## 📘 Entendendo Debounce por Tempo em Sistemas Digitais

### 🔧 O que é "debounce"?

Em sistemas digitais, **debounce** (ou **anti-repique**) é a técnica usada para **eliminar leituras instáveis** geradas por dispositivos mecânicos, como **botões, teclas, e joysticks**, que oscilam rapidamente entre **ligado/desligado (HIGH/LOW)** ao serem pressionados ou soltos.

---

### 🛠️ Por que isso acontece?

Componentes como **chaves tácteis** não são ideais. Ao pressionar um botão, os contatos internos podem **vibrar ou oscilar por poucos milissegundos**, gerando **pulsos espúrios** que os sistemas digitais interpretam como múltiplos eventos.

**Sem debounce:**
Pressionar uma vez pode gerar **2, 3 ou mais "cliques"** acidentalmente.

---

### ⏱️ Debounce por Tempo (Software)

Uma das formas mais comuns e eficientes de tratar o debounce é **aguardar um tempo mínimo (delay)** após detectar uma borda (subida ou descida), e só então validar o estado do sinal.

---

### 🧠 Lógica geral do debounce por tempo:

1. Detecta-se uma transição no pino (ex: botão foi pressionado → borda de descida).
2. Inicia-se um temporizador (delay).
3. Após o tempo definido (ex: 20ms), verifica-se **se o botão ainda está pressionado**.
4. Se estiver, é considerado um clique **válido**.

---

### ✅ Vantagens do debounce por tempo:

- Fácil de implementar em microcontroladores ou em software.
- Não requer hardware adicional.
- Controlável: você define quanto tempo é “seguro” para eliminar ruídos.

---

### 📊 Quanto tempo usar como delay?

- Valores típicos: **10 a 50 ms**
- Para chaves mecânicas simples: **20 ms** costuma ser suficiente.
- Pode ser ajustado conforme o tipo de botão ou teste em campo.

---

### 🔌 Comparativo com outras técnicas

| Técnica | Vantagem | Desvantagem |
|---|---|---|
| **Debounce por tempo** | Simples, sem hardware | Pode atrasar respostas rápidas |
| Filtro por média | Mais robusto contra ruído | Mais código / processamento |
| Hardware RC | Sem uso de CPU | Requer componentes adicionais |
| Debounce por interrupção + temporizador | Eficiente em energia / eventos | Mais complexo |

---

### 🧪 Aplicações típicas

- Leitura de botões em microcontroladores (Arduino, STM32, RP2040…)
- Interfaces homem-máquina (HMI)
- Contadores com botões
- Jogos com joysticks analógicos e digitais

---