferentes eixos, ajuste de sensibilidade e redução de falsos positivos. Assim, pela alta customização, o próprio servidor poderia alterar a sensibilidade do acelerômetro dinamicamente com base nas características dos itens e das suas localizações.

Além disso, recomenda-se a implementação de notificações automáticas para responsáveis pelo patrimônio, painel administrativo mais completo e integração com rede institucional. Em uma perspectiva mais ampla, o *SIMPAT* pode evoluir para uma plataforma de monitoramento patrimonial escalável, capaz de combinar dispositivos embarcados, histórico de eventos, análise de padrões de movimentação e políticas formais de autorização de remanejamento.

REFERÊNCIAS

Espressif. **Sleep Modes - ESP32**. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep$_{m}odes.html/$¿.

Espressif Systems. **ESP32 Series Datasheet**. 2025. Acesso em: 25 jun. 2026. Disponível em: <https://documentation.espressif.com/esp32$_{d}atasheet_{e}n.pdf$¿.

Espressif Systems. **ESP32 Wi-Fi & Bluetooth SoC**. 2026. Acesso em: 25 jun. 2026. Disponível em: <https://www.espressif.com/en/products/socs/esp32>.

FastAPI. **First Steps - FastAPI**. Disponível em: <https://fastapi.tiangolo.com/tutorial/first-steps/>.

FastAPI. **FastAPI Documentation**. 2026. Acesso em: 25 jun. 2026. Disponível em: <https://fastapi.tiangolo.com/>.

Handson Technology. **SW-420 Vibration Sensor Module User Guide**. 2021. Acesso em: 25 jun. 2026. Disponível em: <https://handsontec.com/dataspecs/sensor/Vibration

LEE, E. A.; SESHIA, S. A. **Introduction to Embedded Systems: A Cyber-Physical Systems Approach**. 2. ed. MIT Press, 2017. Disponível em: <https://ptolemy.berkeley.edu/books/leeseshia/>.

Python Software Foundation. **sqlite3 — DB-API 2.0 interface for SQLite databases**. 2026. Acesso em: 25 jun. 2026. Disponível em: <https://docs.python.org/3/library/sqlite3.html>.

Seeed Studio. **Grove - Vibration Sensor (SW-420)**. 2023. Acesso em: 25 jun. 2026. Disponível em: <https://wiki.seeedstudio.com/Grove-Vibration$_{S}ensor_{S}W-420/$¿.

SQLite. **SQLite Documentation**. 2026. Acesso em: 25 jun. 2026. Disponível em: <https://sqlite.org/docs.html>.

STMicroelectronics. **LIS3DH: MEMS digital output motion sensor, ultra-low-power high-performance 3-axis nano accelerometer**. 2016. Acesso em: 25 jun. 2026. Disponível em: <https://www.st.com/resource/en/datasheet/lis3dh.pdf>.

STMicroelectronics. **AN3308: LIS3DH MEMS digital output motion sensor application note**. 2017. Acesso em: 25 jun. 2026. Disponível em: <https://www.st.com/resource/en/application$_{n}ote/an3308-lis3dh-mems-digital-output-motion-sensor-ultralowpower-highperformance-3axis-nano-accelerometer-stmicroelectronics.pdf$¿.