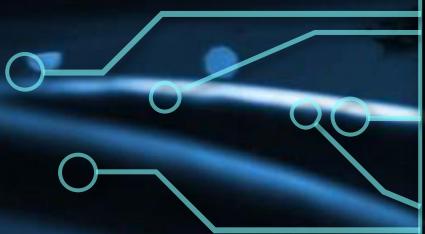
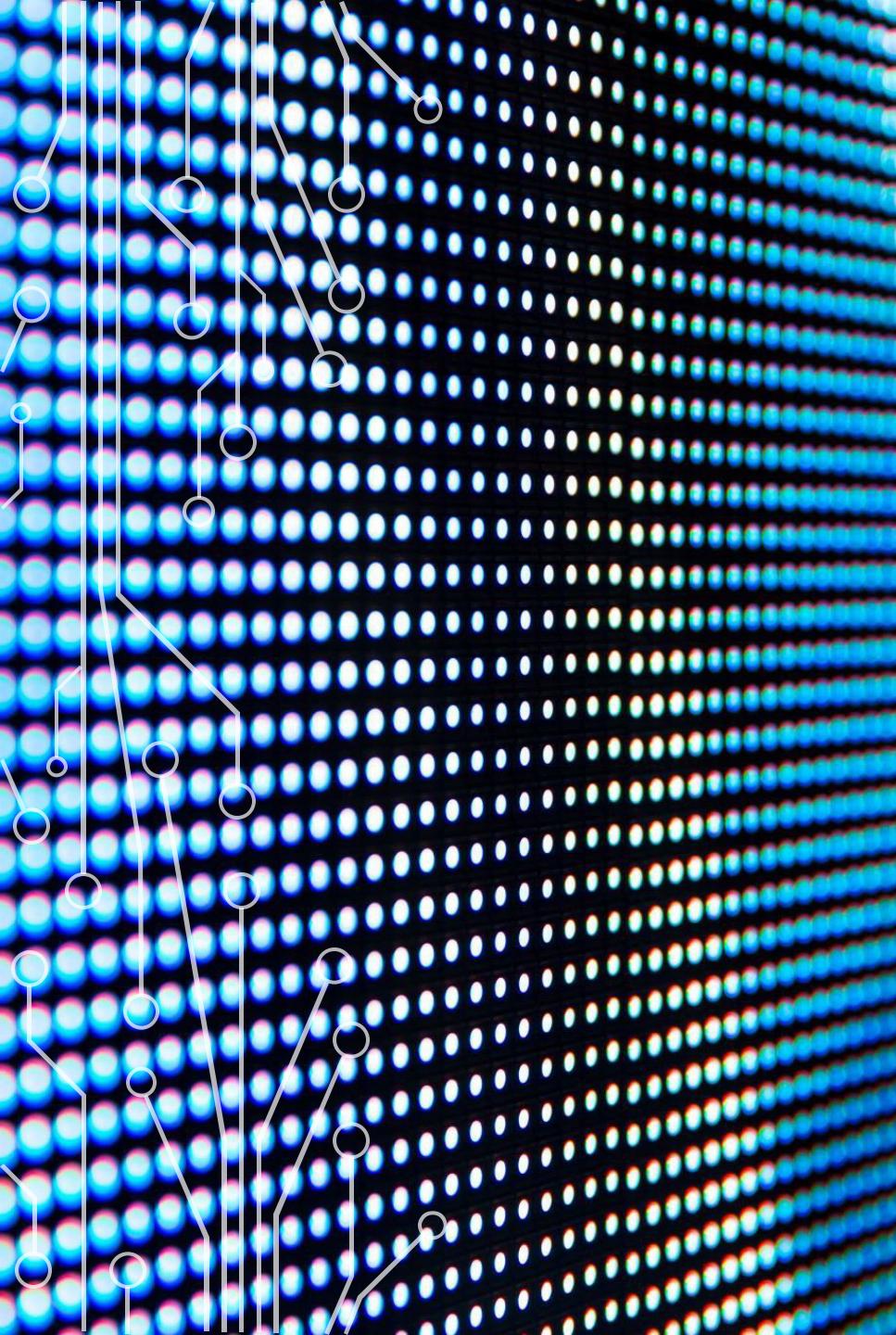




MICROCONTROLADORES

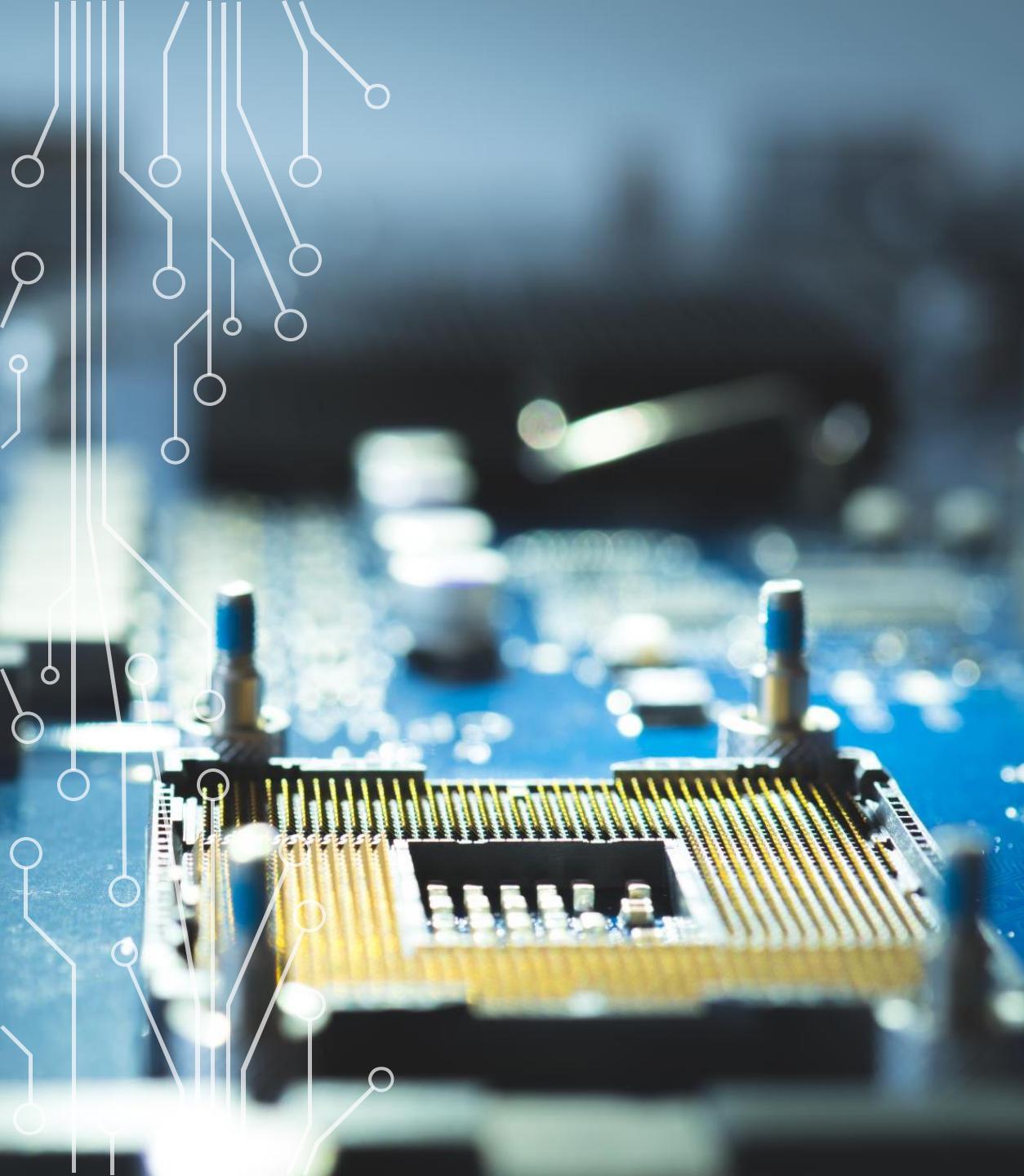
TAREFAS SIMPLES



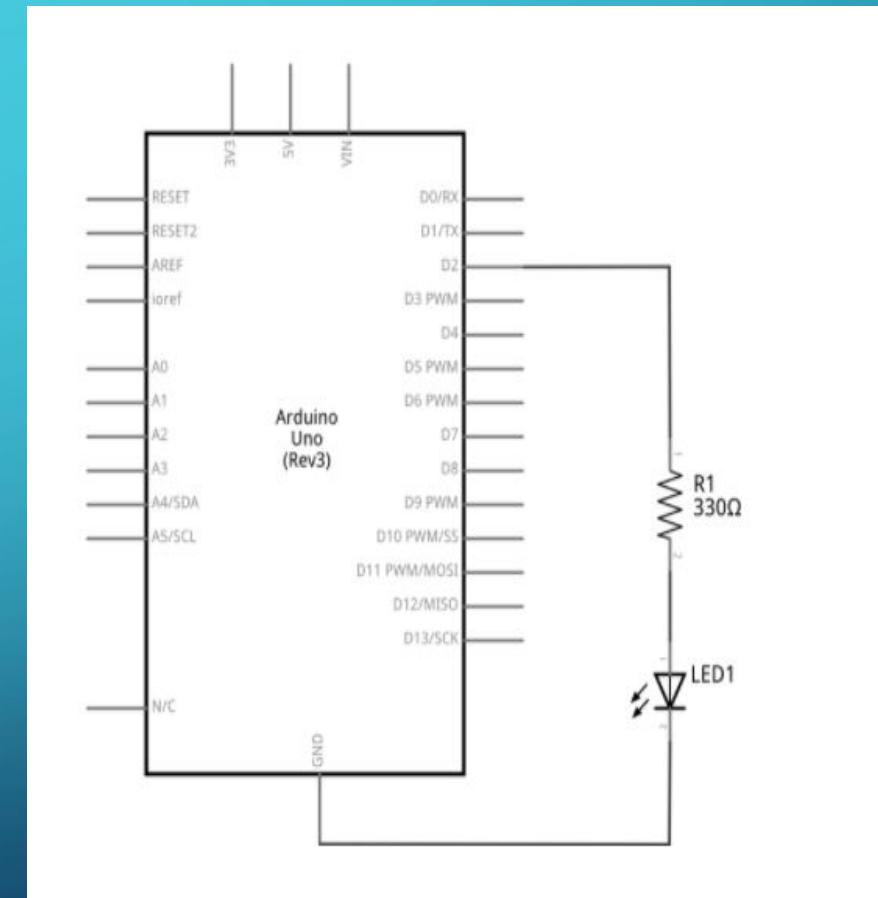


LUZES DE NATAL

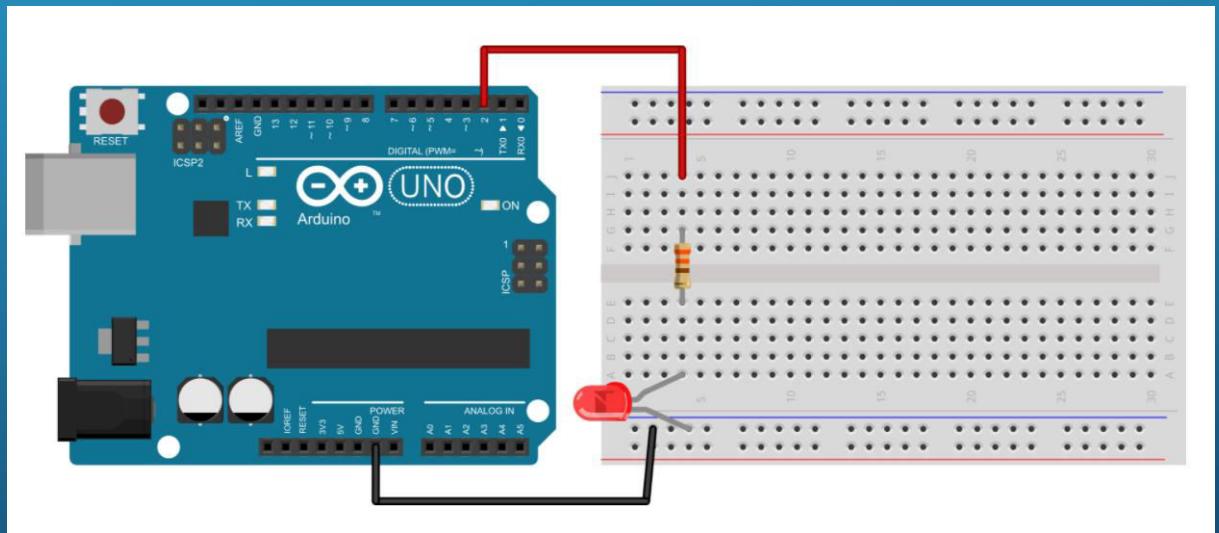
- Conectar **um** LED com um microcontrolador.
Escrever um programa que ligue e desligue o díodo em intervalos de meio segundo para obter um efeito intermitente.



ESQUEMA – UM LED



TELHA EXPERIMENTAL



SOLUÇÃO

```
• int led = 2; //definir led = 2

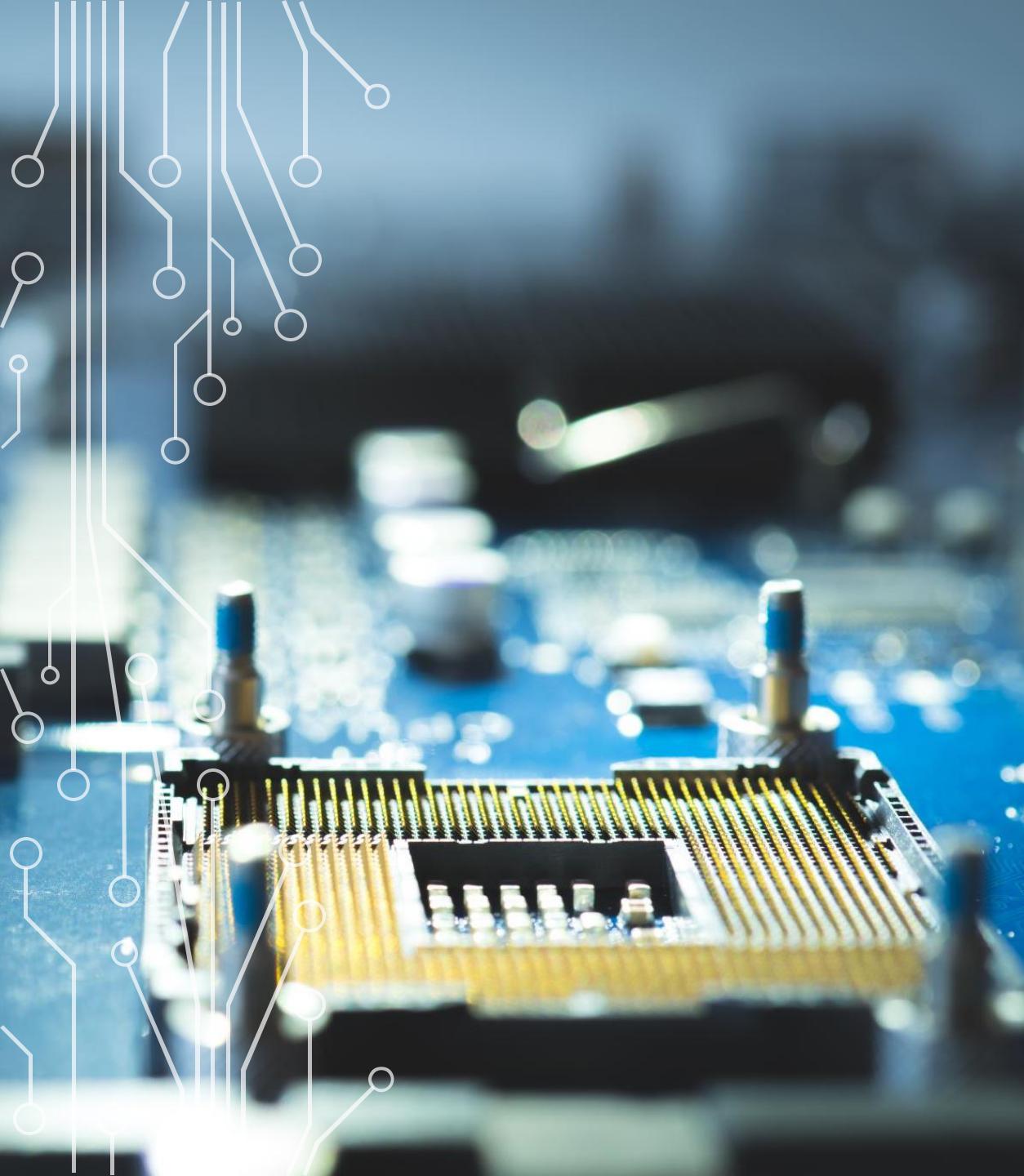
void setup() {
    pinMode(led, OUTPUT); //definir instrução
    led como saída
    digitalWrite(led, LOW); //desligar o LED -
    initial state
}

void loop() {
    digitalWrite(led, HIGH); //ligar o LED
    delay(500); //esperar 500 ms -
    meio segundo
    digitalWrite(led, LOW); //desligar o LED
    delay(500); //esperar 500 ms -
    meio segundo
}
```

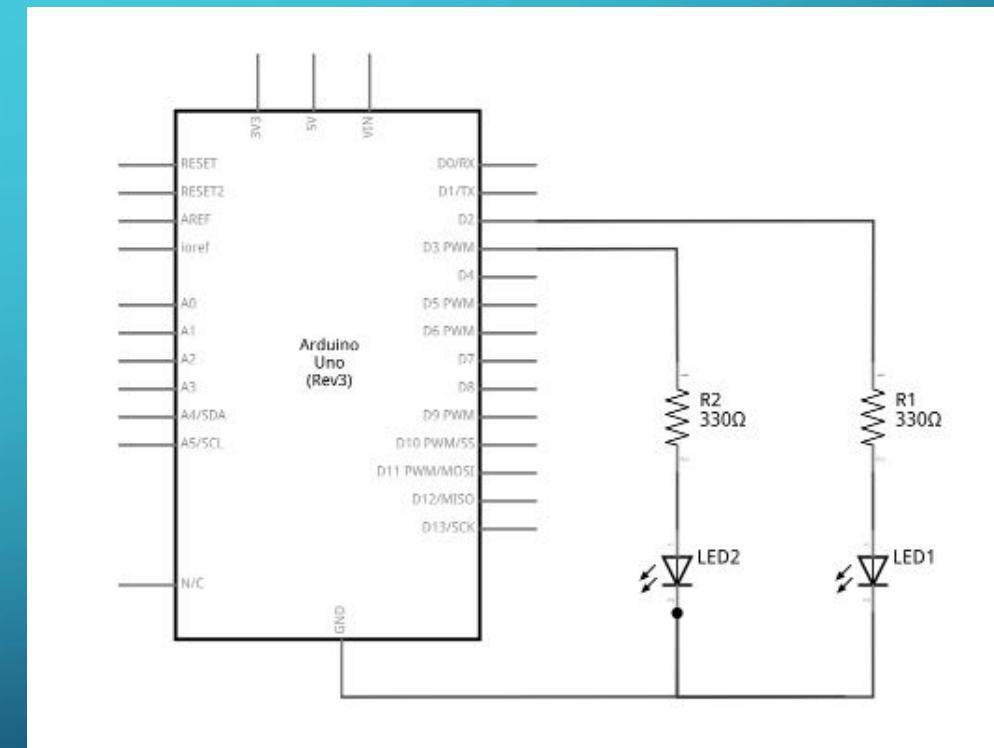


LUZES DE NATAL

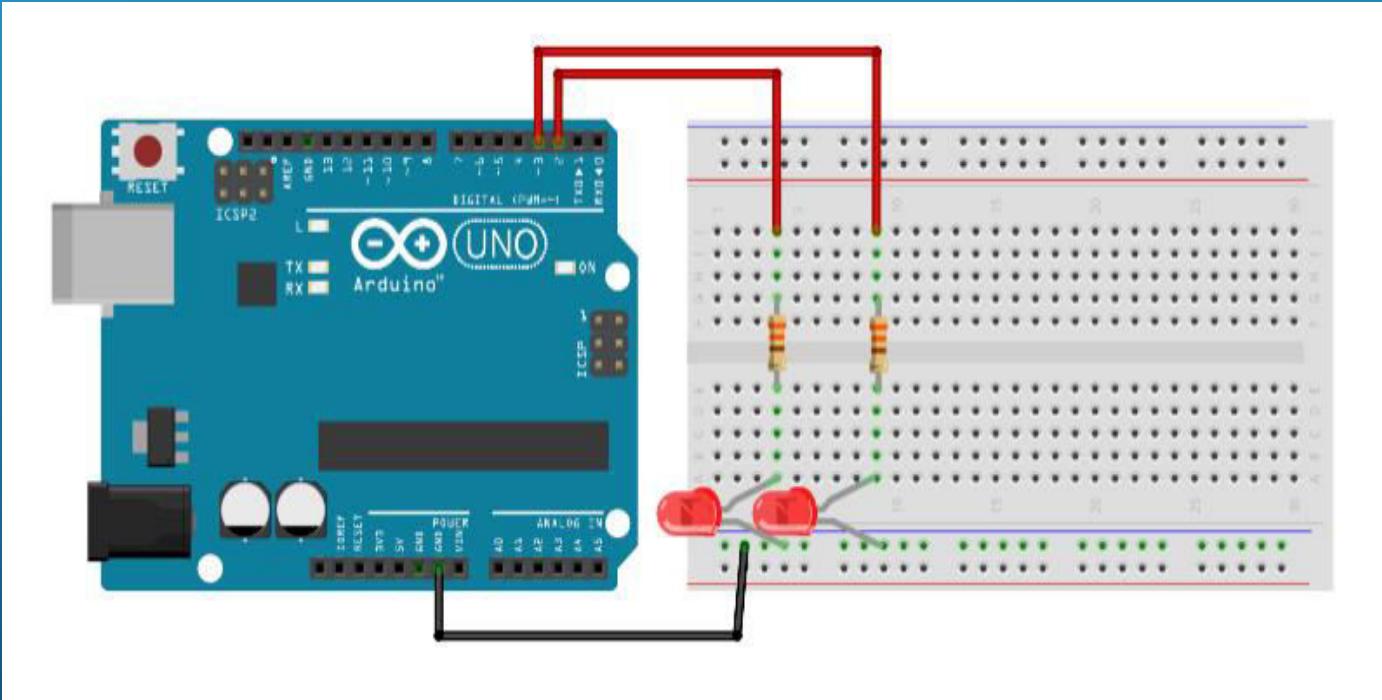
- Conectar **dois** LEDs ao microcontrolador.
Escrever um programa que ligue e desligue os LEDs em intervalos de meio segundo para obter o efeito de que ambos os LEDs pisquem juntos.



ESQUEMA – DOIS LED'S



TELHA EXPERIMENTAL



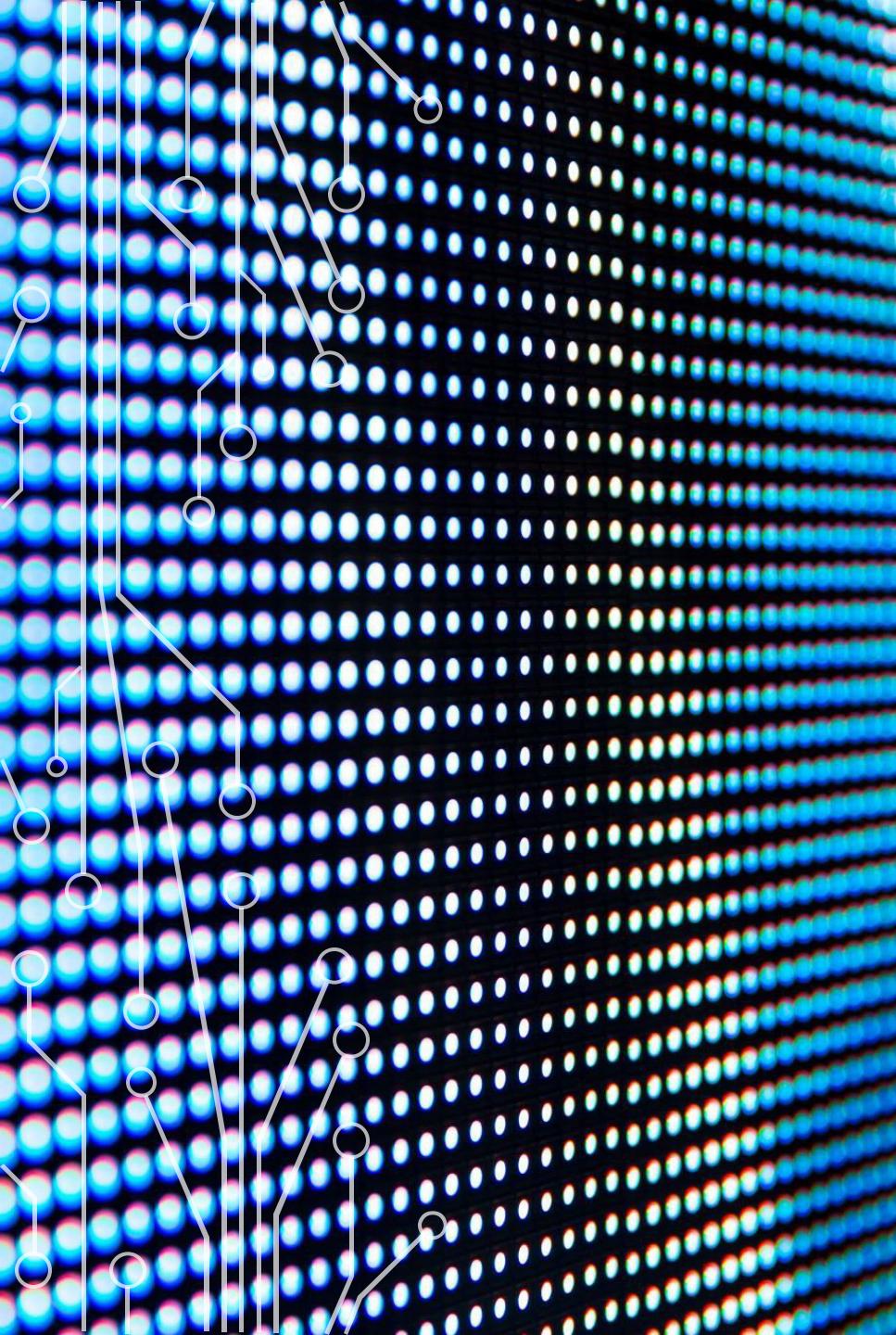
SOLUÇÃO

```
• int led1 = 2; //definir led1 = 2
  int led2 = 3; //definir led2 = 3

void setup() {
  pinMode(led1, OUTPUT); //define a instrução led1 como
  saída
  pinMode(led2, OUTPUT); //define a instrução led2 como
  saída
  digitalWrite(led1, LOW); //desligar o LED 1 - estado
  inicial
  digitalWrite(led2, LOW); //desligar o LED 2 - estado
  inicial
}

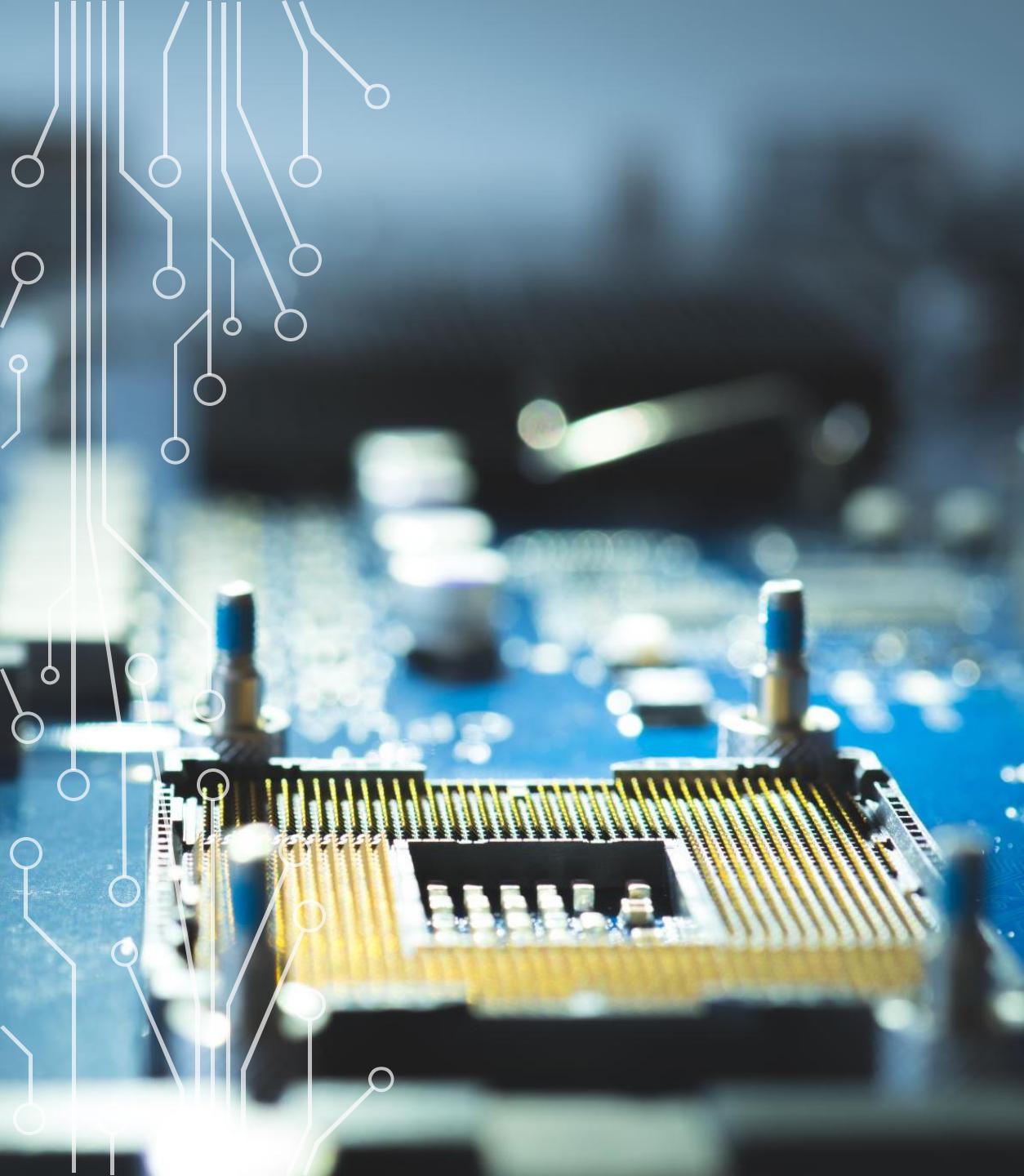
void loop() {
  digitalWrite(led1, HIGH); // ligar o LED 1
  digitalWrite(led2, HIGH); // ligar o LED 2

  delay(500); // esperar 500 ms - meio
  segundo
  digitalWrite(led1, LOW); // desligar o LED 1
  digitalWrite(led2, LOW); // desligar o LED 2
  delay(500); // esperar 500 ms - meio
  segundo
}
```

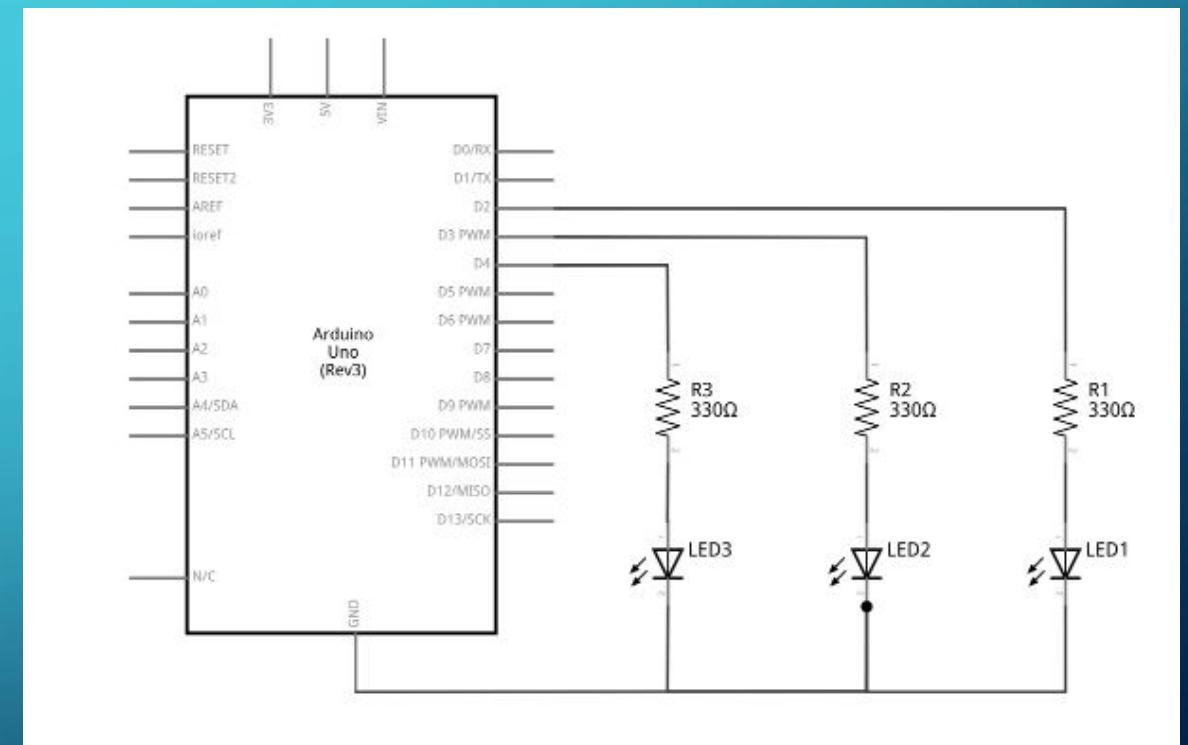


LUZES DE NATAL

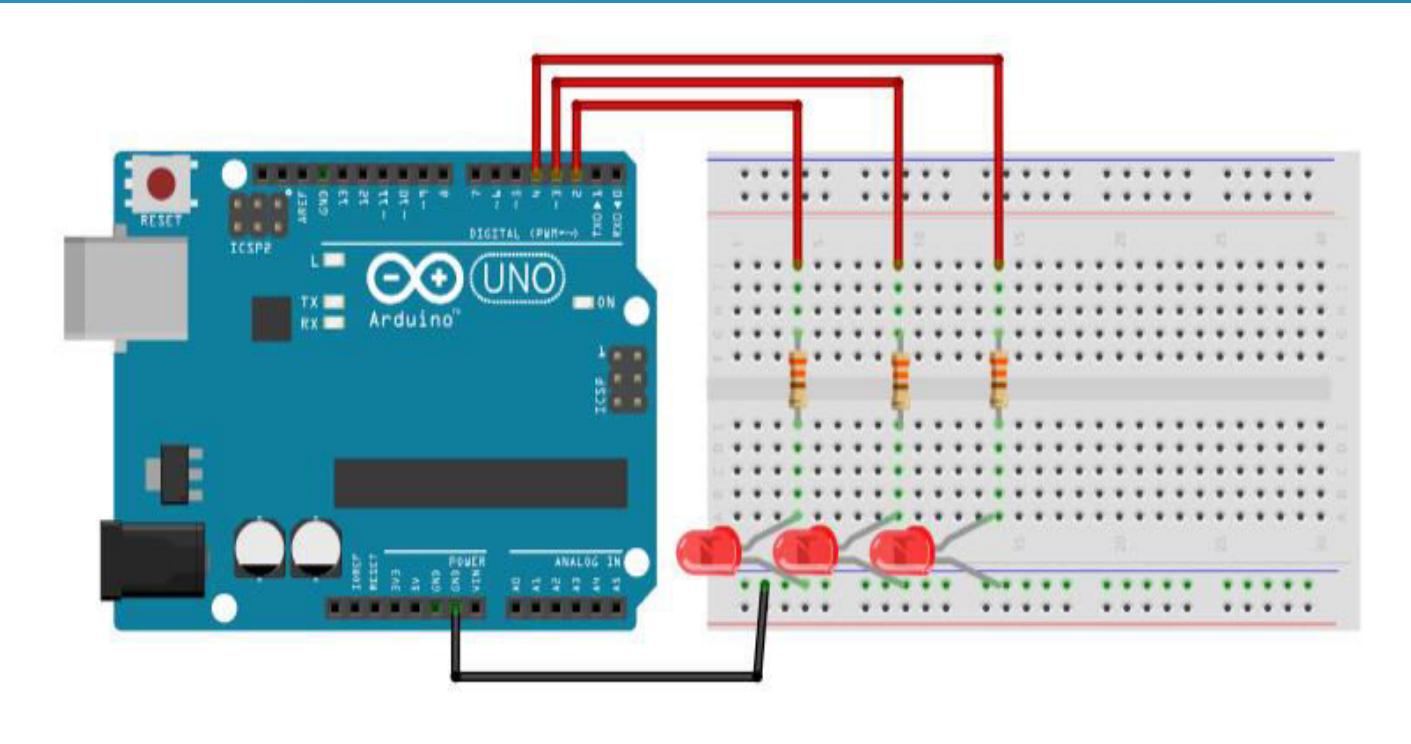
- Conectar **três** LEDs ao microcontrolador.
Escrever um programa que acenda os LEDs da esquerda para a direita sucessivamente.
Apenas um LED está aceso de cada vez. Após desligar o último LED, o primeiro acende-se novamente e o ciclo recomeça desde o início.
Cada LED está aceso por trezentos milissegundos.



ESQUEMA – TRÊS LED'S



TELHA EXPERIMENTAL



SOLUÇÃO

```
• int led1 = 2;          // definir led1 = 2
  int led2 = 3;          // definir led2 = 3
  int led3 = 4;          // definir led3 = 4

void setup() {
  pinMode(led1, OUTPUT); //definir a instrução
  led1 como saída
  pinMode(led2, OUTPUT); // definir a instrução
  led2 como saída
  pinMode(led3, OUTPUT); // definir a instrução
  led3 como saída
  digitalWrite(led1, LOW); //desligar o LED 1 -
  estado inicial
  digitalWrite(led2, LOW); // desligar o LED 2 -
  estado inicial
  digitalWrite(led3, LOW); // desligar o LED 3 -
  estado inicial
}

void loop() {
  digitalWrite(led1, HIGH); //ligar o LED 1
  delay(300);             // esperar 300 ms
  digitalWrite(led1, LOW); // desligar o LED 1

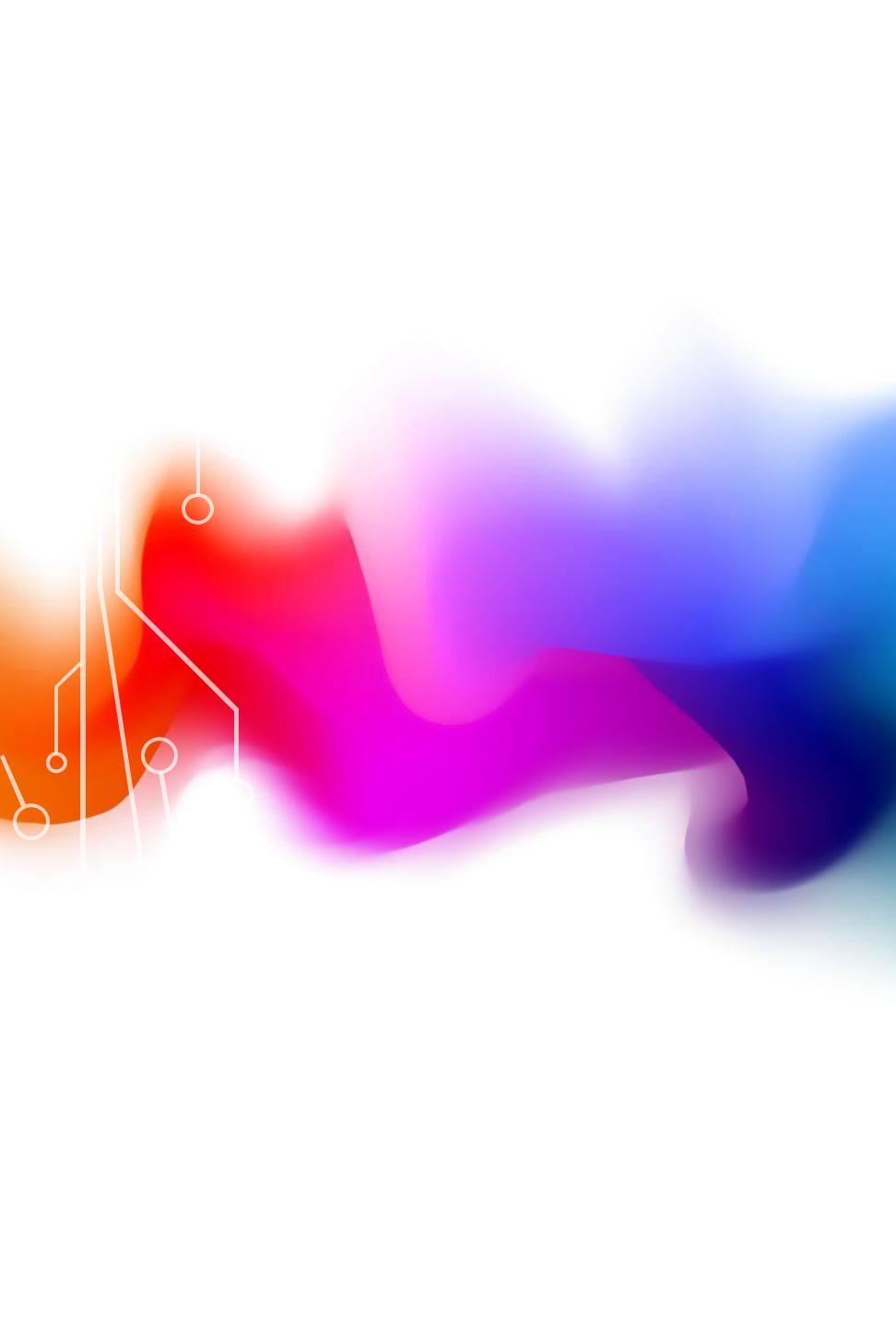
  digitalWrite(led2, HIGH); // ligar o LED 2
  delay(300);             // esperar 300 ms
  digitalWrite(led2, LOW); // desligar o LED 2

  digitalWrite(led3, HIGH); // ligar o LED 3
  delay(300);             // esperar 300 ms
  digitalWrite(led3, LOW); // desligar o LED 3
}
```



SEMÁFOROS

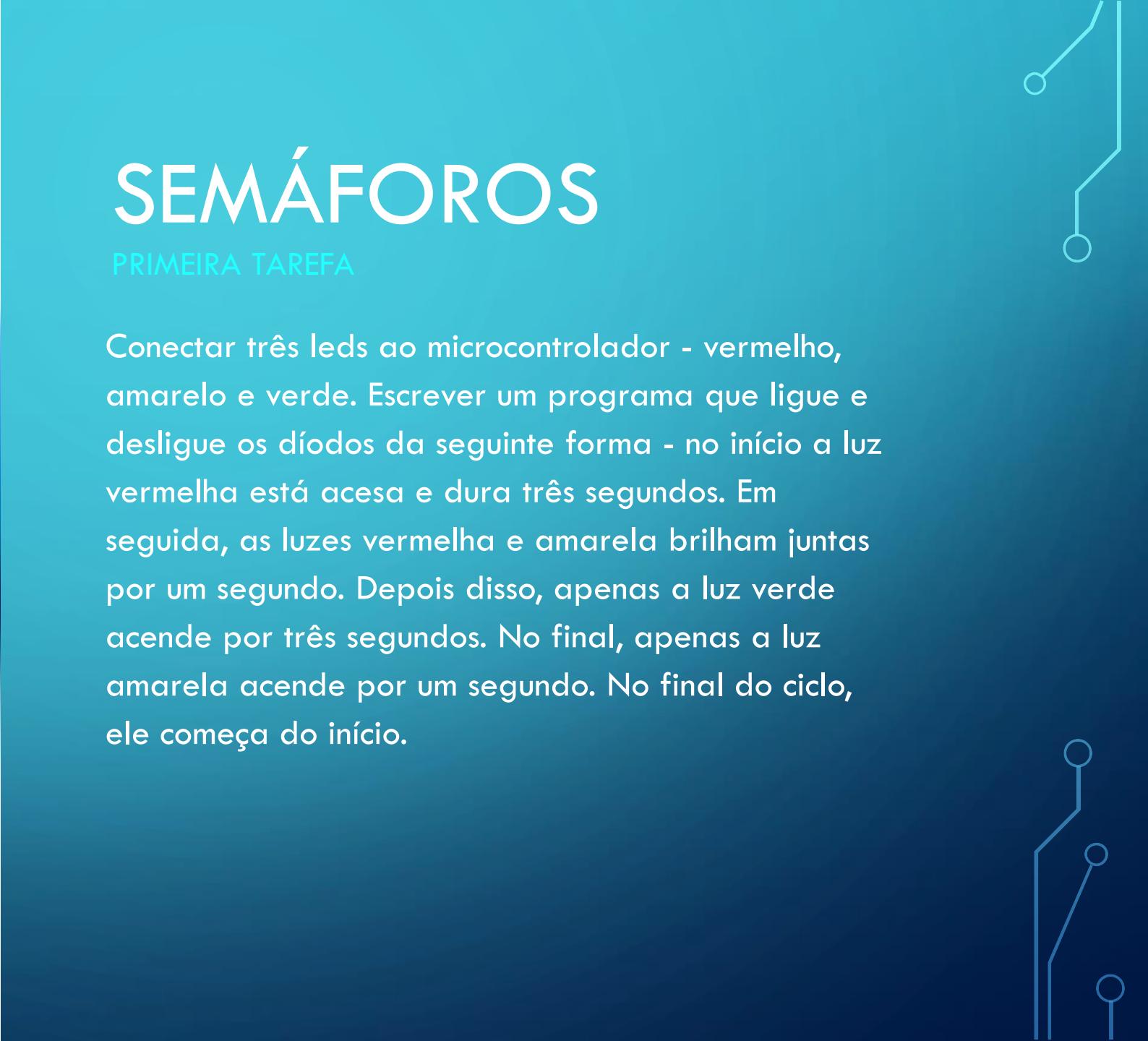
DUAS TAREFAS

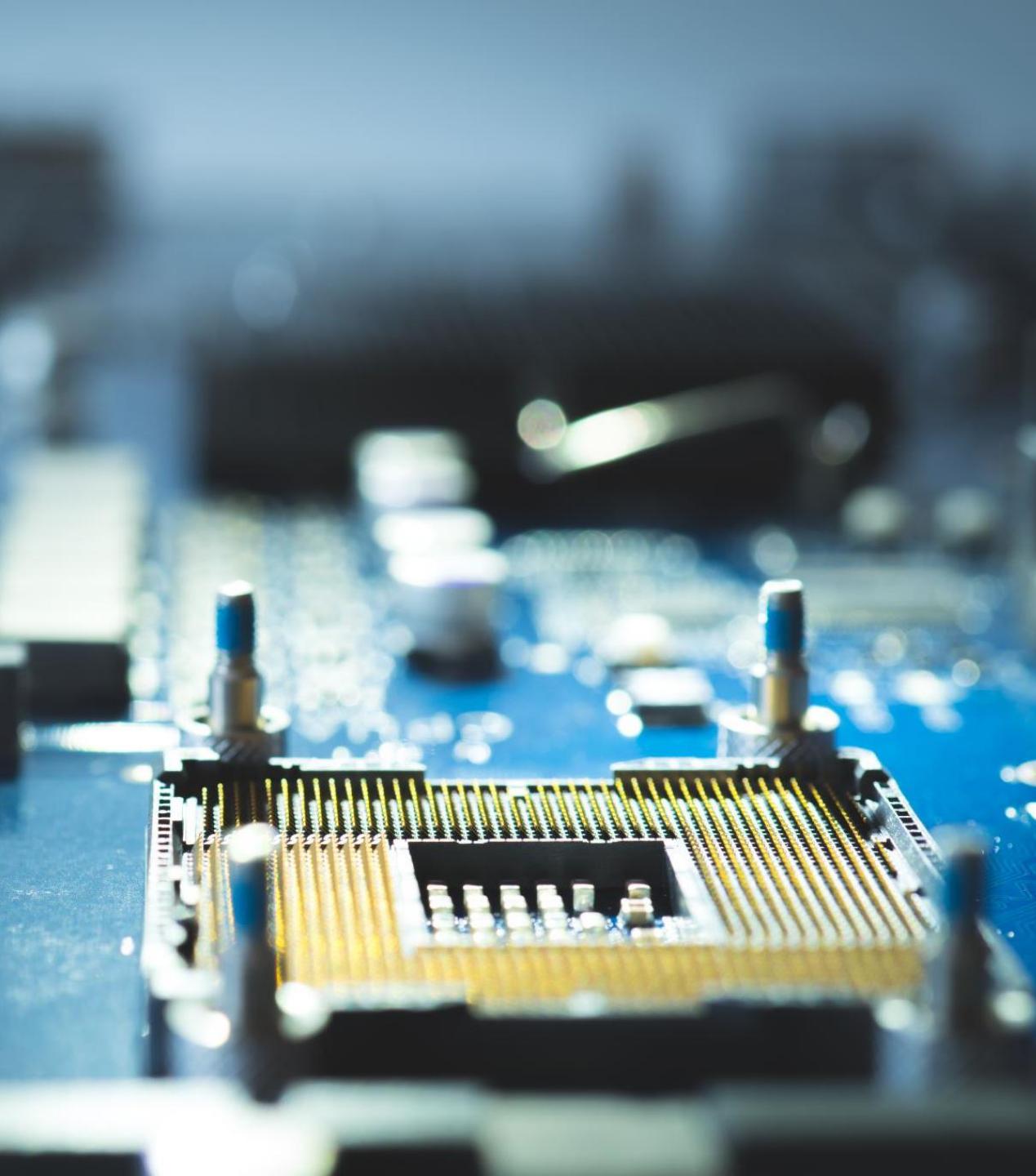


SEMÁFOROS

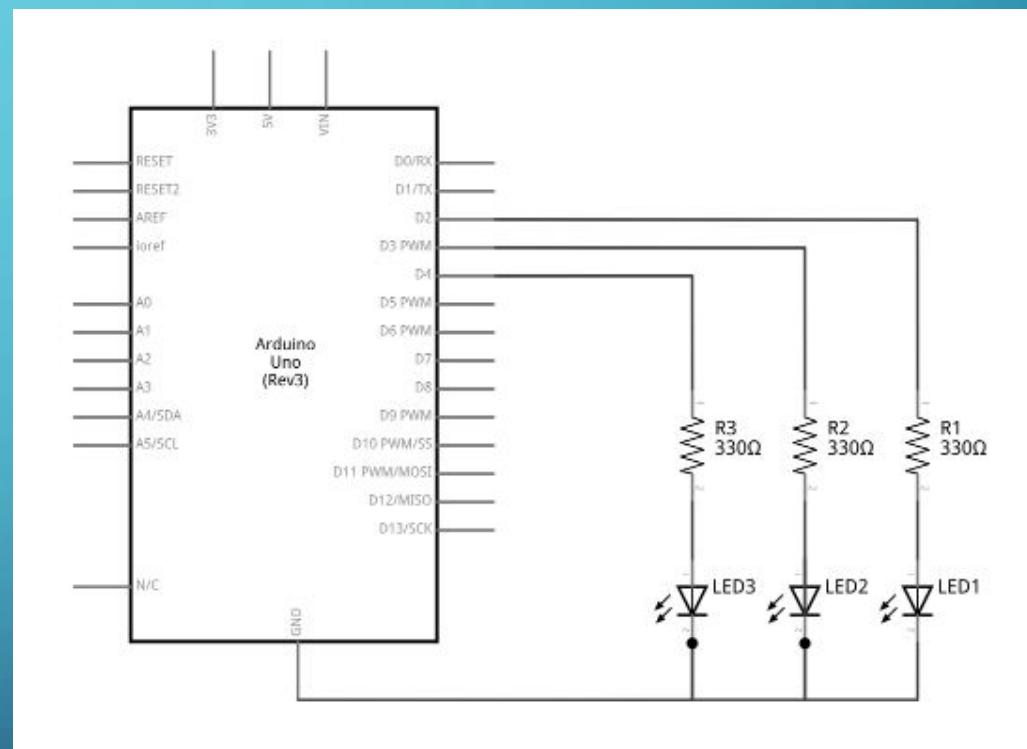
PRIMEIRA TAREFA

Coneectar três leds ao microcontrolador - vermelho, amarelo e verde. Escrever um programa que ligue e desligue os díodos da seguinte forma - no início a luz vermelha está acesa e dura três segundos. Em seguida, as luzes vermelha e amarela brilham juntas por um segundo. Depois disso, apenas a luz verde acende por três segundos. No final, apenas a luz amarela acende por um segundo. No final do ciclo, ele começa do início.

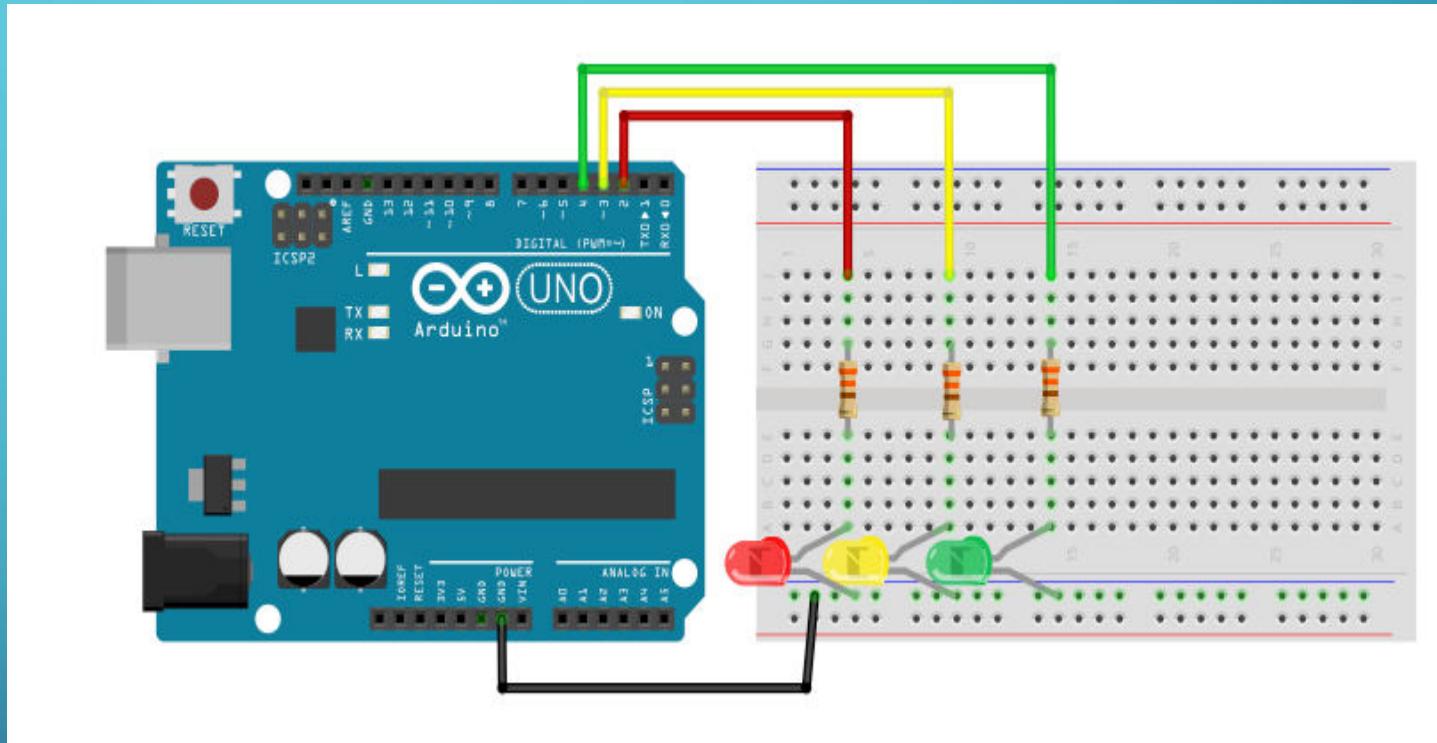




ESQUEMA



TELHA EXPERIMENTAL



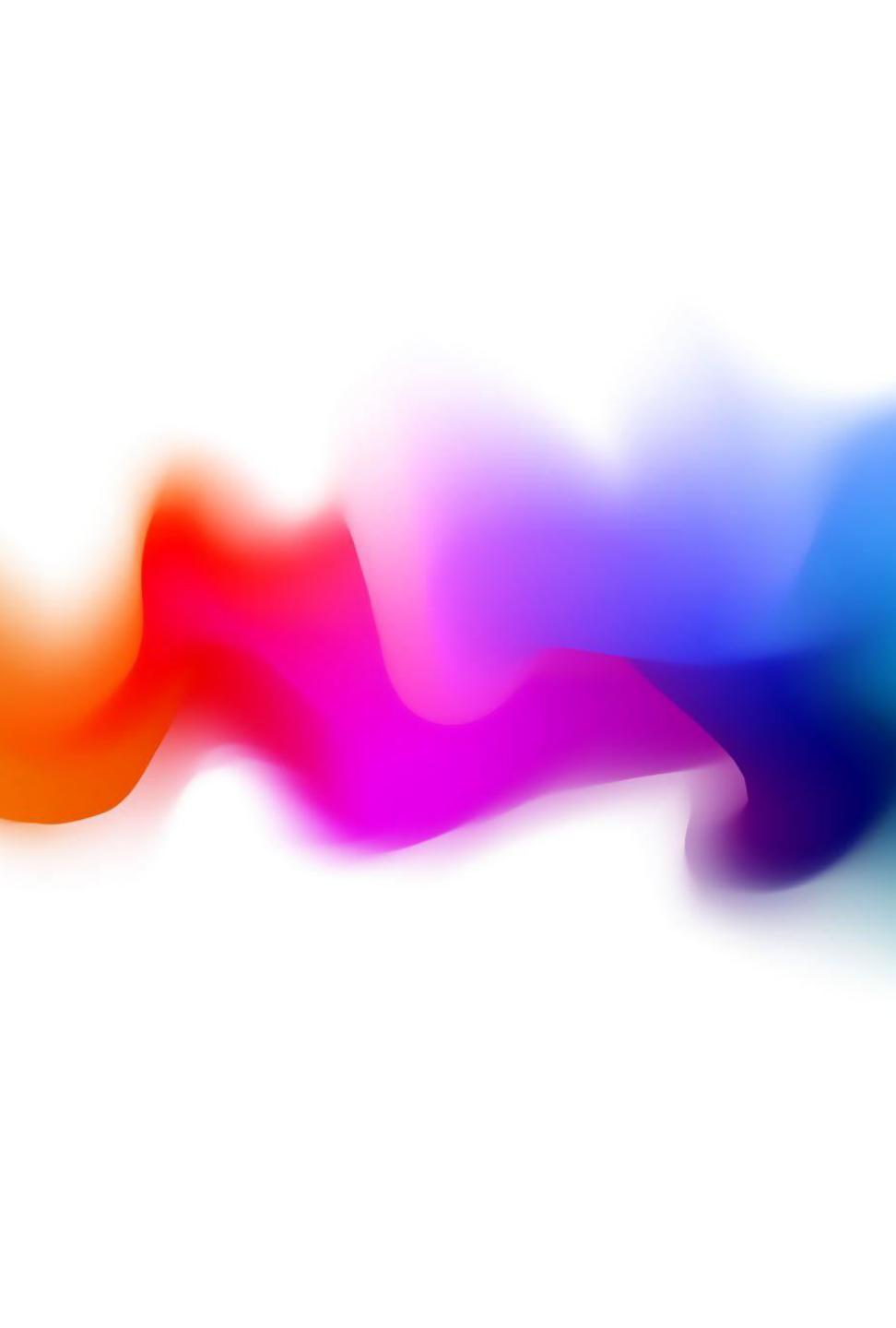
SOLUÇÃO

```
• int ledRed = 2;           //definir ledRed = 2
int ledYellow = 3;          //definir ledYellow = 3
int ledGreen = 4;           // definir ledGreen = 4

void setup() {
    pinMode(ledRed, OUTPUT); //definir a instrução led
    vermelho como saída
    pinMode(ledYellow, OUTPUT); // definir a instrução led
    amarelo como saída
    pinMode(ledZelena, OUTPUT); // definir a instrução led
    verde como saída

    digitalWrite(ledRed, LOW); //desligar o led vermelho
    - estado inicial
    digitalWrite(ledYellow, LOW); // desligar o led amarelo
    - estado inicial
    digitalWrite(ledGreen, LOW); ///// desligar o led verde
    - estado inicial
}
```

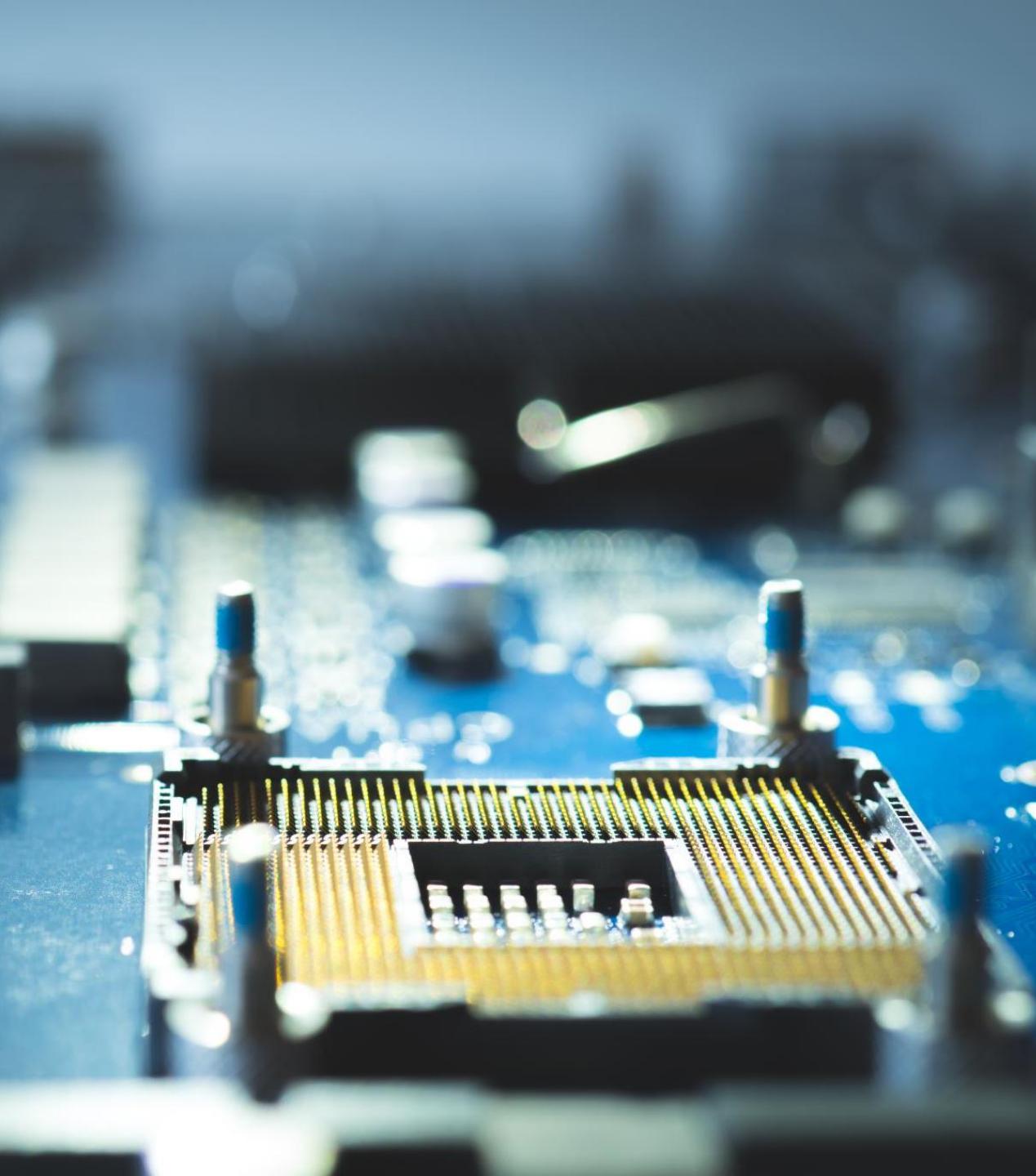
```
void loop() {
    digitalWrite(ledRed, HIGH); // ligar o led
    vermelho delay(3000); //esperar 3 s
    - bright ledRed
    digitalWrite(ledYellow, HIGH); // // ligar o
    led amarelo
    delay(1000); //esperar 1 s - bright R+Y
    digitalWrite(ledRed, LOW); // desligar o led
    vermelho
    digitalWrite(ledYellow, LOW); // desligar o
    led amarelo
    digitalWrite(ledZelena, HIGH); // ligar o led
    verde delay(3000); //wait 3 s -
    bright G
    digitalWrite(ledZelena, LOW); // desligar o led
    verde
    digitalWrite(ledYellow, HIGH); // ligar o led
    amarelo
    delay(1000); //esperar 1 s - bright
    Y
    digitalWrite(ledYellow, LOW); // desligar o led
    amarelo
```



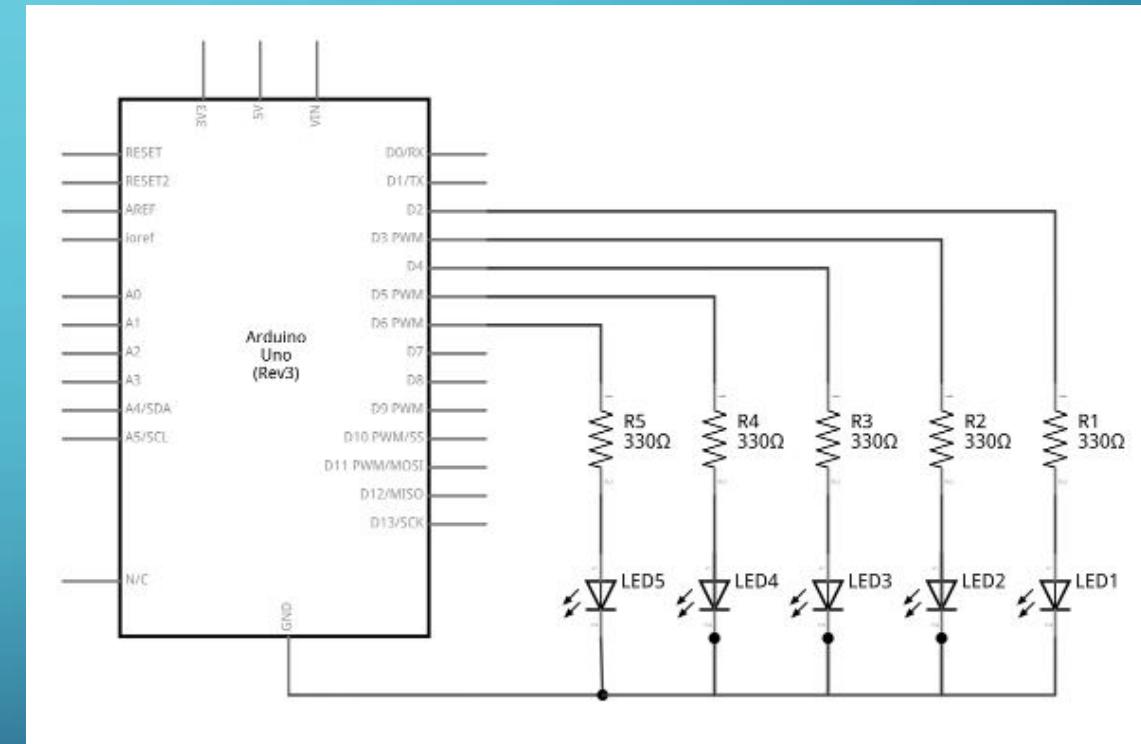
SEMÁFOROS

SEGUNDA TAREFA

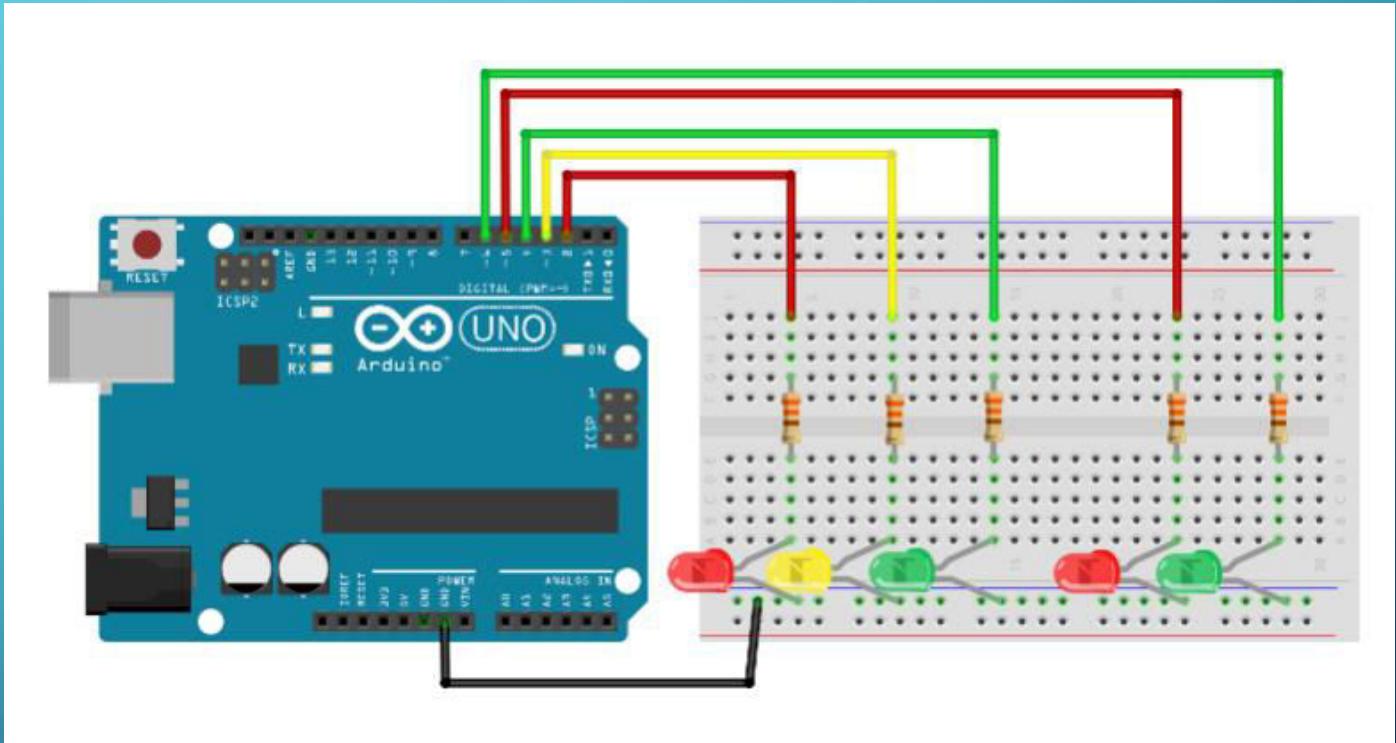
Coneectar dois LEDs adicionais ao microcontrolador - um vermelho e um verde que representem um semáforo de pedestres. O semáforo para carros funciona da mesma forma que na tarefa anterior. A luz verde no semáforo de pedestres só acende quando apenas a luz vermelha está acesa no semáforo de carros. Em todos os outros casos, a luz fica vermelha no semáforo de pedestres.



ESQUEMA



TELHA EXPERIMENTAL



SOLUÇÃO 1/2

```
int ledRed = 2;           //definir ledRed = 2
int ledYellow = 3;         //definir ledYellow = 3
int ledGreen = 4;          // definir ledGreen = 4
int ledCrvenaP = 5;        //definir ledRedP = 5
int ledZelenaP = 6;        //definir ledGreenP = 6

void setup() {
    pinMode(ledRed, OUTPUT); // definir a instrução
    led vermelho como saída
    pinMode(ledYellow, OUTPUT); // definir a
    instrução led amarela como saída
    pinMode(ledGreen, OUTPUT); // definir a instrução
    led verde como saída
    pinMode(ledRedP, OUTPUT); // definir a
    instrução ledRedP como saída
    pinMode(ledGreenP, OUTPUT); // definir a
    instrução ledGreenP como saída

    digitalWrite(ledRed, LOW); // desligar o led
    vermelho - estado inicial
    digitalWrite(ledYellow, LOW); // desligar o led
    amarelo - estado inicial
    digitalWrite(ledZelena, LOW); // desligar o led verde
    - estado inicial
    digitalWrite(ledRedP, LOW); // desligar o ledRedP -
    estado inicial
    digitalWrite(ledGreenP, LOW); // desligar o ledGreenP
    - estado inicial
}
```

SOLUÇÃO 2/2

```
void loop() {  
  
    digitalWrite(ledRed, HIGH);           //ligar o led vermelho  
  
    digitalWrite(ledGreenP, HIGH);        // ligar o ledGreenP  
  
    delay(3000);                      //esperar 3 s - bright R+GP  
  
    digitalWrite(ledGreenP, LOW);         // turn off the  
    ledGreenP  
  
    digitalWrite(ledRedP, HIGH);          // ligar o ledRedP  
  
    digitalWrite(ledYellow, HIGH);        // ligar o led amarelo  
  
    delay(1000);                      //esperar 1 s - bright R+Y+RP  
  
    digitalWrite(ledRed, LOW);           // desligar o led vermelho  
  
    digitalWrite(ledYellow, LOW);         // desligar o led amarelo  
  
    digitalWrite(ledGreen, HIGH);         // ligar o led verde  
  
    delay(3000);                      //esperar 3 s - bright G+RP  
  
    digitalWrite(ledGreen, LOW);          // desligar o led verde  
  
    digitalWrite(ledYellow, HIGH);        // ligar o led amarelo  
  
    delay(1000);                      //esperar 1 s - bright Y+RP  
  
    digitalWrite(ledYellow, LOW);        // desligar o led  
    amarelo  
  
    digitalWrite(ledRedP, LOW);          // desligar o ledRedP  
}  

```