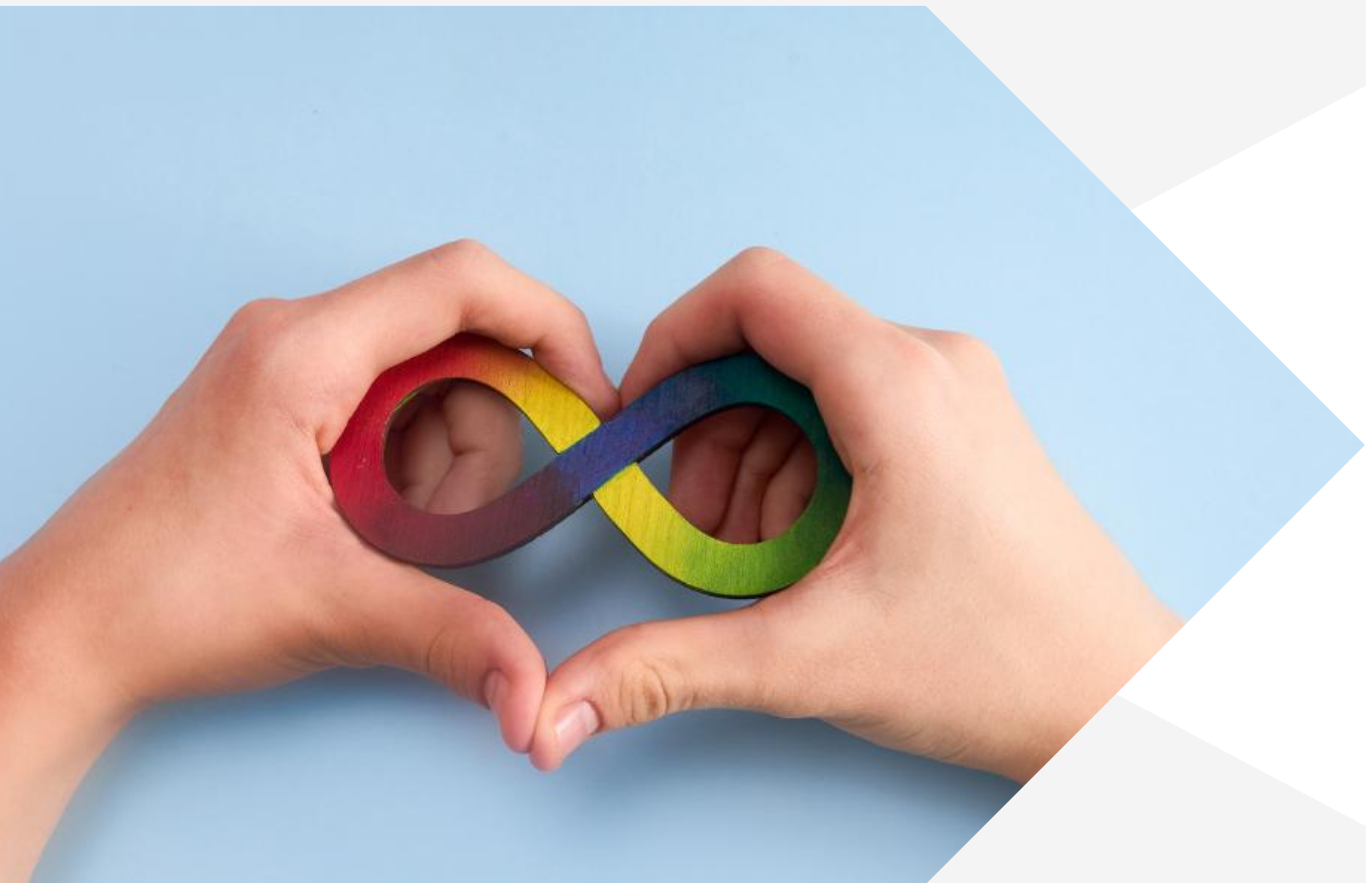


# WHALE

Monitor de Ruídos para  
Ambientes Inclusivos



# Sumário

INTRODUÇÃO	2
OBJETIVOS	3
OBJETIVOS GERAIS	3
OBJETIVOS ESPECÍFICOS	3
FUNDAMENTAÇÃO TEÓRICA	4
HIPERSENSIBILIDADE AUDITIVA	4
TECNOLOGIAS EXISTENTES	4
DIFERENCIAIS DO WHALE	5
HARDWARE	6
DESCRIÇÃO GERAL	6
CONFIGURAÇÃO	7
ESPECIFICAÇÕES	8
MATERIAIS UTILIZADOS	9
CIRCUITO	9
SOFTWARE	11
CAMADAS DO SOFTWARE	11
DESCRIÇÃO DAS FUNCIONALIDADES	13
DEFINIÇÃO DAS VARIÁVEIS	14
INICIALIZAÇÃO DO SOFTWARE	15
ESTRUTURA E FORMATO DE DADOS	17
ORGANIZAÇÃO DA MEMÓRIA	17
PROTOCOLO DE COMUNICAÇÃO	18
FLUXOGRAMA	19
EXECUÇÃO	20
METODOLOGIA	20
TESTES	22
DISCUSSÃO DOS RESULTADOS	23
REFERÊNCIAS	24

# INTRODUÇÃO

Whale | Monitor de Ruídos para Ambientes Inclusivos

A inclusão de alunos neurodivergentes no ambiente escolar é um desafio que requer adaptações para atender às suas necessidades específicas. Indivíduos no espectro autista frequentemente possuem hipersensibilidade auditiva, o que pode causar desconforto extremo em ambientes ruidosos, dificultando seu aprendizado e interação social.

No Brasil, estima-se que cerca de 2 milhões de pessoas estejam dentro do espectro autista, representando aproximadamente 1% da população. Dados do Ministério da Saúde indicam um aumento significativo no número de diagnósticos nos últimos anos, refletindo uma maior conscientização e aprimoramento nos métodos de identificação do transtorno. Entretanto, muitos desses indivíduos ainda enfrentam barreiras significativas no acesso a uma educação inclusiva e de qualidade.

Pesquisas indicam que ambientes escolares inclusivos, quando adequadamente adaptados, podem promover melhorias significativas no desenvolvimento social e acadêmico de alunos autistas. Por exemplo, o uso de tecnologias assistivas, como robôs sociais, tem demonstrado potencial para facilitar interações e promover habilidades sociais em crianças com autismo.

Diante deste problema, o projeto Whale propõe um sistema embarcado para monitoramento do nível de ruído em salas de aula, com o intuito de ser aplicado em ambientes escolares. Utilizando um microfone, um display OLED e LEDs indicadores, o sistema detecta e alerta quando o volume ambiente ultrapassa um limite definido. Além disso, inclui botões físicos para ajuste do limite de alerta, possibilitando uma personalização conforme o contexto da sala. Este projeto foi desenvolvido utilizando o Raspberry Pi Pico W, aproveitando seus recursos de ADC, GPIO e comunicação I2C para garantir uma solução eficiente e responsiva. Embora o sistema tenha sido projetado para aplicação real, os níveis de alerta ainda estão em fase de ajustes experimentais.

# OBJETIVOS

Whale | Monitor de Ruídos para Ambientes Inclusivos

## Objetivos Gerais

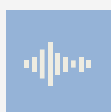
O projeto tem como objetivo desenvolver um sistema embarcado para monitoramento do nível de ruído em ambientes educacionais, com alertas visuais sempre que os valores ultrapassarem um limite pré-configurado. A proposta visa ajudar na criação de ambientes mais inclusivos, especialmente para alunos neurodivergentes, como aqueles no espectro autista, que podem ter hipersensibilidade auditiva. A solução busca não apenas beneficiar esses estudantes, mas também conscientizar os outros alunos sobre o impacto do ruído no ambiente escolar, incentivando práticas mais respeitosas e colaborativas.

Apesar de sua aplicação em salas de aula ser o foco, o sistema ainda está em fase de simulação e precisa de ajustes para garantir que os níveis de alerta e a funcionalidade sejam adequados para o ambiente educacional real.

## Objetivos Específicos



Projetar e implementar um sistema de monitoramento de ruído utilizando um **microfone** conectado ao **ADC** do RP2040.



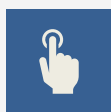
Desenvolver um algoritmo de **processamento de sinal** aplicando um filtro de média móvel para **suavização** das leituras e redução de interferências.



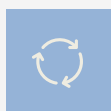
Integrar um **display OLED** para exibição do nível de ruído em tempo real e **mensagens de alerta**.



Implementar um mecanismo de sinalização visual utilizando **LEDs** para indicar se o **nível de ruído** está dentro dos limites aceitáveis.



Desenvolver um sistema de **ajuste dinâmico** dos limiares de alerta utilizando **botões físicos** com tratamento de debounce.



Otimizar o **desempenho e a eficiência** energética do sistema por meio do uso adequado de **timers e interrupções** para processamento eficiente.

# FUNDAMENTAÇÃO TEÓRICA

Whale | Monitor de Ruídos para Ambientes Inclusivos

## Hipersensibilidade Auditiva

Indivíduos com TEA frequentemente apresentam hipersensibilidade auditiva, tornando-os mais suscetíveis a desconfortos em ambientes ruidosos. Estudos indicam que níveis elevados de ruído podem afetar negativamente a concentração, o comportamento e o desempenho acadêmico desses alunos. Portanto, o controle do ambiente sonoro é crucial para promover a inclusão e o bem-estar no contexto escolar.

## Tecnologias Existentes

Diversas iniciativas têm sido desenvolvidas para monitorar e controlar o nível de ruído em ambientes. Entre essas soluções, destaca-se o projeto Silêncio Consciente, desenvolvido para ambientes educacionais, buscando proporcionar um espaço mais acessível e confortável, especialmente para alunos com hipersensibilidade auditiva, como aqueles no espectro autista.

### **Projeto "Silêncio Consciente"**

Com o objetivo de reduzir a poluição sonora em salas de aula, alunas do ensino médio desenvolveram um dispositivo inovador composto por uma bateria de furadeira, microfone, placa de medição de ruído e um giroflex. O sistema detecta quando o som ultrapassa um limite predefinido e aciona um alerta visual, incentivando a autorregulação dos alunos e promovendo um ambiente mais tranquilo. Após sua implementação, observou-se uma redução de 35% nos níveis de ruído, beneficiando especialmente alunos com TEA e melhorando a concentração e o bem-estar de toda a turma.

## Diferenciais do Whale

Embora existam soluções voltadas para o monitoramento de ruído em ambientes educacionais, o Whale apresenta características que o distingue da iniciativa mencionada:

- **Plataforma de Desenvolvimento:**

O Whale utiliza o microcontrolador RP2040, permitindo maior flexibilidade e potencial para futuras expansões do sistema.

- **Processamento Avançado de Sinais:**

Implementa um filtro de média móvel para suavizar as leituras e reduzir interferências.

- **Interface Intuitiva:**

O Whale integra um display OLED que exibe em tempo real os níveis de ruído e mensagens de alerta.

- **Ajuste Dinâmico de Limiares:**

O sistema permite que educadores ajustem dinamicamente os limites de ruído aceitáveis por meio de botões físicos com debounce.

- **Eficiência Energética:**

Com o uso adequado de timers e interrupções, o Whale otimiza o desempenho e a eficiência energética.

Ao analisar as soluções existentes, o Whale surge como uma proposta alternativa e inovadora que combina tecnologias de hardware e software para oferecer uma ferramenta eficaz no monitoramento e controle de ruído em ambientes educacionais inclusivos.

# HARDWARE

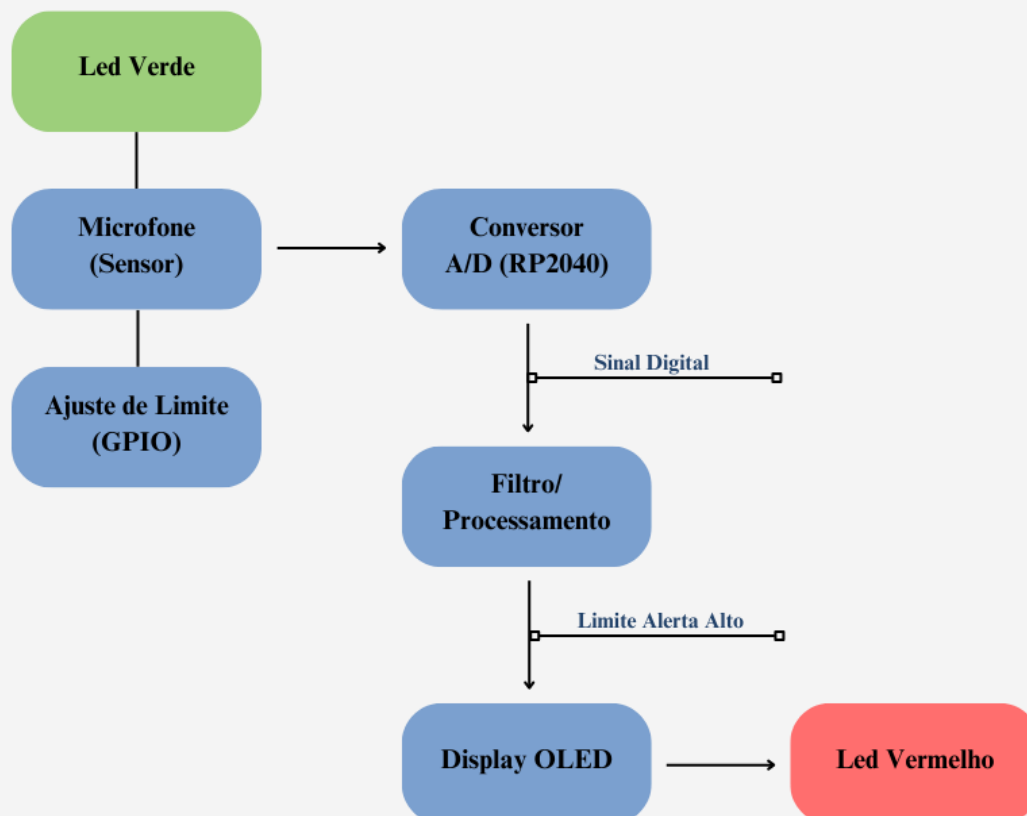
## Whale | Monitor de Ruídos para Ambientes Inclusivos

O sistema monitora o nível de ruído ambiente usando um [microfone](#) e compara com um limite de alerta ajustável. Quando o ruído ultrapassa esse limite, um aviso é exibido em um [display OLED](#) e um [LED](#) vermelho é ativado. Se o ruído diminuir, o LED verde é aceso para indicar um ambiente adequado.

A implementação utiliza a placa BitDogLab, que integra um microcontrolador RP2040, ADC para conversão do sinal do microfone, comunicação I2C para exibição no display OLED, além de GPIOs para controle de LEDs e [botões](#) de ajuste do limite de alerta.

Nesta seção, será apresentada a arquitetura de hardware do Whale, descrevendo cada componente e como ele se interliga no sistema. Será ressaltado, ainda, a variação de consumo de energia associada ao uso de displays maiores, um ponto importante para quem planeja expandir ou modificar o projeto.

## Descrição Geral



- **Microfone (Sensor):** Converte o som ambiente em um sinal elétrico analógico.
- **Conversor A/D (RP2040):** A unidade de conversão analógica-digital do microcontrolador RP2040 lê o sinal do microfone, transformando-o em dados digitais para processamento.
- **Filtro/Processamento (Software no RP2040):** Responsável por aplicar a média móvel e calcular a intensidade do som (RMS ou valor quadrático médio).
- **Ajuste de Limite (GPIO):** Botões físicos conectados ao RP2040 para permitir ao usuário aumentar ou diminuir o limite de detecção de ruído.
- **Display OLED (via I2C):** Exibe as informações de forma legível, como mensagens de alerta ou o valor atual do limite configurado.
- **LED Verde / LED Vermelho (GPIO):** Fornecem indicação imediata do estado de ruído; o LED vermelho acende em caso de alerta e o LED verde permanece aceso quando o ruído está dentro do limite.

## Configuração

- **Microfone:**
  - Alimentação: 3.3 V.
  - Saída: Sinal analógico para o canal ADC 2 (pino 28 físico no RP2040)
- **ADC (RP2040):**
  - Seleciona o canal 2 para leitura do microfone.
  - Resolução de 12 bits para leitura do sinal analógico.
  - Amostras: 200 amostras por ciclo de leitura (no exemplo do código).
- **Filtros/Processamento (Software):**
  - Cálculo do valor quadrático médio (RMS).
  - Filtro de média móvel de 5 amostras para suavização.
  - Limite de alerta ajustável de 0.2 a 1.8 (valores de simulação).
- **Ajuste de Limite (GPIO):**
  - Botão A: Diminui o limite em incrementos de 0.1.
  - Botão B: Aumenta o limite em incrementos de 0.1.



- Uso de pull-up interno para manter estado lógico alto quando não pressionado.

- **Display OLED (I2C):**

- Velocidade I2C: 100 kHz.
- Pinos de SDA e SCL: GPIO14 (SDA) e GPIO15 (SCL) no RP2040.
- Resolução do display: 128 x 64 pixels.

- **LED Verde / LED Vermelho (GPIO):**

- Configurados como saída digital (GPIO).
- LED Verde: Ligado normalmente, desligado em caso de alerta.
- LED Vermelho: Desligado normalmente, ligado em caso de alerta.

## Especificações

- **Medição de ruído:**

- O microfone + ADC do RP2040 tem precisão suficiente para detectar variações de ruído em sala de aula.
- A resolução de 12 bits (0-4095) oferece granularidade para medir sons de baixa a moderada intensidade.

- **Tempo de resposta:**

- O loop principal lê 200 amostras e realiza a média móvel rapidamente (em menos de 200 ms), garantindo detecção quase em tempo real.
- O debounce dos botões (300 ms) evita leituras falsas ao ajustar o limite.

- **Alertas visuais:**

- O display OLED de 128 x 64 pixels é nítido e permite exibir texto com boa legibilidade.
- LEDs coloridos fornecem sinalização simples e rápida sobre o estado de alerta.

- **Consumo de energia:**

- O sistema funciona em 3.3 V, consumindo relativamente pouca corrente, viável para uso em salas de aula.
- **Variação de consumo para aumentar a tela:** Se for utilizado um display maior (p. ex., 1,3” ou 2,42”), o consumo tende a aumentar de forma proporcional ao número de LEDs (pixels) e ao tempo que a tela permanece ligada. Um display maior pode ter **acréscimo de até 10-20 mA** em relação ao modelo de 0,96” ou 1,3”. Esse valor pode ser mais alto se o brilho ou a taxa de atualização forem maiores.

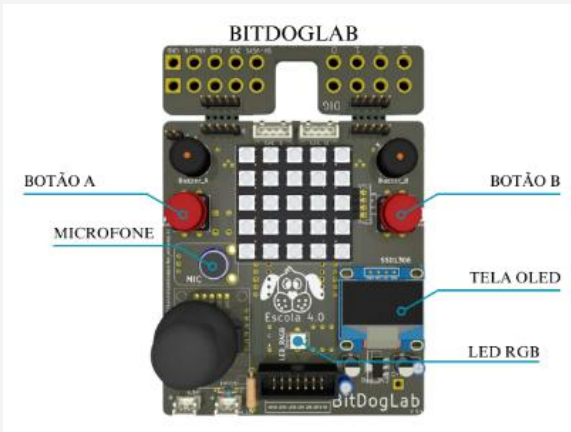
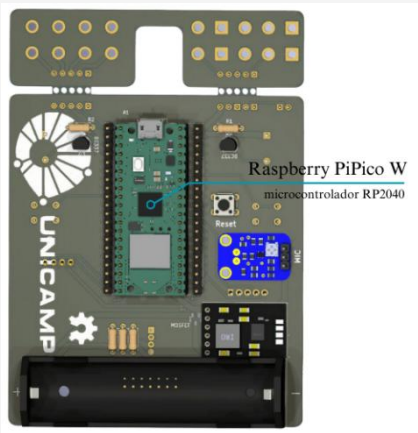
▪ **Ajuste de limite:**

- Os botões permitem configurar um intervalo de 0.2 a 1.8, cobrindo situações de ruído baixo a relativamente alto.

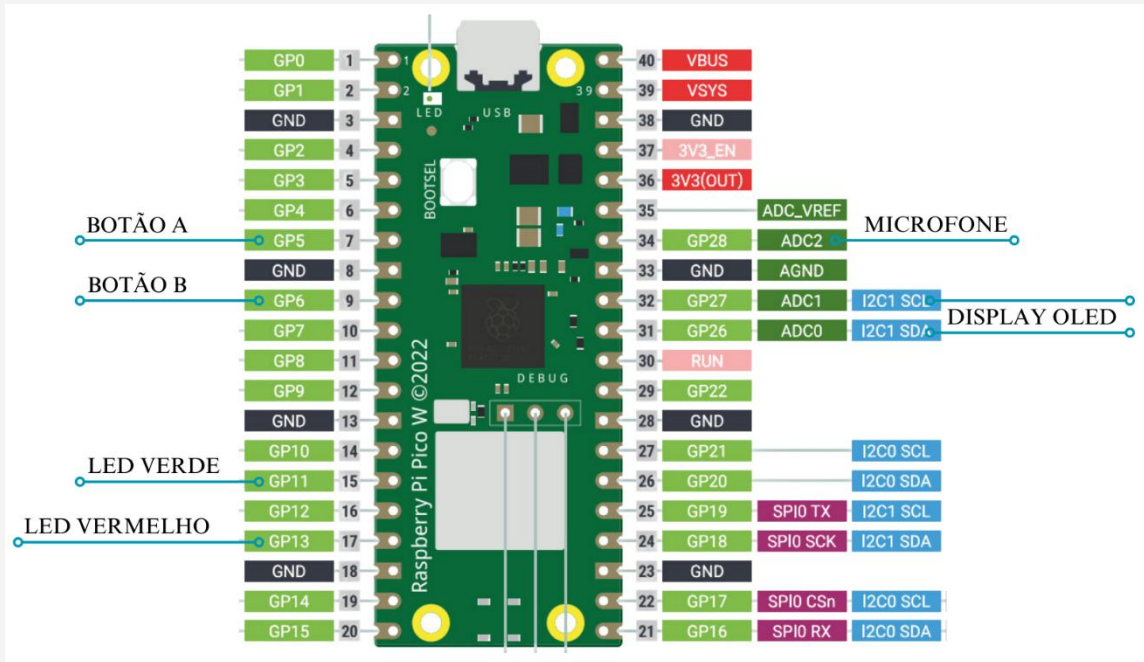
## Materiais Utilizados

COMPONENTES DA BITDOGLAB	WHALE
Led RGB	✓
Display OLED	✓
Botões	✓
Buzzer	✗
Matriz de Leds Coloridos	✗
Microfone	✓
Joystick	✗

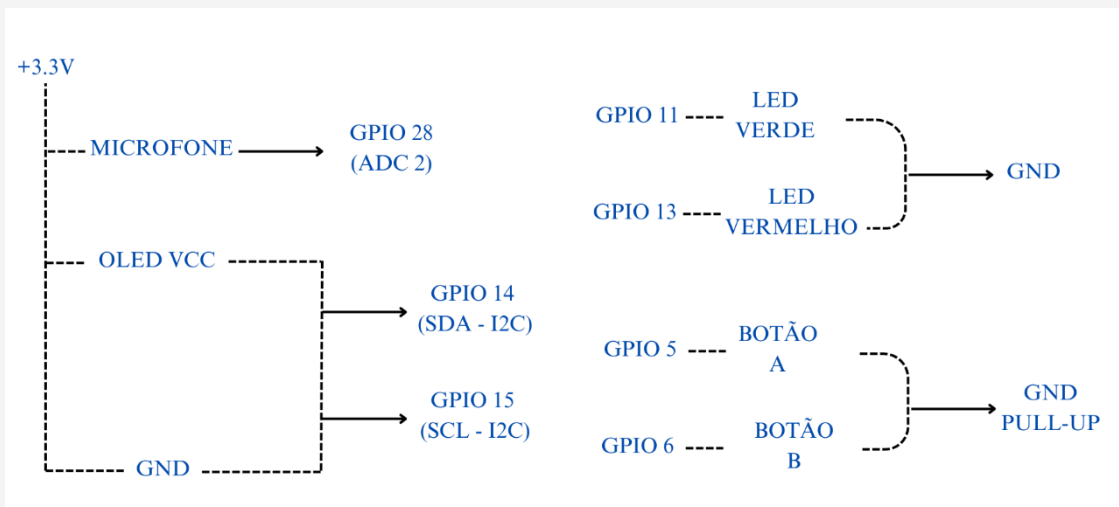
## Circuito

COMPONENTES NA BITDOGLAB	
 <p>BITDOGLAB</p> <p>BOTÃO A</p> <p>MICROFONE</p> <p>BOTÃO B</p> <p>TELA OLED</p> <p>LED RGB</p>	 <p>Raspberry Pi Pico W microcontrolador RP2040</p>

## COMPONENTES NO RASPBERRY PI PICO W



## CIRCUITO SIMPLIFICADO



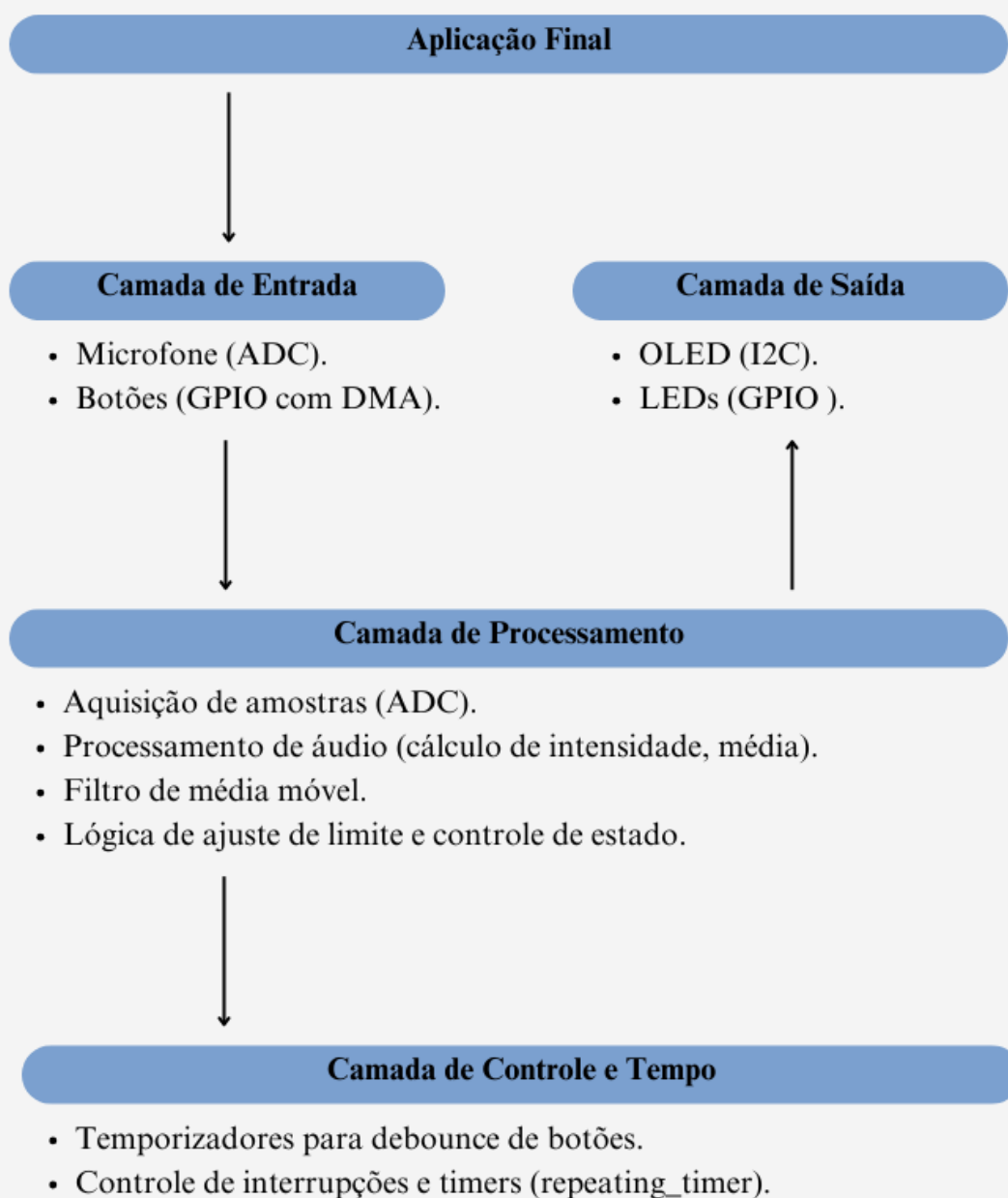
# SOFTWARE

Whale | Monitor de Ruídos para Ambientes Inclusivos

[https://github.com/luanavarelaa/Embarcatech/tree/main/Projeto\\_Final](https://github.com/luanavarelaa/Embarcatech/tree/main/Projeto_Final)

Esta seção descreve a estrutura do software desenvolvido para monitoramento de ruído com foco no bem-estar de alunos autistas. O sistema utiliza o RP2040 para leitura de som via ADC, ajuste dinâmico de limites via botões, exibição de informações em display OLED e indicação de estado por LEDs.

## Camadas do Software



### Camada de Entrada

**Microfone (ADC):** Captura as amostras do ambiente e converte o sinal analógico em digital.

**Botões (GPIO):** Permite o ajuste do limite de alerta com tratamento de debounce para evitar múltiplas detecções.

### Camada de Processamento

**Aquisição e Processamento do Som:** Captura e processa as amostras para calcular a intensidade sonora usando a média quadrática e um filtro de média móvel para suavização.

**Lógica de Controle:** Determina se o ambiente excedeu o limite de alerta, disparando mensagens e acendendo os LEDs correspondentes.

### Camada de Saída

**OLED (I2C):** Exibe informações como o limite de alerta e mensagens (“Silêncio”).

**LEDs (GPIO):** Indicadores visuais (verde para normal, vermelho para alerta).

### Camada de Controle e Tempo

**Temporizadores:** Gerenciam as leituras periódicas e o debounce dos botões para garantir a estabilidade das entradas.

## Descrição das Funcionalidades

### Leitura e Processamento do Som:

O sistema amostra o sinal do microfone utilizando o ADC do RP2040. Em seguida, calcula a intensidade sonora por meio da média quadrática das amostras e aplica um filtro de média móvel para suavizar as variações bruscas.

### Ajuste do Limite de Alerta:

Dois botões (A e B) permitem reduzir ou aumentar o limite de alerta, respectivamente, de 0,2 a 1,8, com incremento de 0,1. Cada botão possui debounce implementado via timer para evitar leituras errôneas. O valor atualizado é exibido na tela OLED.

*Os valores sugeridos para o ajuste dinâmico são baseados em simulações, pois uma calibração precisa requer uma pesquisa mais aprofundada sobre os limiares sonoros específicos de cada grau de hipersensibilidade auditiva.*

### Exibição e Indicadores Visuais:

O display OLED, utilizando protocolo I2C, exibe o valor atual do limite e mensagens de alerta (“Silêncio”) quando a intensidade sonora ultrapassa o limite. Simultaneamente, LEDs indicam o estado do sistema: LED verde aceso em condições normais e LED vermelho acionado em alerta.

### Controle de Temporização e Interrupções:

Timers configurados com a função *repeating\_timer* garantem a execução periódica das leituras e verificações dos botões, assegurando um sistema responsivo e com baixa latência.

## Definição das Variáveis

VARIÁVEL	TIPO	DESCRIÇÃO	CONTRIBUIÇÃO
limite_alerta	float	Limiar de intensidade sonora para alerta.	Define o gatilho para alerta.
Mensagem_exibida	bool	Indica se o alerta já foi exibido.	Evita reexibição desnecessária do alerta.
historico	float[5]	Últimos 5 valores de intensidade sonora (média móvel).	Suaviza as medições.
indice_historico	uint8_t	Índice circular para o array historico.	Atualiza a posição do filtro.
adc_buffer	uint16_t[SAMPLES]	Buffer com amostras do ADC do microfone.	Armazena dados de som para processamento.
botao_timer_A	struct repeating_timer	Timer para debounce do botão A.	Garante leitura estável do botão A.
botao_timer_B	struct repeating_timer	Timer para debounce do botão B.	Garante leitura estável do botão B.
LIMITE_ALERTA_PADRAO	float	Valor padrão do limite de alerta.	Valor inicial do alerta.
LIMITE_RESET	float	Valor para reset do alerta.	Remove alerta e restaura estado normal.
MIC_PIN	uint	Pino do microfone (canal ADC 2).	Define a entrada do microfone.
LED_VERDE_PIN	uint	Pino do LED verde.	Indica estado normal.
LED_VERMELHO_PIN	uint	Pino do LED vermelho.	Indica alerta.

## Inicialização do Software

A inicialização do software acontece em várias etapas para configurar os componentes de hardware e preparar o sistema para a execução do monitoramento.

### Bibliotecas:

As bibliotecas necessárias para operar com o microcontrolador RP2040, manipulação de strings, matemática, comunicação I2C, ADC e temporizadores são incluídas logo no início do código. Isso garante que todas as funções e definições sejam disponibilizadas para o compilador.

```
#include <stdio.h>
#include <string.h> // Permite o uso de memset e outras funções de
manipulação de strings
#include <math.h> // Necessária para cálculos matemáticos (ex.: sqrt)
#include "pico/stdlib.h" // Funções básicas do SDK Pico
#include "hardware/adc.h" // Funções para configuração e uso do ADC
#include "inc/ssd1306.h" // Funções para operar com o display OLED SSD1306
#include "hardware/i2c.h" // Funções para a comunicação via I2C
#include "hardware/timer.h" // Funções para temporizadores e gerenciamento
de interrupções
```

### Configuração dos LEDs:

Os LEDs são configurados para indicar o estado do sistema. O LED verde é aceso por padrão, indicando que o sistema está em funcionamento normal.

```
gpio_init(LED_VERDE_PIN);
gpio_set_dir(LED_VERDE_PIN, GPIO_OUT);
gpio_init(LED_VERMELHO_PIN);
gpio_set_dir(LED_VERMELHO_PIN, GPIO_OUT);

gpio_put(LED_VERDE_PIN, 1); // Acende LED verde
gpio_put(LED_VERMELHO_PIN, 0); // Desliga LED vermelho
```



### Configuração dos Botões A e B:

Os botões são configurados como entradas com pull-up interno, para garantir que os botões sejam lidos corretamente (nível lógico baixo quando pressionados).

```
gpio_init(BOTAO_A_PIN);  
gpio_set_dir(BOTAO_A_PIN, GPIO_IN);  
gpio_pull_up(BOTAO_A_PIN);  
  
gpio_init(BOTAO_B_PIN);  
gpio_set_dir(BOTAO_B_PIN, GPIO_IN);  
gpio_pull_up(BOTAO_B_PIN);
```

### Configuração do ADC (Microfone):

A configuração do ADC é feita para ler o microfone que está conectado no pino do canal 2. O ADC é inicializado, e a entrada do microfone é selecionada para leituras.

```
adc_gpio_init(MIC_PIN);  
adc_init();  
adc_select_input(MIC_CHANNEL);
```

### Configuração do OLED (Tela de Exibição):

A tela OLED é configurada para exibir informações sobre o limite de alerta e mensagens de status. A interface I2C é utilizada para comunicação com o display.

```
// Função de configuração do OLED  
void oled_setup() {  
    i2c_init(i2c1, 100000); // Inicializa o I2C na velocidade de 100kHz  
    gpio_set_function(14, GPIO_FUNC_I2C); // Configura pino 14 para I2C SDA  
    gpio_set_function(15, GPIO_FUNC_I2C); // Configura pino 15 para I2C SCL  
    gpio_pull_up(14);  
    gpio_pull_up(15);  
    ssd1306_init(); // Inicializa o display OLED  
}
```

```
oled_setup();  
oled_clear();           // Limpa a tela do OLED  
oled_display_alert_limit(); // Exibe o limite de alerta inicial
```

#### Configuração de Timers para Botões:

São configurados timers repetitivos para monitorar os botões A e B a cada 100ms, para aplicar o debounce e garantir que os botões sejam lidos corretamente sem erros de pulsação.

```
add_repeating_timer_ms(100, verificar_botao_A_callback, NULL, &botao_timer_A);  
add_repeating_timer_ms(100, verificar_botao_B_callback, NULL, &botao_timer_B);
```

## Estrutura e Formato de Dados

#### Dados de Entrada:

- Valores analógicos provenientes do microfone, convertidos pelo ADC (12 bits).

#### Dados de Processamento:

- Intensidade sonora: Calculada pela média quadrática das amostras.
- Dados filtrados: Utilização de um array (`historico[]`) para calcular a média móvel, suavizando as variações.

#### Dados de Saída:

- Mensagens e limites exibidos no OLED.
- Indicadores de estado (LEDs) que refletem a condição do ambiente.

## Organização da Memória

#### Buffer do ADC:

- O array `adc_buffer[SAMPLES]` armazena as amostras de áudio, sendo alocado na memória RAM.

#### Histórico para Filtro:

- O array `historico[FILTRO_MEDIA]` é usado para o cálculo da média móvel e reside na memória RAM.

#### Buffer do Display:

- A memória destinada à renderização do display OLED é utilizada para construir os pacotes de dados que serão enviados via I2C.

#### Registros de Periféricos:

- As configurações dos pinos e dos módulos (ADC, I2C, Timers) residem em endereços específicos mapeados pelo hardware do RP2040, conforme a documentação técnica.

## Protocolo de Comunicação

### Protocolo I2C para o OLED

#### Configuração:

- O protocolo I2C é utilizado para comunicação com o display OLED. O barramento é inicializado com a velocidade definida (100 kHz) e os pinos são configurados adequadamente.

#### Transmissão de Dados:

- Os pacotes enviados ao display seguem o formato do controlador SSD1306:
  - **Comando de Inicialização:** Inicia o display e define os parâmetros de exibição.
  - **Comandos de Controle:** Configuram a área de renderização (colunas, páginas) e enviam comandos para limpar ou atualizar a tela.
  - **Dados Gráficos:** Dados que representam os caracteres ou gráficos a serem desenhados no OLED.

### Formato dos Pacotes de Dados

#### Cabeçalho do Pacote:

- Contém o endereço I2C do display e um byte de controle que define se o próximo dado será um comando ou um valor para exibição.

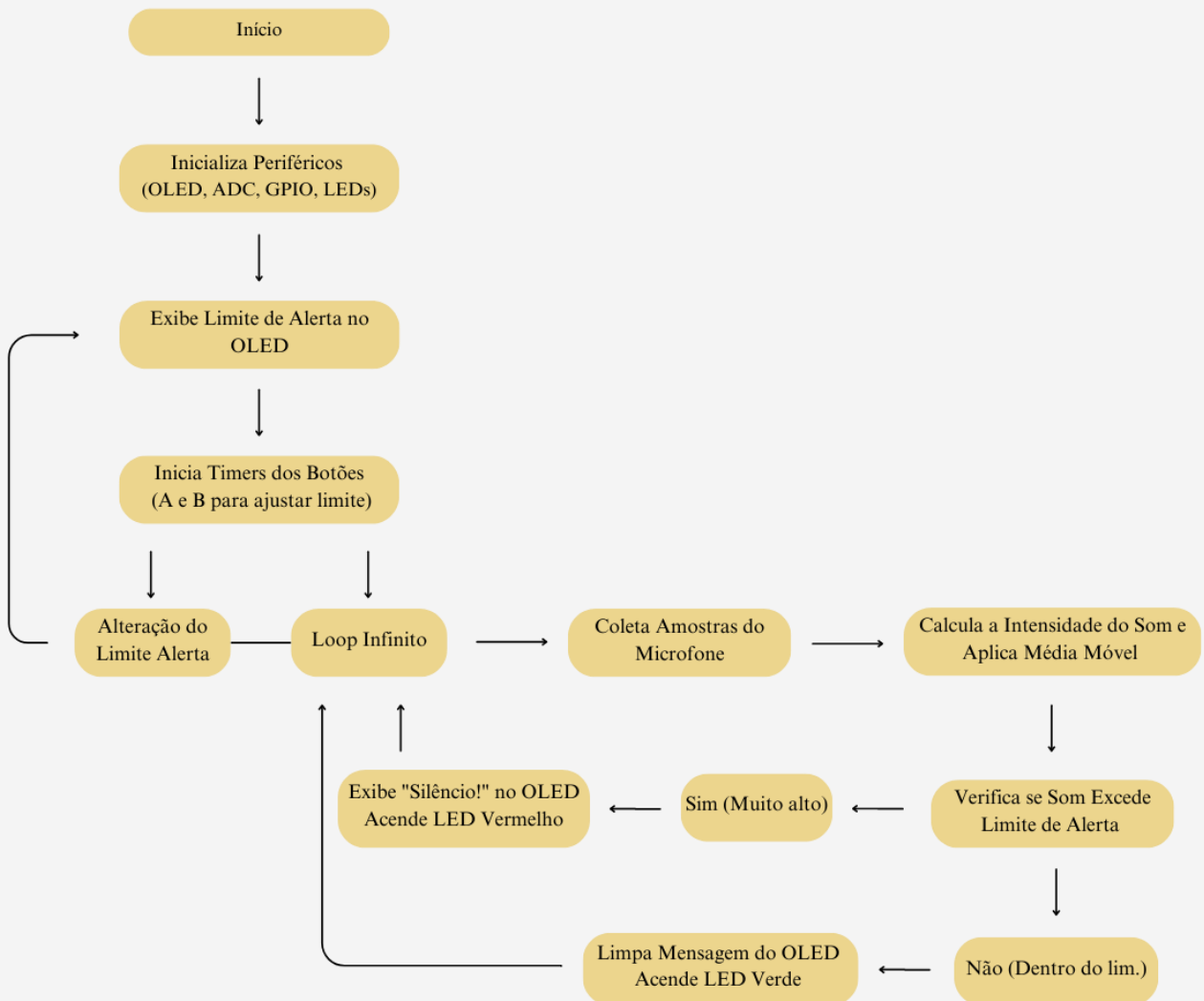
#### Corpo do Pacote:

- Contém os bytes correspondentes aos pixels que serão exibidos, organizados conforme o buffer de renderização.
- Pode incluir comandos intercalados para ajustes de posição ou atualização da tela.

#### Finalização:

- A transmissão é encerrada com a condição de parada do I2C.
- O display responde com um ACK/NACK para indicar se os dados foram recebidos corretamente.

## Fluxograma



# EXECUÇÃO

Whale | Monitor de Ruídos para Ambientes Inclusivos

## Metodologia

O desenvolvimento do projeto seguiu uma abordagem baseada na pesquisa e na experimentação incremental, explorando os recursos da BitDogLab e garantindo a viabilidade da solução proposta. As etapas envolvem desde a concepção da ideia até a implementação final, passando por testes e ajustes contínuos.

### Concepção do Projeto e Pesquisa Inicial

O projeto nasceu da necessidade de criar um sistema de monitoramento de ruído em salas de aula, visando auxiliar pessoas com sensibilidade auditiva. Inicialmente, foram realizadas pesquisas sobre métodos de captação de áudio e estratégias para fornecer feedback acessível ao usuário.

Além disso, buscou-se compreender as capacidades da BitDogLab para atender a esses requisitos. Como a placa já conta com um microfone embutido, um display OLED, LEDs e botões, sua escolha se mostrou adequada, pois elimina a necessidade de módulos externos e simplifica a implementação.

Também foi conduzida uma análise de viabilidade para garantir que os componentes suportariam a funcionalidade esperada, incluindo a precisão da captura de áudio, a capacidade de processamento do Raspberry Pi Pico W e as possibilidades de interação com o usuário.

### Exploração de Recursos e Desenvolvimento Inicial

Com a escolha do hardware definida, a etapa seguinte envolveu a exploração dos recursos disponibilizados pela BitDogLab. Como a documentação da placa é limitada, o ponto de partida foi a busca por exemplos de código no repositório oficial da BitDogLab no GitHub, que fornecem implementações básicas para componentes como OLED e microfone.

Os primeiros testes foram realizados a partir desses exemplos, garantindo que cada periférico operasse corretamente e pudesse ser integrado ao projeto. O código inicial validou a capacidade do microfone de capturar som e do display OLED de exibir informações de forma dinâmica.

## Implementação e Modificação do Código

A partir da base funcional obtida nos testes iniciais, o código foi aprimorado para incorporar novos elementos e tornar o sistema mais interativo e eficiente. As principais modificações incluem:

- **Processamento do Sinal de Áudio:**
  - Ajuste da captura do microfone para determinar níveis médios de ruído ao longo do tempo, evitando oscilações bruscas.
  - Implementação de um limite ajustável de ruído, permitindo que o usuário personalize os níveis de alerta conforme o ambiente.
- **Interação e Feedback:**
  - LEDs foram programados para indicar visualmente diferentes faixas de ruído, com cores que representam estados como "seguro" e "excessivo".
  - Botões físicos foram incorporados para permitir ajustes no limite de detecção sonora sem a necessidade de reprogramação.
- **Gerenciamento de Tempo e Performance:**
  - Uso de timers para otimizar a leitura dos botões e eliminar falsos acionamentos (técnica de debounce).
  - Implementação de um sistema de atualização periódica do display OLED para evitar sobrecarga de processamento.

## Compilação, Testes e Depuração

A implementação do código foi realizada no VS Code, utilizando as extensões apropriadas para o desenvolvimento embarcado no Raspberry Pi Pico. O processo de desenvolvimento envolveu sucessivas compilações e testes incrementais, com ajustes progressivos até alcançar uma versão estável e funcional.

Os principais desafios encontrados e resolvidos nesta fase foram:

- Ajuste da sensibilidade do microfone para garantir que a captação do ruído fosse representativa do ambiente.
- Sincronização entre os componentes para evitar atrasos na resposta do sistema.
- Depuração de falhas de compilação e comunicação com o OLED via protocolo I2C, garantindo que os dados fossem corretamente interpretados pelo display.

## Testes

Para garantir a funcionalidade e confiabilidade do sistema, foram conduzidos diversos testes abordando tanto o desempenho técnico quanto a usabilidade do dispositivo. Os testes foram realizados em diferentes condições ambientais e com variados níveis de ruído para avaliar a precisão da detecção sonora e a resposta dos alertas visuais.

É importante destacar que os níveis de ruído estabelecidos no software (0.2 - 1.8) foram utilizados de forma simulatória, apenas para fins de calibração do sistema. O teste validou exclusivamente a captação e resposta do dispositivo a diferentes níveis de ruído e sensibilidades ajustáveis pelo usuário, sem necessariamente contextualizar os valores para a hipersensibilidade auditiva. Para um ajuste preciso voltado ao público-alvo, como autistas com sensibilidade ao som, seriam necessárias pesquisas e validações adicionais, considerando estudos clínicos e testes com pessoas dentro do espectro autista.

### Testes de Captação e Processamento do Áudio

- O microfone embutido foi testado em ambientes com diferentes níveis de ruído, como salas silenciosas, ambientes com conversas normais e locais com sons elevados.
- A calibração do limiar de detecção sonora foi ajustada para evitar falsos positivos ou negativos, garantindo que o sistema identificasse corretamente os níveis de ruído.
- Foram analisadas as variações na captação de som, verificando se o sistema respondia de forma consistente a mudanças graduais e picos de ruído.

### Testes de Feedback Visual e Interação

- O display OLED foi testado para exibir informações de forma clara e legível, ajustando a taxa de atualização para evitar sobrecarga de processamento.
- Os LEDs indicadores foram avaliados para garantir que as cores representassem corretamente as diferentes faixas de ruído, oferecendo um feedback intuitivo.

### Testes de Funcionalidade dos Botões

- Os botões físicos foram submetidos a testes de acionamento repetitivo para verificar a resposta do sistema e evitar leituras errôneas.
- Foi implementada uma técnica de debounce via software para eliminar falsos acionamentos e garantir uma interação mais precisa.
- Foram realizadas simulações de uso prolongado para validar a durabilidade dos botões e sua resposta ao longo do tempo.

## Testes de Estabilidade e Desempenho

- O código foi testado em execução contínua por longos períodos, monitorando a estabilidade do sistema e a presença de possíveis falhas ou travamentos.

## Ajustes e Otimizações Finais

Com base nos resultados dos testes, foram feitas otimizações para melhorar a precisão do sistema e a experiência do usuário. Entre os principais ajustes estão:

- Refinamento do algoritmo de detecção para tornar a resposta mais estável.
- Melhorias no código para evitar processamento desnecessário e garantir um funcionamento eficiente.

## Vídeo

[https://www.youtube.com/watch?v=O-yyBE2\\_KIY](https://www.youtube.com/watch?v=O-yyBE2_KIY)

## Discussão dos Resultados

Os testes realizados demonstraram que o sistema foi capaz de captar e interpretar variações de ruído de maneira eficaz, respondendo de forma coerente aos diferentes níveis estabelecidos. O funcionamento dos LEDs e do display OLED validou a capacidade do dispositivo de fornecer feedback visual imediato, enquanto os botões permitiram ajustes na sensibilidade conforme necessário. Durante a execução, o software manteve estabilidade e desempenho satisfatório, sem falhas críticas ou comprometimento da resposta do sistema.

Entretanto, é importante ressaltar que os níveis de ruído implementados no software (0.2 - 1.8) são apenas simulatórios, utilizados para verificar a resposta do sistema a diferentes intensidades sonoras. O teste validou apenas a captação do som e a resposta do dispositivo a esses níveis predefinidos, sem garantir que tais valores estejam contextualizados para a hipersensibilidade auditiva. Para uma aplicação mais precisa nesse cenário, seria necessário realizar pesquisas com autistas e especialistas, a fim de definir faixas sonoras mais representativas e ajustar os limiares de ativação do sistema.

Dessa forma, os resultados indicam que o sistema cumpre sua função essencial e apresenta um potencial significativo para aprimoramento, especialmente se direcionado a um público-alvo específico. Com ajustes adicionais e validações mais aprofundadas, ele pode se tornar uma ferramenta eficaz para auxiliar indivíduos com hipersensibilidade auditiva em ambientes variados.



# REFERÊNCIAS

Whale | Monitor de Ruídos para Ambientes Inclusivos

- [1] BRIGHENTI, S. et al. *Social Assistive Robotics for Autistic Children*. 2022. Disponível em: [https://arxiv.org/abs/2209.12289?utm\\_source=chatgpt.com](https://arxiv.org/abs/2209.12289?utm_source=chatgpt.com).
- [2] MINISTÉRIO DA SAÚDE. *Política Nacional de Saúde da Pessoa com Transtorno do Espectro do Autismo*. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-mental/autismo>.
- [3] ORGANIZAÇÃO PAN-AMERICANA DA SAÚDE. *Transtornos do Espectro Autista: uma visão geral*. 2017. Disponível em: <https://www.paho.org/pt/topicos/transtornos-do-espectro-autista>.
- [4] FEBRACE. *Projeto de Redução de Ruídos em Sala de Aula*. Disponível em: <https://virtual.febrace.org.br/2024/HUM/4207/>.
- [5] GOVERNO DO ESTADO DE SÃO PAULO. *Estudantes da rede estadual de Franca criam projeto de redução de ruídos em sala de aula e são finalistas de premiação global*. Disponível em: <https://www.educacao.sp.gov.br/estudantes-da-rede-estadual-de-franca-criam-projeto-de-reducao-de-ruídos-em-sala-de-aula-e-sao-finalistas-de-premiacao-global/>.
- [6] CANAL AUTISMO. *Estudantes criam dispositivo que reduz ruídos em sala de aula*. Disponível em: <https://www.canalautismo.com.br/noticia/estudantes-criam-dispositivo-que-reduz-ruídos-em-sala-de-aula/>.
- [7] GOOGLE DOCUMENTS. *Unidade 4- Glossario de Siglas e Recursos do RP2040*. Disponível em: [https://docs.google.com/document/d/11ZK1vOQOY3kn\\_Ox4ppb5nJMUT5G6lgx\\_9BOly\\_wThlw/edit?tab=t.0](https://docs.google.com/document/d/11ZK1vOQOY3kn_Ox4ppb5nJMUT5G6lgx_9BOly_wThlw/edit?tab=t.0).
- [8] ADAFRUIT. *SSD1306 OLED Datasheet*. Disponível em: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>.
- [9] HP INFOTECH. *SSD1309 Datasheet*. Disponível em: <https://www.hpinfotech.ro/SSD1309.pdf>.
- [10] BITDOGLAB. *BitDogLab*. Disponível em: <https://github.com/BitDogLab>.