Anexo A3 – Temporizador com add\_alarm\_in\_ms()

# Descrição

Este exemplo mostra o uso de um temporizador cooperativo baseado em callback, utilizando a função add\_alarm\_in\_ms(). A versão foi corrigida para garantir que o LED pisque de forma visível ao olho humano com um pequeno atraso após a mudança de estado.

# Objetivo

- Alternar o estado de um LED (GPIO 12) a cada 1 segundo  
- Usar um callback temporizado com controle de estado  
- Tornar a transição visual perceptível com sleep\_ms(100)

# Código-fonte Corrigido e Visível (main.c)

#include <stdio.h>  
#include "pico/stdlib.h"  
#include "hardware/timer.h"  
  
#define LED\_PIN 12  
  
volatile bool alternar\_led = false;  
bool estado\_led = false;  
  
int64\_t temporizador\_callback(alarm\_id\_t id, void \*user\_data) {  
 alternar\_led = true;  
 return 1000;  
}  
  
int main() {  
 stdio\_init\_all();  
  
 gpio\_init(LED\_PIN);  
 gpio\_set\_dir(LED\_PIN, GPIO\_OUT);  
  
 add\_alarm\_in\_ms(1000, temporizador\_callback, NULL, true);  
  
 while (true) {  
 if (alternar\_led) {  
 estado\_led = !estado\_led;  
 gpio\_put(LED\_PIN, estado\_led);  
 printf("LED = %d\n", estado\_led);  
 sleep\_ms(100);  
 alternar\_led = false;  
 }  
 tight\_loop\_contents();  
 }  
}

# Notas sobre a Visibilidade do LED

- O uso de gpio\_put com sleep\_ms(100) garante que a mudança de estado do LED seja visível.  
- Esse comportamento melhora a percepção do funcionamento correto do sistema embarcado.  
- tight\_loop\_contents() mantém o sistema responsivo sem bloquear o loop principal.