

# Relatório do projeto de TI

O meu código é constituído por 5 partes principais, a parte do contador de clicks, os modos vitória e derrota, os temporizadores, o modo reset e a parte da operação entre números.

## Contador de clicks

Aqui para fazer com que o contador fosse limitado por 7 segundos criei uma variável “tempoINICIO” para guardar o valor do tempo quando o código “chega aí”. Depois a variável “tempoAGORA” vai guardar o valor do tempo a cada novo ciclo que o código faz para comparar a diferença entre o tempoAGORA e o tempoINICIO e verificar se está dentro dos 7 segundos, contabilizando assim os clicks apenas durante os 7 segundos, este mesmo raciocínio foi aplicado em todos os outros temporizadores. Também foi implementado o debounce para fazer a contagem.

```
int tempoINICIO=millis(); //guarda o v
int tempoAGORA=millis();
while((tempoAGORA-tempoINICIO)<7000){
    tempoAGORA=millis(); //tempo que é

int reading = digitalRead(buttonPin); //debouncezinho basico
if (reading != lastButtonState){
    lastDebounceTime = millis();
}
if ((millis()-lastDebounceTime) < debounceDelay){
    if (reading!= buttonState){
        buttonState=reading;
        if (buttonState != HIGH){ //contador se o botao for despresio
            cont = cont + 1;
            Serial.print("Numero de cliks :"); Serial.println(cont);
        }
    }
}
```

## Modos Vitória e derrota

Para fazer os modos vitória e derrota criei uma função para cada um dos modos, chamando-a quando o jogador ganha-se (contador = conta) ou quando perde-se (contador != conta).

Esta parte do código é o varrimento dos leds no modo vitória. Basicamente o que acontece é que ligo primeiro os leds de 6 a 9 e de 13 a 10 e apos ligar os leds desligo os leds de 9 a 6 e de 10 a 13. Pus tudo isso dentro de um ciclo para repetir o processo 3 vezes.

```
for (int i = 0; i <4; i++){ //varrimento 3 vezes
    for (int digit = 0; digit<4; digit++){
        digitalWrite(6+digit, HIGH); //liga o led
        digitalWrite(13-digit, HIGH); //liga os le
        delay(500);
    }
    for (int digit = 0; digit<4; digit++){ //dep
        digitalWrite(9-digit, LOW);
        digitalWrite(10+digit, LOW);
        delay(500);
    }
}
```

## Temporizadores

Neste trabalho haviam os temporizadores dos modos vitoria e derrota, do reset e dos 7 segundos de tempo de jogo. Em todos os temporizadores usei o mesmo raciocínio do temporizador de 7 segundos menos no reset mudando apenas o "tempo limite". Contudo no reset que consiste em primir o botão durante 1 ou mais segundos ativando assim o modo reset, após o varrimento dos leds, quando chamo a função setup() para dar reset, o código dava skip ao jogo. Na verdade ele não dava skip ao jogo, estava era a ler o próximo código porque estava tudo dentro de um if. Para resolver apenas disse que quando o código passa-se pelo reset o a =3 , assim bastou adicionar que o modo vitoria e derrota só corriam se a != 3 além das condições obvias.

```
a = 3; //yeeiee conseqüência if (conta == cont && a!=3){ else if (a!=3){ //se nao
num2 = random(1,17); Serial.println("Muita Serial.println("E
num1 = random(1,17); vitoria()); //chama a f Serial.print("A re
setup(); // não sei setup(); //chama a fur derrota()); //chama
                                         setup(); //chama a
```

## Operação entre os números

Primeiramente "escolhi" um número de 1 a 3 random para que correspondesse aos operadores and, or e xor, respetivamente. Criei também uma variável "corRGB" para corresponder ao pin onde as cores do RGB estão ligados, para poder fazer o RGB piscar no modo derrota e vitoria. Depois de ser feita essa escolha, o led RGB liga correspondendo as cores azul, vermelho e verde a cada uma dessas operações. Depois faz-se a operação correspondente entre o número 1 e o número 2 que foram previamente randomizados, esse valor da conta vai ser posteriormente comparado com o número de clicks que o jogador fez para verificar se ele ganhou ou perdeu.

```
oper = random(1,4); //valor random para corresponde
if (oper == 1){
    corRGB = 5;
    Serial.println("O operador binario e (and)");
    digitalWrite(vermelho,1);
    conta = (num2 & num1); if (conta == cont && a!=3){
}
else if (oper == 2){
    corRGB = 4;
    Serial.println("O operador binario e (or)");
    digitalWrite(azul,1);
    conta = (num2 | num1);
}
else {
    corRGB = 3;
    Serial.println("O operador binario e (xor)");
    digitalWrite(verde,1);
    conta = (num2 ^ num1);
}
```

