

## Laboratório (Intermediário)

### Pisca-Pisca com Padrões Alternáveis via Botão

Neste experimento, você criará um sistema de iluminação com 4 LEDs onde é possível alternar entre dois padrões de pisca-pisca utilizando um botão de pressão.

#### Materiais Necessários:

- Arduino Uno.
- 4 LEDs (de cores diferentes, se possível).
- 4 resistores de 220 ohms.
- 1 Botão de pressão.
- Cabos jumper (fios de conexão).
- Protoboard.

### 1. Requisitos do Projeto

- Ao iniciar, o sistema deve executar um padrão fixo de pisca-pisca.
- Ao pressionar o botão, o padrão deve mudar (ex: piscar em sequência → piscar pares alternados).
- O botão deve alternar continuamente entre os modos.
- Use lógica condicional e controle de estados.

## Laboratório (Avançado)

### Pisca-Pisca Natalino com Animação Aleatória

Neste laboratório, o desafio será programar 4 LEDs com padrões de pisca-pisca que se alteram de forma aleatória, simulando luzes natalinas dinâmicas.

#### Materiais Necessários:

- Arduino Uno.
- 4 LEDs (de cores variadas).
- 4 resistores de 220 ohms.
- Cabos jumper (fios de conexão).
- Protoboard.

### 1. Requisitos do Projeto

- Os LEDs devem piscar em combinações aleatórias (ex: 1 acende, depois 2 e 4, depois 3, etc.).
- Utilize a função `random()` para gerar diferentes padrões a cada iteração.
- Os tempos de piscar também devem ser aleatórios dentro de um intervalo definido.
- Não é necessário entrada do usuário, apenas lógica aleatória contínua.

### 2. Dicas

- Você pode usar `random(0, 2)` para decidir se um LED fica ligado ou desligado em cada ciclo.
- Adicione uma pequena pausa entre ciclos (100ms a 500ms, por exemplo).

## Laboratório (Desafio)

### Sistema de Luzes Natalinas com Controle de Efeito e Brilho

Neste desafio, você vai projetar um sistema completo de luzes natalinas com controle de padrão e de intensidade de brilho usando um botão e um potenciômetro.

**Objetivo:** Criar um sistema com múltiplos efeitos de iluminação e controle de brilho:

- O botão deve alternar entre diferentes padrões de pisca-pisca.
- O potenciômetro deve controlar o nível de brilho global dos LEDs (via PWM).
- Os LEDs devem piscar de forma suave e responsiva.
- É obrigatório o uso de `analogWrite()` e `millis()` para controle não bloqueante.

### Materiais Necessários:

- Arduino Uno.
- 4 LEDs.
- 4 resistores de 220 ohms.
- 1 botão de pressão.
- 1 potenciômetro (10k).
- Cabos jumper.
- Protoboard.

### Restrições:

- Evite usar `delay()` no `loop()`. Prefira `millis()`.
- O controle de brilho deve ser proporcional ao valor do potenciômetro.
- O botão deve trocar os efeitos suavemente sem reiniciar o programa.

Boa sorte! Desafios com múltiplos modos de funcionamento exigem criatividade e organização do código.