Controle de LEDs com Medição de Tempo de CPU e Simulação de Carga no FreeRTOS (Raspberry Pi Pico W)

# 1) Tempo de execução sem carga (tarefas "leves")

No Raspberry Pi Pico W, a operação básica de uma tarefa que apenas alterna o estado de um GPIO (acender ou apagar um LED) e imprime o tempo no terminal é extremamente rápida. Usando `get\_absolute\_time()` e `absolute\_time\_diff\_us()`, observamos tempos de execução em torno de 2 a 5 microssegundos (us) para essas tarefas, sem nenhuma carga computacional adicional.

# 2) Ativando simulação de carga computacional

Para simular tarefas mais "pesadas", adicionamos laços de iteração vazios, que ocupam a CPU por alguns microssegundos. Isso nos permite medir e comparar a utilização efetiva da CPU.  
  
Ative as diretivas de carga:  
#define CARGA\_LED11 55000  
#define CARGA\_LED12 75000  
#define CARGA\_LED13 80000  
  
E utilize a função:  
void carga\_computacional(int ciclos) {  
 for (volatile int i = 0; i < ciclos; i++) {  
 \_\_asm volatile("");  
 }  
}

# 3) Inserindo carga em cada tarefa

Em cada uma das três tarefas, ativamos a simulação de carga com a chamada:  
carga\_computacional(CARGA\_LEDxx);  
Isso é feito logo após o gpio\_put(), antes da medição de tempo final.

# 4) Medindo o tempo de execução com carga

Após ativar a carga, utilizamos get\_absolute\_time() antes e depois da alternância do LED e da carga. O tempo medido representa o tempo total de CPU ocupado pela tarefa em uma execução.  
  
Exemplo de saída esperada no terminal:  
LED11 - Tempo de CPU: 1450 us  
LED12 - Tempo de CPU: 1800 us  
LED13 - Tempo de CPU: 1950 us

# 5) Cálculo da utilização do processador

Com os tempos de CPU por tarefa e seus respectivos períodos (vTaskDelay), calculamos a utilização da CPU:  
  
U = Σ(Ci / Ti)  
  
Exemplo de cálculo:  
Tarefa | Execução (Ci) | Período (Ti) | Utilização  
LED11 | 1,450 ms | 500 ms | 0,29%  
LED12 | 1,800 ms | 700 ms | 0,26%  
LED13 | 1,950 ms | 900 ms | 0,22%  
TOTAL | | | 0,77%