

## **NOISEGUARD: SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE RUÍDO EM AMBIENTES ACADÊMICOS**

**Diego Nascimento de Oliveira**

**Área temática:** Sistemas Inteligentes de Monitoramento

### **RESUMO**

O monitoramento de níveis de ruído em ambientes coletivos, como bibliotecas e salas de estudo, é essencial para garantir um ambiente adequado para concentração e produtividade. Este trabalho apresenta o desenvolvimento do NoiseGuard, um sistema embarcado baseado na placa BitDogLab, que detecta níveis sonoros e fornece alertas visuais e sonoros para incentivar o silêncio em ambientes acadêmicos. O sistema utiliza os periféricos integrados da BitDogLab, incluindo um microfone para captar o som ambiente, um display OLED para exibir mensagens, uma matriz de LEDs para indicar o nível de ruído por cores e um buzzer para emitir sinais de alerta. Os dados coletados são processados e enviados para a plataforma ThingSpeak via Wi-Fi, permitindo o monitoramento remoto. Além disso, uma interface web exibe as informações de ruído em tempo real e fornece alertas visuais aos usuários. O sistema foi validado em um ambiente simulado, demonstrando sua eficácia na detecção de ruídos e na geração de feedback imediato.

**Palavras-chave:** Sistemas Embarcados. IoT. Monitoramento de Ruído. BitDogLab. ThingSpeak.

## 1. INTRODUÇÃO

O ruído excessivo em ambientes de estudo pode comprometer significativamente a concentração e a produtividade dos alunos. Bibliotecas e salas de estudo, que deveriam ser locais silenciosos, frequentemente enfrentam dificuldades para manter um nível sonoro adequado, pois alguns usuários não percebem que estão gerando ruídos incômodos. Em muitos casos, a necessidade de intervenção manual, como a atuação de funcionários para solicitar silêncio, pode ser ineficiente e depender da disponibilidade de mediadores.

Diante desse problema, este trabalho apresenta o desenvolvimento do NoiseGuard, um sistema embarcado baseado na placa BitDogLab, projetado para monitorar os níveis de ruído em ambientes acadêmicos e fornecer alertas visuais e sonoros, incentivando o silêncio de forma automatizada. O sistema utiliza o microfone integrado da placa para captar o som ambiente, processa os dados e classifica os níveis de ruído em três categorias: ambiente silencioso, ruído moderado e ruído alto. Para fornecer feedback imediato aos usuários, a solução emprega uma matriz de LEDs, um display OLED e um buzzer, que indicam visualmente e sonoramente a necessidade de redução do barulho. Além disso, os dados são enviados para a plataforma ThingSpeak e exibidos em uma interface web interativa, que pode ser acessada remotamente e projetada em um monitor dentro da sala de estudo. Isso permite que os usuários visualizem os alertas em tempo real, incentivando a redução do ruído sem necessidade de intervenção de terceiros.

A motivação para a escolha deste tema surgiu a partir da observação direta desse problema em uma biblioteca universitária, onde alguns estudantes manifestavam incômodo com o ruído excessivo gerado por outros usuários. Dessa forma, o sistema busca oferecer uma solução prática, de fácil implementação e que contribua para a melhoria da experiência de estudo em locais compartilhados.

O principal objetivo deste trabalho é desenvolver um protótipo funcional de um sistema de monitoramento de ruído para ambientes acadêmicos, capaz de detectar, classificar e alertar sobre níveis sonoros inadequados. Para isso, são empregados conceitos de sistemas embarcados, comunicação sem fio, processamento de sinais e Internet das Coisas (IoT). A implementação segue boas práticas de desenvolvimento de software e hardware, utilizando metodologias de teste para validar a eficácia do sistema.

Com essa abordagem, espera-se que a solução proposta contribua para a criação de ambientes mais silenciosos e confortáveis para estudo, promovendo a conscientização dos usuários sobre a importância da redução de ruído em locais coletivos.

## 2. METODOLOGIA

O desenvolvimento do NoiseGuard seguiu uma abordagem modular e estruturada, utilizando os recursos disponíveis na placa BitDogLab para capturar, processar e exibir informações sobre os níveis de ruído em um ambiente. A seguir, são descritos os principais componentes e tecnologias empregados, a abordagem utilizada no desenvolvimento, as estratégias de teste e validação, além de um fluxograma representando o funcionamento do sistema.

### 2.1 Recursos Utilizados

Para implementar o sistema de monitoramento de ruído, diversos componentes integrados à BitDogLab foram utilizados, cada um desempenhando um papel essencial no funcionamento do projeto.

O microfone embutido na placa foi responsável por capturar os sons do ambiente. O sinal analógico gerado pelo microfone foi convertido para digital utilizando o Conversor Analógico-Digital (ADC), permitindo a análise dos níveis de ruído. Para melhorar a precisão das medições, um filtro de média móvel foi aplicado aos dados coletados, reduzindo oscilações abruptas e garantindo leituras mais estáveis.

A matriz de LEDs RGB foi utilizada como um indicador visual dos níveis de ruído, alterando sua cor conforme a intensidade do som detectado. O display OLED serviu como uma interface textual para exibir mensagens informando o status do ambiente. Já o buzzer foi empregado para fornecer alertas sonoros curtos, ativando-se em situações de ruído moderado, servindo apenas de alerta.

Para comunicação externa, foi utilizado o módulo Wi-Fi CYW43, que permitiu o envio dos dados para a plataforma ThingSpeak via protocolo HTTP (HyperText Transfer Protocol). Esse protocolo foi escolhido devido à sua simplicidade de implementação, ampla

compatibilidade e eficiência para a aplicação proposta. O HTTP é um protocolo bem estabelecido e suportado por diversas plataformas, incluindo o ThingSpeak, que foi utilizado para armazenar e visualizar os dados coletados. A API da ThingSpeak permite o envio de informações por meio de requisições HTTP, o que facilitou a integração do sistema sem a necessidade de utilizar protocolos mais complexos, como o MQTT. Além disso, como o monitoramento de ruído requer apenas o envio periódico de leituras, sem necessidade de comunicação bidirecional contínua, o HTTP se mostrou uma solução eficiente e de baixo consumo de recursos, atendendo perfeitamente às necessidades do projeto.

O sistema também fez uso do temporizador interno (função `time_us_64` da biblioteca "hardware/timer.h"), que controlou a execução de ações temporizadas, como a ativação do buzzer e a coleta de leituras em intervalos regulares.

## 2.2 Abordagem do Desenvolvimento

A estrutura do sistema foi dividida em cinco principais etapas:

- a. Aquisição e processamento do sinal: os dados captados pelo microfone foram convertidos para digital, processados para obter o nível de ruído em decibéis (dB) e foi feito uma média para calibração dos dados.
- b. Classificação do ruído: os valores de dB foram analisados e categorizados em três níveis: Ambiente Silencioso ( $\leq 52$  dB), Ruído Moderado (52-70 dB) e Ruído Alto ( $> 70$  dB).
- c. Exibição de feedback visual e sonoro: os LEDs e o display OLED são atualizados para refletir o status do ambiente, e o buzzer foi acionado conforme necessário.
- d. Envio de dados para a nuvem: os valores médios dos últimos 15 segundos são enviados para o ThingSpeak para monitoramento remoto.
- e. Integração com página web: os dados são consumidos da API do ThingSpeak e exibidos em tempo real em uma interface web interativa.

## 2.3 Estratégias de Teste e Validação

Para garantir o correto funcionamento do sistema, foram realizados testes em diferentes cenários. O microfone e o ADC foram testados coletando amostras de som em diversos

ambientes, calibrando os limites de classificação dos níveis de ruído. As medições obtidas foram comparadas com um decibelímetro digital para validar a precisão do sistema.

Os periféricos de saída foram testados individualmente para garantir a resposta correta aos diferentes níveis de ruído. A matriz de LEDs foi verificada para acender com as cores apropriadas, o display OLED foi testado para garantir que as mensagens fossem exibidas corretamente e o buzzer foi validado para operar conforme esperado.

A comunicação Wi-Fi e o envio de dados para o ThingSpeak foram testados monitorando a atualização dos valores na plataforma. Além disso, a página web foi acessada em diferentes dispositivos (PCs, smartphones e tablets) para garantir que a interface fosse responsiva e os dados fossem atualizados corretamente.

## **2.4 Fluxograma do Sistema**

A Fig. 1 apresenta o fluxo de funcionamento do projeto, detalhando as etapas desde a captura do áudio até a atualização da página web.

- O sistema inicia com a captura do áudio utilizando o microfone da BitDogLab.
- O sinal é convertido para digital através de um Conversor Analógico-Digital (ADC).
- Em seguida, o nível de ruído é calculado por meio da conversão RMS  $\rightarrow$  dB.
- O sistema então classifica o ruído e executa ações correspondentes.
- A cada 15 segundos, os dados passam por um processamento do sinal (cálculo da média), são enviados ao ThingSpeak e a página web é atualizada.

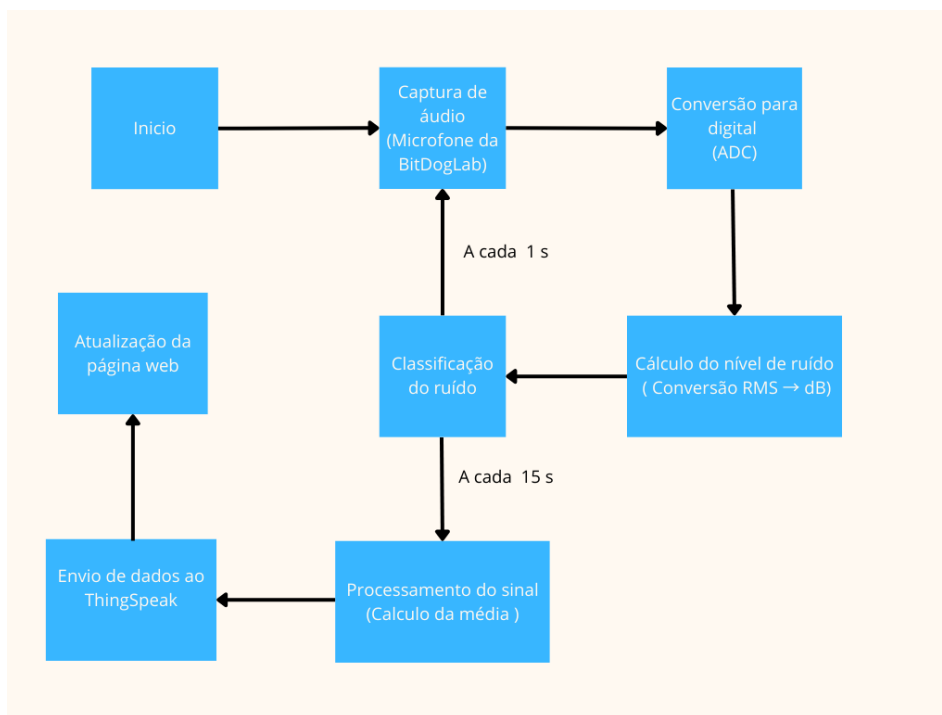


Fig. 1 – Imagem do fluxo de funcionamento do sistema.

## 2.5 Considerações sobre a Metodologia

A divisão do projeto em etapas bem definidas possibilitou uma implementação clara e organizada, assegurando que cada componente do sistema operasse corretamente dentro de suas respectivas funções.

O uso da BitDogLab e seus periféricos permitiu a criação de um sistema funcional e interativo, integrando monitoramento local e remoto para fornecer alertas visuais e sonoros conforme os níveis de ruído. A validação do projeto mostrou que o sistema é capaz de detectar variações sonoras e responder adequadamente, garantindo sua aplicabilidade para ambientes como bibliotecas, salas de estudo e escritórios compartilhados.

### 3. RESULTADOS ALCANÇADOS E DISCUSSÕES

Este projeto foi estruturado para garantir a integração eficiente entre sensores, atuadores e comunicação com a nuvem. Esta seção apresenta as funcionalidades implementadas, descrevendo a interação entre os componentes e os testes realizados para validar a precisão e eficácia do sistema.

#### 3.1 Implementação do Monitoramento de Ruído

Como falado anteriormente, a captura dos níveis de ruído foi realizada pelo microfone embutido da BitDogLab, cujo sinal analógico foi convertido para digital utilizando o Conversor Analógico-Digital (ADC) da placa. Como a leitura bruta do microfone apresenta oscilações, foi aplicado um filtro de média móvel para suavizar os valores e reduzir flutuações instantâneas que poderiam gerar leituras errôneas.

A equação utilizada para converter os valores de RMS (Root Mean Square) para decibéis (dB) foi a seguinte:

$$dB = 20 \times \log_{10} \left( \frac{RMS}{0.00002} \right)$$

Essa conversão permitiu a obtenção de valores aproximados aos capturados por um decibelímetro digital, garantindo maior precisão no monitoramento.

Para evitar reações exageradas a ruídos momentâneos, como uma pessoa espirrando ou movimentando uma cadeira, foi utilizada a média móvel de 15 leituras (15 segundos), antes do envio dos dados ao servidor ThingSpeak. Isso garantiu que apenas mudanças consistentes no nível de ruído fossem consideradas na ativação dos alertas.

#### 3.2 Indicação Visual e Sonora

A matriz de LEDs RGB foi utilizada como um indicador visual dos níveis de ruído, com três estados distintos:

- Ambiente Silencioso ( $\leq 52$  dB)  $\rightarrow$  LEDs verdes acesos.
- Ruído Moderado (52 - 70 dB)  $\rightarrow$  LEDs amarelos acesos.
- Ruído Alto ( $>70$  dB)  $\rightarrow$  LEDs vermelhos acesos.

O buzzer foi configurado para emitir alertas sonoros apenas no estado "Ruído Moderado" por 1 segundo, servindo como um aviso preventivo. No estado "Ruído Alto", o sistema não emite som para evitar aumentar ainda mais o ruído no ambiente. Essa abordagem foi validada com usuários em uma sala de estudos e demonstrou ser eficiente para alertar sem causar incômodo adicional.

O display OLED exibiu mensagens informativas, reforçando o feedback visual. As mensagens variaram conforme o nível de ruído, exibindo "Ambiente Silencioso", "Por favor, fazer silêncio", ou "Ruído Alto, silêncio necessário".

### 3.3 Comunicação com o ThingSpeak e Página Web

A comunicação externa foi implementada via Wi-Fi, utilizando a biblioteca CYW43 para conectar a BitDogLab a uma rede sem fio. Os dados foram enviados ao servidor ThingSpeak utilizando o protocolo HTTP, respeitando a limitação da plataforma na versão gratuita, que permite uma atualização a cada 15 segundos.

Os dados coletados foram exibidos em uma página web interativa hospedada no GitHub Pages. A Fig. 2 mostra uma captura de tela da interface web que foi desenvolvida utilizando HTML, CSS e JavaScript, e consumiu os dados da API do ThingSpeak para exibir:

- O nível atual de ruído.
- Um gráfico histórico atualizado dinamicamente com os últimos valores coletados.
- Mensagens de alerta e mudanças de cor conforme o nível de ruído.
- Um áudio de uma pessoa pedindo silêncio quando o ruído estiver moderado ou alto.





Fig. 2– Captura de tela quando a pagina web estava no modo ambiente silencioso.

### 3.4 Temporização e Controle de Fluxo

Para garantir a correta execução do sistema sem bloqueios, foi utilizada a função de temporização disponibilizada pela biblioteca "hardware/timer.h", como dito anteriormente, permitindo controlar o tempo de execução do buzzer e demais eventos de forma precisa.

A implementação do temporizador garantiu que o buzzer funcionasse de maneira não bloqueante, permitindo que o sistema continuasse coletando dados e exibindo informações sem atrasos.

### 3.5 Testes e Validação

Os testes do sistema foram realizados em diferentes ambientes, incluindo:

- Sala de estudos em biblioteca → Avaliação do comportamento do sistema com variação de pessoas e ruídos ocasionais.
- Laboratórios de pesquisa → Teste da resposta do sistema a conversas paralelas e uso de equipamentos eletrônicos.

Os resultados demonstraram que:

- A média móvel de 15 leituras evitou falsos positivos, garantindo que apenas mudanças consistentes no ruído ativassem os alertas.
- A classificação do ruído foi precisa, com LEDs e display respondendo corretamente a cada variação detectada.
- O buzzer foi eficaz em alertar sem ser invasivo, e o áudio na página web reforçou a notificação para usuários remotos.
- A atualização da página web ocorreu sem atrasos significativos, garantindo acesso contínuo às informações em tempo real.

## 4. CONCLUSÃO

O desenvolvimento do NoiseGuard demonstrou a viabilidade de um sistema de monitoramento de ruído eficiente e acessível para ambientes como salas de estudo e bibliotecas. O projeto alcançou os objetivos propostos ao integrar sensores, atuadores e comunicação em nuvem, permitindo a detecção precisa do nível de ruído e fornecendo feedback visual, sonoro e remoto por meio de uma página web interativa

Os resultados obtidos confirmam que a abordagem adotada foi eficaz na filtragem de ruídos momentâneos e na classificação dos níveis de som ambiente. A implementação da média móvel contribuiu significativamente para a estabilidade das medições, reduzindo falsos alertas e garantindo que o sistema respondesse apenas a variações consistentes no ruído. Além disso, a interface web se mostrou uma solução prática para monitoramento remoto, permitindo que administradores acompanhem os dados em tempo real e recebam alertas visuais e sonoros conforme necessário.

### 4.1 Desafios e Aprendizados

Durante o desenvolvimento do projeto, alguns desafios foram enfrentados, como:

- a. Filtragem de ruídos inesperados → Inicialmente, o sistema reagia a picos de ruído momentâneos, como cadeiras sendo arrastadas ou objetos caindo. A implementação da média móvel solucionou esse problema.
- b. Sincronização entre os alertas físicos e a página web → Ajustes foram necessários para minimizar a defasagem entre as indicações da matriz de LEDs e as atualizações na interface web.
- c. Limitações da versão gratuita do ThingSpeak → O tempo mínimo de atualização de 15 segundos exigiu a implementação de um mecanismo eficiente para condensar os dados coletados sem perda de precisão.

Esses desafios proporcionaram um aprendizado valioso sobre o tratamento de sinais em sistemas embarcados, a importância da escolha de protocolos de comunicação adequados e a necessidade de otimização para integração com serviços em nuvem.

## 4.2 Limitações do Projeto

Embora tenha atendido aos requisitos estabelecidos, o NoiseGuard possui algumas limitações que podem ser trabalhadas em futuras versões:

- a. Ajuste manual dos níveis de alerta → Atualmente, os limiares de ruído são fixos. A inclusão de um sistema configurável permitiria adaptar o sistema a diferentes ambientes.
- b. Integração com outros sensores → A adição de sensores de presença poderia tornar o monitoramento mais inteligente, correlacionando o nível de ruído ao número de pessoas no ambiente.
- c. Melhorias na experiência do usuário na interface web → Implementar gráficos mais detalhados e permitir a personalização das notificações pode tornar o sistema mais eficiente para gestores de ambientes compartilhados.

## 4.3 Contribuição do Projeto

A ferramenta desenvolvida representa uma solução prática para a gestão de ambientes coletivos, reduzindo a necessidade de intervenção manual para controle de ruído. A combinação de sensores embarcados, conectividade Wi-Fi e uma interface web interativa permitiu a criação de um sistema acessível e de fácil utilização.

Além disso, o projeto pode servir como base para aplicações mais amplas, como controle de ruído em ambientes industriais, monitoramento acústico urbano ou até mesmo adaptação para dispositivos voltados à acessibilidade, auxiliando pessoas com sensibilidade auditiva.

Dessa forma, o NoiseGuard não apenas atendeu aos objetivos iniciais, mas demonstrou potencial para futuras expansões, contribuindo para a aplicação de tecnologias IoT em soluções reais.

## REFERÊNCIAS

BITDOGLAB. **BitDogLab/doc at main · BitDogLab/BitDogLab**. Disponível em: <<https://github.com/BitDogLab/BitDogLab/tree/main/doc>>. Acesso em: 12 fev. 2025.

RASPBERRY PI LTD. **Raspberry Pi Pico C/C++ SDK**. 2024. Disponível em: <https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-c-sdk.pdf>. Acesso em: 14 fev. 2025.

RASPBERRY PI LTD. **RP2040 Datasheet**. 2024. Disponível em: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>. Acesso em: 15 fev. 2025.

JULIANA. **Estudar em Silêncio: Como Afeta Sua Concentração e Retenção | Dr. Silêncio**. Disponível em: <<https://www.drsilencio.com.br/estudar-em-silencio-como-afeta-concentracao-e-retencao/>>. Acesso em: 13 fev. 2025.