

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO
INTRODUÇÃO À ENGENHARIA DE COMPUTAÇÃO

PROF. DR. ADRIANO CÉSAR SANTANA
adriano@ufg.br

INTRODUÇÃO AO ARDUINO – AULA 3





```
void setup() {  
  // put your setup code here, to run once:  
  
}
```

Parte do código que é executada apenas uma vez.

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Parte do código que é executada enquanto o arduino estiver ligado.

- Variável: Uma variável é um recurso utilizado para armazenar dados em um programa de computador. Todo computador possui algum tipo de memória, e uma variável representa uma região da memória usada para armazenar uma determinada informação.

```
int led = 13;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

- Tipo de dado: Significa, como o próprio nome diz, o tipo de informação que se pode armazenar naquela variável.
- Boolean: valor verdadeiro (true) ou falso (false).
- Char: Um caractere.
- Float: Número real de precisão simples (ponto flutuante).
- Void: tipo vazio (não tem tipo).

```
int led = 13;  
  
void setup() {  
    pinMode(led, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
}
```

- Atribuição: Atribuir um valor a uma variável significa armazenar o valor nela para usar posteriormente. O comando de atribuição em C++ é o =. Para atribuirmos o valor 13 à variável led que criamos acima, fazemos assim:

```
int led = 13;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

- Operadores aritméticos:
 - `+`: adição ("mais")
 - `-`: subtração ("menos")
 - `*`: multiplicação ("vezes")
 - `/`: divisão ("dividido por")
- Operadores de atribuição:
 - `=`: atribui um valor a uma variável, como vimos acima.

- Operadores lógicos:
 - `&&`: conjunção ("e")
 - `||`: disjunção ("ou")
 - `==`: igualdade ("igual a")
 - `!=`: desigualdade ("diferente de")
 - `!`: negação ("não")
 - `>`: "maior que"
 - `<`: "menor que"
 - `>=`: "maior ou igual a"
 - `<=`: "menor ou igual a"

- Função: é uma sequência de comandos que pode ser reutilizada várias vezes ao longo de um programa.

```
int led = 13;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

- Chamada de função: Chamar uma função significa executar os comandos que foram definidos na sua declaração. Uma vez declarada, uma função pode ser chamada várias vezes no mesmo programa para que seus comandos sejam executados novamente.

```
int led = 13;

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

void loop() {
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
}
```


- While:

```
while(condição) {  
  
    ...  
  
}
```

- For:

```
for(i = 0; i < 3; i++) {  
  
    comandos;  
  
}
```

- If:

```
if(condição) {  
  
    ...  
  
}else {  
  
    ...  
  
}
```

Estrutura de condição IF/ELSE



sketch_sep07a §

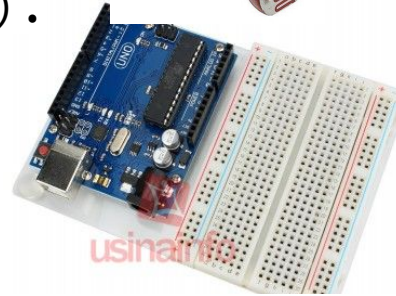
```
int botao = 2; // Define a porta do led como 3;
int led = 3; // Define a porta do led como 2;

void setup() {
  pinMode(botao, INPUT); // Define que a porta 2 sera de entrada/leitura;
  pinMode(led, OUTPUT); //Define que a porta 3 será de saida/escrita;
}

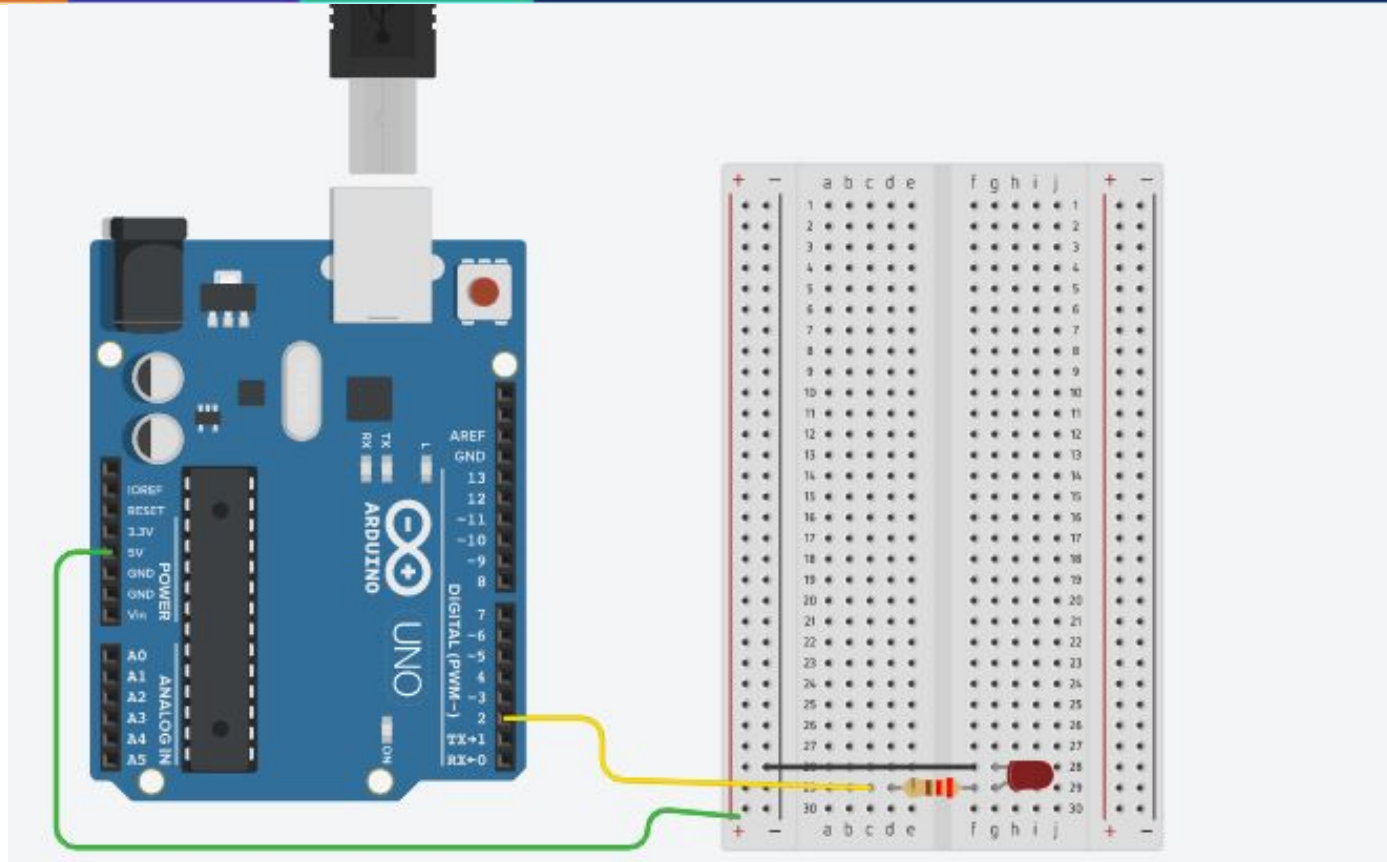
void loop() {
  if(digitalRead(botao) == HIGH){    // Verifica se o botão da porta 2 estiver pressionado;
    digitalWrite(led, HIGH); //Se sim, liga o led;
  }
  else {
    digitalWrite(led, LOW); //Se não, não liga o led;
  }
}
```

Componentes necessários

- Diodo Emissor de Luz (LED):
- Botão Não Retentivo.
- Resistores Simples.
- Light Dependent Resistor(LDR).
- Protoboard e Arduino



1 - Monte o seguinte circuito



2 - Abra a programação na IDE

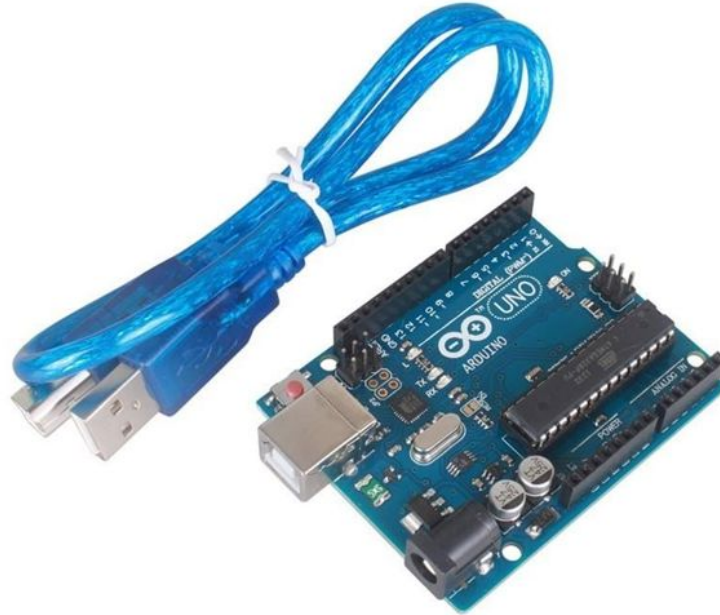


```
p_ligar_led

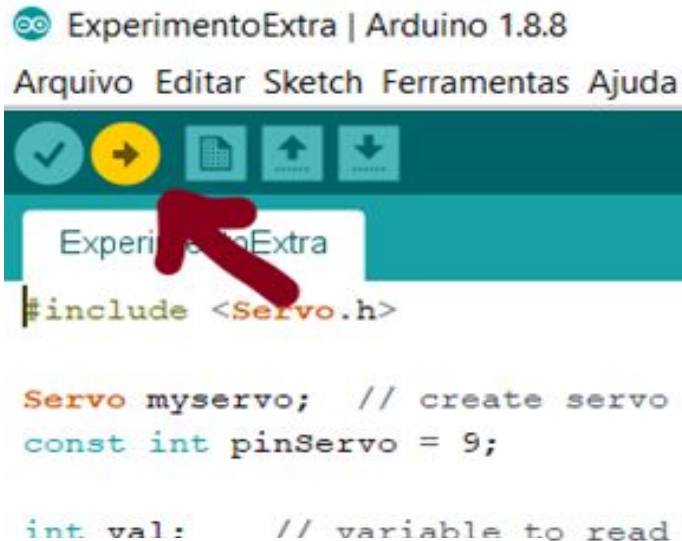
void setup()
{
    // Configura o pino dl3 como saída
    pinMode(2, OUTPUT);
}

void loop()
{
    // Configura o pino 13 como HIGH
    digitalWrite(2, HIGH);
}
```

3 - Conecte o Arduino ao computador utilizando o cabo de conexão USB



4 - Carregue o código a placa



```
void setup() {
  pinMode(buttonPin, INPUT);
```

Problema ao carregar para a placa. Veja <http://www.arduino.cc/en/Guide/Troubleshooting#upload> para sugestões.

Problema ao carregar para a placa. Veja <http://www.arduino.cc/en/Guide/Troubleshooting>

Certifique-se de que não foram encontrados erros.

5 - Faça a alteração de HIGH para LOW

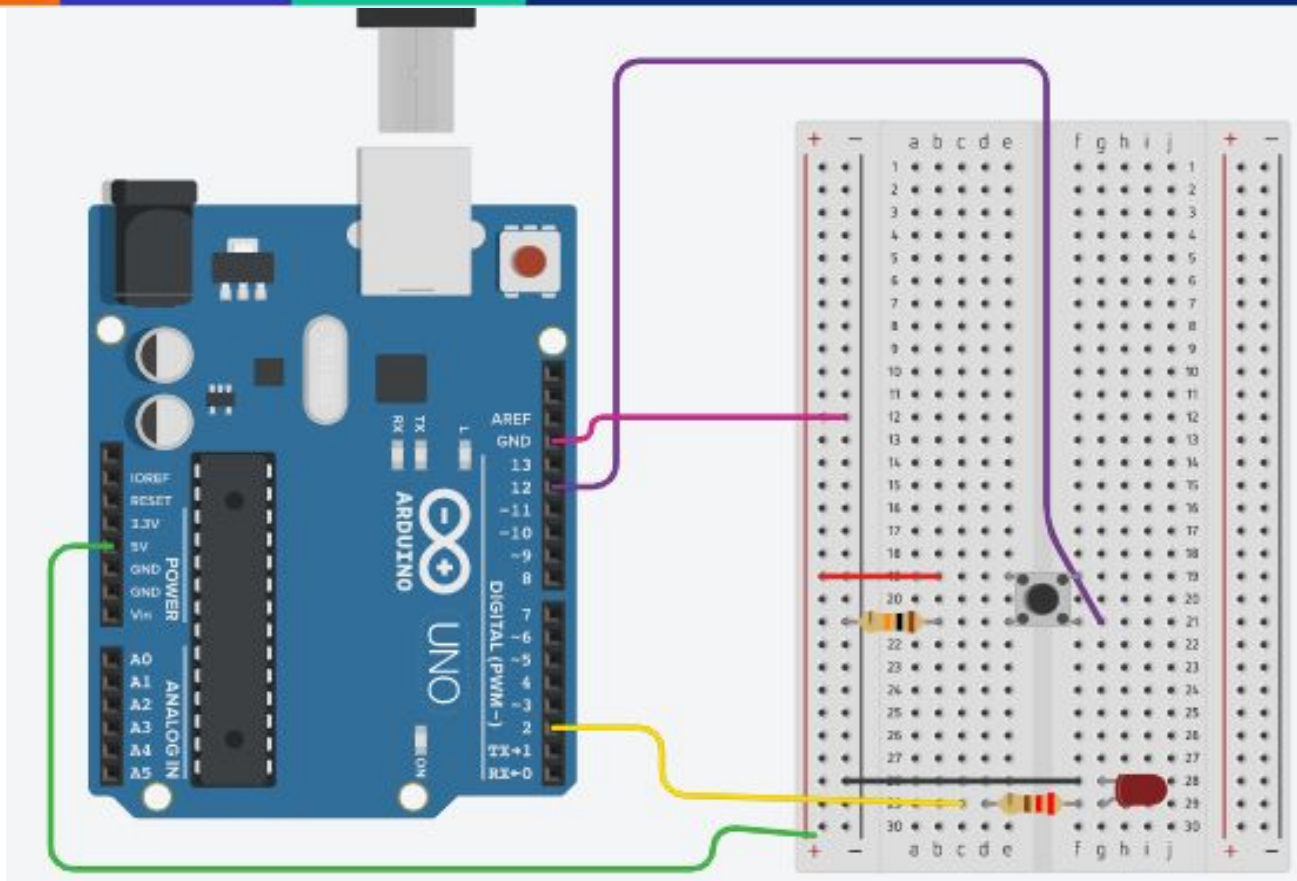


p_ligar_led

```
void setup()
{
    // Configura o pino d13 como saída
    pinMode(2, OUTPUT);
}

void loop()
{
    // Configura o pino 13 como HIGH
    digitalWrite(2, HIGH);
}
```


6 - Acrescente o botão ao circuito



7 - Abra o arquivo “p_botao_led” e faça a compilação

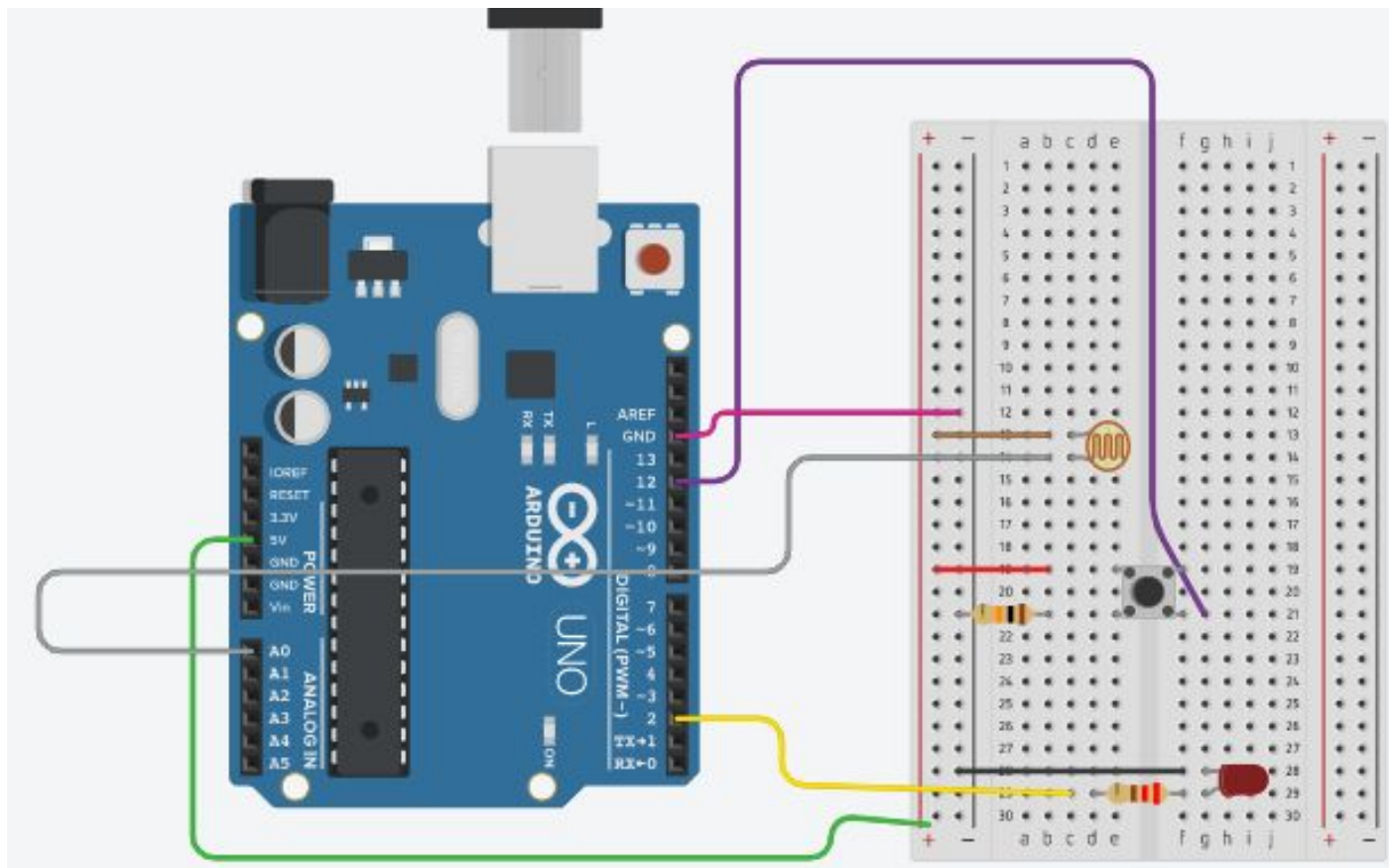


p_botao_led \$

```
const int ledPin = 2; //led no pino 2
const int Botao = 12; //botao no pino 12
int estadoBotao; //Variavel para ler o status do pushbutton

void setup(){
    pinMode(ledPin, OUTPUT); //Pino do led será saída
    pinMode(Botao, INPUT); //Pino com botão será entrada
}
void loop(){
    estadoBotao = digitalRead(Botao); //le o estado do botão - HIGH OU LOW
    if (estadoBotao == HIGH){ //Se botão estiver pressionado (HIGH)
        digitalWrite(ledPin, HIGH); // acende o led do pino 2.}
    else{ //se não estiver pressionado (LOW)
        digitalWrite(ledPin, LOW); //deixa o led do pino 2 apagado}
    }
}
```

8 - Acrescente o LDR



9 - Abra o arquivo p_ldr_botao_led



p_ldr_botao_led\$

```
const int ledPin = 2; //led no pino 2
const int Botao = 12; //botao no pino 12
int estadoBotao; //Variavel para ler o status do pushbutton
int ldrPin = A0; //LDR no pino analógico 8
int ldrValor = 0; //Valor lido do LDR

void setup(){
    pinMode(ledPin, OUTPUT); //Pino do led será saída
    pinMode(Botao, INPUT); //Pino com botão será entrada}
void loop(){
    ///ler o valor do LDR
    ldrValor = analogRead(ldrPin); //O valor lido será entre 0 e 1023
    //se o valor lido for maior que 500, liga o led
    if (ldrValor >= 800){
        estadoBotao = digitalRead(Botao); //le o estado do botão - HIGH OU LOW
        if (estadoBotao == HIGH){ //Se botão estiver pressionado (HIGH)
            digitalWrite(ledPin, HIGH); // acende o led do pino 2.}
        else{ //se não estiver pressionado (LOW)
            digitalWrite(ledPin, LOW); //deixa o led do pino 2 apagado    }}
    else if (ldrValor <= 800){
        digitalWrite(ledPin, HIGH); // acende o led do pino 2.}}
```



- Melyssa: melyssa.mariana@outlook.com
- Antonio: antonio_pereira@outlook.com
- Monitoria: monitoria89@gmail.com