Análise Completa da Atividade 3 – Interrupções com Verificação de Estado e Debounce Seguro

# 1. Explicação Detalhada dos Códigos

Na Atividade 3, o projeto visa controlar três LEDs com três botões físicos (A, B e o botão do joystick) usando uma lógica aprimorada de interrupções.  
  
Diferente das versões anteriores, aqui temos:  
• Desabilitação temporária da interrupção para evitar múltiplos disparos (bouncing).  
• Debounce tratado fora da ISR (interrupt service routine).  
• Verificação adicional do estado do botão antes de executar a ação.  
• Reativação da interrupção apenas após o evento ser processado.  
• Economia de energia via instrução \_\_wfi().

A função gpio\_callback apenas sinaliza o evento e desabilita a interrupção do pino correspondente. O loop principal verifica se há eventos pendentes, aguarda o tempo de debounce, confirma se o botão ainda está pressionado e, então, alterna o estado do LED. Ao final, a interrupção é reabilitada.

# 2. Respostas às Perguntas do Autor

## 2.1 Questão 1 – Teste de Funcionamento

Durante a execução do código, ao pressionar os botões A, B e do joystick de forma repetida, rápida ou em diferentes sequências, observa-se que os LEDs reagem corretamente: alternam o estado apenas quando o botão correspondente é realmente pressionado.  
  
Não são detectadas mudanças inesperadas de estado. Isso confirma que a lógica de debounce e verificação adicional do botão no loop principal estão funcionando corretamente. O sistema permanece estável mesmo sob pressões rápidas ou simultâneas.

## 2.2 Questão 2 – Alterações na Função gpio\_callback

A função gpio\_callback sofreu uma mudança importante: agora ela apenas sinaliza o evento e desativa a interrupção do botão, deixando o tratamento completo para o loop principal. Isso representa uma separação clara entre a detecção (interrupção) e o processamento (loop principal).  
  
Na lógica anterior, o debounce era feito na própria ISR ou dependia do tempo. Essa abordagem poderia falhar em casos de bouncing ou ruído elétrico, causando acionamentos indesejados dos LEDs mesmo sem o botão ser pressionado de fato.  
  
Agora, com a interrupção desativada temporariamente e a verificação adicional após o tempo de debounce, evita-se múltiplos disparos e garante-se que apenas uma pressão física real altere o estado do LED. Essa mudança foi essencial para tornar o sistema confiável e estável.

# 3. Conclusão

A Atividade 3 representa um avanço significativo no uso de interrupções com GPIO no Raspberry Pi Pico. A lógica é robusta, segura e escalável para aplicações embarcadas reais. O uso de \_\_wfi(), desativação/reabilitação de interrupções, e verificação pós-debounce mostra boas práticas de engenharia embarcada.