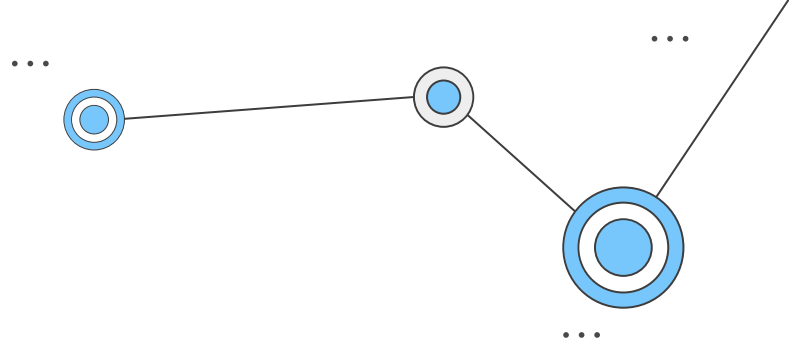




PUC Minas



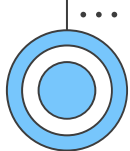
# Laboratório de Iniciação à Programação

Prof. Dr. João Paulo Aramuni

# Aula 13

## Exercícios

LIP - Manhã



# Exercícios

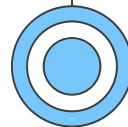
1 - Alarme de intruso: Crie um sistema de alarme simples onde um LED acende quando um botão é pressionado e mantém-se aceso até que o botão seja pressionado novamente para desativá-lo.

2- Contador de cliques de botão com LED: Faça um circuito que conte quantas vezes um botão é pressionado em 2s e exiba esse número piscando em um LED.

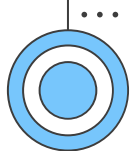
...



...



...



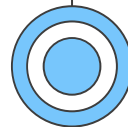
# Exercícios

3 - Simulação de semáforo com botão: Crie um semáforo simples com três LEDs representando vermelho, amarelo e verde. Use um botão para simular o botão de pedestres. Quando o botão é pressionado, o semáforo deve mudar para a luz vermelha durante alguns segundos (para o pedestre passar) antes de voltar ao ciclo normal.

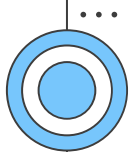
...



...



...



# Resolução ex1

```
int LED = 2;
int BOTAO = 7;

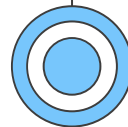
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(BOTAO, INPUT);
}
bool ledON = false;

void loop()
{
  if(digitalRead(BOTAO) == HIGH)
  {
    ledON = !ledON;
    delay(500);
  }
  if(ledON)
  {
    digitalWrite(LED, HIGH);
  } else
  {
    digitalWrite(LED, LOW);
  }
}
```

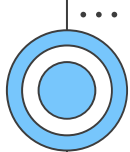
Mostrar/ocultar monitor serial



...



...



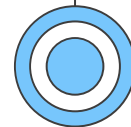
# Resolução ex2

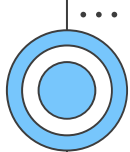
```
// Definição dos pinos
const int buttonPin = 2; // Pino do botão
const int ledPin = 13;   // Pino do LED

// Variáveis de controle
int clickCount = 0;      // Contador de cliques
unsigned long lastClick; // Armazena o momento do último clique
unsigned long currentTime; // Armazena o tempo atual

// Função setup() - Executa uma vez ao iniciar
void setup() {
  pinMode(buttonPin, INPUT_PULLUP); // Configura o pino do botão como entrada com pull-up
  pinMode(ledPin, OUTPUT);          // Configura o pino do LED como saída
  Serial.begin(9600);               // Inicia a comunicação serial (opcional, para debug)
}
```

...





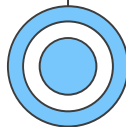
# Resolução ex2

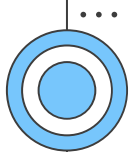
```
// Função loop() - Executa repetidamente
void loop() {
    currentTime = millis(); // Obtém o tempo atual em milissegundos

    // Verifica se o botão foi pressionado
    if (digitalRead(buttonPin) == HIGH) {
        // Verifica se o último clique foi há mais de 2 segundos
        if (currentTime - lastClick > 2000) {
            clickCount = 0; // Reinicia o contador de cliques
        }

        clickCount++; // Incrementa o contador de cliques
        lastClick = currentTime; // Atualiza o momento do último clique
    }

    // Exibe o número de cliques no LED
    for (int i = 0; i < clickCount; i++) {
        digitalWrite(ledPin, HIGH); // Liga o LED
        delay(500); // Aguarda 500ms
        digitalWrite(ledPin, LOW); // Desliga o LED
        delay(500); // Aguarda 500ms
    }
}
```





# Resolução ex3

```
int ledVermelho = 2;
int ledAmarelo = 3;
int ledVerde = 4;
int BOTAO = 7;
```

```
void setup()
{
  pinMode(ledVermelho, OUTPUT);
  pinMode(ledAmarelo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(BOTAO, INPUT);
}
```

```
void loop()
{

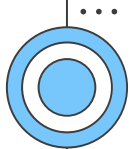
  digitalWrite(ledVerde, HIGH);
  for(int i = 0; i < 12; i++){
    if(digitalRead(BOTAO) == HIGH)
    {
      break;
    }
    delay(500);
  }
  digitalWrite(ledVerde, LOW);

  digitalWrite(ledAmarelo, HIGH);
  delay(1500);
  digitalWrite(ledAmarelo, LOW);

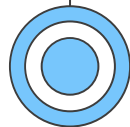
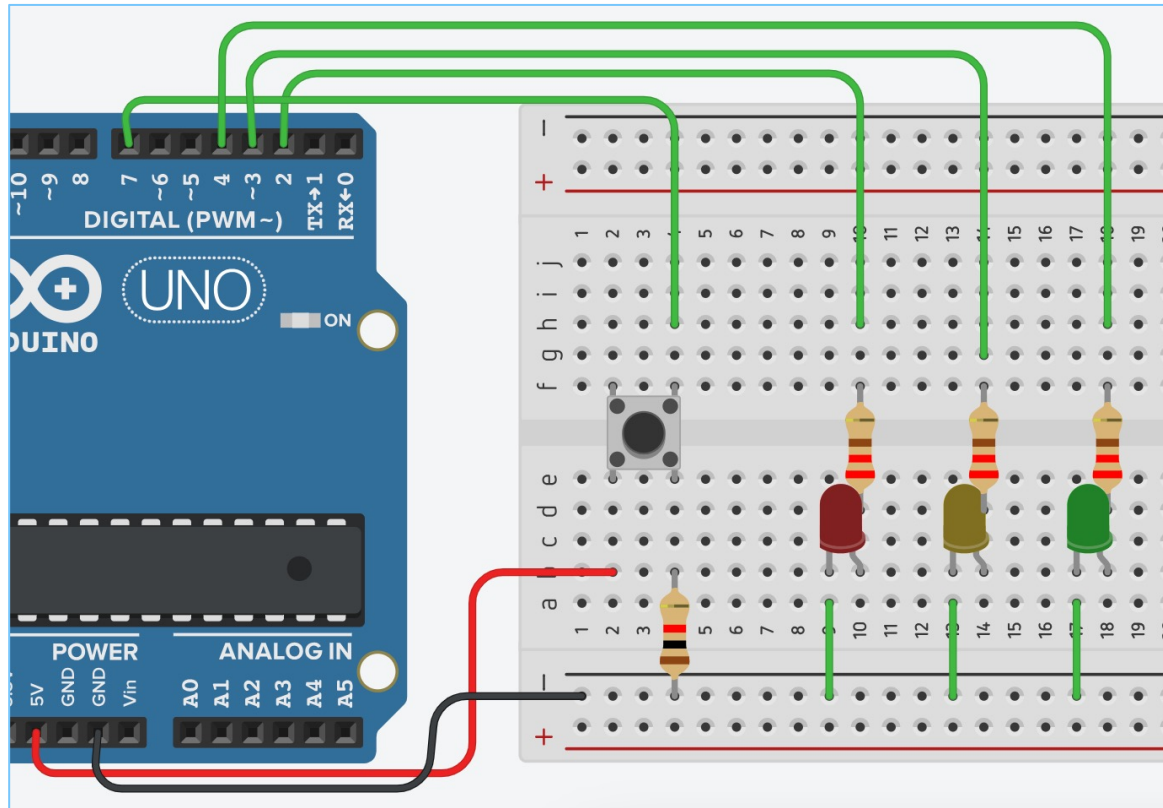
  digitalWrite(ledVermelho, HIGH);
  delay(3000);
  digitalWrite(ledVermelho, LOW);

}
```





# Resolução ex3



# Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...