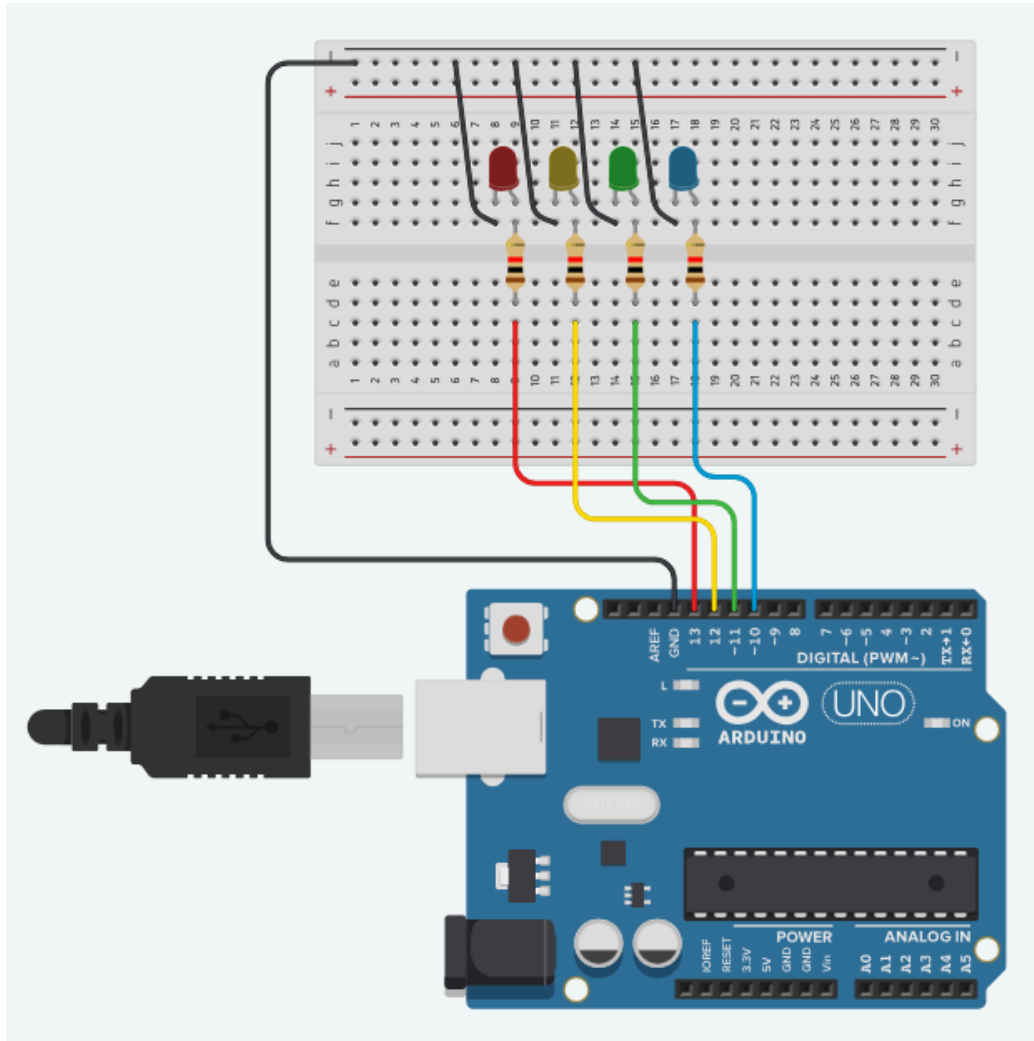


EP03 - Arquitetura de Computadores ||

1- Semáforo tinkercad



1- Código semáforo tinkercad

```
const int pinoVermelho = 13;
const int pinoAmarelo = 12;
const int pinoVerde = 11;
const int pinoAzul = 10;

void setup() {
  pinMode(pinoVermelho, OUTPUT);
  pinMode(pinoAmarelo, OUTPUT);
  pinMode(pinoVerde, OUTPUT);
  pinMode(pinoAzul, OUTPUT);
}

void loop() {
  for (int ciclo = 0; ciclo < 3; ciclo++) {
    digitalWrite(pinoVermelho, HIGH); // Acende o LED Vermelho
    delay(1000); // Espera 1000 equivale a 1 seg ligado
    digitalWrite(pinoVermelho, LOW); // Apaga o LED Vermelho
    delay(1000); // aguarda 1 seg

    digitalWrite(pinoAzul, HIGH); // faz o led azul acender
    delay(1000); // aguarda 1 seg
    digitalWrite(pinoAzul, LOW); // faz o led azul apagar
    delay(1000); // aguarda 1 seg
  }

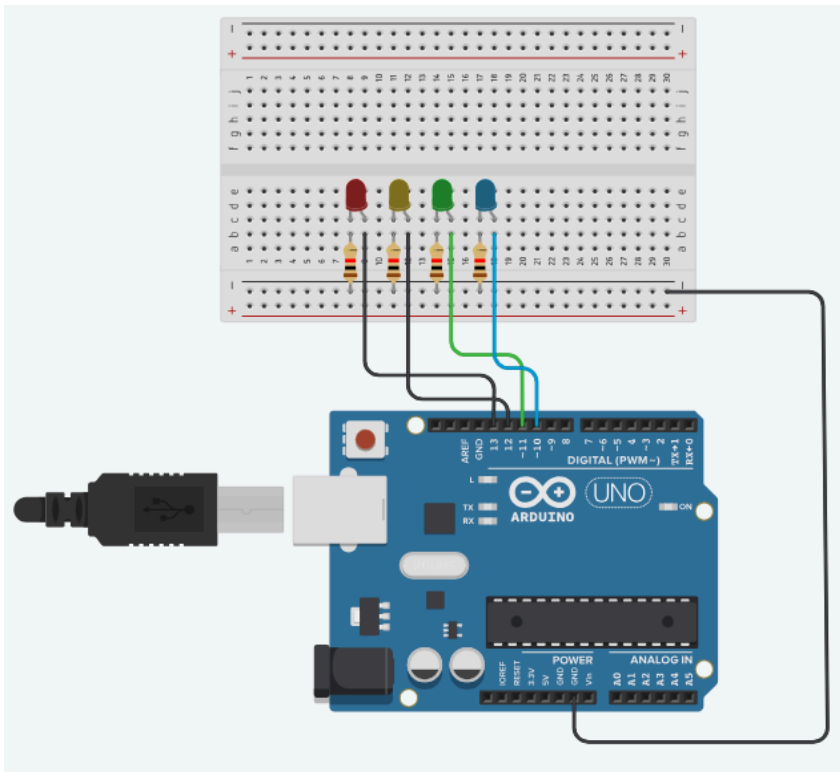
  for (int ciclo = 0; ciclo < 4; ciclo++) {
    digitalWrite(pinoVerde, HIGH); // faz o led verde acender
    delay(1000); // aguarda 1 seg
    digitalWrite(pinoVerde, LOW); // faz o led verde apagar
    delay(1000); // aguarda 1 seg

    digitalWrite(pinoAzul, HIGH); // faz o led azul acender
    delay(1000); // aguarda 1 seg
    digitalWrite(pinoAzul, LOW); // faz o led azul apagar
    delay(1000); // aguarda 1 seg
  }

  for (int ciclo = 0; ciclo < 2; ciclo++) {
    digitalWrite(pinoAmarelo, HIGH); // faz o led amarelo acender
    delay(1000); // Espera 1 segundo
    digitalWrite(pinoAmarelo, LOW); // faz o led amarelo apagar
    delay(1000); // Espera 1 segundo

    digitalWrite(pinoAzul, HIGH); // faz o led azul acender
    delay(1000); // aguarda 1 seg
    digitalWrite(pinoAzul, LOW); // faz o led azul apagar
    delay(1000); // aguarda 1 seg
  }
}
```

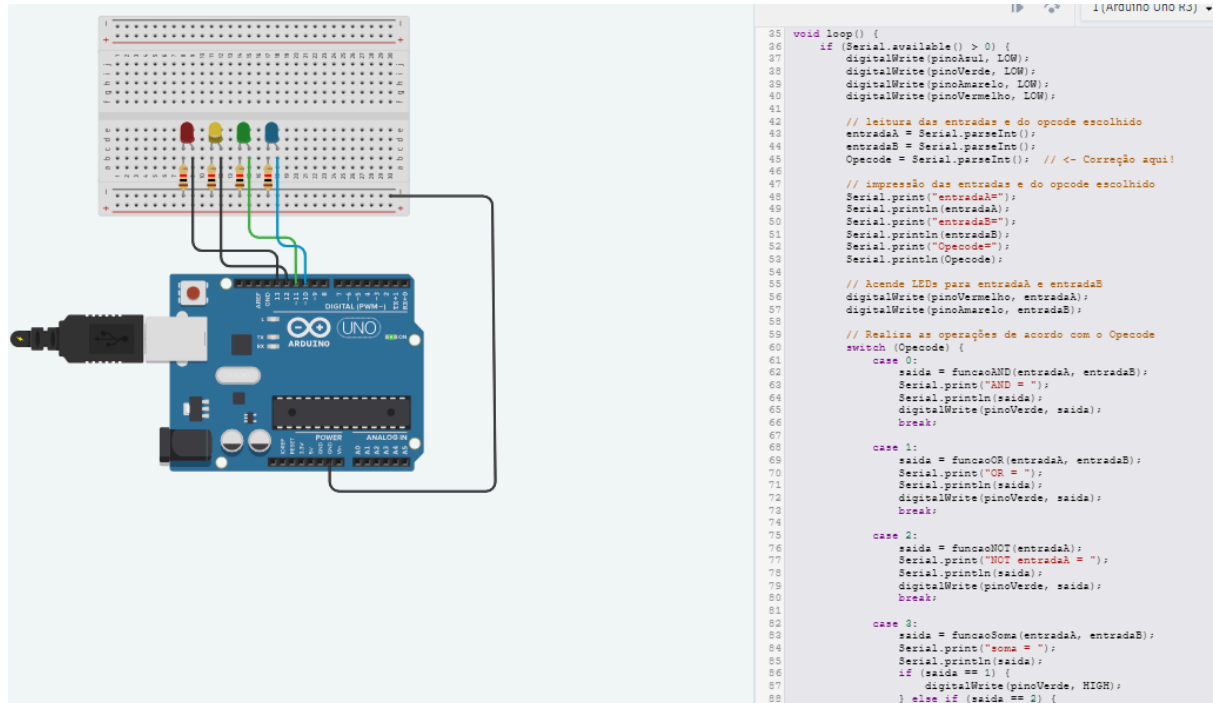
2 -



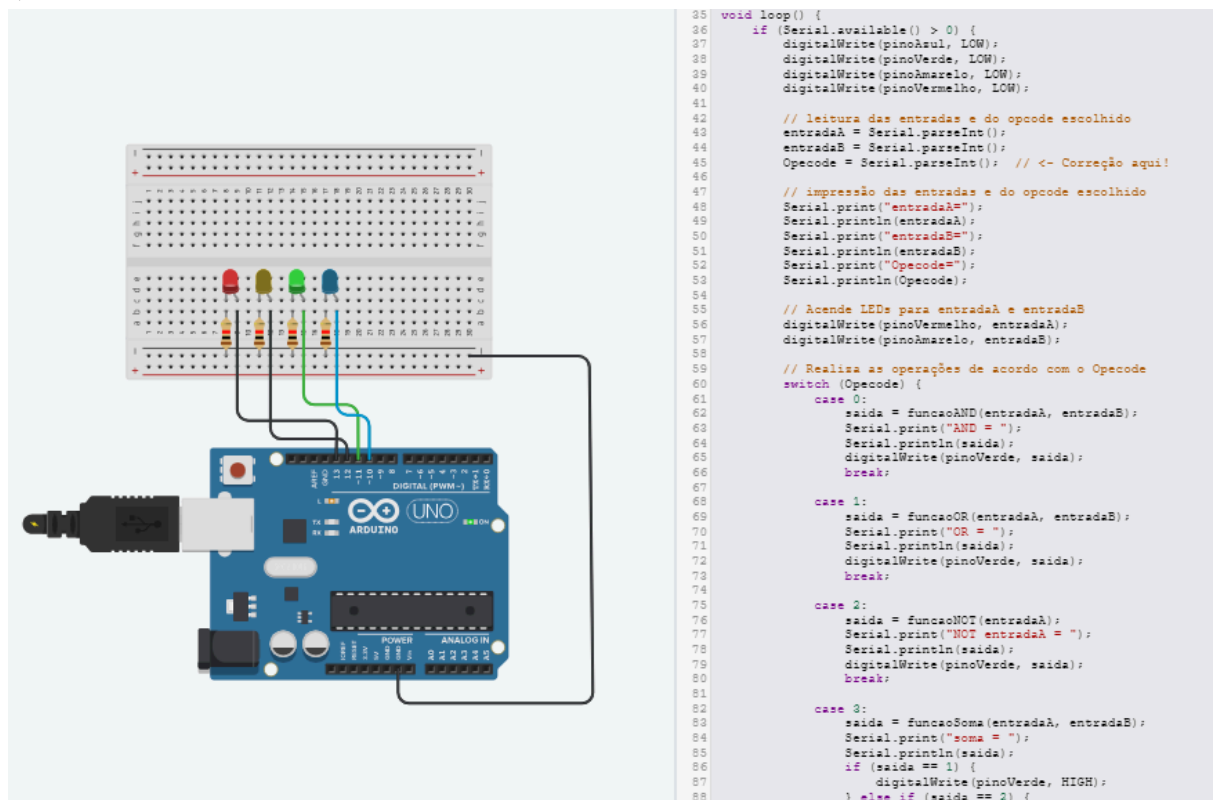
```
1 const int pinoVermelho = 13;
2 const int pinoAmarelo = 12;
3 const int pinoVerde = 11;
4 const int pinoAzul = 10;
5
6 // Variáveis globais
7 int entradaA, entradaB, Opecode, saida;
8
9 void setup() {
10     pinMode(pinoVermelho, OUTPUT);
11     pinMode(pinoAmarelo, OUTPUT);
12     pinMode(pinoVerde, OUTPUT);
13     pinMode(pinoAzul, OUTPUT);
14     Serial.begin(9600);
15 }
16
17 // Funções lógicas
18 int funcaoAND(int a, int b) {
19     return a & b;
20 }
21
22 int funcaoOR(int a, int b) {
23     return a | b;
24 }
25
26 int funcaoNOT(int a) {
27     return !a;
28 }
29
30 int funcaoSoma(int a, int b) {
31     return (a + b) % 2;
32 }
33
34 // Loop principal
35 void loop() {
36     if (Serial.available() > 0) {
37         // Desliga todos os LEDs
38     }
39 }
40
```

Testes no arduino

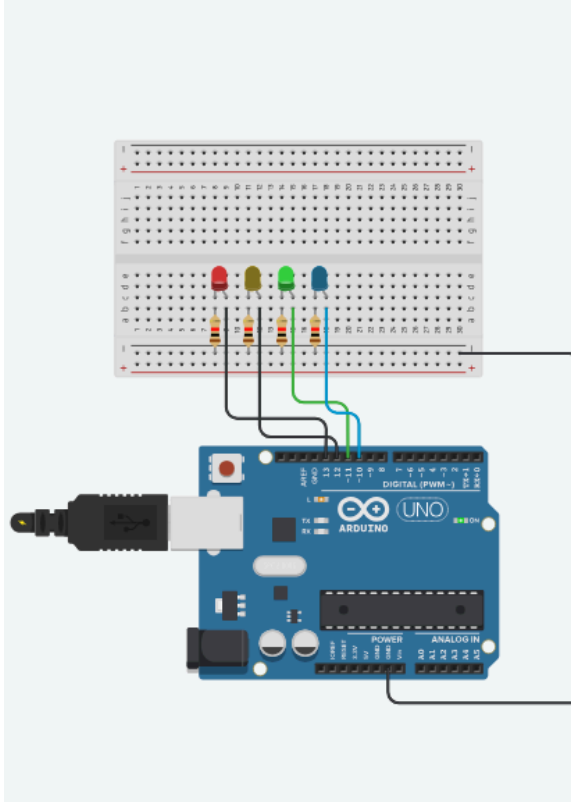
1) 0 1 0



2) 1 0 1



3) 1 0 3

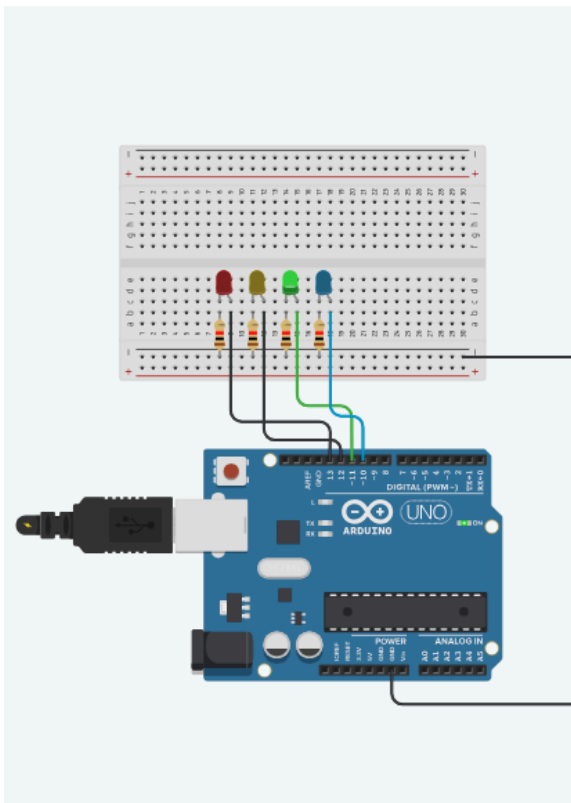


```

25 void loop() {
26   if (Serial.available() > 0) {
27     digitalWrite(pinoAzul, LOW);
28     digitalWrite(pinoVerde, LOW);
29     digitalWrite(pinoAmarelo, LOW);
30     digitalWrite(pinoVermelho, LOW);
31
32     // leitura das entradas e do opcode escolhido
33     entradaA = Serial.parseInt();
34     entradaB = Serial.parseInt();
35     OPCODE = Serial.parseInt(); // <- Correção aqui!
36
37     // impressão das entradas e do opcode escolhido
38     Serial.print("entradaA=");
39     Serial.println(entradaA);
40     Serial.print("entradaB=");
41     Serial.println(entradaB);
42     Serial.print("OPCODE=");
43     Serial.println(OPCODE);
44
45     // Acende LEDs para entradaA e entradaB
46     digitalWrite(pinoVermelho, entradaA);
47     digitalWrite(pinoAmarelo, entradaB);
48
49     // Realiza as operações de acordo com o OPCODE
50     switch (OPCODE) {
51       case 0:
52         saida = funcaoAND(entradaA, entradaB);
53         Serial.print("AND = ");
54         Serial.println(saida);
55         digitalWrite(pinoVerde, saida);
56         break;
57
58       case 1:
59         saida = funcaoOR(entradaA, entradaB);
60         Serial.print("OR = ");
61         Serial.println(saida);
62         digitalWrite(pinoVerde, saida);
63         break;
64
65       case 2:
66         saida = funcaoNOT(entradaA);
67         Serial.print("NOT entradaA = ");
68         Serial.println(saida);
69         digitalWrite(pinoVerde, saida);
70         break;
71
72       case 3:
73         saida = funcaoSoma(entradaA, entradaB);
74         Serial.print("soma = ");
75         Serial.println(saida);
76         if (saida == 1) {
77           digitalWrite(pinoVerde, HIGH);
78         } else if (saida == 2) {
79

```

4) 0 0 2



```

25 void loop() {
26   if (Serial.available() > 0) {
27     digitalWrite(pinoAzul, LOW);
28     digitalWrite(pinoVerde, LOW);
29     digitalWrite(pinoAmarelo, LOW);
30     digitalWrite(pinoVermelho, LOW);
31
32     // leitura das entradas e do opcode escolhido
33     entradaA = Serial.parseInt();
34     entradaB = Serial.parseInt();
35     OPCODE = Serial.parseInt(); // <- Correção aqui!
36
37     // impressão das entradas e do opcode escolhido
38     Serial.print("entradaA=");
39     Serial.println(entradaA);
40     Serial.print("entradaB=");
41     Serial.println(entradaB);
42     Serial.print("OPCODE=");
43     Serial.println(OPCODE);
44
45     // Acende LEDs para entradaA e entradaB
46     digitalWrite(pinoVermelho, entradaA);
47     digitalWrite(pinoAmarelo, entradaB);
48
49     // Realiza as operações de acordo com o OPCODE
50     switch (OPCODE) {
51       case 0:
52         saida = funcaoAND(entradaA, entradaB);
53         Serial.print("AND = ");
54         Serial.println(saida);
55         digitalWrite(pinoVerde, saida);
56         break;
57
58       case 1:
59         saida = funcaoOR(entradaA, entradaB);
60         Serial.print("OR = ");
61         Serial.println(saida);
62         digitalWrite(pinoVerde, saida);
63         break;
64
65       case 2:
66         saida = funcaoNOT(entradaA);
67         Serial.print("NOT entradaA = ");
68         Serial.println(saida);
69         digitalWrite(pinoVerde, saida);
70         break;
71
72       case 3:
73         saida = funcaoSoma(entradaA, entradaB);
74         Serial.print("soma = ");
75         Serial.println(saida);
76         if (saida == 1) {
77           digitalWrite(pinoVerde, HIGH);
78         } else if (saida == 2) {
79

```

Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0 1 00	0x4	0
OR(A,B)	1 0 0 1	0x9	1
SOMA(A,B)	1 0 1 1	0xb	1
NOT(A)	0 0 1 0	0x2	1
AND(B,A)	0 1 0 0	0x4	0