



## **EmbarcaTech - Projeto Final**

### **Residência em Sistemas Embarcados**

**Autor:**

**Tífany de Paula Severo**

**Dispenser Automatizado Para Portadores de Alzheimer**

**Brasília**

**2024**



Introdução	1
Objetivos	2
Principais Requisitos	3
Funcionamento	3
Justificativa do projeto	4
Hardware	5
Especificações Técnicas	9
Lista de Materiais	10
Pinagem	11
Circuito do Hardware	12
Software	14
Definição de variáveis	18
Inicialização do Software	19
Configuração dos registros e estrutura e formato dos dados	20
Execução do Projeto	22
Metodologia	22
Testes de Validação	23
Discussão dos Resultados	23
Referências	24

## Dispenser Automatizado Para Portadores de Alzheimer

### Introdução

O Dispenser Automatizado de Medicamentos para Pessoas com Alzheimer é um dispositivo inovador desenvolvido para auxiliar pacientes e cuidadores no controle preciso da administração de medicamentos. O projeto busca minimizar os riscos associados ao esquecimento ou à ingestão inadequada de remédios, desafios esses, comuns enfrentados por pessoas com Alzheimer e outras doenças cognitivas.

O sistema automatizado permite programar horários para a liberação dos comprimidos, garantindo que o paciente receba sua medicação corretamente e sem atrasos. Além disso, o

Coordenação:



Financiadores:





dispenser emite alertas sonoros e visuais para notificar o usuário e conta com sensores que monitoram se o medicamento foi efetivamente retirado do dispenser. Caso o paciente não tome o remédio dentro do tempo estipulado, o sistema reforça os alertas, proporcionando um nível adicional de segurança.

Com uma interface intuitiva baseada em botões e joystick, o dispositivo permite que cuidadores ou familiares programem os horários de maneira simples e rápida. Para tornar a experiência ainda mais acessível, a tecnologia pode incluir reconhecimento de voz, permitindo a confirmação dos horários programados e dos nomes dos medicamentos.

Essa proposta combina tecnologia acessível e funcionalidades avançadas para oferecer uma solução prática, segura e eficiente na gestão da medicação, proporcionando maior autonomia ao paciente e tranquilidade aos cuidadores.

O título "Dispenser Automatizado Para Portadores de Alzheimer" foi escolhido para comunicar de forma clara e objetiva o propósito do projeto. A palavra "Dispenser" destaca a funcionalidade do dispositivo de armazenar e liberar os medicamentos de forma controlada. "Automatizado" reforça o caráter inovador do projeto, evidenciando o uso da tecnologia para facilitar o processo de medicação. Por fim, a menção a "Portadores de Alzheimer" direciona o foco para o público-alvo principal, evidenciando a importância de oferecer soluções que tragam mais segurança e autonomia a pacientes que enfrentam desafios cognitivos. Este título reflete o compromisso do projeto em aliar tecnologia e acessibilidade para melhorar a qualidade de vida dos pacientes e reduzir a sobrecarga dos cuidadores, oferecendo um dispositivo funcional, intuitivo e seguro.

## Objetivos

Os objetivos do projeto são:

- 01. Assegurar a administração correta dos medicamentos:** Garantir que pacientes com Alzheimer tomem seus remédios nos horários certos, reduzindo esquecimentos e erros na dosagem.
- 02. Fornecer notificações e alertas intuitivos:** Utilizar sinais sonoros e visuais para alertar o paciente e os cuidadores sobre o horário da medicação.
- 03. Liberar medicamento no horário programado:** Utilizar um mecanismo que libera o medicamento de forma automática no horário programado.

Coordenação:



Financiadores:





- 04. Monitorar a retirada do medicamento:** Implementar sensores para verificar se o comprimido foi retirado, ativando alertas caso contrário.
- 05. Aumentar a autonomia do paciente:** Oferecer uma interface simples e acessível para que o próprio paciente ou seu cuidador possa programar o dispenser com facilidade.
- 06. Facilitar o acompanhamento por cuidadores e familiares:** Possibilitar o monitoramento do consumo da medicação para garantir a adesão ao tratamento.
- 07. Desenvolver uma solução acessível e viável:** Criar um dispositivo com componentes de fácil aquisição e custo reduzido, permitindo sua implementação em larga escala.

### Principais Requisitos

Tendo isso em vista, os principais requisitos para o projeto são os seguintes:

- 01. Interface intuitiva:** a programação de horários de medicação deve ser de forma simples e acessível.
- 02. Exibição clara das informações:** o sistema deve mostrar o horário atual, o horário programado, o status do medicamento e notificações relevantes.
- 03. Sistema de alerta eficiente:** o sistema deve emitir alertas sonoros, e se possível visuais, para garantir que o usuário perceba o horário da medicação.
- 04. Mecanismo de dispensação automatizado:** o sistema deve ser capaz de liberar automaticamente o medicamento no horário programado.
- 05. Liberar o medicamento apenas se o usuário estiver perto do dispenser:** a abertura do compartimento do dispenser só deverá ocorrer se houver usuário próximo ao dispenser, evitando assim que o remédio fique exposto desnecessariamente.
- 06. Monitoramento da retirada do medicamento:** o sistema deve ser capaz de identificar se o remédio foi retirado do compartimento ou não, emitindo alertas enquanto o comprimido não for retirado.
- 07. Precisão no controle de horário:** o sistema deve ser capaz de mostrar o horário atual de forma automática, sem a necessidade de configurá-lo manualmente.
- 08. Baixo consumo de energia e acessível:** a alimentação do sistema de ter um baixo consumo e ser acessível.
- 09. Possibilidade de comando por voz (opcional):** uso de um microfone que permita que o paciente ou cuidador registre o nome do medicamento e confira a programação sem necessidade de interação manual.

Coordenação:



Financiadores:





## Funcionamento

Descrição do funcionamento:

### 01. Inicialização:

- a. O display mostra o horário atual e em seguida a seguinte frase: *"Gostaria de programar um horário? Aperte o botão A para sim e o botão B para não"*.
- b. Se o usuário pressionar o Botão A, entra no modo de programação de alarme.

### 02. Programação do Horário:

- a. O display mostra o horário atual (via RTC).
- b. O usuário ajusta as horas movendo o joystick para cima/baixo e confirma com Botão A.
- c. Em seguida, ajusta os minutos com o joystick e confirma com Botão A.
- d. O horário programado é salvo e informado no display.

### 03. Alarme e Dispenser:

- a. Quando o horário atual (RTC) coincide com o programado:
- b. O buzzer é ativado como despertador.
- c. O sensor ultrassônico monitora a distância.
- d. Se detectar um objeto a  $\leq 40$  cm, o buzzer para e o servo motor abre a tampa, liberando o comprimido.

### 04. Verificação do Comprimido:

- a. O comprimido cai em uma bandeja com o LDR.
- b. Se o LDR detecta que o comprimido está presente (luz bloqueada), o sistema espera.
- c. Se o comprimido não for retirado em 5 minutos:

### 05. O buzzer toca novamente.

- a. Display mostra: *"O remédio ainda não foi tomado"*.

### 06. Se o comprimido for retirado (luz detectada):

- a. Display mostra: *"Remédio tomado"*.

### 07. Após a retirada do comprimido, o servo motor fecha a tampa do dispenser.

## Justificativa do projeto

O Alzheimer é uma doença neurodegenerativa que compromete a memória e a capacidade cognitiva, dificultando a adesão correta ao tratamento médico. O esquecimento da

Coordenação:



Financiadores:





medicação ou o uso inadequado dos medicamentos pode agravar a condição do paciente, comprometendo sua qualidade de vida. Diante desse cenário, um dispenser automatizado **contribui** para garantir a administração correta dos medicamentos, oferecendo maior segurança tanto para os pacientes quanto para os cuidadores. O sistema reduz a necessidade de supervisão constante, promovendo autonomia ao paciente e otimizando o tempo dos cuidadores.

De acordo com estudos realizados por Alzheimer's Association (2021), aproximadamente 55 milhões de pessoas no mundo são diagnosticadas com demências [1], sendo o Alzheimer a principal causa. Um controle rigoroso da medicação é fundamental para um tratamento adequado e para o retardamento da progressão da doença.

Além disso, a escolha da abordagem técnica do projeto foi baseada em conceitos aprendidos durante o programa, permitindo a aplicação prática dos seguintes tópicos:

- 01. PWM** para o controle do buzzer e do servo motor.
- 02. RTC** interno para manter a precisão do horário real.
- 03. Botões** para a interação com o usuário para configuração do horário.
- 04. Comunicação I2C** para a interface para o display OLED e RTC.
- 05. Conversor ADC** para a leitura precisa do sensor de luminosidade e controle do Joystick.
- 06. Interrupções** para rotinas do sistema.

Atualmente, há dispositivos eletrônicos que auxiliam na administração de remédios, mas a maioria apenas notifica o paciente do horário da medicação, sem garantir que a dose realmente foi tomada. O Dispenser Automatizado de Medicamentos para Pessoas com Alzheimer se diferencia ao integrar:

- 01.** Sensores para detectar a retirada do medicamento.
- 02.** Monitoramento da presença do paciente no momento da administração.
- 03.** Possibilidade de confirmação por voz, tornando o sistema mais acessível para pacientes com dificuldades motoras ou visuais.

Essa abordagem garante um sistema mais eficiente e humanizado, melhorando a segurança e a qualidade de vida dos pacientes.

Coordenação:



Financiadores:



## Hardware

O sistema utiliza sensores para monitorar a presença do paciente e confirmar a retirada do comprimido, além de um atuador para liberar o comprimido no horário programado, reduzindo riscos de esquecimento ou ingestão incorreta. A seguir, o Diagrama de Blocos (Figura 1) ilustra a estrutura e funcionamento do sistema:

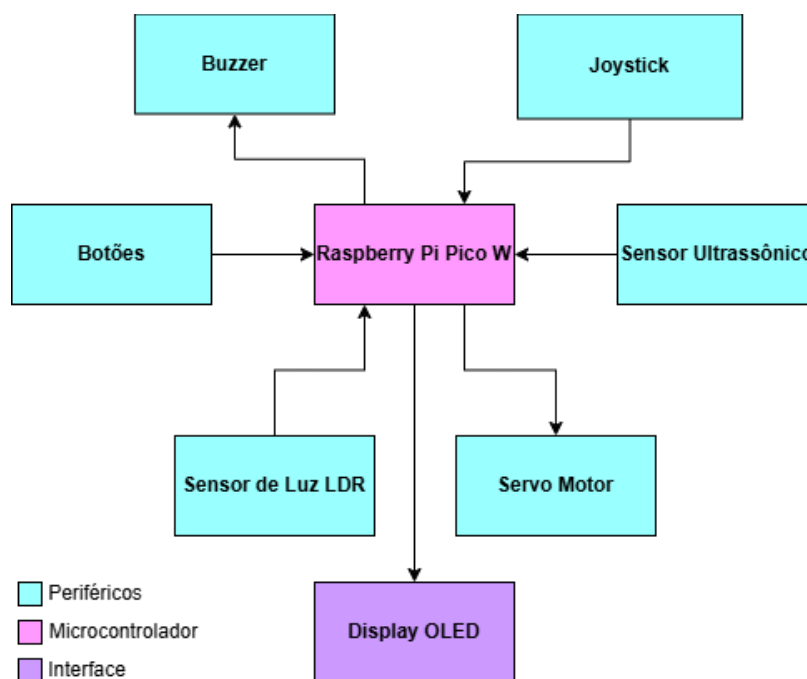


Figura 1 - Diagrama de Blocos

A tabela a seguir apresenta a função de cada bloco do sistema, descrevendo seu papel na automação do dispenser de medicamentos:

Função dos Blocos
-------------------



Bloco	Função
Microcontrolador	Processar dados dos sensores, controlar atuador e buzzer, RTC interno, módulo WiFi para obter informações de horário em tempo real, conversor ADC.
Buzzer	Emitir alerta sonoro para notificar o usuário.
Botões/Joystick	Permite configurar os horários do medicamento.
Sensor Ultrassônico	Detecta a presença do usuário para liberação do medicamento.
Sensor de Luz LDR	Confirma se o medicamento foi retirado do compartimento através da diferença de luminosidade.
Servo Motor	Abre o compartimento do dispenser após ser detectado a presença do usuário e libera o medicamento.
Display OLED	Exibe informações sobre horários e status do sistema.

Tabela 1 - Função de cada bloco

A tabela abaixo detalha as configurações utilizadas em cada bloco do sistema, garantindo seu correto funcionamento e integração:

Coordenação:



Financiadores:





Configuração dos Blocos e Pinagem	
Bloco	Configuração
Microcontrolador RP2040 (BitDogLab)	Habilitado o módulo RTC interno, o módulo WiFi interno, utilização dos slices para PWM, configuração dos GPIO como entradas ou saídas, timers interrupções, ativação de pull-ups internos, conversores ADC.
Buzzer (BitDogLab)	Conectado ao GPIO21 via transistor para controle, com frequência de 100Hz e duty cycle de 50%
Botões (BitDogLab)	<p>Botão A: GPIO5 (pull-up interno, HIGH por padrão, LOW ao pressionar).</p> <p>Botão B: GPIO6 (pull-up interno, HIGH por padrão, LOW ao pressionar).</p>
Joystick (BitDogLab)	<p>VRy: GPIO26 (eixo vertical, analógico).</p> <p>VRx: GPIO27 (eixo horizontal, analógico).</p> <p>SW: GPIO22 (botão, pull-up interno, HIGH por padrão, LOW ao pressionar).</p>
Sensor Ultrassônico HC-SR04	Trig: GPIO18 (saída para disparar o pulso).

	Echo: GPIO17 (entrada para medir o tempo do eco).
Sensor de Luz LDR	Pino: GPIO28 (ADC2, entrada analógica para medir a tensão variada pela luz).
Servo Motor SG90	Pino: GPIO16 (PWM disponível para controle do servo). O SG90 usa pulsos PWM para girar (500-2500 $\mu$ s para 0-180°; 1500 $\mu$ s é o centro, ~90° será ajustado).
Display OLED (BitDogLab)	SDA: GPIO14 e SCL: GPIO15

Tabela 2 - Configuração de cada bloco

## Especificações Técnicas

As especificações do hardware que garantem que o sistema atenda aos requisitos do usuário, proporcionando um funcionamento confiável são:

- 01. Programar alarmes e mostrar horário atual:** o uso do RTC interno e do Wifi, permitem a coleta de horário atual através da consulta em um servidor NTP, garantindo que os horários atuais sejam configurados de forma automática e confiável. O RTC interno permite a configuração de alarmes e mantém a contagem do tempo real, garantindo assim a verificação correta do horário agendado com o horário atual.
- 02. Interface intuitiva:** O uso do display OLED e botões facilita a interação do usuário proporcionando uma interface intuitiva. Este requisito precisou ser negociado e as informações serão mostradas no monitor serial no modelo MVP.
- 03. Abrir o compartimento apenas se detectar a presença do usuário:** O sensor ultrassônico HC-SR04 evita o acionamento indevido do servo motor, garantido que o

compartimento só será aberto se houver presença de um usuário há 40 cm ou menos do dispenser.

- 04. Liberar o medicamento de forma automática:** o servo motor SG90 faz o acionamento para abrir o compartimento do dispenser, automatizando assim o sistema, sem precisar da intervenção humana para abrir o compartimento.
- 05. Confirmar a retirada do remédio:** O sensor LDR, através da diferença de luminosidade, confirma a retirada do medicamento garantindo a ingestão do comprimido e aumentando a confiança do sistema.
- 06. Alerta sonoro:** O buzzer emite alertas sonoros que garante que o usuário não perca o horário da medicação.
- 07. Baixo consumo de energia:** uso de uma bateria recarregável de 3.7 V, garantindo mobilidade e operação contínua, além de ser acessível.

### Lista de Materiais

A tabela a seguir apresenta os componentes utilizados no projeto, especificando cada item necessário para a construção do dispenser automatizado:

Lista de Materiais		
Componente	Descrição	Quantidade
Microcontrolador	<b>RP2040 - Raspberry Pi Pico W BitDogLab</b>	1
Sensor Ultrassônico	<b>HC-SR04</b>	1
Sensor de Luz	<b>LDR</b>	1

Buzzer	<b>Buzzer Passivo da BitDogLab</b>	1
Servo Motor	<b>Micro servo gg - SG90 - Tower Pro</b>	1
Display OLED	<b>SSD1306 128x64 da BitDogLab</b>	1
Botões	<b>Push Botton da BitDogLab</b>	2
Joystick	<b>KY023 da BitDogLab</b>	1
Alimentação	<b>Pilha recarregável de 3,7 V</b>	1

Tabela 3 - Lista de Materiais

## Pinagem

### 01. Sensor LDR:

- VCC - 3.3 V
- D0 - GPIO 28
- GND - GND

### 02. Sensor Ultrassônico:

- VCC - VBUS
- TRIG - GPIO 18
- ECHO - GPIO 17
- GND - GND

### 03. Buzzer Passivo:

- VCC - 3.3 V
- D - GPIO 21 por meio de transistor

Coordenação:



Financiadores:





**04. Botão A:**

- a. VCC - GPIO 5
- b. GND - GND

**05. Botão B:**

- a. VCC - GPIO 5
- b. GND - GND

**06. Joystick:**

- a. VCC - 3.3 V
- b. VRy - GPIO26
- c. VRx - GPIO27
- d. SW - GPIO22
- e. GND - GND

**07. Servo Motor:**

- a. Vermelho - 5 V
- b. Amarelo - GPIO16
- c. Marrom - GND

**08. Display Oled:**

- a. VCC - 3.3 V
- b. SDA - GPIO14
- c. SCL - GPIO15
- d. GND - GND

## Circuito do Hardware

A figura a seguir apresenta o circuito completo do hardware e o esquemático, ilustrando a interconexão entre os componentes eletrônicos:

Coordenação:



Financiadores:



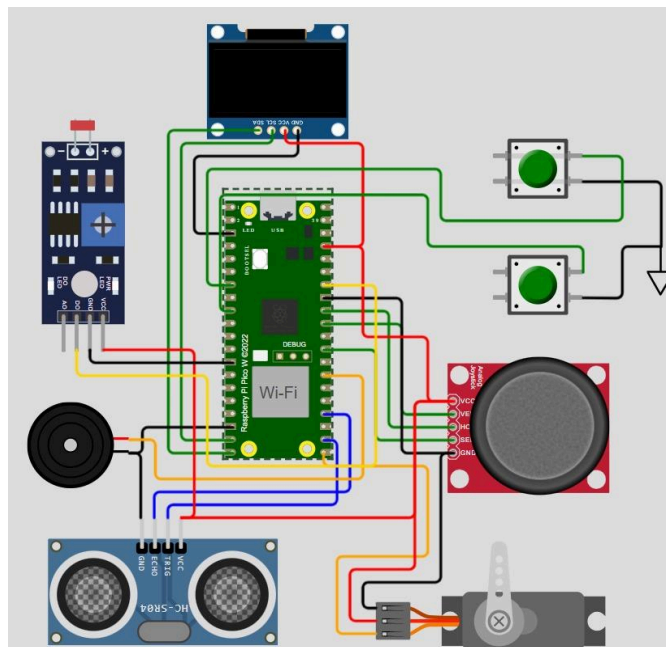


Figura 2 - Circuito completo do Hardware

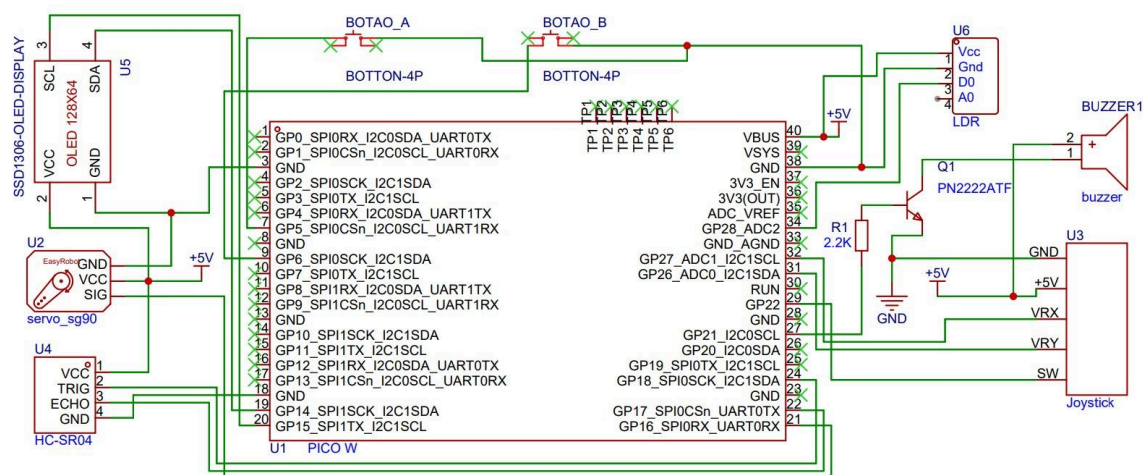


Figura 3 - Esquemático do Circuito da Eletrônica Embarcada



## Software

Nesta seção, será documentada a estrutura de software do sistema, detalhando suas principais camadas e funcionalidades. A figura 4 apresenta o diagrama de Blocos Funcionais com as camadas de Software:

- 01. Interface com o Usuário:** Responsável por exibir informações no Display OLED e gerenciar a interação com botões e joystick.
- 02. Lógica de Controle:** Processa os dados recebidos, gerencia os horários programados e ativa os alarmes quando necessário.
- 03. Sensores e Atuadores:** Controla o funcionamento do sensor ultrassônico, sensor LDR e servo motor para garantir a correta dispensação do medicamento.
- 04. Comunicação:** Gerencia a troca de informações utilizando protocolos como I2C e Módulo wi-fi.

Explicação das camadas e blocos:

### Interface com o Usuário:

- 01.** Permite que o usuário programe os horários de administração do medicamento por meio de botões e joystick.
- 02.** Oferece a opção de entrada de comandos de voz através de um microfone.
- 03.** Exibe mensagens e horários programados no display OLED para facilitar a interação.

### Controle do Dispenser:

- 01.** Gerenciado pelo microcontrolador Raspberry Pi Pico W, que armazena os horários programados.
- 02.** Ativa os alertas para lembrar o paciente de tomar o medicamento.
- 03.** Libera a medicação automaticamente no horário correto por meio do servo motor.

Coordenação:



Financiadores:





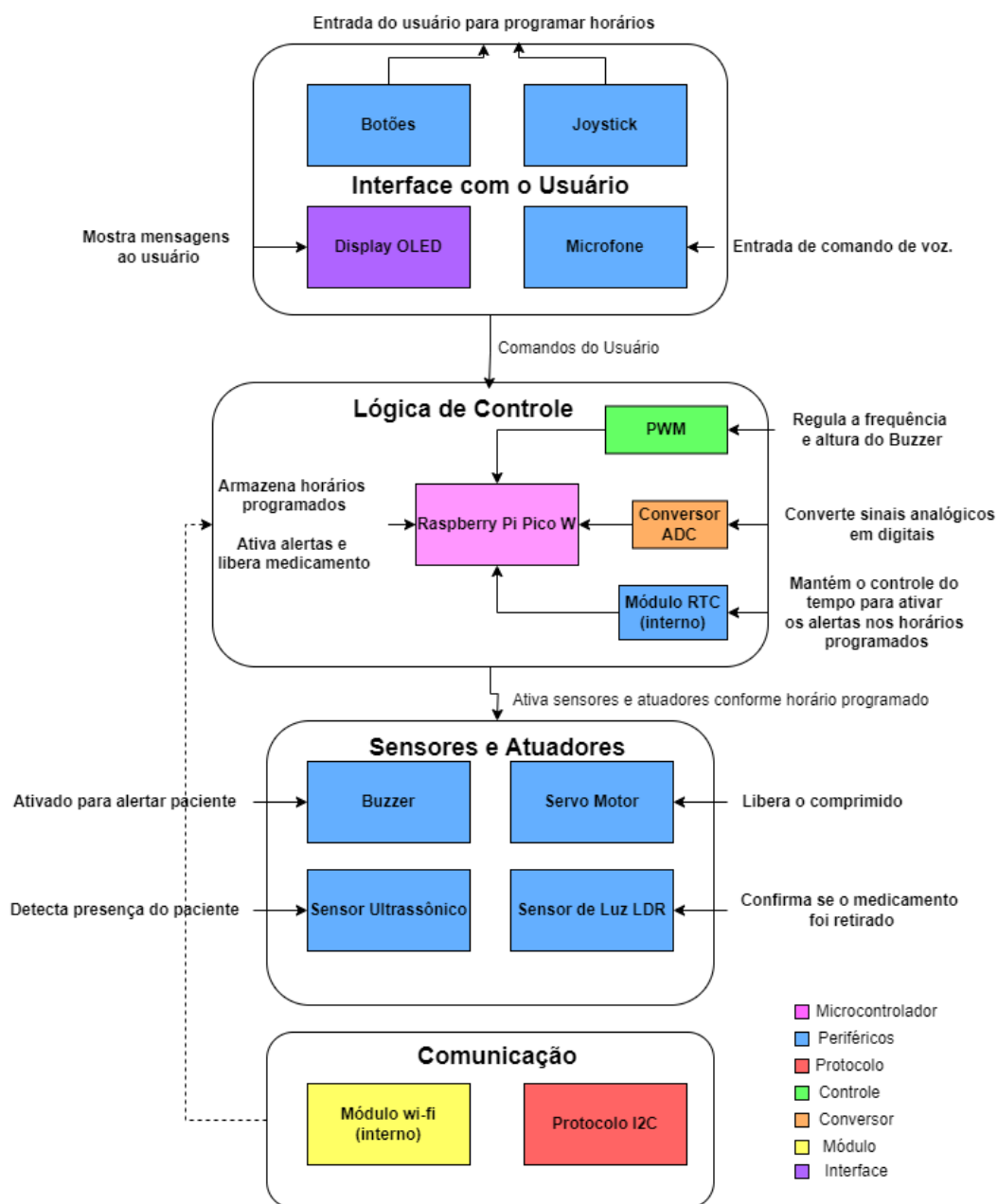


Figura 4 - Diagrama de Blocos Funcionais



### **Monitoramento de Presença:**

01. Utiliza um sensor ultrassônico para detectar a proximidade do paciente antes da liberação da medicação.
02. Garante que o medicamento seja disponibilizado apenas quando o paciente estiver presente.

### **Verificação da Medicação:**

01. O sensor de luz LDR confirma se o paciente retirou o comprimido após a liberação.
02. Caso o medicamento não seja retirado, um alerta pode ser acionado para reforçar a necessidade da ingestão.

### **Controle de Tempo:**

01. O módulo RTC (interno) mantém o controle do tempo para garantir que os alertas sejam ativados nos horários programados.

### **Alertas e Notificações:**

01. Um buzzer é ativado para emitir alertas sonoros quando for o horário do medicamento.
02. O PWM controla a frequência e altura do som do buzzer para diferentes tipos de alertas.

### **Comunicação e Expansão:**

01. Possui um módulo Wi-Fi interno, permitindo sincronização remota dos horários e notificações.
02. Utiliza o protocolo I2C para comunicação eficiente entre os diferentes módulos do sistema.

Coordenação:



Financiadores:



## Definição de variáveis

```
#define BUTTON_PIN 5 // Botão A para configurar alarme
#define JOYSTICK_VRY_PIN 26 // Pino ADC para o eixo Y do joystick
#define BUZZER_PIN 21 // Pino PWM para o buzzer passivo (GPIO 21)
#define TRIG_PIN 18 // Pino TRIG do sensor ultrassônico
#define ECHO_PIN 17 // Pino ECHO do sensor ultrassônico
#define SERVO_PIN 16 // Pino PWM para o servo SG90
#define LDR_DO_PIN 4 // Pino digital DO do módulo LDR (GPIO 4)

// Configuração da frequência do buzzer (em Hz)
#define BUZZER_FREQUENCY 100 // Frequência de 2 kHz para buzzer passivo (ajuste se necessário)

// Configurações da rede Wi-Fi
#define WIFI_SSID "nome_wifi"
#define WIFI_PASSWORD "sua_senha"

// Configurações NTP
#define NTP_SERVER "time.google.com"
#define NTP_PORT 123
#define NTP_MSG_LEN 48 // Tamanho do pacote NTP
#define NTP_DELTA 2208988800ULL // Segundos entre 1900 e 1970 (epoch Unix)

// Constantes para o ultrassônico
#define SOUND_SPEED 34300 // Velocidade do som em cm/s (a 20°C)
#define MAX_DISTANCE 400 // Distância máxima em cm (ajuste conforme necessário)

// Constantes para o servo
#define SERVO_MIN_WIDTH 550 // Ajuste para 550 us (teste valores entre 500 e 600 para 0°)
#define SERVO_MAX_WIDTH 2500 // Pulso máximo em us (180°)
#define PWM_FREQ 50 // Frequência do PWM (50Hz)

// Variáveis globais
static struct udp_pcb *udp;
static ip_addr_t ntp_server_addr;
static bool ntp_request_sent = false;

volatile bool servo_state = false; // Estado do servo (false = 0°, true = 180°)
volatile bool ldr_active = false; // Estado da leitura do LDR (false = desativado, true = ativado)
volatile bool distance_active = false; // Estado da leitura da distância (false = desativado, true = ativado)
volatile bool buzzer_active = false; // Estado do buzzer (false = desativado, true = ativado)

// Estados do sistema
enum State {
    INITIAL,
    SHOW_TIME,
    ASK_ALARM,
    SET_ALARM_HOURS,
    SET_ALARM_MINUTES,
    ALARM_SET,
    ALARM_TRIGGERED,
    MONITORING // Novo estado para monitoramento após o alarme
} current_state = INITIAL;

// Variáveis para o alarme
int alarm_hours = 0; // 0-23
int alarm_minutes = 0; // 0-59
static rtc_callback_t alarm_callback = NULL;
```

O diagrama a seguir representa o **fluxo de execução do software**, detalhando as etapas do funcionamento do dispenser automatizado:

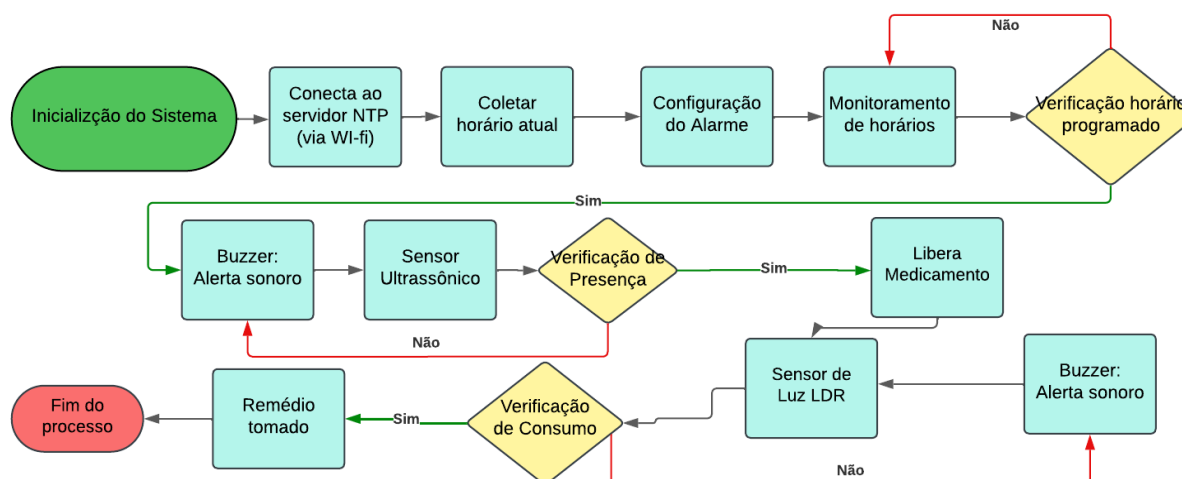


Figura 5 - Fluxograma de Software

## Inicialização do Software

A inicialização do software é a primeira etapa do processo, garantindo que o sistema esteja pronto para operar corretamente. Essa fase envolve os seguintes passos:

01. **Inicialização do Sistema** – O software inicia seu funcionamento e prepara os componentes para a execução.
02. **Conexão ao Servidor NTP (via Wi-Fi)** – O sistema estabelece conexão com um servidor NTP (Network Time Protocol) para obter o horário real atualizado.
03. **Coleta do Horário Atual** – Após a conexão, o software obtém e armazena o horário exato no RTC para controle dos alarmes.
04. **Configuração do Alarme** – O usuário pode definir os horários de administração do medicamento, que serão monitorados pelo sistema.

Após essa fase, o software entra no modo de monitoramento de horários, verificando continuamente se um alarme programado coincide com o horário atual.

## Configuração dos registros e estrutura e formato dos dados

No desenvolvimento do software para o dispenser automatizado, a configuração dos periféricos do microcontrolador **RP2040** foi realizada por meio de funções fornecidas pela SDK do Raspberry Pi Pico, que abstraem o acesso direto aos registradores de hardware. As principais funções de configuração utilizadas são descritas abaixo:

### 01. `setup_joystick()`:

- Configura o conversor analógico-digital (ADC) para leitura do eixo Y do joystick no pino GPIO 26. A função `adc_init()` inicializa o módulo ADC, enquanto `adc_gpio_init(JOYSTICK_VRY_PIN)` configura o pino como entrada analógica, ajustando os registradores internos do ADC para habilitar a conversão de sinais analógicos em valores digitais.

### 02. `setup_button()`:

- Configura o pino GPIO 5 como entrada digital para o botão. A função `gpio_init(BUTTON_PIN)` inicializa o pino, `gpio_set_dir(BUTTON_PIN, GPIO_IN)` define como entrada, e `gpio_pull_up(BUTTON_PIN)` ativa o resistor de pull-up interno.

### 03. `setup_buzzer()`:

- Configura o pino GPIO 21 como saída PWM para o buzzer passivo. A função `gpio_set_function(BUZZER_PIN, GPIO_FUNC_PWM)` associa o pino ao módulo PWM, a função `pwm_set_clkdiv()` ajusta os registradores do PWM para definir a frequência de 100 Hz e o duty cycle inicial de 50%.

### 04. `setup_ultrasonic()`:

- Configura os pinos GPIO 18 (TRIG) e GPIO 17 (ECHO) para o sensor ultrassônico. `gpio_init()` e `gpio_set_dir()` ajustam os registradores de GPIO, definindo TRIG como saída e ECHO como entrada.

### 05. `setup_servo()`:

- Configura o pino GPIO 16 como saída PWM para o servo SG90. A função `gpio_set_function(SERVO_PIN, GPIO_FUNC_PWM)` associa o pino ao PWM, e `pwm_set_wrap()` e `pwm_set_clkdiv()` configuram os registradores do PWM para uma frequência de 50 Hz, adequada ao controle do servo, inicializando-o na posição de 0°.

### 06. `setup_ldr()`:

- a. Configura o pino GPIO 4 como entrada digital para o módulo LDR. *gpio\_init()*, *gpio\_set\_dir(LDR\_D0\_PIN, GPIO\_IN)* e *gpio\_pull\_down(LDR\_D0\_PIN)* ajustam os registradores de GPIO para leitura do sinal digital do LDR, com pull-down interno para estabilidade.

#### 07. *rtc\_init()* e Configurações do RTC:

- a. Inicializa o módulo RTC (Real-Time Clock) do **RP2040**. A função *rtc\_init()* configura os registradores do RTC para operação básica, enquanto *rtc\_set\_datetime()* e *rtc\_set\_alarm()* ajustam os registradores de hora e alarme, sincronizando o sistema com o horário NTP e definindo o disparo do alarme.

Os dados utilizados foram ***alarm\_hours*** e ***alarm\_minutes*** para armazenar o horário configurado, ***datetime\_t*** para sincronizar o RTC com o horário NTP e configurar o alarme com *rtc\_set\_alarm()*. ***distance*** do tipo float para retornar o valor da distância. ***Ldr\_state*** do tipo bool para indicar o estado do LDR, ***valor\_y*** do tipo uint16\_t usado para ajustar o horário, ***current\_state*** do tipo enum State para controlar o fluxo do programa na máquina de estados. ***servo\_state***, ***ldr\_active***, ***buzzer\_active*** do tipo bool para indicar o estado do servo, ldr e o buzzer. ***Ntp\_seconds*** do tipo uint32\_t que é extraído do pacote NTP em *ntp\_recv\_callback()* e convertido para horário Unix. ***unix\_seconds*** do tipo uint32\_t convertido para a estrutura ***datetime\_t*** para sincronizar o RTC.

Os **protocolos de comunicação** utilizados no projeto foram o Protocolo I2C e o Módulo Wi-Fi interno, cada um desempenhando um papel fundamental na transmissão de dados entre os dispositivos. O Protocolo I2C (Inter-Integrated Circuit) é um protocolo de comunicação serial que permite a interconexão de múltiplos dispositivos utilizando apenas duas linhas de comunicação: uma para o clock (SCL) e outra para os dados (SDA). Já o Módulo Wifi interno foi empregado para estabelecer a conectividade sem fio, permitindo que o sistema se comunique com redes externas para obter o horário atual.

## Execução do Projeto

### Metodologia

A execução do projeto do dispenser automatizado seguiu uma metodologia estruturada, dividida em etapas distintas que envolveram pesquisa, seleção de hardware, definição de





funcionalidades, configuração do ambiente de desenvolvimento e programação, culminando na depuração do software. Abaixo, cada etapa é descrita:

#### **01. Pesquisas Realizadas:**

- a. Pesquisa sobre protocolos de comunicação (I2C e wifi) para integração eficiente dos componentes.
- b. Investigou-se o protocolo NTP (Network Time Protocol) para sincronização de horário, resultando na escolha do servidor `time.google.com` por sua confiabilidade.
- c. Foram pesquisados sensores e atuadores compatíveis: o sensor ultrassônico HC-SR04 para medição de distância, o servo SG90 para dispensação, o LDR para detecção de remédio e um buzzer passivo para alertas sonoros.
- d. Consulta a documentações [2] para entendimento da programabilidade e manuais de operação dos dispositivos [3].

#### **02. Escolha do Hardware:**

- a. Utilizando a Raspberry Pi Pico W na BitDogLab recebida no curso.
- b. Escolha do sensor ultrassônico HC-SR04 devido a sua precisão e alta documentação disponível.
- c. Escolha do sensor LDR devido a fácil utilização para o requisito estabelecido.
- d. Uso de um buzzer para alerta sonoro e um display OLED para interação com o usuário.
- e. Uso dos botões disponíveis na BitDogLab buscando minimizar conexões de fios com periféricos.

#### **03. Definição das Funcionalidades do Software:**

- a. Sincronização do horário via NTP e configurar um alarme usando o RTC interno.
- b. Permitir ao usuário definir o horário do alarme com um joystick e botão.
- c. Desenvolvimento de um sistema para gerenciar horários programados.
- d. Acionar o buzzer ao disparar o alarme, monitorar a distância com o sensor ultrassônico e abrir o servo a 100° quando a distância for  $\leq 40$  cm
- e. Continuar o som do buzzer até que o remédio seja tomado (detectado pelo LDR como "Tomado"), mantendo o servo aberto enquanto a distância for  $\leq 40$  cm e retornando-o a 0° apenas quando o remédio for retirado e a distância exceder 40 cm.

#### **04. Inicialização da IDE:**

Coordenação:



Financiadores:





- a. Foi utilizada a IDE Visual Studio Code com o plugin CMake para desenvolvimento em C, configurada com o Pico SDK. O ambiente foi inicializado seguindo o guia oficial do Raspberry Pi
- b. Instalação do Pico SDK e ferramentas de compilação (arm-none-eabi-gcc)
- c. Configuração do CMake para gerar o arquivo de build (CMakeLists.txt) com as bibliotecas necessárias (pico\_stdlib, pico\_cyw43\_arch, lwip).
- d. Conexão do Pico via USB para gravação do firmware.

#### 05. Programação na IDE:

- a. O código foi estruturado em C usando uma máquina de estados (enum State) para gerenciar as etapas.
- b. Funções específicas foram implementadas para cada periférico (ex.: *setup\_buzzer()*, *set\_servo\_angle()*, *measure\_distance()*), utilizando a Pico SDK para configurar GPIO, PWM, ADC e RTC.
- c. A sincronização NTP foi programada com as bibliotecas **lwip** para comunicação **UDP**, e o controle do alarme foi integrado ao **RTC** com um *callback* (*alarm\_handler()*).
- d. Interrupção de hardware do RTC. A interrupção é configurada pela função *rtc\_set\_alarm()*, que associa o evento de alarme ao *callback alarm\_handler()*.
- e. Implementação incremental e modular para facilitar testes e depuração.

#### 06. Depuração:

- a. Testes unitários foram realizados em cada módulo, verificando seu funcionamento individual antes da integração completa.
- b. A cada etapa, testes de comunicação e resposta dos sensores foram verificados utilizando printf() via monitor serial.
- c. Após a integração dos módulos, ajustes foram feitos para otimizar desempenho e confiabilidade.

### Testes de Validação

Para validar o funcionamento do dispenser automatizado, foram realizados diversos testes práticos, simulando condições reais de uso. Os testes abrangeram cada funcionalidade principal do sistema:

#### 01. Sincronização NTP e Configuração do Alarme com RTC:

- a. **Teste:** Conexão à rede Wi-Fi e sincronização com time.google.com.

- b. **Procedimento:** O Pico foi conectado a uma rede Wi-Fi conhecida, e o horário atual foi exibido no monitor serial.
- c. **Resultado:** O horário foi sincronizado corretamente (ex.: "Horário definido (UTC-3): 2025-02-22 12:36:44"), validando a comunicação NTP.
- d. **Teste:** Configuração do alarme com joystick e botão.
- e. **Procedimento:** O botão foi pressionado para iniciar a configuração, e o joystick foi movido para definir um alarme (ex.: 12:37).
- f. **Resultado:** O alarme foi configurado com sucesso ("Alarme configurado"), confirmado no monitor serial.

## 02. Disparo do Alarme e Ativação do Buzzer:

- a. **Teste:** Disparo automático do alarme no horário configurado.
- b. **Procedimento:** O relógio foi ajustado próximo ao horário do alarme via NTP, e o sistema aguardou o disparo.
- c. **Resultado:** O buzzer foi ativado ("Buzzer ativado no GPIO 21...") no horário correto, validando o RTC e o *callback*.

## 03. Controle do Servo e Monitoramento da Distância:

- a. **Teste:** Abertura do servo a 100° quando distância  $\leq 40$  cm.
- b. **Procedimento:** Um objeto foi aproximado a menos de 40 cm do sensor ultrassônico.
- c. **Resultado:** O servo abriu a 100° ("Servo: 100°" no print tabular)
- d. **Teste:** Fechamento do servo apenas quando o remédio for tomado e distância  $> 40$  cm.
- e. **Procedimento:** O objeto foi afastado após simular a retirada do remédio (LDR ajustado para "Tomado").
- f. **Resultado:** O servo voltou a 0° ("Servo: 0°") somente após "Tomado" e a distância ficou  $> 40$  cm, confirmando a lógica.

## 04. Teste do Sensor Ultrassônico:

- a. Medidas da distância foram verificadas com print() no monitor serial para garantir detecção precisa da presença do usuário.

## 05. Teste do Sensor LDR:

- a. Confirmação da retirada do medicamento através da detecção da mudança de luz foi confirmada com printf().

## 06. Teste de Integração:

- a. Simulação completa do ciclo do sistema para validar o funcionamento integrado.



- b. Verificação da correta comunicação entre sensores, atuadores e microcontrolador.

## Discussão dos Resultados

Os testes de validação demonstraram que o dispenser automatizado atende às funcionalidades planejadas com alta confiabilidade em condições controladas. A sincronização NTP funcionou consistentemente, garantindo precisão no disparo do alarme e confirmando o funcionamento do RTC. O controle do servo e do buzzer, integrado aos sensores de distância e LDR, respondeu adequadamente às condições definidas: o servo abriu a 100° quando a distância era  $\leq 40$  cm, e o buzzer permaneceu ativo até a retirada do remédio, retornando ao tocar se o remédio não fosse tomado e a distância excedesse 40 cm. A saída tabular no monitor serial facilitou a análise em tempo real, aumentando a usabilidade do sistema para depuração e monitoramento.

A aplicação prática do projeto demonstrou que o sistema é confiável em cenários simples, garantindo que o alarme seja disparado no horário correto e que os periféricos funcionem conforme esperado. No entanto, sua confiabilidade pode ser influenciada por fatores externos, como a estabilidade da conexão Wi-Fi para sincronização via NTP e a precisão do sensor ultrassônico em ambientes com interferências acústicas (ex.: reflexões sonoras). O sensor LDR também requer calibração para evitar falsos positivos em condições de iluminação variáveis.

O sistema se destaca pela interface intuitiva, facilitando o uso tanto para o usuário final quanto para os cuidadores. Além disso, representa uma solução de baixo custo para a dispensação automatizada de medicamentos, sendo especialmente útil para pacientes que necessitam de lembretes sonoros persistentes até a medicação ser retirada.

Sua modularidade permite diversas adaptações, como a adição de sensores adicionais ou a integração com sistemas de automação residencial. Há também potencial para expansão, incluindo a implementação de conectividade remota via Wi-Fi para envio de notificações a smartphones, aumentando ainda mais sua aplicabilidade no acompanhamento da administração de medicamentos.

Durante o desenvolvimento, foi necessário negociar os requisitos de exibição de informações no display OLED devido a dificuldades na utilização das funções para imprimir



variáveis dinamicamente, ao invés de caracteres fixos como visto em vários exemplos. Além disso, o requisito opcional de reconhecimento de voz por meio de um microfone não foi implementado, devido a priorização de outras funcionalidades essenciais do sistema.

## Conclusão

O dispenser automatizado é um protótipo funcional e confiável para uso doméstico em cenários simples, com potencial de expansão para aplicações mais avançadas.

Melhorias futuras podem incluir a implementação de reconhecimento de voz por meio de um microfone, o uso de uma câmera integrada para identificar ações e garantir que o medicamento foi devidamente administrado, e a substituição do display atual por um modelo mais intuitivo e funcional para facilitar a visualização das informações.

Além disso, a integração com um aplicativo poderia permitir o armazenamento dos dados da administração dos medicamentos e o envio de alertas para cuidadores, aprimorando o acompanhamento e a gestão da medicação.

[Vídeo do projeto](#)

[Repositório com o projeto](#)

## Referências

- [1] Alzheimer's Association. (2021). 2021 Alzheimer's Disease Facts and Figures. Disponível em: <https://www.alz.org/alzheimers-dementia/facts-figures>
- [2] Repositório oficial da BitDogLab. Disponível em: <https://github.com/BitDogLab/BitDogLab-C>
- [3] RTC - 4.8.5.4. Configuring an Alarm. Disponível em: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>

Coordenação:



Financiadores:

