

# RELATÓRIO DO EXERCÍCIO DE AVALIAÇÃO

## Introdução:

Para implementar a minha solução para o jogo intitulado “Descobre o *LED*” comecei por criar o circuito no TinkerCad e, assim, começar a desenvolver o programa. Vou falar um pouco de como desenvolvi cada função do programa como a de Acender os *LEDS* sequencialmente, a função *Reset*, a função de descoberta do *LED* e, por fim, a função *Vitória*.

## Variáveis:

- Criei 7 variáveis inteiras (*WINNING\_TIME*, *LED\_TRANSITION\_TIME*, *ATTEMPTS*, *BLINKING\_DELAY*, *buttonState*, *buttonPlay*, *buttonReset*), 2 arrays inteiros (*LEDS[]*, *LEDS\_ON[]*) e uma variável *long* (*timepass*), isto tudo com o intuito de tornar o código mais modular e fácil de alterar e usar no futuro.

## Acender os *LEDS* sequencialmente e Descoberta:

- Comecei por criar uma função chamada *LED\_SEQUENCE\_ATTEMPT()* (tem implementada a função de ligar os *LEDS* de forma sequencial e de descobrir os *LEDS*), em seguida, usei um *for* para ligar os *LEDS* sequencialmente, usando também a função *millis()* para calcular o tempo em que o utilizador pode clicar no botão de jogar para manter o *LED* aceso, também usei o array *LEDS\_ON[]* para gravar o estado de cada *LED* a cada *loop* efetuado. Nesta função, também, é gravado o número de cliques dados pelo utilizador tendo em conta *button debouncing*.

## *Reset*:

- Criei uma função *reset()* que, quando o botão respetivo for clicado, coloca as tentativas do utilizador em 0, desliga todos os *LEDS* e coloca o array *LEDS\_ON[]* todo em 0. Coloquei esta função no início do *for* da função anterior para poder ser executada mais frequentemente e quando necessário.

## *Vitória*:

- Por fim criei uma função *LED\_VICTORY()* que, cria uma variável local *LEDS\_ON\_HIGH* e executa um *for* para ler o array *LEDS\_ON[]*, caso *LEDS\_ON[i]* para qualquer índice *i*, seja igual a 1 incrementa 1 valor à variável local e quando esse valor for igual a 9 (números de *LEDS* total) usando a função *millis()* faz os *LEDS* piscar por quanto tempo (milissegundos) for colocado na variável *WINNING\_TIME* e imprime o número de cliques dados pelo utilizador.

A implementação deste programa mostrou-se importante para avaliar as capacidades individuais na criação de código e resolução de problemas.

Penso que a minha maior dificuldade foi implementar o contador de cliques do utilizador e que este talvez não esteja a funcionar totalmente bem.

### Código do Jogo “Descobre o LED”:

```
int WINNING_TIME=20000,LED_TRANSITION_TIME=500,ATTEMPTS=0,BLINKING_DELAY=400;
int lastbuttonState=0,buttonState=0,buttonPlay=11,buttonReset=12;
int LEDS[]={2,3,4,5,6,7,8,9,10};
int LEDS_ON[]={0,0,0,0,0,0,0,0,0};
long timepass=0, lastDebounceTime=0;
void LED_SEQUENCE_ATTEMPT(){
  for(byte i=0; i<=8; i++){
    if(buttonState==digitalRead(buttonReset)){
      LED_RESET();
      i=0;
    }
    if(digitalRead(buttonPlay)==lastbuttonState)
      lastDebounceTime=millis();
    if((millis()-lastDebounceTime)>50)
      if (digitalRead(buttonPlay)!=buttonState){
        buttonState=digitalRead(buttonPlay);
        if (buttonState==1)
          ATTEMPTS++;
      }
    buttonState=0;
    digitalWrite(LEDS[i], HIGH);
    timepass=millis();
    while(millis()-timepass<=LED_TRANSITION_TIME)
      if(buttonState==digitalRead(buttonPlay) &&
buttonState!=digitalRead(LEDS[i])){
        digitalWrite(LEDS[i], HIGH);
        LEDS_ON[i]=1;
      }
    if(LEDS_ON[i]==0)
      digitalWrite(LEDS[i], LOW);
  }
}
void LED_RESET(){
  ATTEMPTS=0;
  for(byte i=0; i<=8; i++){
    digitalWrite(LEDS[i], LOW);
    LEDS_ON[i]=0;
  }
}
void LED_VICTORY(){
  int LEDS_ON_HIGH=0;
  for(byte i=0; i<=8; i++)
    if(LEDS_ON[i]==1)
      LEDS_ON_HIGH++;
  if(LEDS_ON_HIGH==9){
    timepass=millis();
    while(millis()-timepass<=WINNING_TIME){
      for(byte i=0; i<=8; i++){
        digitalWrite(LEDS[i], HIGH);
        LEDS_ON[i]=1;
      }
      delay(BLINKING_DELAY);
    }
  }
}
```

```

        for(byte i=0; i<=8; i++){
            digitalWrite(LED_S[i], LOW);
            LED_ON[i]=0;
        }
        delay(BLINKING_DELAY);
    }
    Serial.println("O seu numero de cliques foi:");
    Serial.println(ATTEMPTS);
    ATTEMPTS=0;
}
}
void setup() {
    for(byte i=0; i<=8; i++)
        pinMode(LED_S[i], OUTPUT);
    pinMode(11, INPUT_PULLUP);
    pinMode(12, INPUT_PULLUP);
    Serial.begin(9600);
}
void loop(){
    LED_SEQUENCE_ATTEMPT();
    LED_VICTORY();
}

```