

Texto de Estudo - Eletrônica - 1DN

Sistemas Embarcados

Prof. Eduardo Cesar Alves Cruz

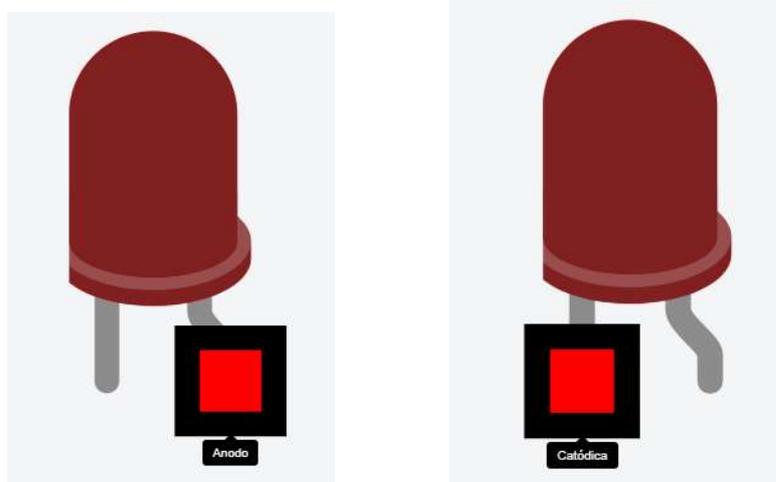
Junho / Julho

ARDUINO: SAÍDAS DIGITAIS

1 – Polarização de um LED

Polarizar um LED significa ligá-lo corretamente para que ele acenda a partir de uma fonte de alimentação, de modo que a corrente que o atravessa seja compatível com as suas especificações. Para isso, utilizamos um resistor conectado em série com o LED.

O LED tem dois terminais: **anodo** e **catodo**. No Tinkercad, o catodo é denominado **catódica**, provavelmente por erro de tradução.



O terminal catodo é aquele que fica próximo da parte achatada da base do LED, como se vê na figura acima à direita.

Para que o LED **acenda**, o anodo tem que estar com potencial positivo em relação ao catodo, caracterizando a **polarização direta**.

Etec Jorge Street

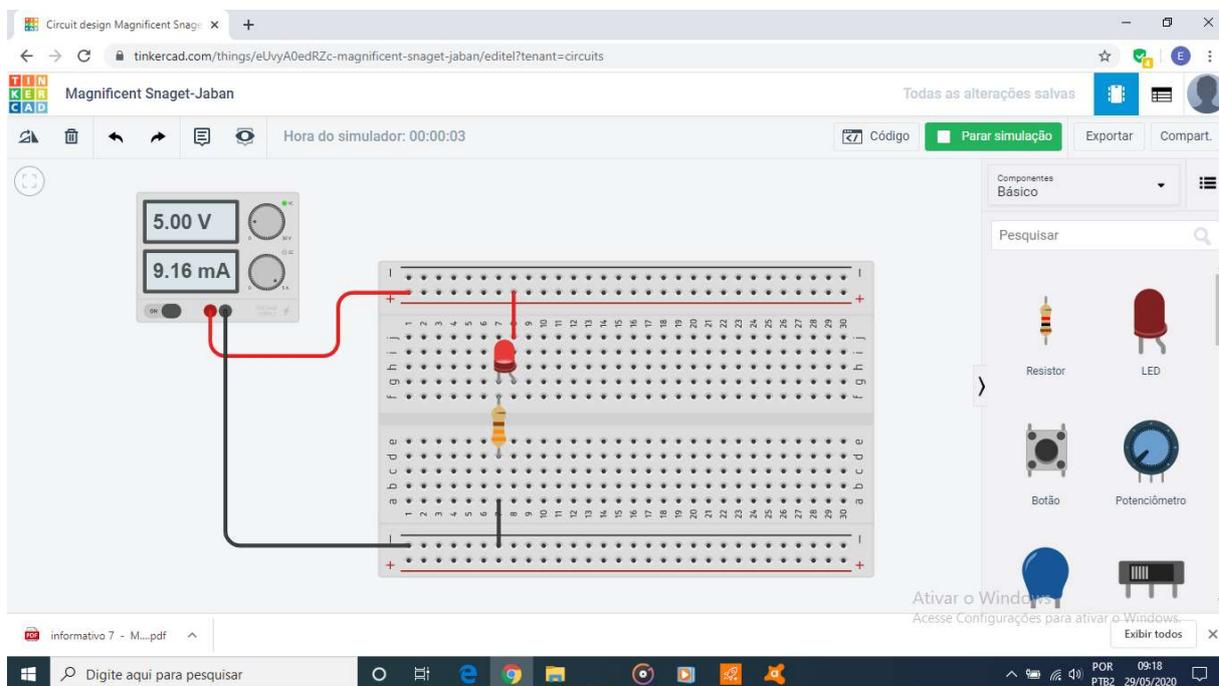
Essa diferença de potencial ou tensão direta no LED vale **aproximadamente 2 V**. Nos manuais de especificações, a tensão direta é denominada V_F (*forward voltage*).

A corrente direta I_F (*forward current*) para que o LED acenda, depende do tipo de LED:

- LED de 3 mm → $I_F \approx 1 \text{ mA}$ com $V_F \approx 2 \text{ V}$
- LED de 5 mm → $I_F \approx 10 \text{ mA}$ com $V_F \approx 2 \text{ V}$

Veja que o LED de 3 mm de diâmetro precisa, para acender, de uma corrente dez vezes menor que o de 5 mm, o que é ótimo para projetos que operam com bateria.

Observe a montagem abaixo em que um LED de 5 mm está com o anodo ligado no polo positivo da fonte de alimentação de 5 V, o catodo está ligado ao resistor de polarização (330Ω) que, por sua vez, está conectado ao terminal negativo da fonte, fechando o circuito:

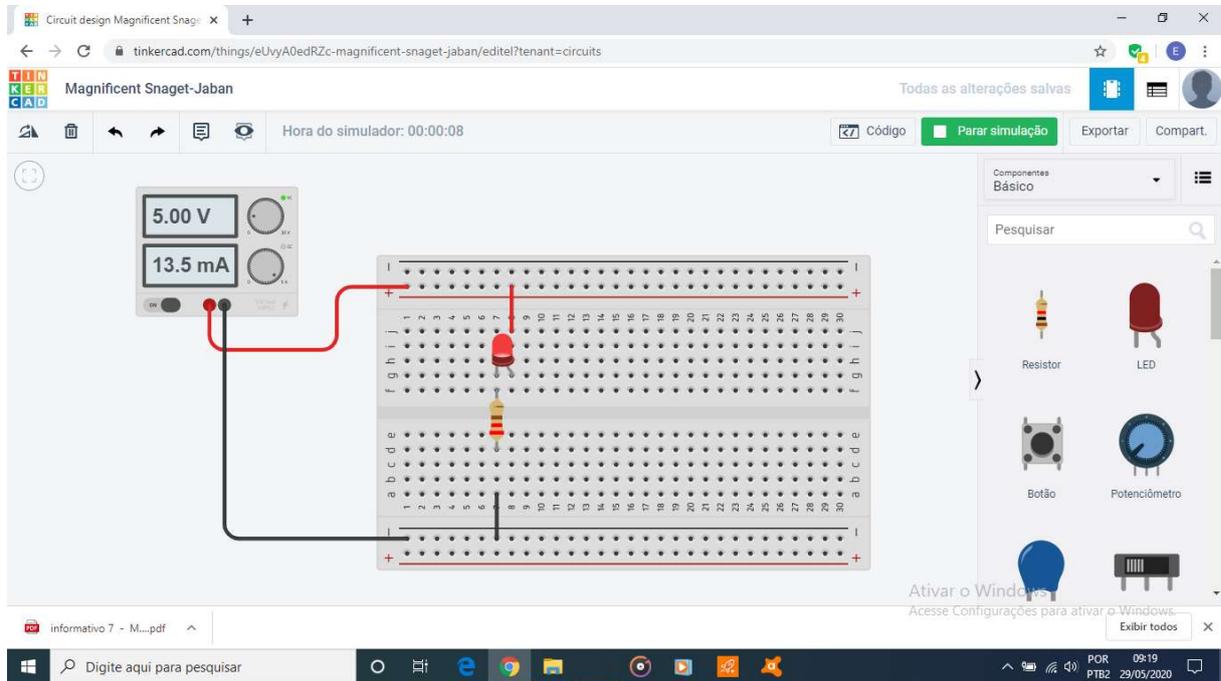


Veja que a corrente que a fonte de alimentação está fornecendo para o circuito vale $I_F = 9,16 \text{ mA}$, que é a corrente de polarização direta do LED e é um pouco maior do valor padrão (10 mA), o que não acarreta nenhum problema.

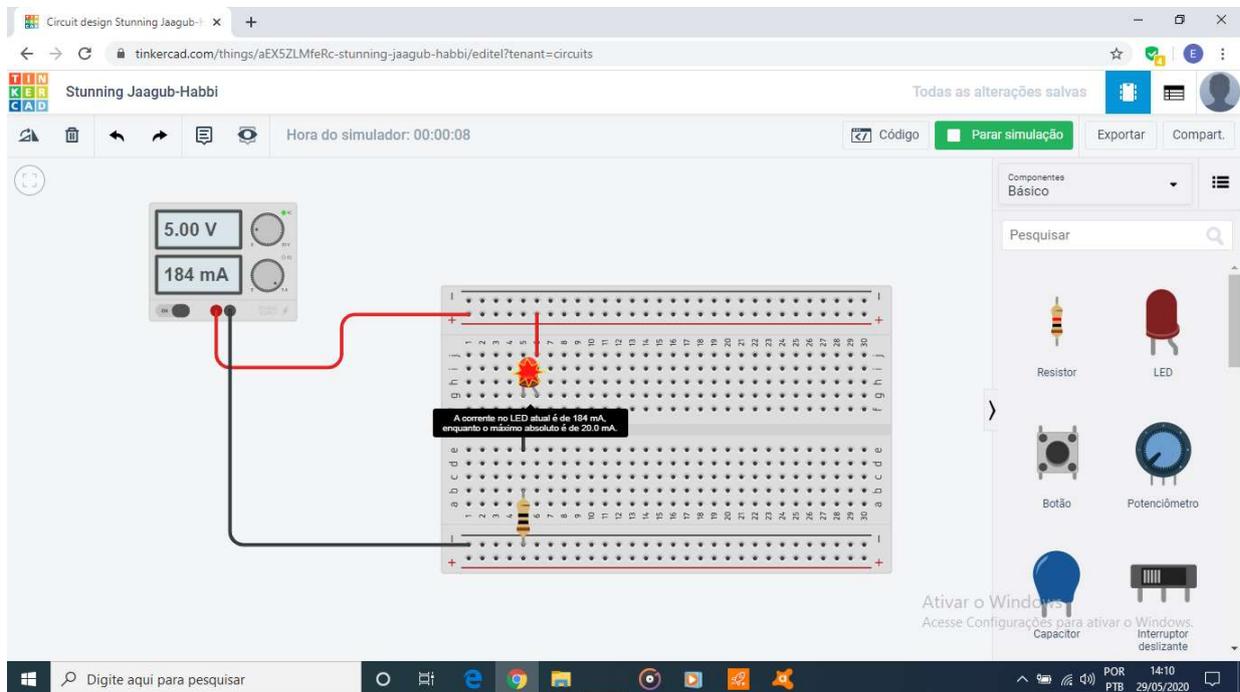
Se o resistor for de valor menor, a corrente será maior e seu brilho um pouco mais forte, como se vê na figura seguinte, em que o resistor de 330Ω foi substituído pelo resistor de 220Ω , fazendo com que a corrente aumente para $13,5 \text{ mA}$.

Esse valor também não acarreta nenhum problema pois, embora esteja um pouco acima do valor padrão (10 mA), ele está bem abaixo do máximo valor especificado, que é acima de 20 mA .

Etec Jorge Street



Agora, veja na figura seguinte o que ocorre quando o valor do resistor é muito baixo. Nesse caso, foi usado um resistor de 10Ω , fazendo com que a corrente seja de 184 mA.

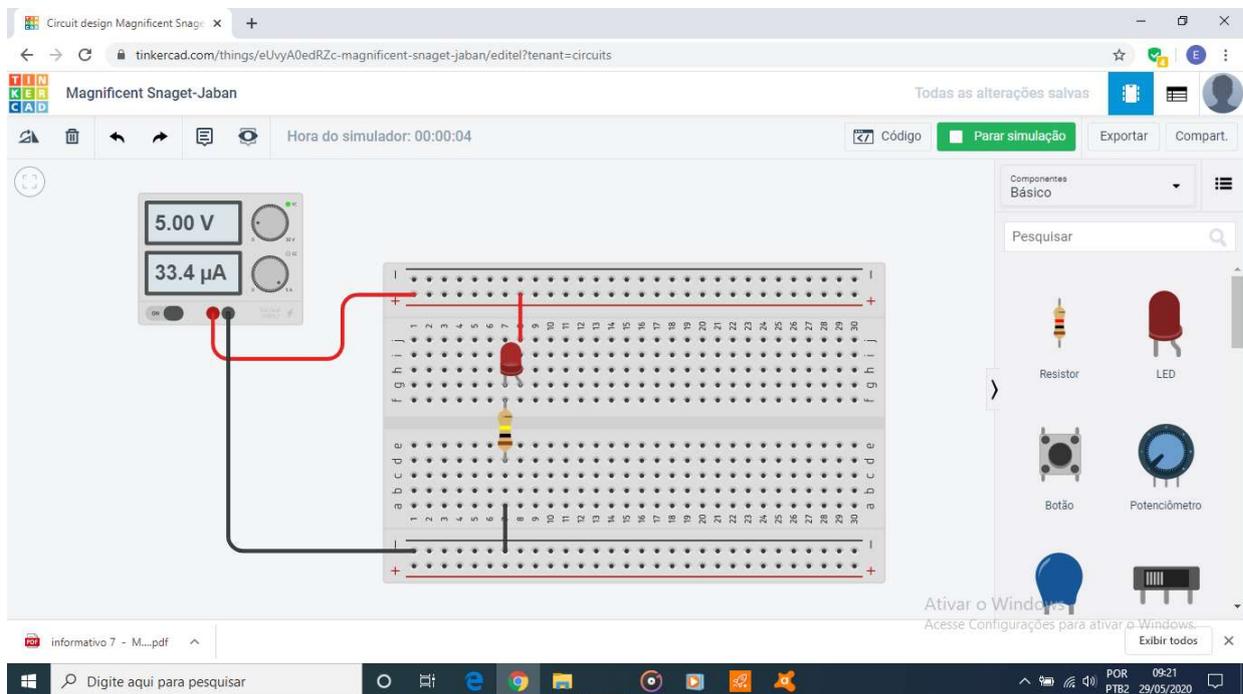


O Tinkercad indica essa sobrecarga de corrente simbolizando uma “explosão” no LED. Colocando o ponteiro do mouse sobre o LED, aparece a mensagem:

A corrente no LED atual é de 184 mA, enquanto o máximo absoluto é de 20,0 mA.

Observação: Na prática, o LED de 5 mm suporta mais que 20 mA, mas não é desejável polarizá-lo para esse valor de operação, pois corresponde a um consumo de energia desnecessário.

Na figura seguinte, está mostrado o que ocorre quando o valor do resistor é muito elevado, como neste caso em que foi usado um resistor de 100 kΩ.



O resultado é uma corrente muito baixa, $I_F = 33,4 \mu\text{A}$, e seu brilho é quase nulo.

Para se calcular o valor do resistor de polarização de um LED em função de seu tipo (3 mm ou 5 mm) e da tensão de alimentação (V_{CC}), deve-se aplicar a fórmula abaixo e adotar um valor comercial de resistor que seja próximo ao calculado, podendo ter um valor nominal superior ou inferior.

Cálculo do resistor R de polarização:

$$R = \frac{V_{CC} - V_F}{I_F}$$

em que:

V_{CC} = tensão de alimentação	→	$V_{CC} = 5 \text{ V}$ para o Arduino
V_F = tensão direta no LED de 3 e 5 mm	→	$V_F \approx 2 \text{ V}$ para ambos
I_F = corrente direta no LED	→	$I_F \approx 10 \text{ mA}$ para LED de 5 mm

→ $I_F \approx 1 \text{ mA}$ para LED de 3 mm

Etec Jorge Street

Exemplos:

- ◆ Resistor para um LED de 5 mm ligado ao pino de saída digital de um Arduino:

Nesse caso: $V_{cc} = 5\text{ V}$ - $V_F = 2\text{ V}$ - $I_F = 10\text{ mA}$

$$R = \frac{V_{cc} - V_F}{I_F} = \frac{5 - 2}{10 \cdot 10^{-3}} \rightarrow R = 300\Omega$$

Neste caso, pode-se usar um resistor comercial de 220 Ω , 270 Ω , 330 Ω ou 470 Ω .

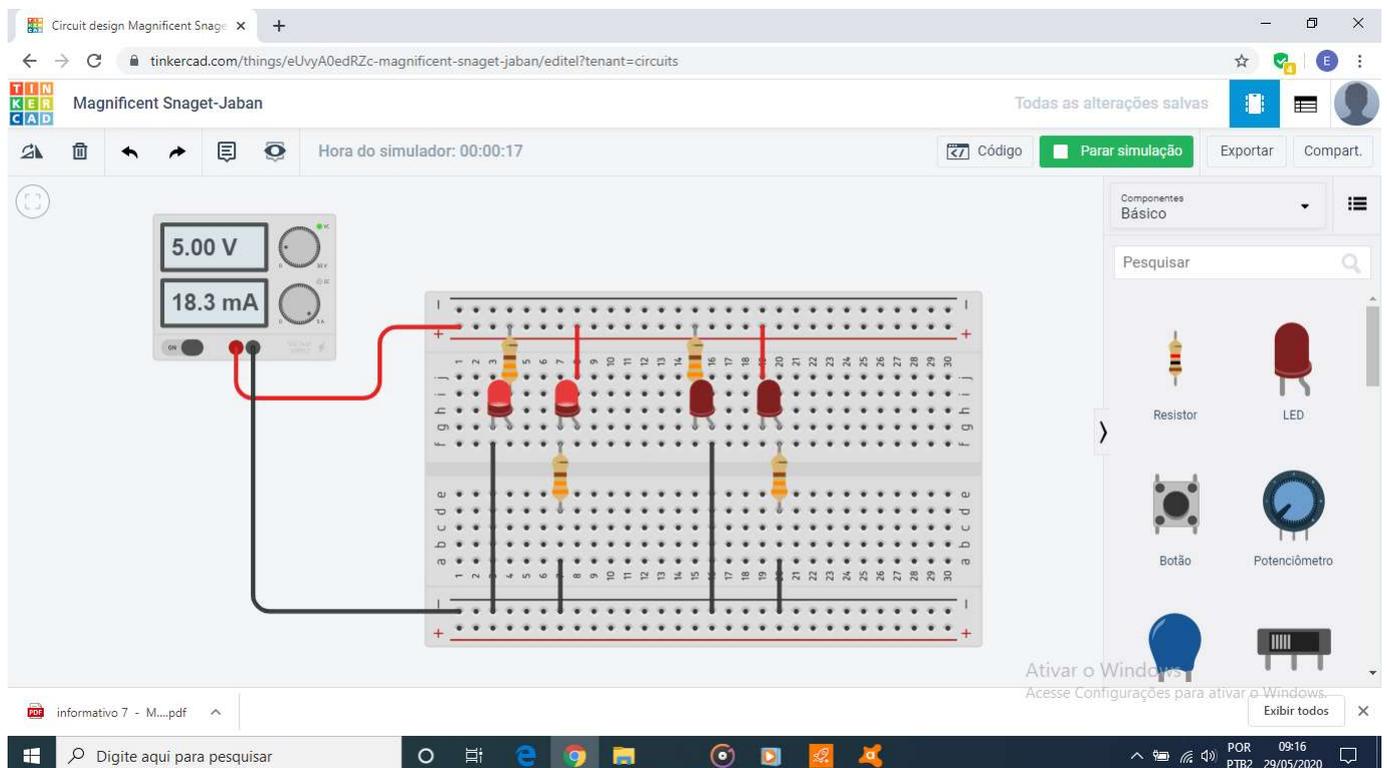
- ◆ Resistor para um LED de 3 mm ligado ao pino de saída digital de um Arduino:

Nesse caso: $V_{cc} = 5\text{ V}$ - $V_F = 2\text{ V}$ - $I_F = 1\text{ mA}$

$$R = \frac{V_{cc} - V_F}{I_F} = \frac{5 - 2}{1 \cdot 10^{-3}} \rightarrow R = 3000\Omega$$

Neste caso, pode-se usar um resistor comercial de 2k2 Ω , 2k7 Ω , 3k3 Ω ou 4k7 Ω .

Agora, vamos abordar as formas corretas e incorretas de se ligar os LEDs. A figura seguinte apresenta dois LEDs acesos (à esquerda) e dois apagados (à direita).



Observação: Lembre-se que os pontos superiores e inferiores da matriz de contatos (*protoboard*) estão conectados horizontalmente, enquanto que os pontos centrais estão conectados verticalmente.

Veja que nos dois primeiros casos, as posições do LED e do resistor estão invertidas, mas a polarização dos LEDs está correta, pois ambos estão com os anodos ligados ao potencial positivo, seja por meio de um

Etec Jorge Street

resistor, seja diretamente, enquanto que os catodos estão ligados ao potencial negativo. Por isso, ambos os LEDs acendem.

Nos dois últimos casos, as posições do LED e do resistor estão invertidas e suas polarizações estão incorretas, pois ambos estão com os catodos ligados ao potencial positivo e os anodos, ao potencial negativo. Neste caso, dizemos que os LEDs estão reversamente polarizados, de modo que não acendem.

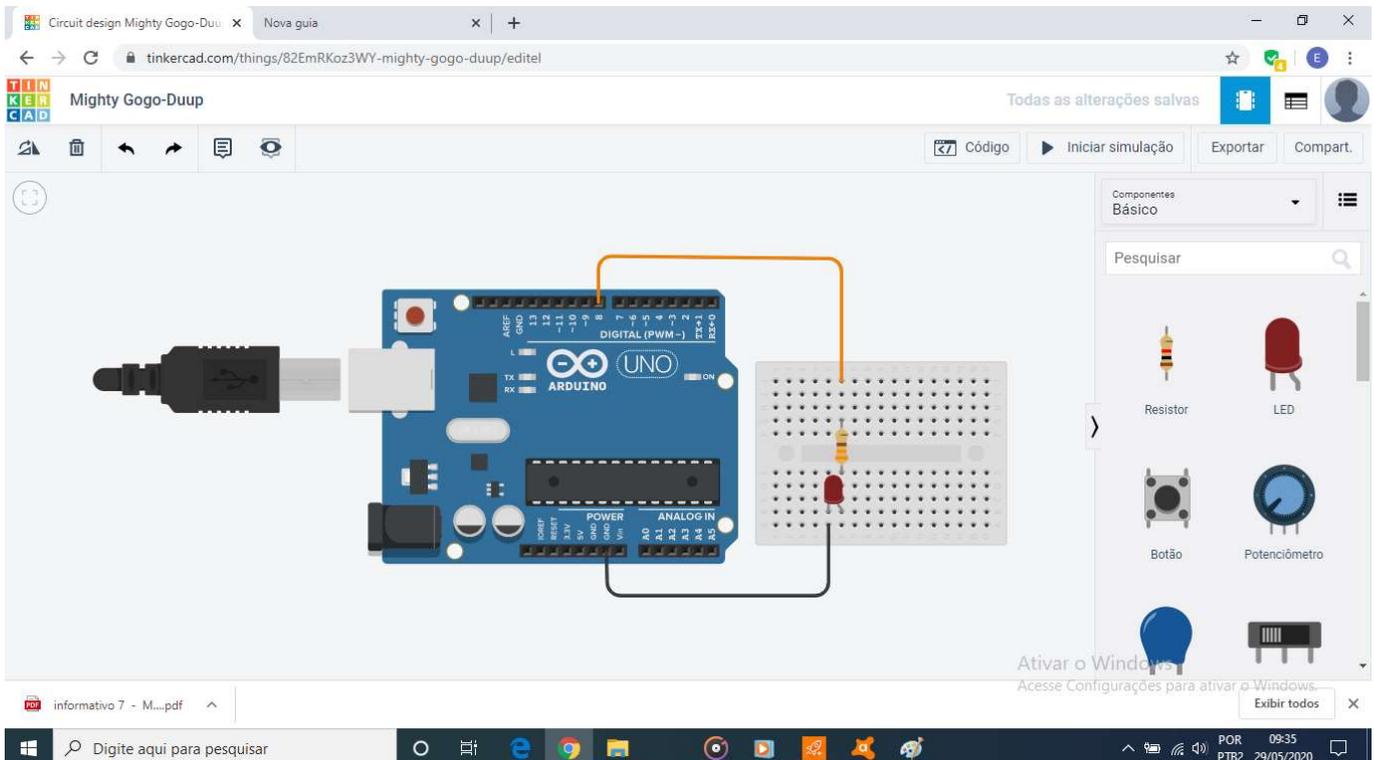
Outra coisa importante que podemos observar na figura anterior é a corrente fornecida pela fonte de alimentação. Como há apenas dois LEDs conduzindo corrente e, portanto, acesos, a fonte envia ao circuito um valor total de corrente para os dois LEDs, ou seja, 9,15 mA para cada um.

Conforme vimos na figura da página 4, o Tinkercad mostrou que o LED de 5 mm ligado a uma fonte de 5 V consumia 9,16 mA, valor equivalente ao observado aqui.

2 – Atividade 02-A – Pisca-pisca com LEDs externos

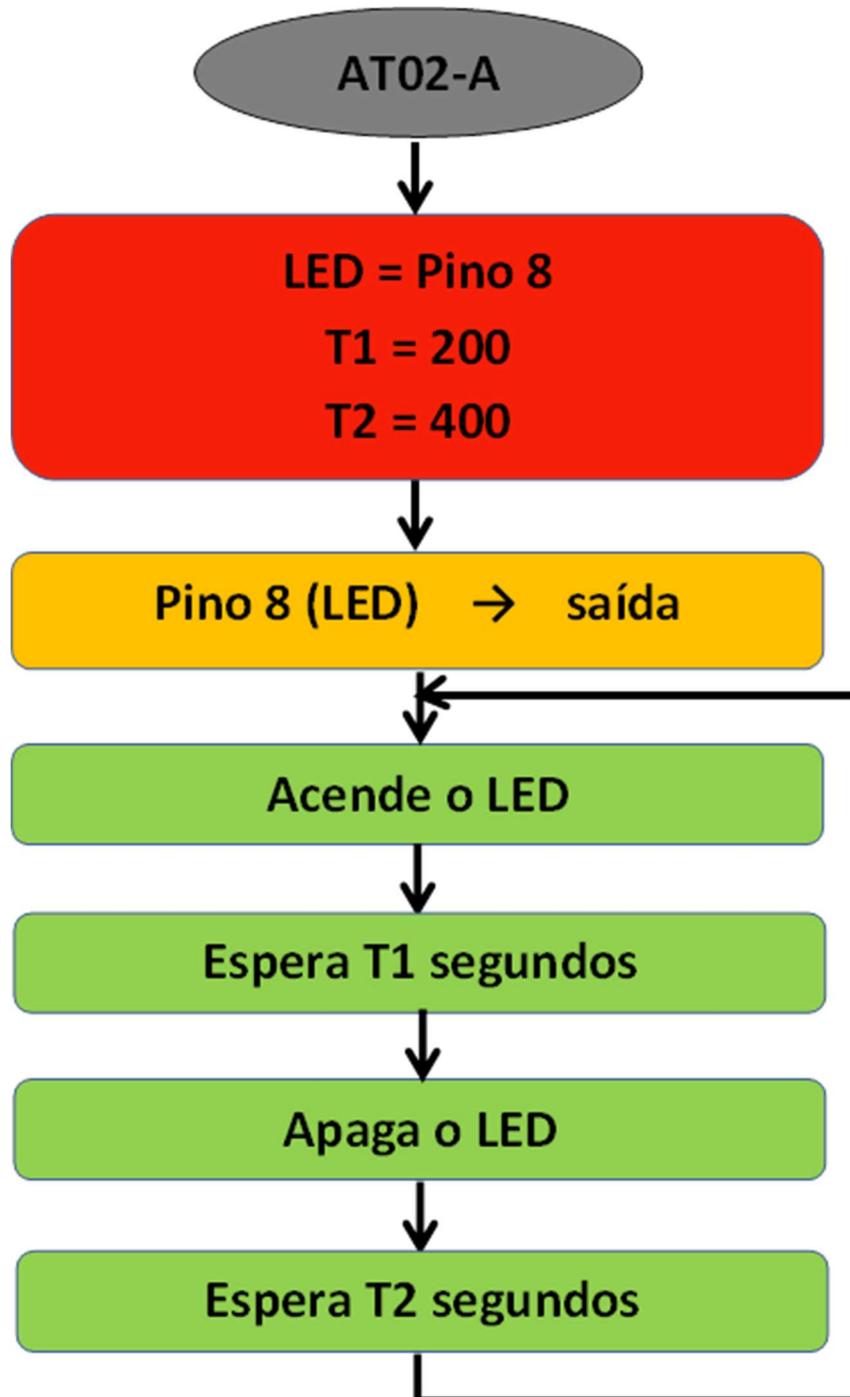
AT02-A

A Atividade 02-A mostra um Arduino acionando um LED externo pelo pino 8, de modo a fazê-lo piscar.



O fluxograma da atividade AT02-A é bem simples, como se vê abaixo:

Etec Jorge Street



Na etapa do **loop**, o LED é aceso por um tempo igual a T1 e em seguida ele é apagado por um tempo T2.

Portanto, o programa, incluindo comentários para orientação, fica como segue:

AT02-A

// AT02-A

// ETAPA DE PRÉ-CONFIGURAÇÃO

```
int LED = 8;           // Cria a variável LED e atribui valor 8.  
int T1 = 200;         // Cria a variável T1 e atribui valor 200.  
int T2 = 400;         // Cria a variável T2 e atribui valor 400.
```

// ETAPA DE CONFIGURAÇÃO

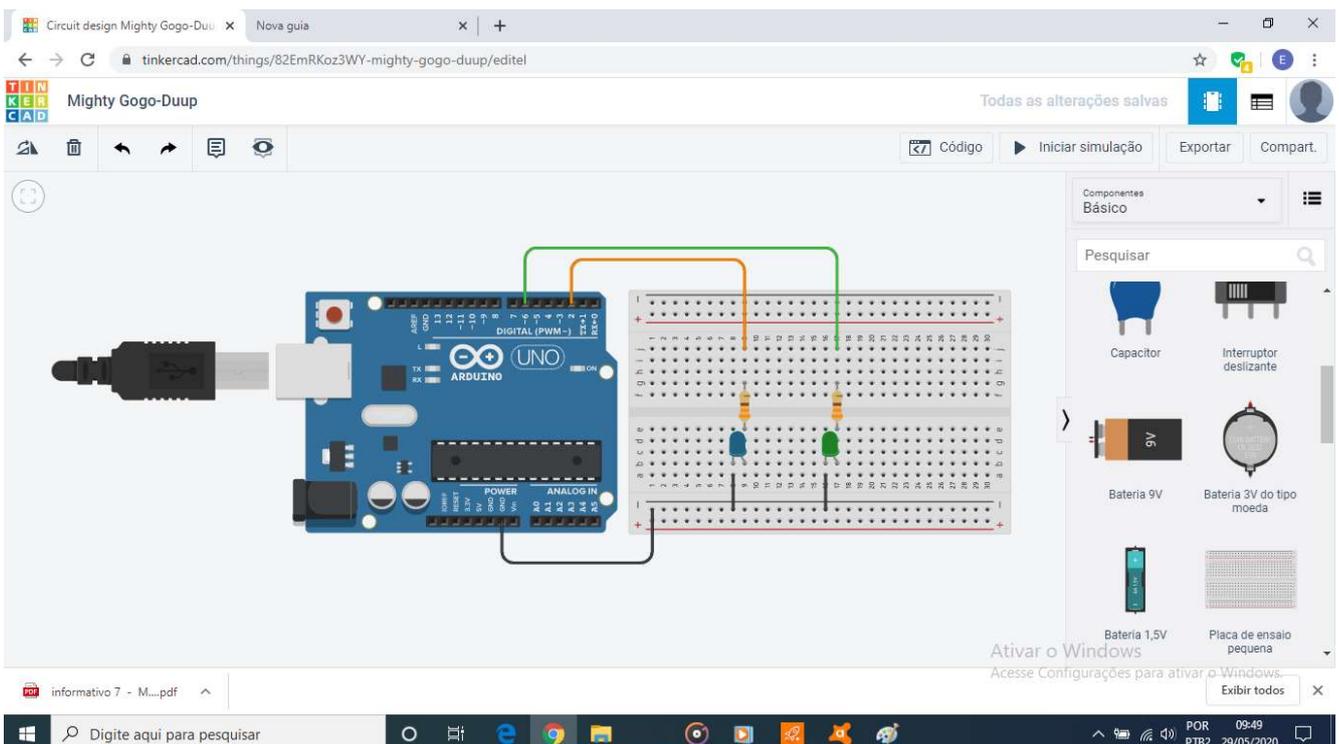
```
void setup ()  
{  
  pinMode (LED, OUTPUT); // Configura o pino 8, denominado LED, como saída.  
}
```

// PROGRAMA PRINCIPAL

```
void loop ()  
{  
  digitalWrite (LED, HIGH); // Acende o LED.  
  delay (T1);               // Espera T1 = 200 ms.  
  digitalWrite (LED, LOW);  // Apaga o LED.  
  delay (T2);               // Espera T2 = 400 ms.  
}
```

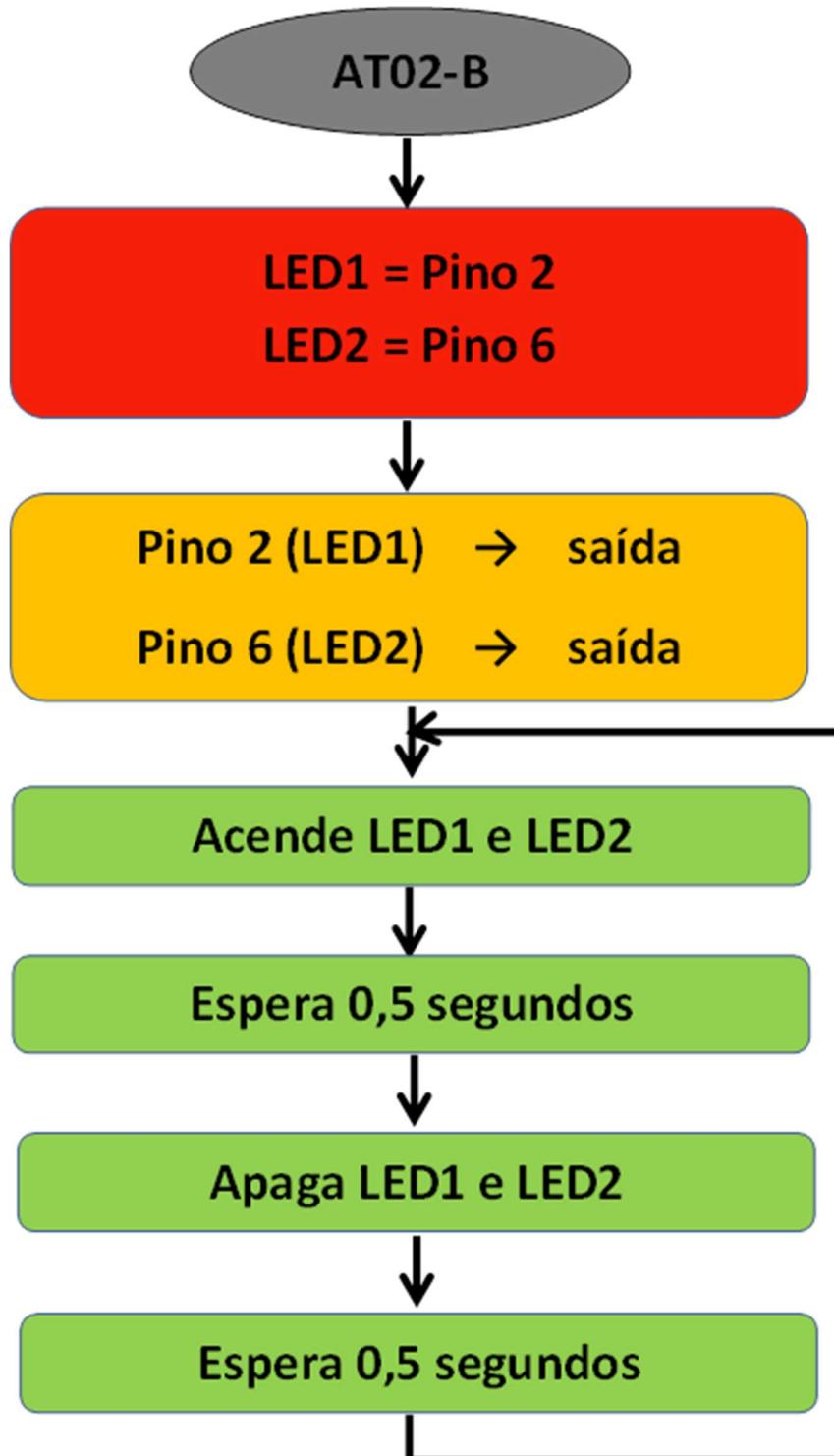
AT02-B – Ações simultâneas

A Atividade 02-B mostra um Arduino controlado dois LEDs: o azul pelo pino 2 e o verde pelo pino 6.



Etec Jorge Street

O objetivo é fazer os dois LEDs piscarem simultaneamente de 0,5 em 0,5 segundos, conforme mostra o fluxograma seguinte:



Portanto, o programa, incluindo comentários para orientação, fica como segue:

Etec Jorge Street

AT02-B

```
// AT02-B

// ETAPA DE PRÉ-CONFIGURAÇÃO

int LED1 = 2;           // Cria a variável LED1 e atribui valor 2.
int LED2 = 6;           // Cria a variável LED2 e atribui valor 6.

// ETAPA DE CONFIGURAÇÃO

void setup ()
{
  pinMode (LED1, OUTPUT); // Configura o pino 2, denominado LED1, como saída.
  pinMode (LED2, OUTPUT); // Configura o pino 6, denominado LED2, como saída.
}

// PROGRAMA PRINCIPAL

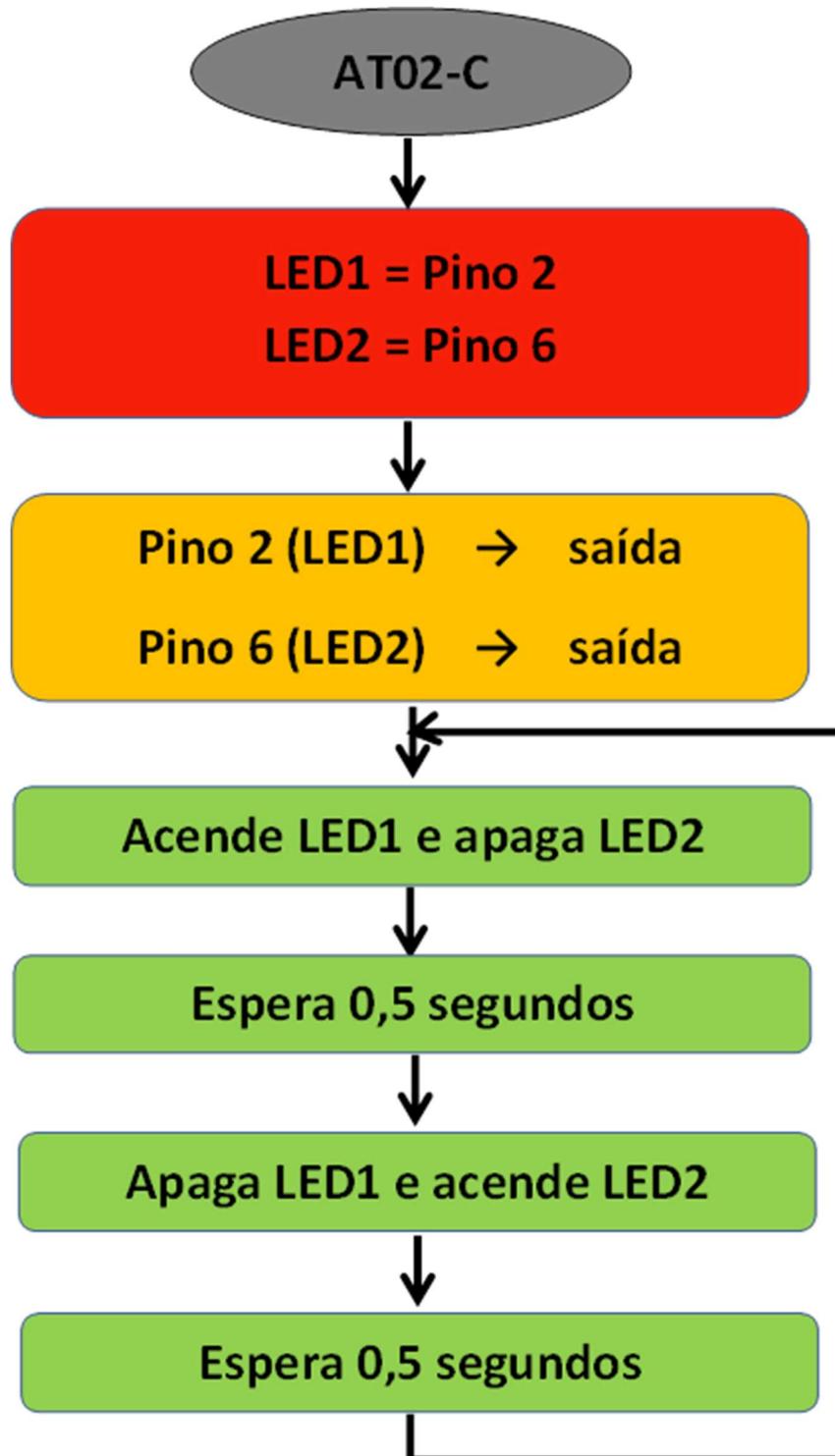
void loop ()
{
  digitalWrite (LED1, HIGH); // Acende o LED1.
  digitalWrite (LED2, HIGH); // Acende o LED2.
  delay (500);               // Espera 500 ms.
  digitalWrite (LED1, LOW);  // Apaga o LED1.
  digitalWrite (LED2, LOW);  // Apaga o LED2.
  delay (500);               // Espera 500 ms.
}
```

AT02-C – Ações alternadas

Na Atividade 02-C será usado o mesmo circuito anterior, só que, neste caso, o objetivo é fazer os LEDs acenderem alternadamente, em intervalos de 0,5 em 0,5 segundos.

Observe pelo fluxograma que a mudança é bem sutil, mas suficiente para que o comportamento seja diferente da atividade anterior.

Etec Jorge Street



Portanto, o programa, incluindo comentários para orientação, fica como segue:

AT02-C

// AT02-C

// ETAPA DE PRÉ-CONFIGURAÇÃO

```
int LED1 = 2; // Cria a variável LED1 e atribui valor 2.  
int LED2 = 6; // Cria a variável LED2 e atribui valor 6.
```

// ETAPA DE CONFIGURAÇÃO

```
void setup ()  
{  
  pinMode (LED1, OUTPUT); // Configura o pino 2, denominado LED1, como saída.  
  pinMode (LED2, OUTPUT); // Configura o pino 6, denominado LED2, como saída.  
}
```

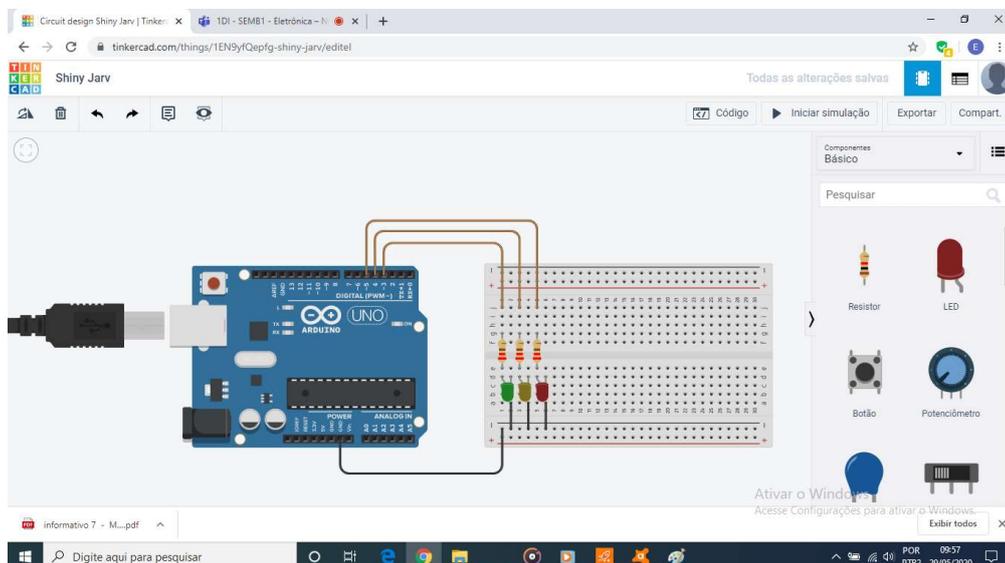
// PROGRAMA PRINCIPAL

```
void loop ()  
{  
  digitalWrite (LED1, HIGH); // Acende o LED1.  
  digitalWrite (LED2, LOW); // Apaga o LED2.  
  delay (500); // Espera 500 ms.  
  digitalWrite (LED1, LOW); // Apaga o LED1.  
  digitalWrite (LED2, HIGH); // Acende o LED2.  
  delay (500); // Espera 500 ms.  
}
```

3 – Atividade 03 – Semáforo

AT03-A – Semáforo de uma via

A Atividade 03-A mostra um Arduino controlando um semáforo de uma via.

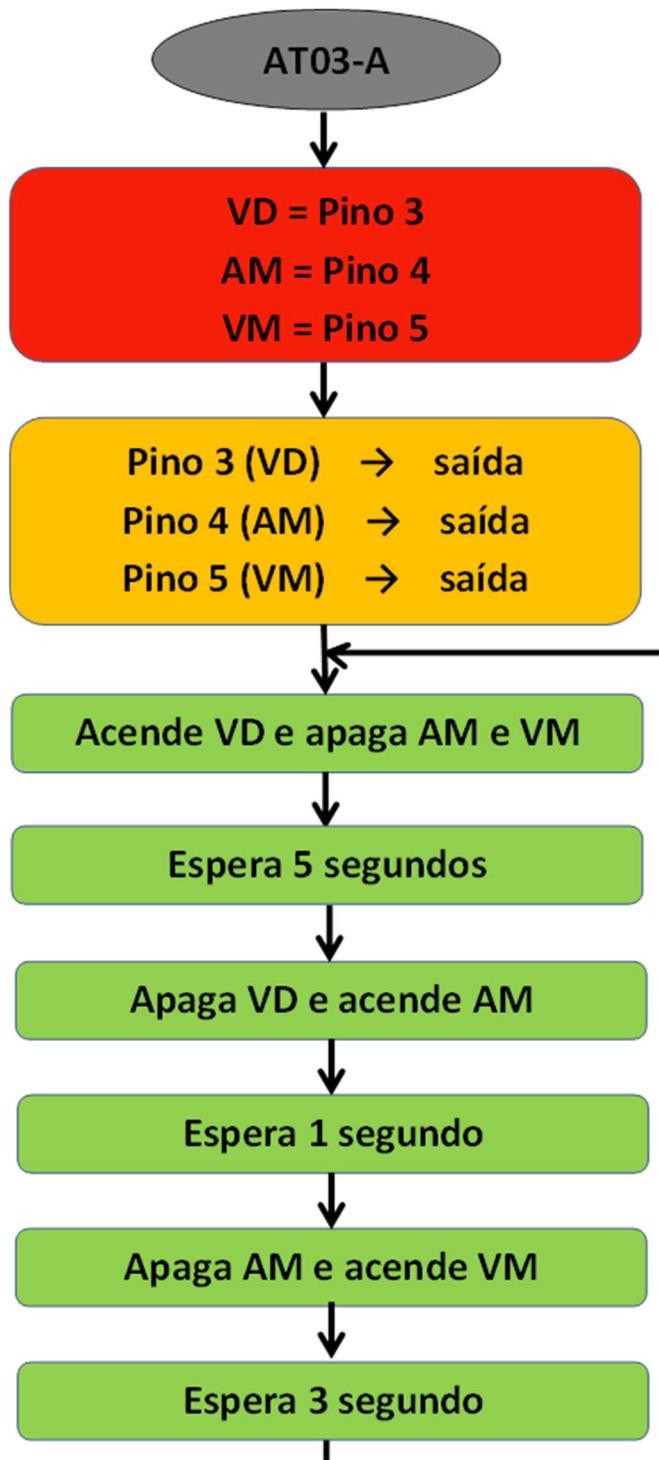


Etec Jorge Street

Os tempos de cada fase são os seguintes:

- Verde (VD) = 5 segundos;
- Amarelo (AM) = 1 segundo;
- Vermelho (VM) = 3 segundos.

O fluxograma da atividade AT03-A está mostrado abaixo:



Etec Jorge Street

O programa, incluindo comentários para orientação, fica como segue:

```
AT03-A
// AT03-A

// ETAPA DE PRÉ-CONFIGURAÇÃO

int VD = 3;           // Cria a variável VD e atribui valor 3.
int AM = 4;           // Cria a variável AM e atribui valor 4.
int VM = 5;           // Cria a variável VM e atribui valor 5.

// ETAPA DE CONFIGURAÇÃO

void setup ()
{
  pinMode (VD, OUTPUT); // Configura o pino 3, denominado VD, como saída.
  pinMode (AM, OUTPUT); // Configura o pino 4, denominado AM, como saída.
  pinMode (VM, OUTPUT); // Configura o pino 5, denominado VM, como saída.
}

// PROGRAMA PRINCIPAL

void loop ()
{
  digitalWrite (VD, HIGH); // Acende o verde.
  digitalWrite (AM, LOW);  // Apaga o amarelo.
  digitalWrite (VM, LOW);  // Apaga o vermelho.
  delay (5000);            // Espera 5 segundos.

  digitalWrite (VD, LOW);  // Apaga o verde.
  digitalWrite (AM, HIGH); // Acende o amarelo (o vermelho continua apagado).
  delay (1000);           // Espera 1 segundo.

  digitalWrite (AM, LOW);  // Apaga o amarelo (o verde continua apagado).
  digitalWrite (VM, HIGH); // Acende o vermelho.
  delay (3000);           // Espera 3 segundos.
}
```

Se cuidem e, se possível, FIQUEM EM CASA!

Abraços



Prof. Eduardo Ino