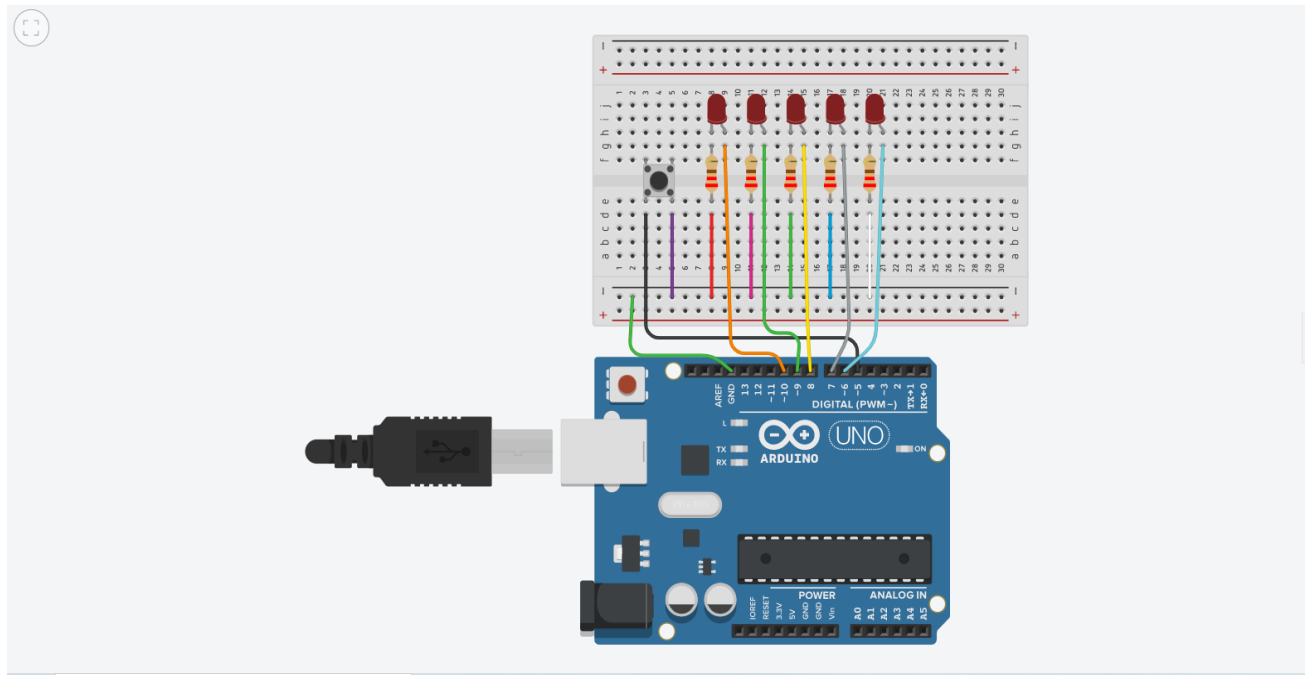


# Relatório:

## Montagem de circuito:



Na montagem do circuito procedeu-se à implementação de um botão de pressão ligado ao pino 5 e ao GND, o mesmo dos leds para que o circuito possa funcionar corretamente e as funcionalidades do botão de pressão se tornem utilizáveis. Este botão de pressão está ligado em modo *input pullup*, logo utiliza as resistências internas do Arduino.

Os 5 leds estão ligados entre o pino 6 e o 10, e a corrente transmitida pelo GND a cada um dos leds fica sujeita a uma resistência de 220  $\Omega$ .

## Código:

```
1 int buttonPin=5;
2 bool trueReading=false;
3 bool lastState=false;
4 long debounceDelay=50;
5 long bounce;
6 int contaprima=0;
7 void setup() {
8     // put your setup code here, to run once:
9     pinMode(buttonPin, INPUT_PULLUP);
10    pinMode(6, OUTPUT);
11    pinMode(7, OUTPUT);
12    pinMode(8, OUTPUT);
13    pinMode(9, OUTPUT);
14    pinMode(10, OUTPUT);
15    Serial.begin(9600);
16    randomSeed(analogRead(0));
17 }
```

Inicialmente introduz-se as variáveis globais que serão necessárias para o funcionamento do botão, do debouncing no botão de pressão e do contador de cliques. No setup() procede-se à configuração do botão em modo input pullup, dos leds como dispositivos de saída, à inicialização do monitor

série(`Serial.begin(9600)`) e ainda a introdução de `randomSeed` para que a função `random` utilizada em `loop()` funcione.

```
22 void loop() {
23   // put your main code here, to run repeatedly:
24   unsigned long timer = millis();
25   int numero_aleatorio=random(1,32);
26   for(int i=0; i<=4; i++){
27     digitalWrite(6+i, (numero_aleatorio >> i) & 1);
28   }
29   Serial.println("TENS 15 SEGUNDOS FORÇA:");
30   while((millis()-timer)<15000){
31     bool buttonState=!digitalRead(buttonPin);
32     if (buttonState!=lastState){
33       bounce=millis();
34     }
35   }
36   if(millis()-bounce>1000 && buttonState==HIGH){
37     while(digitalRead(buttonPin)==LOW){
38     }
39   }
40   loop();
41 }
42
43 if(millis()-bounce>deBounceDelay){
44   if (buttonState!=trueReading){
45     trueReading=buttonState;
46     if (trueReading==true){
47       contaprima=contaprima+1;
48       Serial.println(contaprima);
49       if (contaprima==32){
50         contaprima=0;
51       }
52 }
53 }}}
54 lastState=buttonState;
55 }
```

Dentro da função `loop()`, que corresponde ao código que será executado repetidamente, encontra-se a variável inteira `numero_aleatorio`, na qual a função `random` irá escolher aleatoriamente um número entre 1 e 32, esta função só funciona devido à presença da seed no `setup()`. Na linha 26 encontra-se um ciclo `for` que vai permitir apresentar o número aleatório em binário, uma vez que este ciclo vai deslocar o número de bits correspondentes ao iterador sendo que por cada repetição do ciclo o pino do led a ser verificado também avança, ou seja, cada led ficará associado a apenas um bit, a utilização o "e" lógico e da máscara 1 define se o led acende ou não, uma vez que caso o bit deslocado corresponda a 1 o valor introduzido no led será HIGH(acende) e caso o bit deslocado seja 0 o valor introduzido no led será LOW (não acende). O ciclo `while` presente na linha 30 define o tempo disponível ao utilizador para acertar o número, neste caso 15 segundos, uma vez que assim que a contagem do tempo for igual a 15000 ms o programa sai do ciclo `while`(onde se encontra o debouncing do botão (linha 43), contador de cliques (linha 46) e o reset) e testa as condições seguintes. A condição `if` da linha 36 corresponde ao reset do botão, uma vez que quando o tempo de clique do botão supere um segundo e este seja clicado entra no ciclo `while` em que caso a leitura do botão corresponda ao botão a ser pressionado (valor LOW, uma vez que se encontra configurado em

input pullup), nada é executado, mas assim que o valor de leitura do botão se alterar a função loop() será chamada e o programa voltará ao início.

```
61  if (numero_aleatorio==contaprima){
62  Serial.println("Parabéns acertou o numero");
63  contaprima=0;
64  for(int led=6;led<=10;led++){
65    digitalWrite(led,LOW);
66
67  }
68
69  for(int led=6;led<=10;led++){
70    digitalWrite(led,HIGH);
71    delay(500);
72  }
73  }else{
74    Serial.println("Bem erraste, o número certo:");
75    Serial.println(numero_aleatorio,DEC);
76    contaprima=0;
77  }
78
79
80
81
82
83 }
```

---

Na secção final do código encontram-se as condições para caso o utilizador acerte o número ou caso falhe, depois de passados os 15 segundos impostos pela função while (linha 30). Essencialmente, caso o número apresentado no contador seja igual ao gerado pela função random será impressa uma mensagem de vitória, o contador reinicializará e todos os leds serão desligados (ciclo for, linha 64) para imediatamente a seguir iniciar o efeito de varrimento definido pelo ciclo for da linha 69 em que se acenderá cada um dos leds com um intervalo de tempo de meio segundo. Caso a condição de vitória não se verifique o programa entrará automaticamente na condição else para o qual se imprime a mensagem de derrota, a resposta correta e se reinicializa o contador.