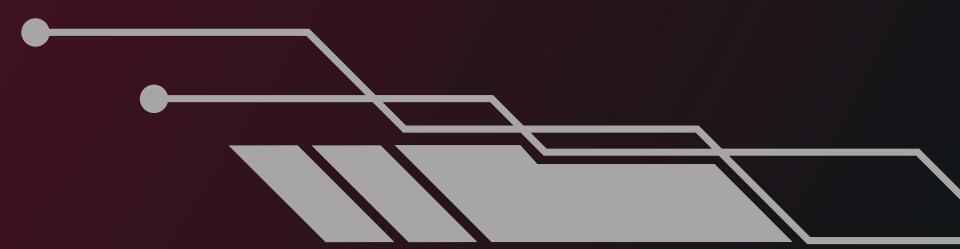




TRAINING





RETROSPECTIVA

- O QUE É UMA PROTOBOARD?
- COMO USAR UMA PROTOBOARD
- CORRENTE, TENSÃO E RESISTÊNCIA
- PRIMEIRA LEI DE OHM

MONTAGEM NA PROTOBOARD



BOOTCAMP DA
ROBOCAMP

- USO DE CONSTANTES E VARIÁVEIS
- LACOS CONDICIONAIS

SIRENE LED



BOTÕES E FUNÇÕES NO ARDUINO

- USO DE BOTÕES
- USO DE FUNÇÕES EM ARDUINO

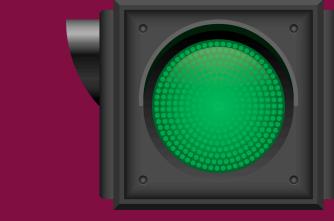
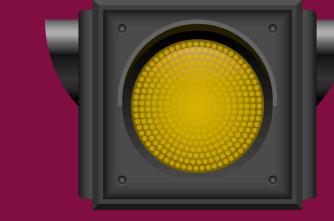
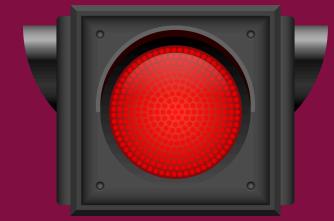


EXERCÍCIOS DE FIXAÇÃO

USO DE BOTÕES

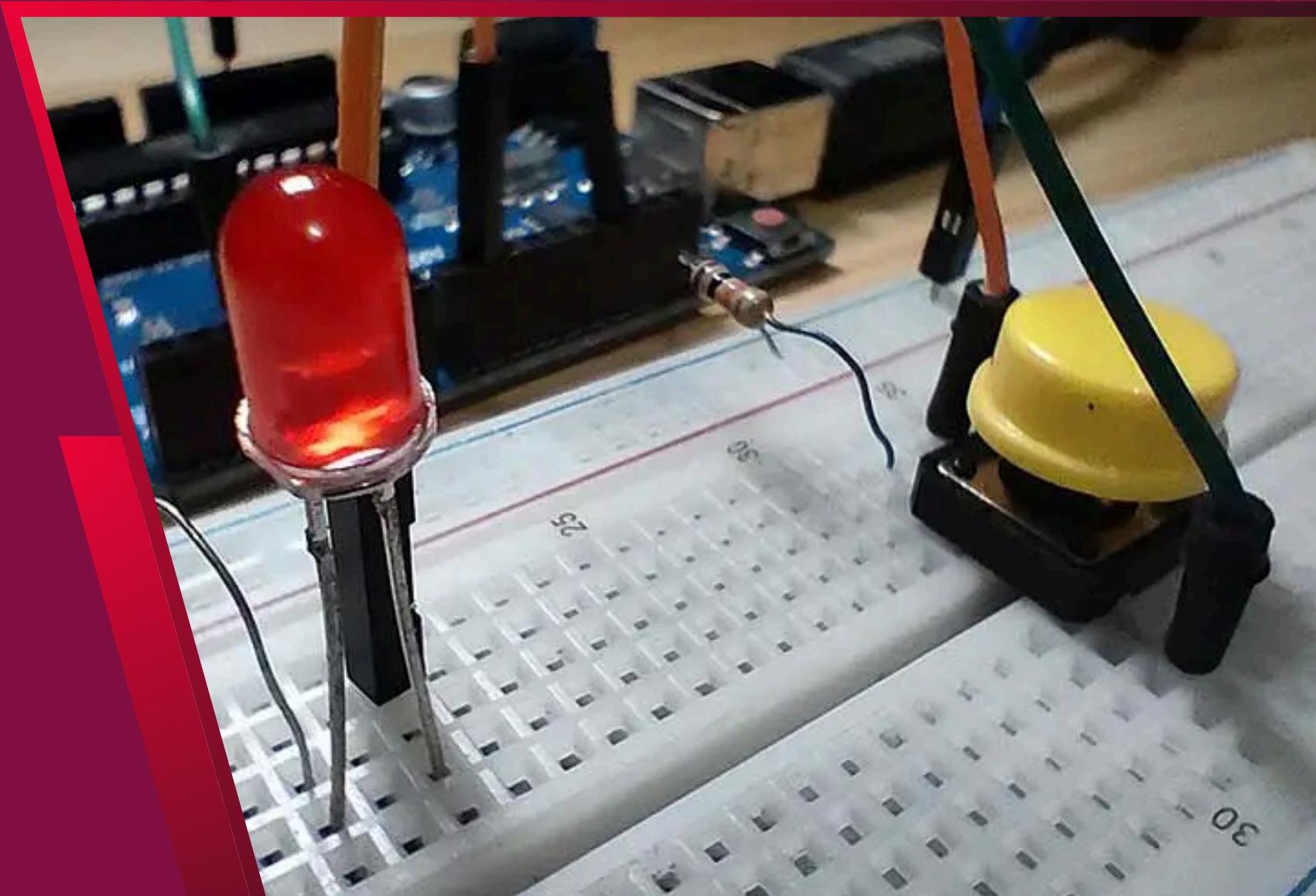


SEMÁFORO SIMPLES



USO DE BOTÕES NO ARDUINO

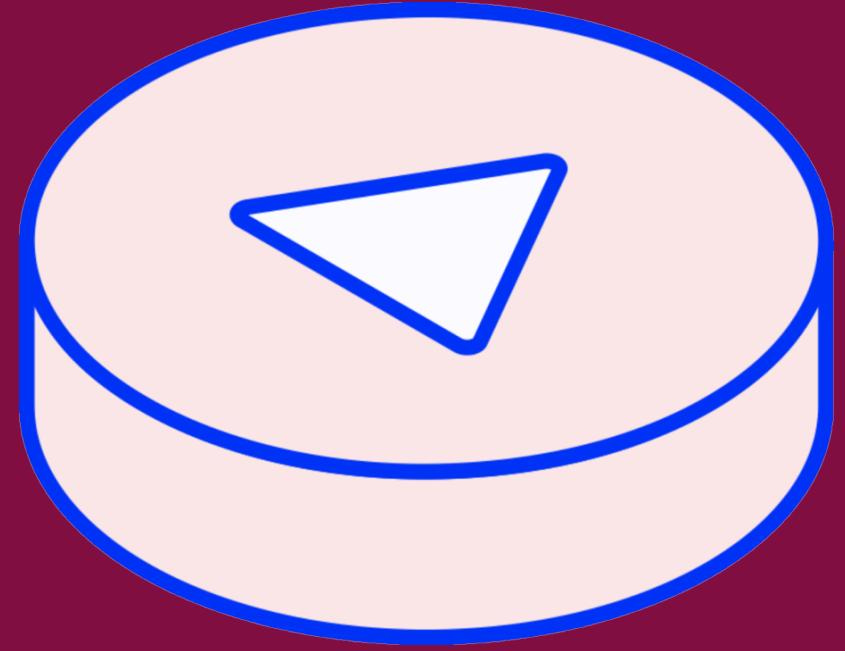
- INTRODUÇÃO AO PUSHBUTTON E SEU FUNCIONAMENTO
- CONEXÃO DO BOTÃO AO ARDUINO
- DIAGRAMA DE CONEXÃO
- PROGRAMAÇÃO DE BOTÕES



PUSH BUTTON

O QUE É UM BOTÃO PUSHBUTTON

- COMPONENTE QUE CONTROLA O FLUXO DA CORRENTE ELÉTRICA EM UM CIRCUITO.
- ATUA COMO UMA CHAVE:
 - PRESSIONADO
 - SOLTO



TIPOS DE BOTÕES PUSHBUTTON

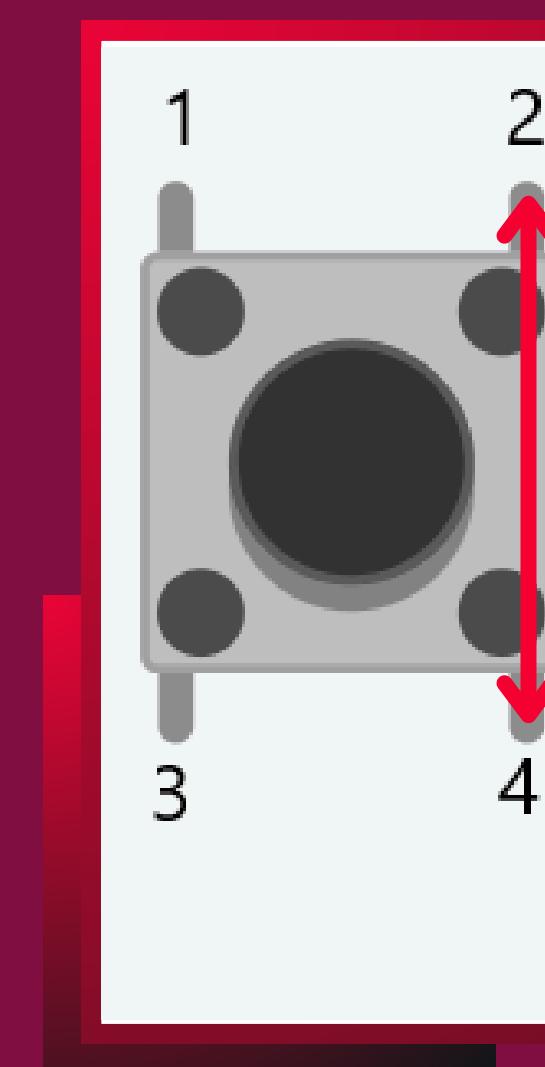
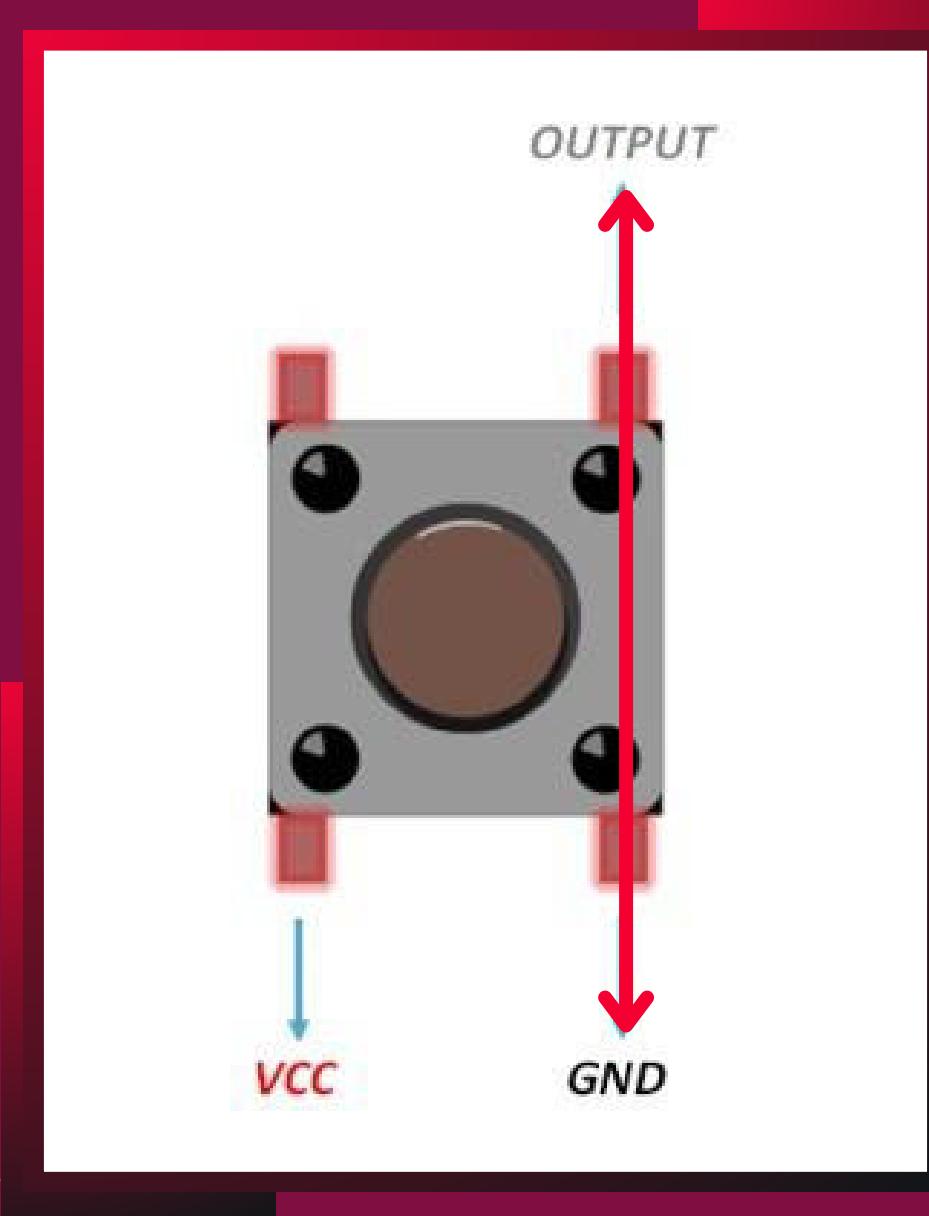
NORMALMENTE ABERTO (NA) :

- NÃO CONDUZ CORRENTE QUANDO EM REPOUSO
- CONDUZ CORRENTE APENAS QUANDO PRESSIONADO

NORMALMENTE FECHADO (NF) :

- CONDUZ CORRENTE QUANDO EM REPOUSO
- INTERROMPE A CONDUÇÃO QUANDO PRESSIONADO

DIAGRAMA BOTÃO EM REPOUSO



Pin 1 & 3 are connected
Pin 2 & 4 are connected

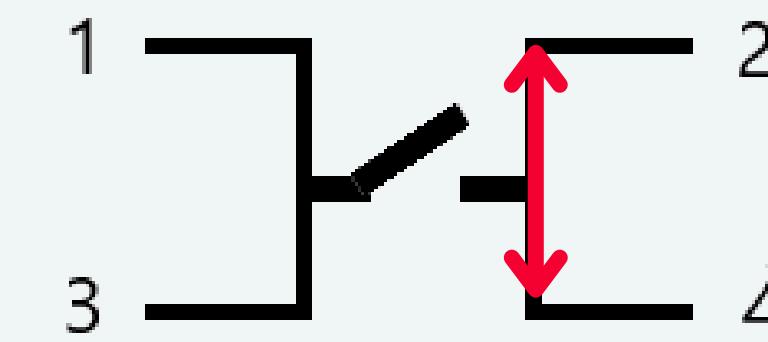
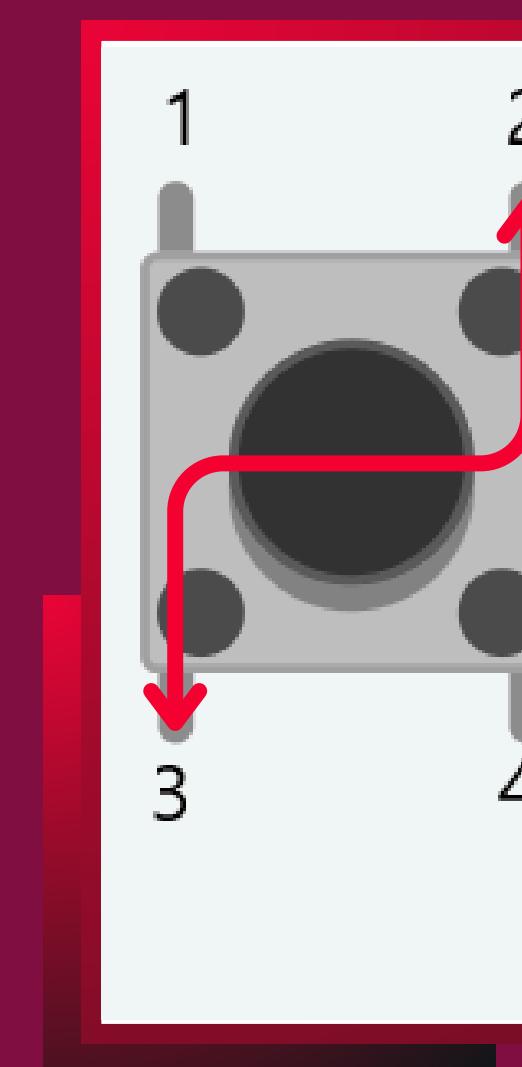
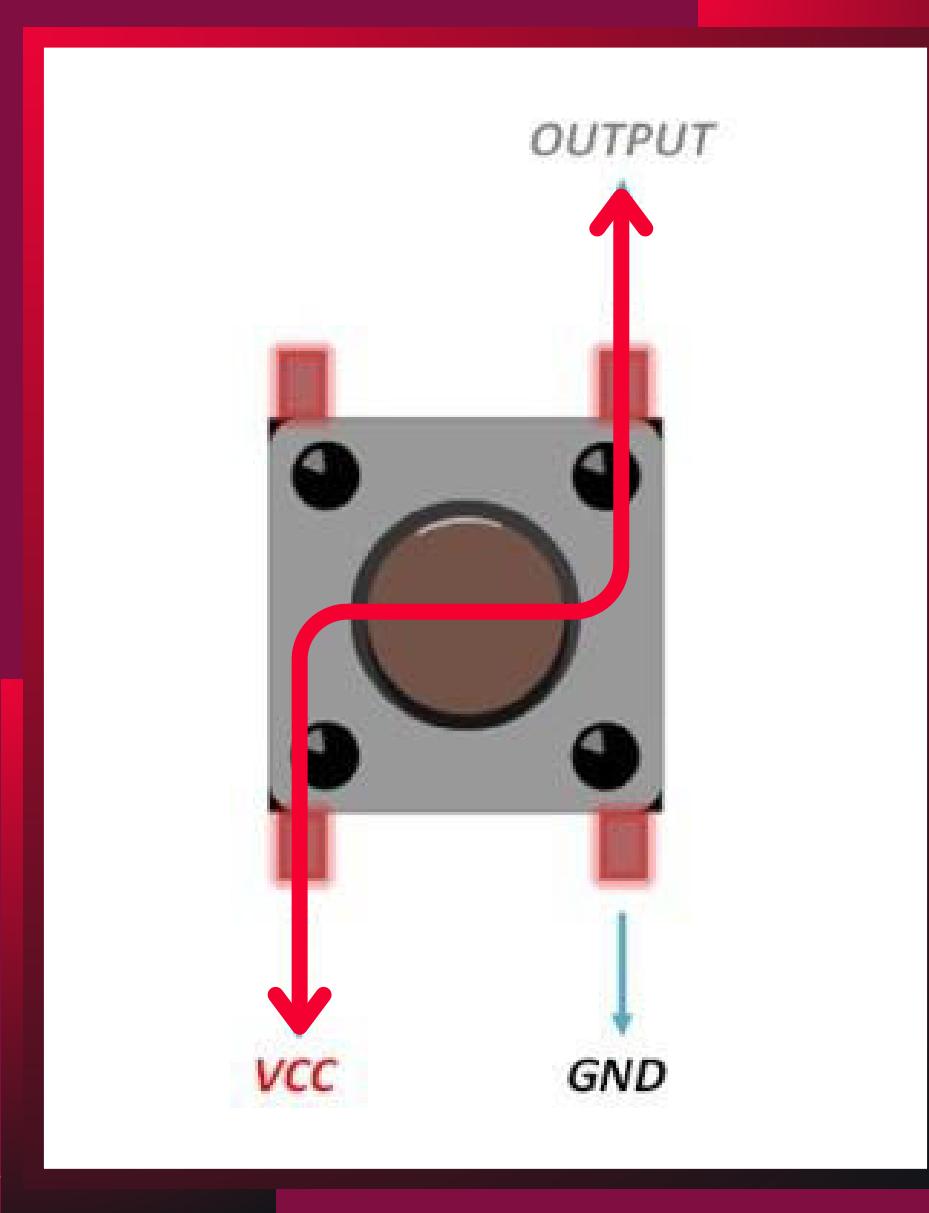
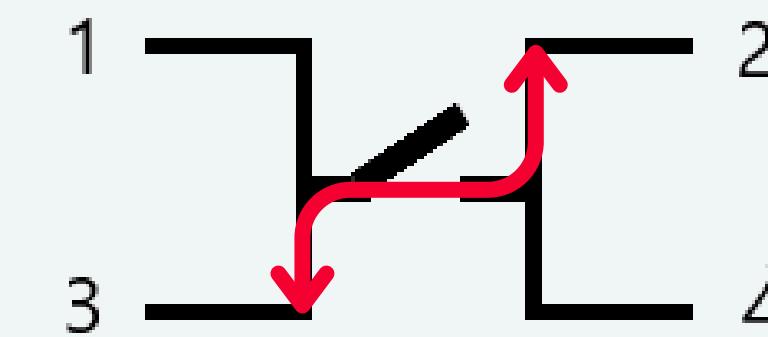


DIAGRAMA BOTÃO PRESSIONADO



Pin 1 & 3 are connected
Pin 2 & 4 are connected



CONECTANDO UM BOTÃO AO ARDUINO

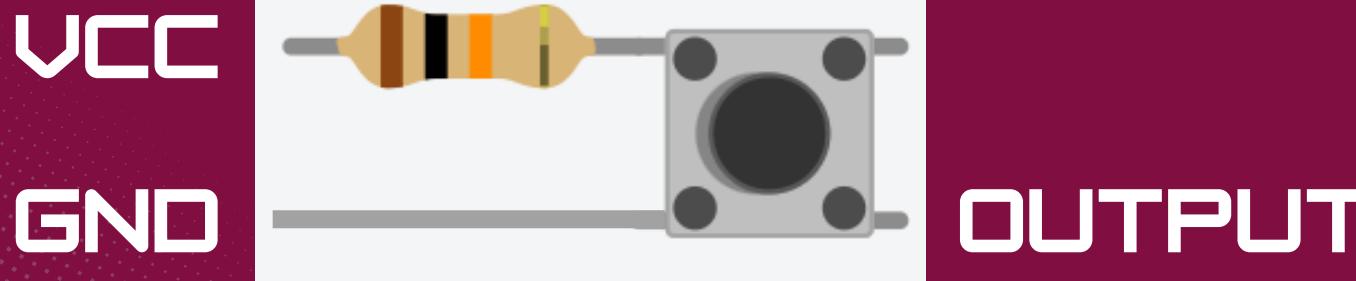
RESISTOR PULL-UP

CONEXÃO:

- BOTÃO ENTRE PINO DIGITAL E GND
- RESISTOR (DE 4 A $10K\Omega$) ENTRE PINO DIGITAL (OUTPUT) E 5V

LEITURA DO PINO DIGITAL:

- BOTÃO SOLTO → HIGH (5V)
- BOTÃO PRESSIONADO → LOW (0V)



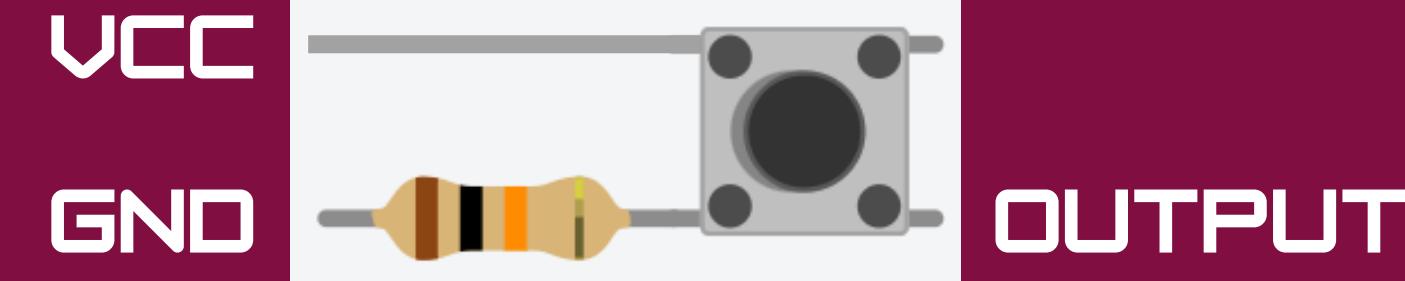
RESISTOR PULL-DOWN

CONEXÃO:

- BOTÃO ENTRE PINO DIGITAL E 5V
- RESISTOR (DE 4 A $10K\Omega$) ENTRE PINO DIGITAL (OUTPUT) E GND

LEITURA DO PINO DIGITAL:

- BOTÃO SOLTO → LOW (0V)
- BOTÃO PRESSIONADO → HIGH (5V)



PROGRAMAÇÃO DE BOTÕES

FUNÇÃO DIGITALREAD ()

- LÊ O SINAL DO PINO DE ENTRADA

CONFIGURAÇÃO INICIAL (SETUP)

- PINMODE (BOTAOPIN, INPUT) :
DEFINE O BOTÃO COMO ENTRADA

USO NO CÓDIGO PRINCIPAL (LOOP)

- DIGITALREAD (BOTAOPIN) : LÊ O
ESTADO DO BOTÃO

```
#define botaoPin 2 // Pino do botão
#define ledPin 13 // Pino do LED

void setup() {
    pinMode(botaoPin, INPUT); // Configura o pino como
    // entrada com resistor pull-down
    pinMode(ledPin, OUTPUT); // Configura o pino do
    // LED como saída
}

void loop() {
    int estadoBotao = digitalRead(botaoPin); // Lê o
    // estado do botão

    // Com resistor pull-down, botão pressionado retorna
    // LOW
    if (estadoBotao == HIGH) {
        digitalWrite(ledPin, HIGH); // Liga o LED
    } else {
        digitalWrite(ledPin, LOW); // Desliga o LED
    }
}
```



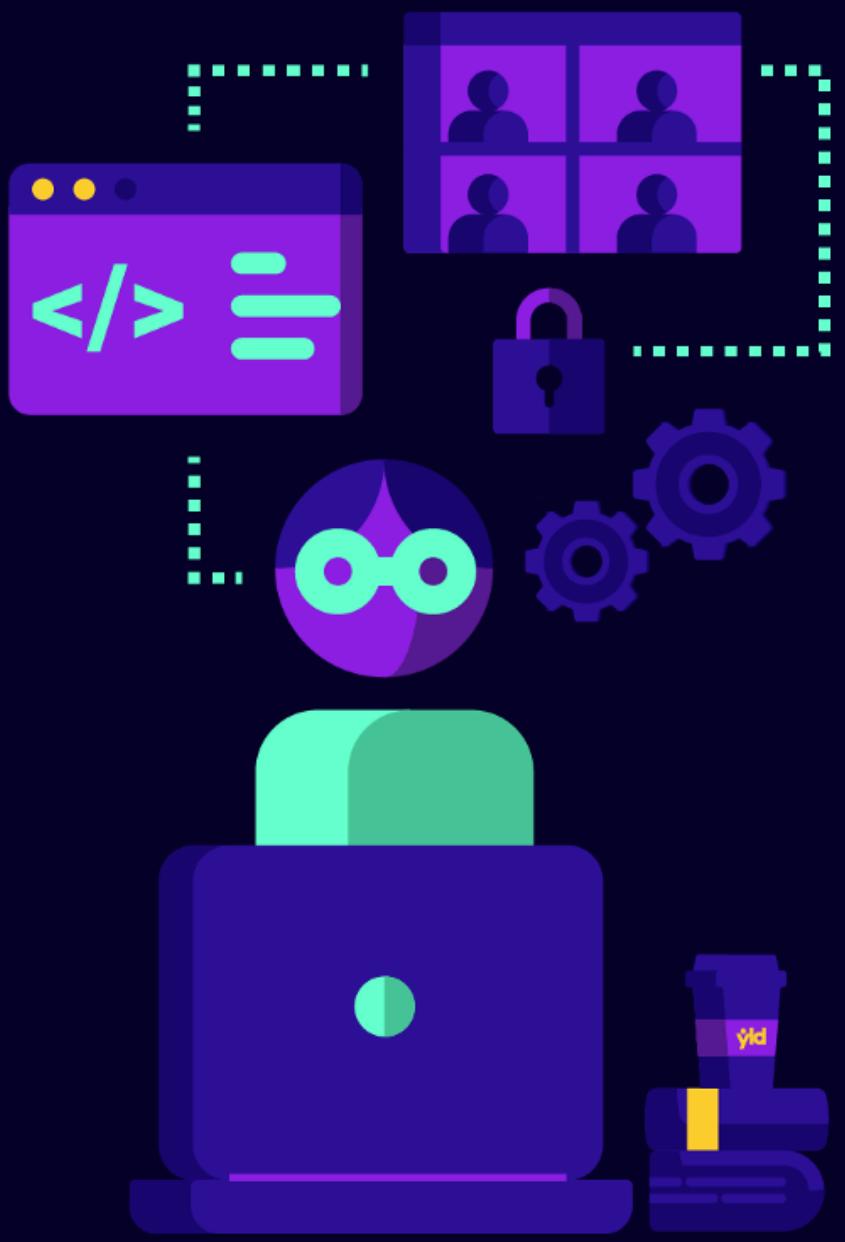
EXERCÍCIO DE FIXAÇÃO

USO DE BOTÕES



USO DE FUNÇÕES EM ARDUINO

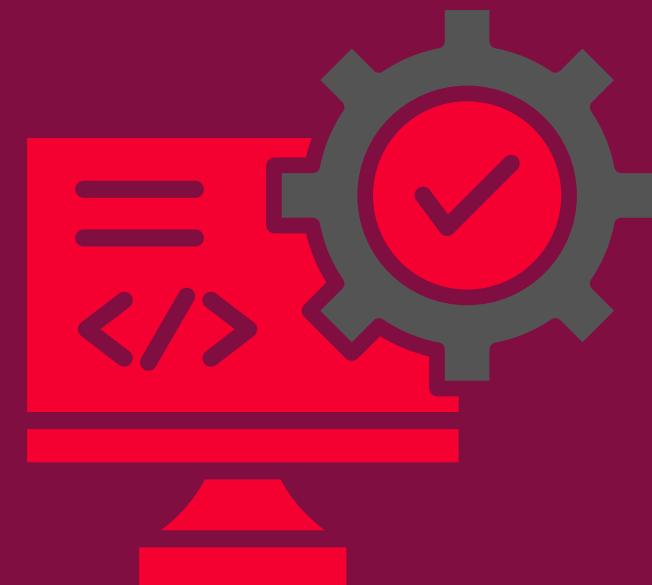
- INTRODUÇÃO AO CONCEITO DE FUNÇÕES E SUAS VANTAGENS
- ESTRUTURA DE CRIAÇÃO DE FUNÇÕES NO ARDUINO
- FUNÇÕES NATIVAS DO ARDUINO
- ESCOPO DE VARIÁVEIS (LOCAL VS. GLOBAL)



O QUE SÃO FUNÇÕES

💡 O QUE SÃO FUNÇÕES?

- BLOCOS DE CÓDIGO QUE EXECUTAM UMA TAREFA
- PARA TAREFAS REPETITIVAS
- ORGANIZAÇÃO DO CÓDIGO



VANTAGENS DO USO DE FUNÇÕES:

- ❖ REUTILIZAÇÃO DE CÓDIGO: ESCREVA UMA VEZ, USE VÁRIAS
- ❖ ORGANIZAÇÃO: DIVIDE PROBLEMAS COMPLEXOS EM PARTES MENORES
- ❖ MANUTENÇÃO: FACILITA CORREÇÕES E ALTERAÇÕES NO CÓDIGO
- ❖ LEGIBILIDADE: Torna o código mais fácil de entender

COMO CRIAR UMA FUNÇÃO

TIPO_RETORNO (TIPO):

- É O TIPO DE DADO QUE A FUNÇÃO VAI DEVOLVER (INT, FLOAT, VOID, ETC.)

NOME_DA_FUNCÃO (NOME):

- É O NOME QUE VOCÊ ESCOLHE PARA SUA FUNÇÃO

PARAMETROS:

- SÃO VALORES QUE A FUNÇÃO PODE RECEBER PARA TRABALHAR

RETURN:

- É USADO PARA DEVOLVER UM VALOR, QUANDO NECESSÁRIO

```
tipo nome (parametros) {  
    // código a ser executado  
    return valor; // se necessário  
}
```



COMO CRIAR UMA FUNÇÃO

FUNÇÃO SEM PARÂMETROS E SEM RETORNO:

```
void piscarLed() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
}
```

FUNÇÃO COM PARÂMETROS

```
void piscarLed (int pinLed, int  
tempoLigado, int tempoDesligado) {  
    digitalWrite(pinLed, HIGH);  
    delay(tempoLigado);  
    digitalWrite(pinLed, LOW);  
    delay(tempoDesligado);  
}
```

FUNÇÃO COM RETORNO E COM RETORNO

```
void int somarDoisNumeros(int  
numero1, int numero2) {  
    return numero1 + numero2;  
}
```



ONDE DECLARAR FUNÇÕES

```
// Protótipo da função
void piscarLed(int pinLed, int tempoLigado, int tempoDesligado);

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    piscarLed(LED_BUILTIN, 1000, 500);
}

// Implementação da função
void piscarLed(int pinLed, int tempoLigado, int tempoDesligado) {
    digitalWrite(pinLed, HIGH);
    delay(tempoLigado);
    digitalWrite(pinLed, LOW);
    delay(tempoDesligado);
}
```



ONDE DECLARAR FUNÇÕES

```
// Implementação da função
void piscarLed(int pinLed, int tempoLigado, int tempoDesligado) {
    digitalWrite(pinLed, HIGH);
    delay(tempoLigado);
    digitalWrite(pinLed, LOW);
    delay(tempoDesligado);
}

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    piscarLed(LED_BUILTIN, 1000, 500);
}
```



FUNÇÕES NATIVAS DO ARDUINO

PINMODE () :

- CONFIGURA UM PINO COMO ENTRADA OU SAÍDA

DIGITALWRITE () :

- ENVIA UM SINAL HIGH (5V) OU LOW (0V) PARA UM PINO DIGITAL

DIGITALREAD () :

- LÊ O ESTADO DE UM PINO DIGITAL

ANALOGREAD () :

- LÊ O VALOR DE UM PINO ANALÓGICO

ANALOGWRITE () :

- ENVIA UM VALOR PWM PARA UM PINO

DELAY () :

- PAUSA O PROGRAMA POR UM TEMPO ESPECÍFICO EM MILISSEGUNDOS

MILLIS () :

- RETORNA O TEMPO EM MILISSEGUNDOS DESDE QUE O ARDUINO FOI LIGADO



ESCOPO DE VARIÁVEIS EM FUNÇÕES

```
//Variáveis Locais

void setup() {
    Serial.begin(9600);
}

void loop() {
    int valorLocal = 10;
    mostraValor();
    Serial.println(valorLocal); // Imprime 10
    delay(1000);
}

void mostraValor() {
    int valorLocal = 5;
    Serial.println(valorLocal); // Imprime 5
}

//Saída : 10  5  10
```



ESCOPO DE VARIÁVEIS EM FUNÇÕES

```
//Variáveis Globais

int valorGlobal = 10; // Variável global

void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(valorGlobal);
    mostraValor();
    valorGlobal += 5;
    Serial.println(valorGlobal); // Imprime o valor modificado
    delay(1000);
}

void mostraValor() {
    valorGlobal += 5; // Modifica a variável global
    Serial.println(valorGlobal);
}

//Saída : 10 15 20
```



ESCOPO DE VARIÁVEIS EM FUNÇÕES

```
//Variáveis Globais

int valorGlobal = 10; // Variável global

void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(valorGlobal);
    mostraValor();
    valorGlobal += 5;
    Serial.println(valorGlobal); // Imprime o valor modificado
    delay(1000);
}

void mostraValor() {
    valorGlobal += 5; // Modifica a variável global
    Serial.println(valorGlobal);
}

//Saída : 10 15 20
```



ESCOPO DE VARIÁVEIS EM FUNÇÕES

```
//Variáveis Globais

int valorGlobal = 10; // Variável global

void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(valorGlobal);
    mostraValor();
    valorGlobal += 5;
    Serial.println(valorGlobal); // Imprime o valor modificado
    delay(1000);
}

void mostraValor() {
    valorGlobal += 5; // Modifica a variável global
    Serial.println(valorGlobal);
}

//Saída : 10 15 20
```



ESCOPO DE VARIÁVEIS EM FUNÇÕES

```
//Variáveis Globais

int valorGlobal = 10; // Variável global

void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(valorGlobal);
    mostraValor();
    valorGlobal += 5;
    Serial.println(valorGlobal); // Imprime o valor modificado
    delay(1000);
}

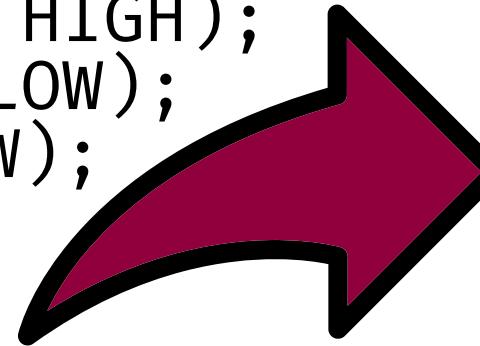
void mostraValor() {
    valorGlobal += 5; // Modifica a variável global
    Serial.println(valorGlobal);
}

//Saída : 10 15 20
```



MELHORANDO O CÓDIGO COM FUNÇÕES

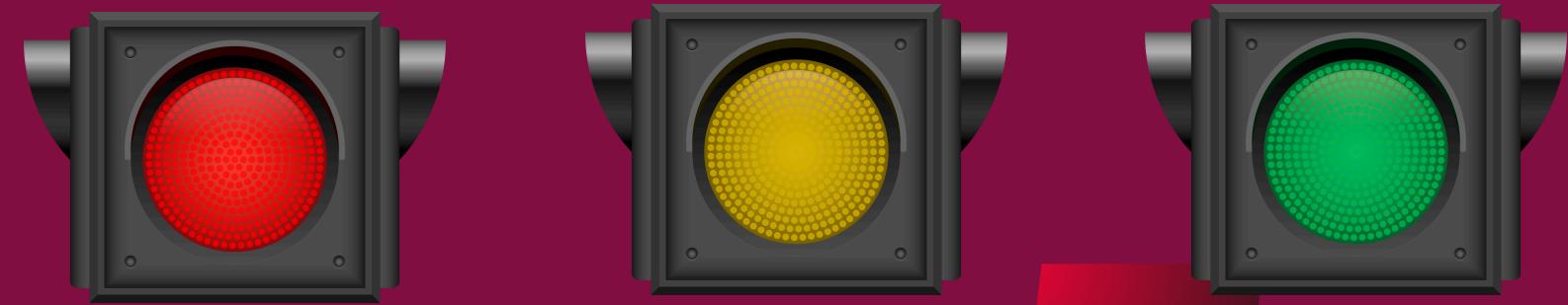
```
void loop() {  
  
    // Acende o LED vermelho  
    digitalWrite(ledVermelho, HIGH);  
    digitalWrite(ledAmarelo, LOW);  
    digitalWrite(ledVerde, LOW);  
    delay(1000);  
  
    // Acende o LED amarelo  
    digitalWrite(ledVermelho, LOW);  
    digitalWrite(ledAmarelo, HIGH);  
    digitalWrite(ledVerde, LOW);  
    delay(1000);  
  
    // Acende o LED verde  
    digitalWrite(ledVermelho, LOW);  
    digitalWrite(ledAmarelo, LOW);  
    digitalWrite(ledVerde, HIGH);  
    delay(1000);  
}
```



```
void loop() {  
    acenderLed(ledVermelho);  
    acenderLed(ledAmarelo);  
    acenderLed(ledVerde);  
}  
  
void acenderLed(int pino) {  
    // Apaga todos os LEDs  
    digitalWrite(ledVermelho, LOW);  
    digitalWrite(ledAmarelo, LOW);  
    digitalWrite(ledVerde, LOW);  
  
    // Acende apenas o LED desejado  
    digitalWrite(pino, HIGH);  
    delay(1000);  
}
```

EXERCÍCIO DE FIXAÇÃO

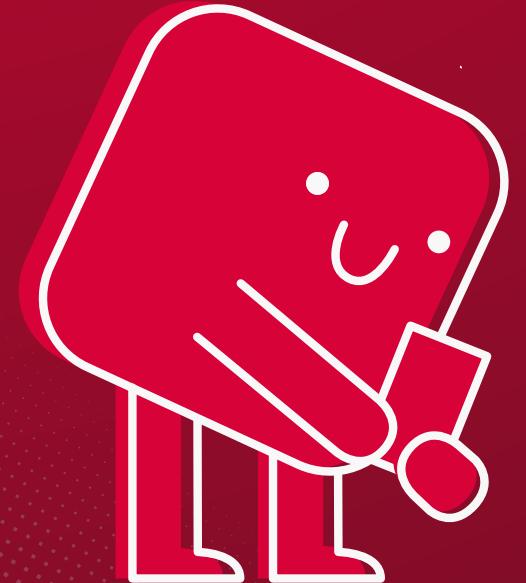
SEMÁFORO SIMPLES



EQUIPE RESPONSÁVEL :

DIRETOR DE TREINAMENTO :

PEDRO S. CONCEIÇÃO



MUITO
OBRIGADO !!