Resumo do Projeto: Três Tarefas com FreeRTOS no Raspberry Pi Pico W

# 1) Objetivos do Projeto

Este projeto tem como objetivo demonstrar o uso de três tarefas concorrentes rodando no sistema operacional de tempo real FreeRTOS, aplicadas ao microcontrolador Raspberry Pi Pico W. Cada tarefa realiza uma função independente, com períodos distintos e prioridades configuradas.  
  
Tarefas implementadas:  
- tarefa\_led: Pisca um LED no GPIO 13 a cada 250ms.  
- tarefa\_usb: Imprime mensagens de status no terminal a cada 500ms.  
- tarefa\_monitoramento: Lê dois potenciômetros via ADC (GPIO26 e GPIO27) e imprime suas tensões a cada 1 segundo.

# 2) Implementação na BitDogLab

O projeto pode ser carregado na placa BitDogLab, que utiliza o Raspberry Pi Pico W como base. A BitDogLab permite acesso fácil aos GPIOs, ADCs e comunicação USB, facilitando:  
- Conexão de LEDs diretamente aos pinos.  
- Conexão de potenciômetros nos canais ADC.  
- Comunicação serial via cabo USB-C.  
  
Para compilar o projeto, utiliza-se o VSCode com suporte ao SDK Pico + FreeRTOS, ou o ambiente automatizado do BitDogLab, se houver suporte integrado.

# 3) Onde ver os resultados

Os resultados são exibidos no terminal serial USB:  
- Ao conectar o Pico W ao computador, um terminal como o PuTTY, TeraTerm, Minicom ou o próprio serial monitor do VSCode pode ser usado.  
- A velocidade padrão da porta é geralmente 115200 bps.  
  
No terminal, você verá:  
- O estado do LED (ON/OFF)  
- Mensagens periódicas do sistema  
- Tensões dos dois potenciômetros convertidas em volts

# 4) Como alterar a prioridade das tarefas

As prioridades são definidas na chamada xTaskCreate():  
  
xTaskCreate(tarefa\_led, "LED", 256, NULL, 5, NULL);  
xTaskCreate(tarefa\_usb, "USB", 256, NULL, 4, NULL);  
xTaskCreate(tarefa\_monitoramento, "Monitor", 256, NULL, 3, NULL);  
  
- O último número antes do NULL é a prioridade da tarefa.  
- Quanto maior o número, maior a prioridade (0 é a menor).  
- Alterar a prioridade pode afetar a ordem de execução e preempção das tarefas.  
  
Exemplos:  
- Dar prioridade máxima à leitura de sensores se for crítico.  
- Reduzir a prioridade do printf da USB se interferir com tarefas mais importantes.