

Aplicações Práticas

Agora que entendemos os conceitos de entradas e saídas digitais, vamos aplicá-los em alguns projetos simples.

- [Projeto 1: Controlar um LED com um botão](#)
- [Projeto 2: Luz Automática com Sensor de Movimento \(PIR\)](#)
- [Desafios](#)
- [Conclusão](#)

Projeto 1: Controlar um LED com um botão

Projeto 1: Controlar um LED com um botão

Objetivo: Acender um LED quando um botão for pressionado, e apagá-lo quando o botão for solto.

Circuito:

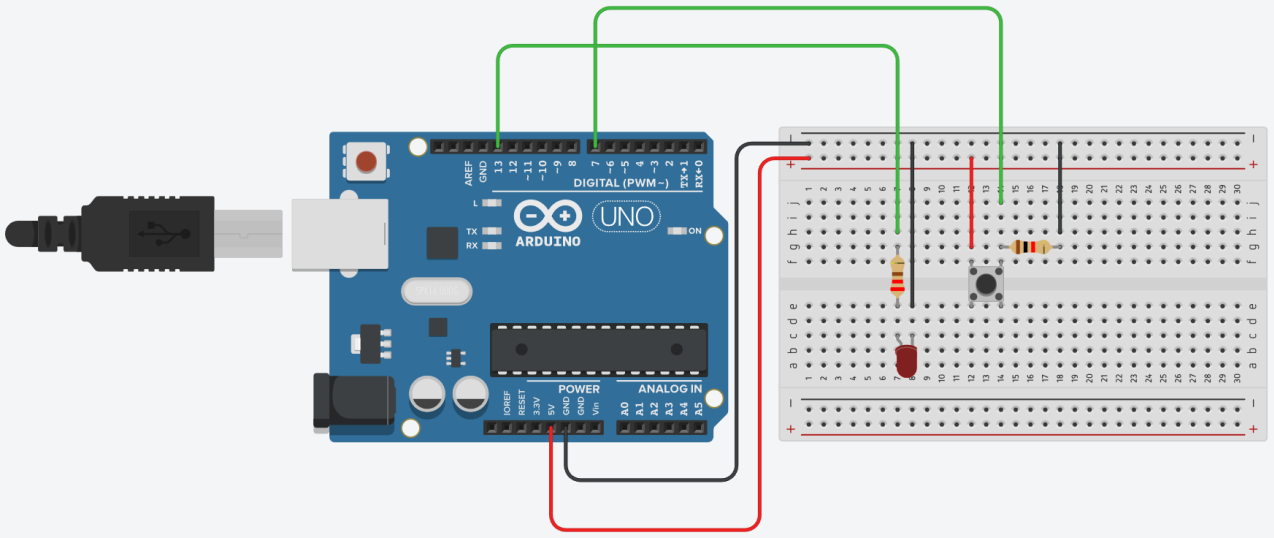
- Conecte um botão ao pino 7 do Arduino e ao GND. Conforme o Desenho
- Conecte um LED ao pino 13 e ao GND (com um resistor de 220Ω para proteger o LED).

Código:

```
int buttonPin = 7;
int ledPin = 13;
int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```



Projeto 2: Luz Automática com Sensor de Movimento (PIR)

Projeto 2: Luz Automática com Sensor de Movimento (PIR)

Objetivo: Ligar uma lâmpada ou LED automaticamente quando detectar movimento com um sensor PIR (Passive Infrared).

Componentes:

- Sensor PIR conectado ao pino 2.
- LED ou lâmpada conectada ao pino 13.

Circuito:

- Conecte o pino de saída do sensor PIR ao pino 2 do Arduino.
- Conecte o LED ao pino 13 e ao GND.

Código:

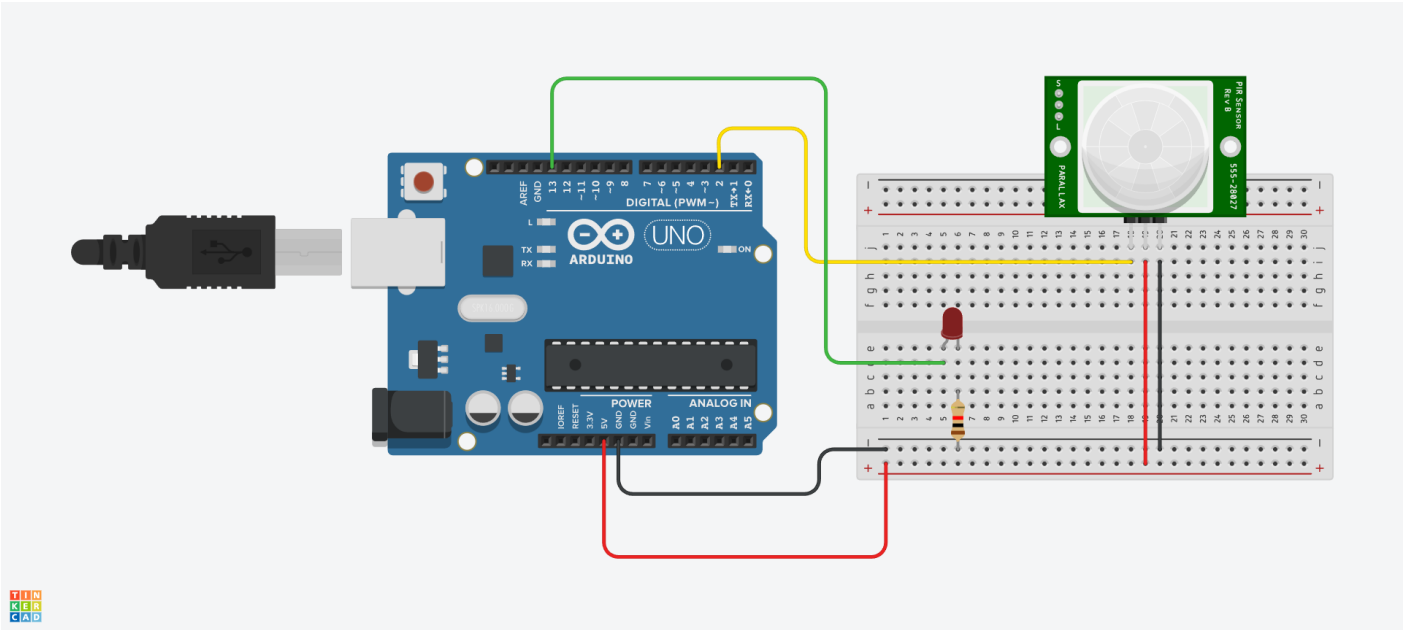
```
int pirPin = 2; // Pino onde o sensor PIR está conectado
int ledPin = 13;

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int pirState = digitalRead(pirPin); // Lê o estado do sensor PIR
  if (pirState == HIGH) {
    digitalWrite(ledPin, HIGH); // Liga o LED se houver movimento
  } else {
    digitalWrite(ledPin, LOW); // Desliga o LED se não houver movimento
  }
}
```

}

}



Desafios

Agora que você aprendeu o básico de entradas e saídas digitais, tente os seguintes desafios:

1. **Desafio 1:** Crie um código que acenda um LED apenas quando o botão for pressionado três vezes consecutivas.
2. **Desafio 2:** Adicione um buzzer ao circuito com o sensor PIR e faça-o emitir um som quando o sensor detectar movimento.

Conclusão

Conclusão

Neste capítulo, você aprendeu a configurar e usar as entradas e saídas digitais no Arduino. Usamos as funções `pinMode()`, `digitalWrite()` e `digitalRead()` para controlar LEDs e ler botões e sensores. Esses são conceitos fundamentais para praticamente todos os projetos com Arduino. No próximo capítulo, vamos explorar as entradas e saídas analógicas, permitindo o controle mais preciso de componentes como LEDs e motores.