

# Trabalho Prático 02 - Arquitetura de Computadores

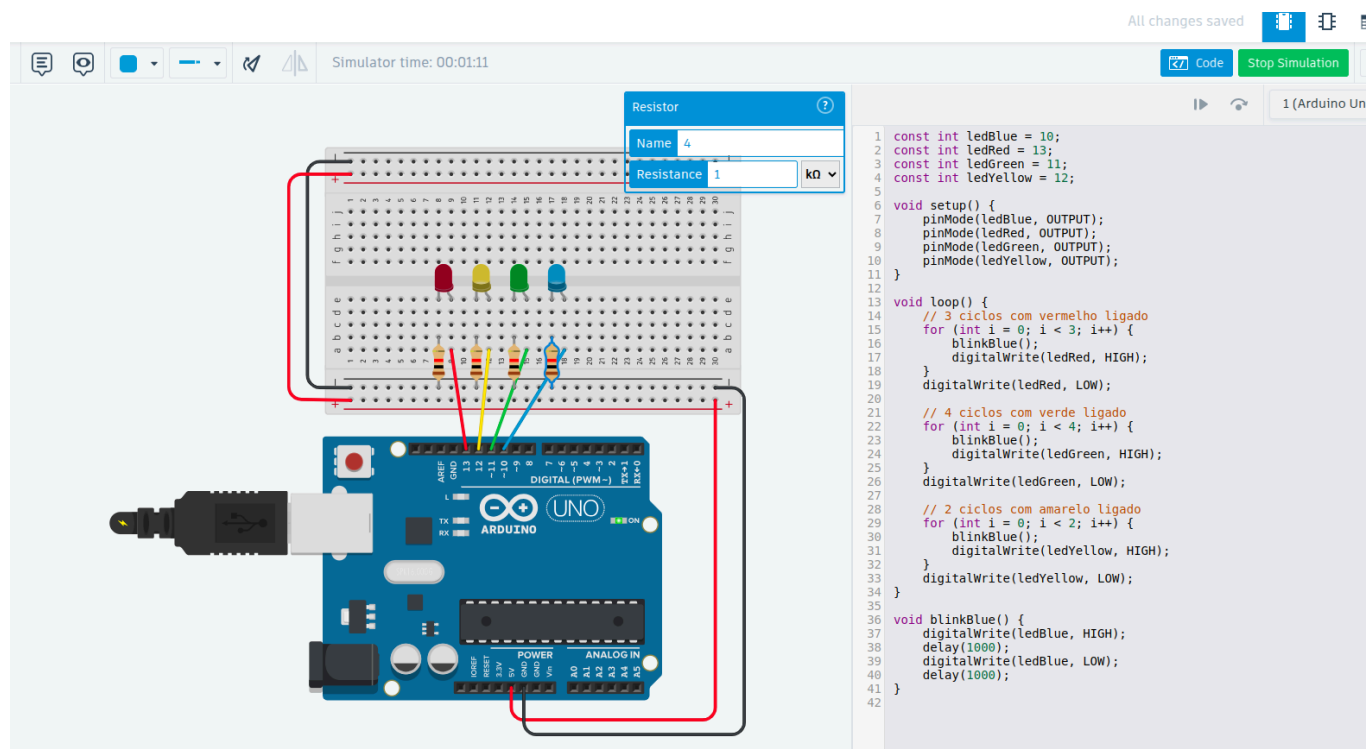
Aluno: Mariana Almeida Mendonça

## Questão 01

Para esse projeto foi usado quatro LEDs (vermelho, verde, amarelo e azul), além de quatro resistências de 1kΩ. Para completar a montagem do circuito, foi utilizada uma protoboard e uma placa Arduino.

Sobre o programa, foram usados três métodos, sendo dois deles padrão do Arduino. O primeiro é o método `setup`, responsável pela configuração dos pinos de cada LED, especificando a porta e o tipo de saída. O segundo é o método `loop`, que define a lógica do funcionamento do semáforo, garantindo que o LED azul pisque a cada 1 segundo e alternando entre os LEDs vermelho, verde e amarelo em seus respectivos tempos.

Por fim, foi criado um método auxiliar chamado `piscaAzul`, que facilita a repetição do efeito de piscar do LED azul, tornando o código mais organizado e modularizado.



## Questão 02

Este projeto utiliza um Arduino com quatro LEDs (vermelho, amarelo, verde e azul) para representar os valores de entrada e saída de uma Unidade Lógica e Aritmética (ULA). Através

The image displays the hardware implementation of a 4-bit adder using an Arduino Uno and a breadboard. The breadboard contains four LEDs (red, yellow, green, blue) and four 220Ω resistors. The Arduino Uno is connected to the breadboard via jumper wires. The code implements a 4-bit adder using bitwise operations (AND, OR, NOT, XOR) to calculate the sum and carry. The Serial Monitor shows the output for three test cases: (0,0), (0,1), and (1,1).

**Hardware Setup:**

- Arduino Uno connected to a breadboard.
- Breadboard contains four LEDs (red, yellow, green, blue) and four 220Ω resistors.
- Wires connect the Arduino pins to the breadboard components.

**Code (Arduino IDE):**

```

1 int ledA = 13; // LED vermelho (a)
2 int ledB = 12; // LED amarelo (b)
3 int ledOutput = 11; // LED verde (Saída ULA)
4 int ledCarryOut = 10; // LED azul (Vail)
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode(ledA, OUTPUT);
9   pinMode(ledB, OUTPUT);
10  pinMode(ledOutput, OUTPUT);
11  pinMode(ledCarryOut, OUTPUT);
12 }
13
14 void loop() {
15   if (Serial.available() >= 3) {
16     // Lendo os três valores
17     int a = Serial.read() - '0'; // Converte char para int
18     int b = Serial.read() - '0';
19     int opCode = Serial.read() - '0';
20
21     int output = 0, carryOut = 0;
22
23     // Operações da ULA
24     switch (opCode) {
25       case 0: // AND (a & b)
26         output = a & b;
27         break;
28       case 1: // OR (a | b)
29         output = a | b;
30         break;
31       case 2: // NOT (a)
32         output = !a;
33         break;
34       case 3: // Soma (a + b)
35         output = (a + b) % 2; // Soma binária
36         carryOut = (a + b) / 2; // carryOut (carry)
37         break;
38     }
39
40     // Exibir no Serial Monitor
41     Serial.print("a = "); Serial.print(a);
42     Serial.print(", b = "); Serial.print(b);
43     Serial.print(", opCode = "); Serial.print(opCode);
44     Serial.print(" -> Saída = "); Serial.print(output);
45     Serial.print(", Vail = "); Serial.println(carryOut);
46
47     // Atualizar LEDs
48     digitalWrite(ledA, a);
49     digitalWrite(ledB, b);
50     digitalWrite(ledOutput, output);
51     digitalWrite(ledCarryOut, carryOut);
52   }
53 }
54

```

**Serial Monitor Output:**

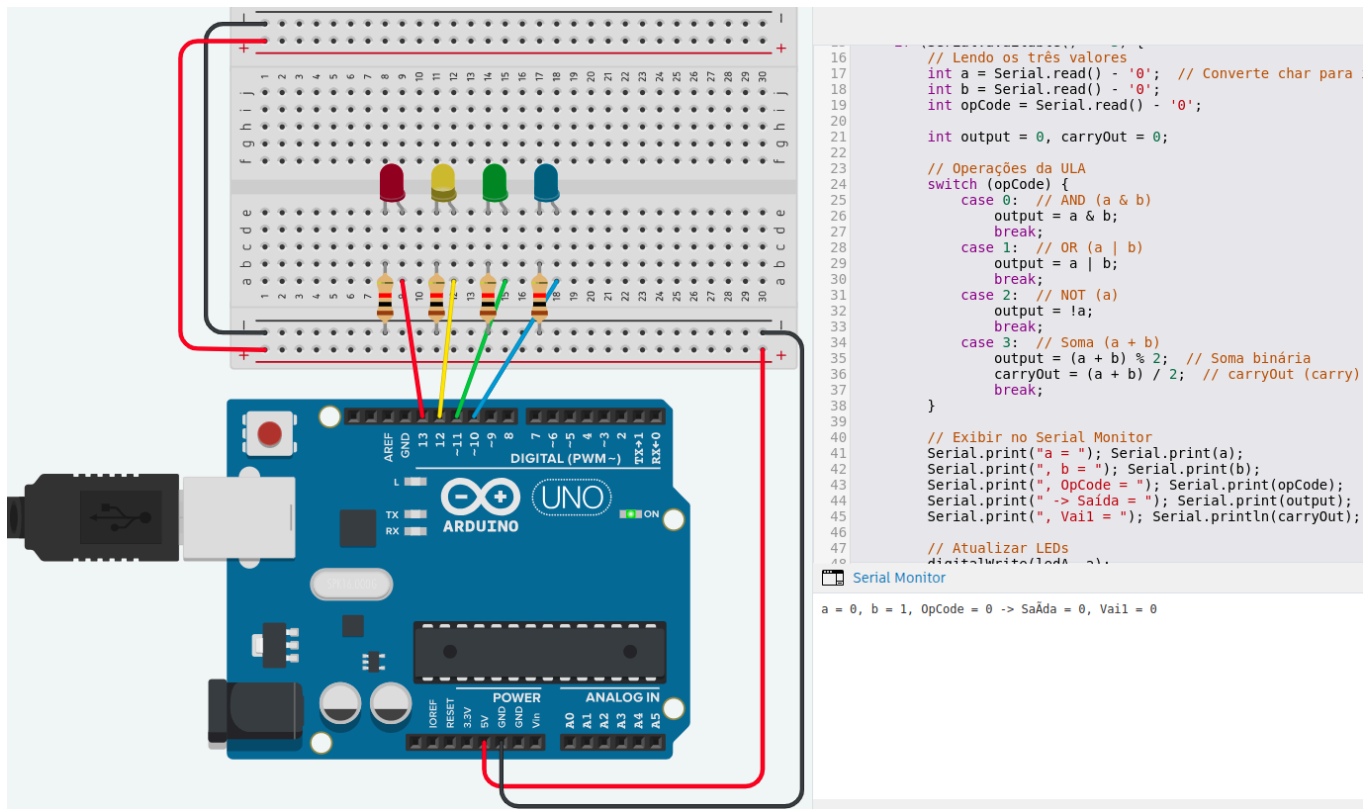
```

a = 1, b = 0, opCode = 0 -> Saída = 0, Vail = 0
a = 0, b = 1, opCode = 2 -> Saída = 1, Vail = 0
a = 1, b = 1, opCode = 3 -> Saída = 0, Vail = 1

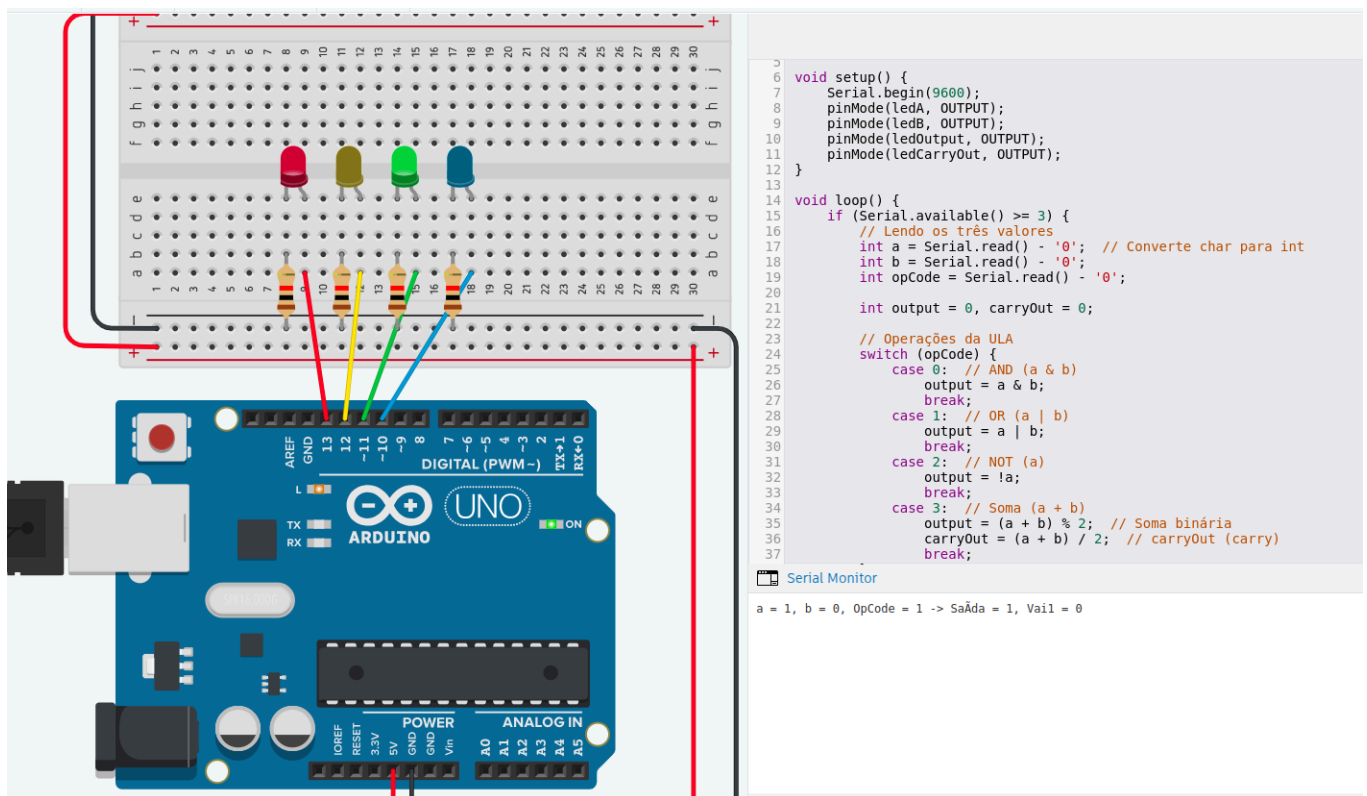
```

x	Instrução	Binário	Hexadecimal	Resultado
1	<b>AND(A,B)</b>	0 1 00	0x4	0
2	<b>OR(A,B)</b>	1 0 01	0x9	1
3	<b>SOMA(A,B)</b>	1 0 11	0xB	1
4	<b>NOT(A)</b>	0 0 10	0x2	1
5	<b>AND(B,A)</b>	1 1 00	0xC	1

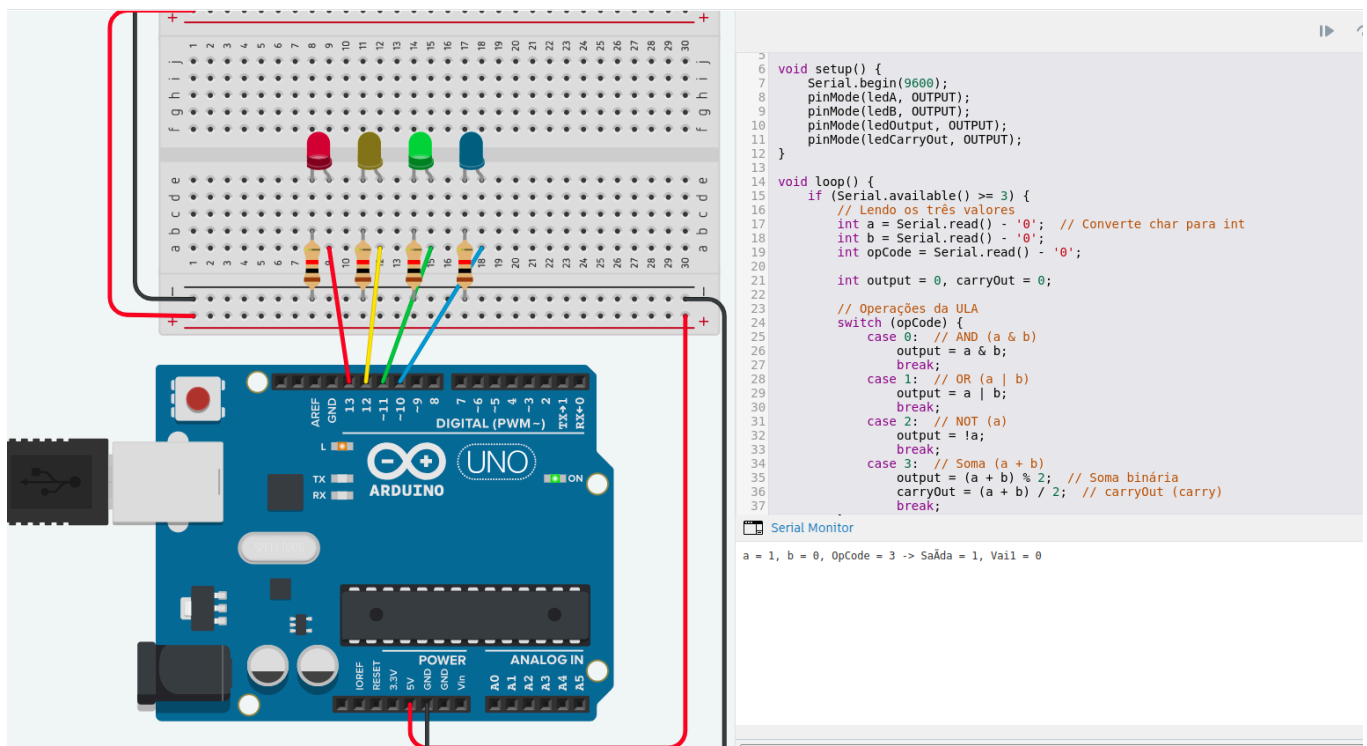
### Instrução 1:



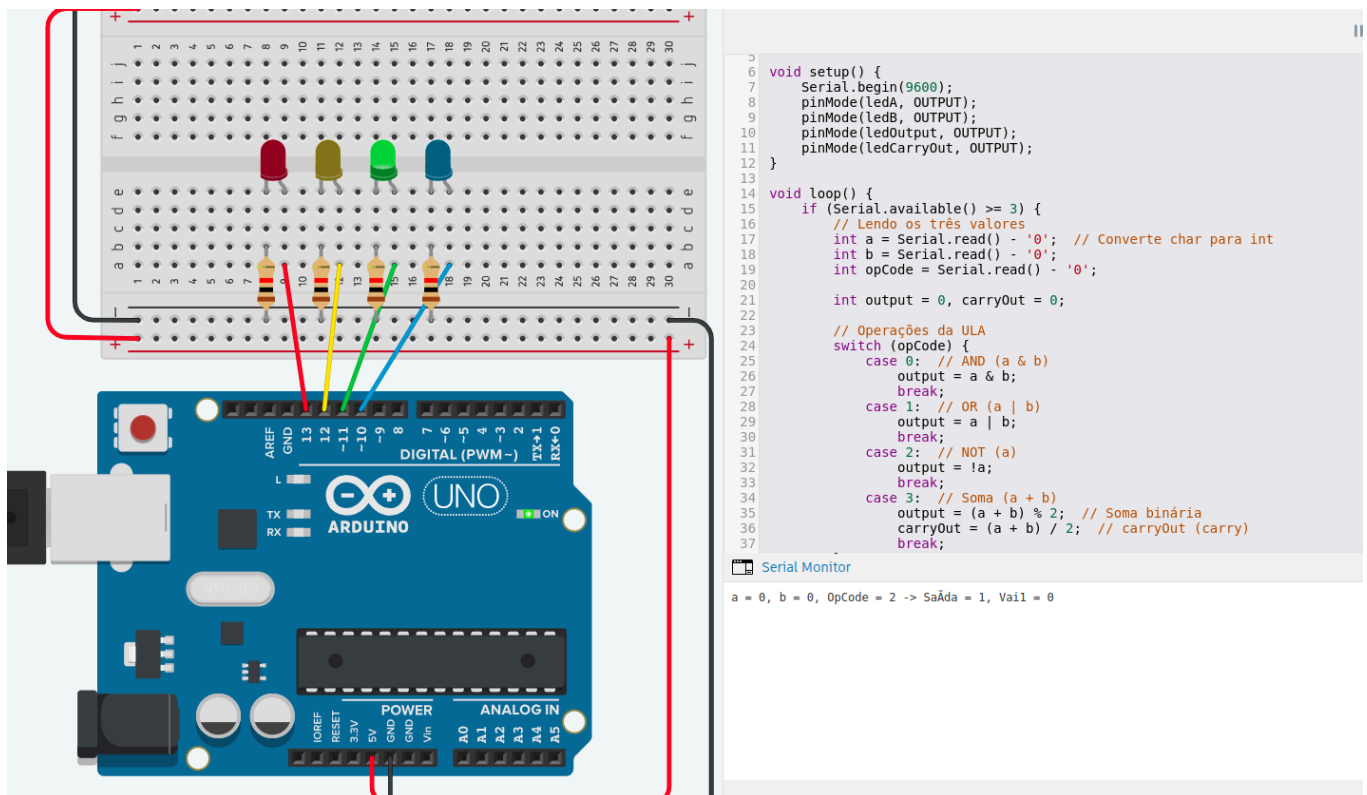
## Instrução 2



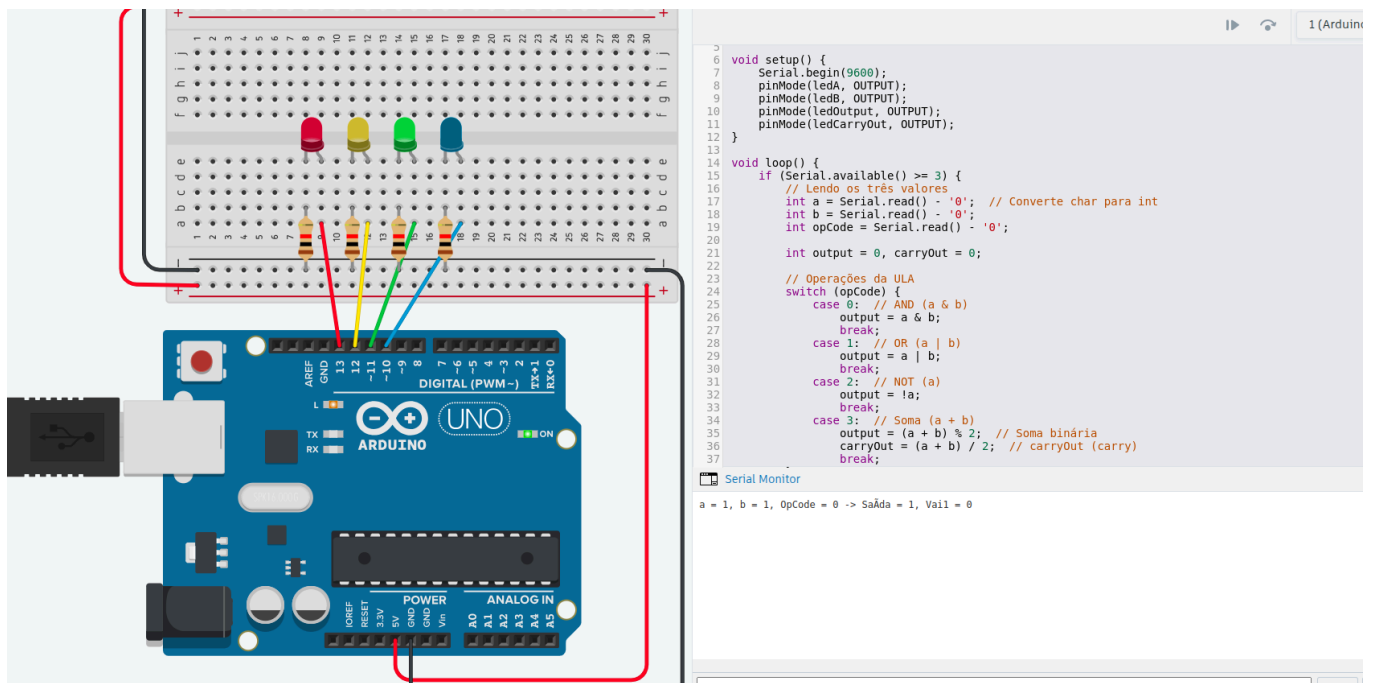
## Instrução 3



## Instrução 4



## Instrução 5



**OBS: Quando há a ocorrência de "Vai1":**

