

DESENVOLVIMENTO DE UM SEMÁFORO UTILIZANDO A PLATAFORMA ARDUINO

Erick Fonseca Boaventura

MANUAL DO ALUNO



Para que os circuitos eletrônicos, que são controlados pela plataforma Arduino, possam funcionar, tem-se duas partes essenciais:

- 1 – O circuito eletrônico montado fisicamente e conectado a placa Arduino (*hardware*);
- 2 – A programação do microcontrolador da placa Arduino (*software*).

1.1. O circuito eletrônico

Um circuito eletrônico é formado pela conexão de vários componentes eletrônicos (resistores, capacitores, LED's, entre outros), a uma fonte de alimentação.

Os circuitos eletrônicos estão presentes em diversos equipamentos, que facilitam nossas vidas e nos trazem conforto, como nos celulares, televisores, computadores, carros, brinquedos, sistemas de automação, etc. Portanto, os circuitos eletrônicos são muito importantes para a criação de inovações tecnológicas e para a evolução da sociedade.

Nesta atividade, você aprenderá a montar circuitos eletrônicos básicos, utilizando a placa Arduino, o protoboard, jumpers, botões, resistores e LED's.

1.2. A programação

A placa Arduino possui um microcontrolador que pode ser programado. O microcontrolador é um componente eletrônico que funciona como um 'cérebro' do sistema, pois capta as informações que chegam para ele, processa essas informações de acordo com a sua programação e realiza algo, como resultado do seu processamento.

Logo, a programação do microcontrolador é uma parte muito importante para os circuitos eletrônicos que utilizam a plataforma Arduino, pois é ela que irá definir o que será feito pelo circuito. Por exemplo, quando um botão for apertado ou um sensor for ativado, qual será o LED a acender ou apagar, se será tocado um sinal sonoro ou não, se irá mandar uma mensagem ou irá ligar o ar condicionado da casa, enfim, uma infinidade de possibilidades que podem ser definidas através do circuito eletrônico montado e da programação associada a ele. Além de presente nos circuitos eletrônicos microcontrolados, a programação pode ser utilizada para o desenvolvimento de sites da internet, de programas para celulares e de jogos para vídeo game, por exemplo.

Nesta atividade, você aprenderá sobre a linguagem de programação C e sobre comandos específicos da plataforma Arduino.

1.3. Situação problema

Luiz desde pequeno sempre foi empolgado com carros. Ao completar 18 anos, logo iniciou o curso para tirar a sua carteira de habilitação para carros. Em uma das aulas do curso de legislação, o professor Bruno estava explicando sobre a importância de se respeitar as cores do semáforo.

Bruno explicou que um semáforo para carros possui 3 lâmpadas, sendo uma da cor vermelha, outra amarela e uma verde e que cada cor tem um significado importante:

- Vermelha: parar;
- Amarela: atenção, mostrando a iminência da parada obrigatória;
- Verde: seguir em frente.

Bruno ainda explicou que em um semáforo para carros a lâmpada verde fica ligada por um tempo e assim que esta lâmpada apaga, a lâmpada amarela acende no mesmo instante. Em seguida a lâmpada amarela fica acesa por um tempo e assim que ela apaga, a lâmpada vermelha acende. Ao acender a lâmpada vermelha, ela fica ligada por um tempo e assim que esta lâmpada apaga, a lâmpada verde acende novamente reiniciando o ciclo. Assim, em um semáforo para carros sempre uma lâmpada estará acesa e as outras duas estarão apagadas, para não confundir os motoristas sobre a atitude a ser tomada.

Luiz ficou tão empolgado com a aula, que após o seu término, ficou imaginando como o semáforo funcionava internamente, para que cada lâmpada acendesse e apagasse no momento exato e com tempos diferentes. Então, correu para a casa do seu avô Francisco, que é Técnico em Eletrônica e Engenheiro de Controle e Automação formado pelo IFMG, que lhe explicou nos mínimos detalhes tudo que estava por trás do funcionamento do semáforo para carros.

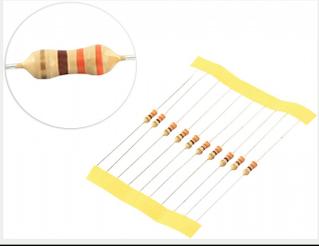
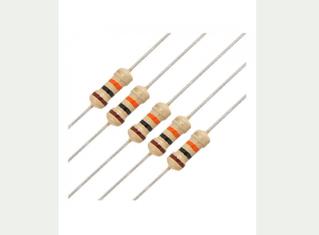
1.4. Questão motriz

Como fazer um semáforo para carros utilizando a plataforma Arduino?

Materiais necessários

O Quadro 1 apresenta a lista de materiais necessária para a atividade.

Item	Nome do item	Descrição	Quant.	Foto	Substituição
1	Notebook	Notebook para a programação do microcontrolador da plataforma Arduino	1		Computador de mesa
2	Arduino + cabo USB	Placa Arduino + cabo USB que vem junto da placa	1		Não há substituto
3	Protoboard	Protoboard/Matriz de contatos	1		Não há substituto
4	LED verde	LED monocromático na cor verde	1		Não há substituto
5	LED amarelo	LED monocromático na cor amarela	1		Não há substituto
6	LED vermelho	LED monocromático na cor vermelha	1		Não há substituto

7	Resistor LED	Resistor de 180Ω à 560Ω - 1/8W	3		Não há substituto
8	Botão	Chave tátil de 2 ou 4 pinos	2		Não há substituto
9	Resistor Botão	Resistor de 1KΩ à 10KΩ - 1/8W	2		Não há substituto
10	Jumpers	Jumper macho-macho	15		Não há substituto

Quadro 1 – Lista de materiais.
Fonte: Próprio autor (2022).

2.1 Placa Arduino

A Figura 1 mostra a placa Arduino, com destaque para:

- Botão de *reset*: possui a função de reiniciar a programação;
- Pinos digitais de 0 a 13: podem ser utilizados para receber um sinal (entrada) ou enviar um sinal (saída) de 5V;
- Entrada para conexão do cabo USB: entrada na qual será ligado o cabo USB entre o notebook/computador e a placa Arduino, para a gravação do código a ser executado pelo microcontrolador;
- Microcontrolador: receberá o código a ser executado;

- Pinos de alimentação: disponibilizam 5V (+) e GND (-), para serem utilizados pelos circuitos que serão conectados a placa Arduino.

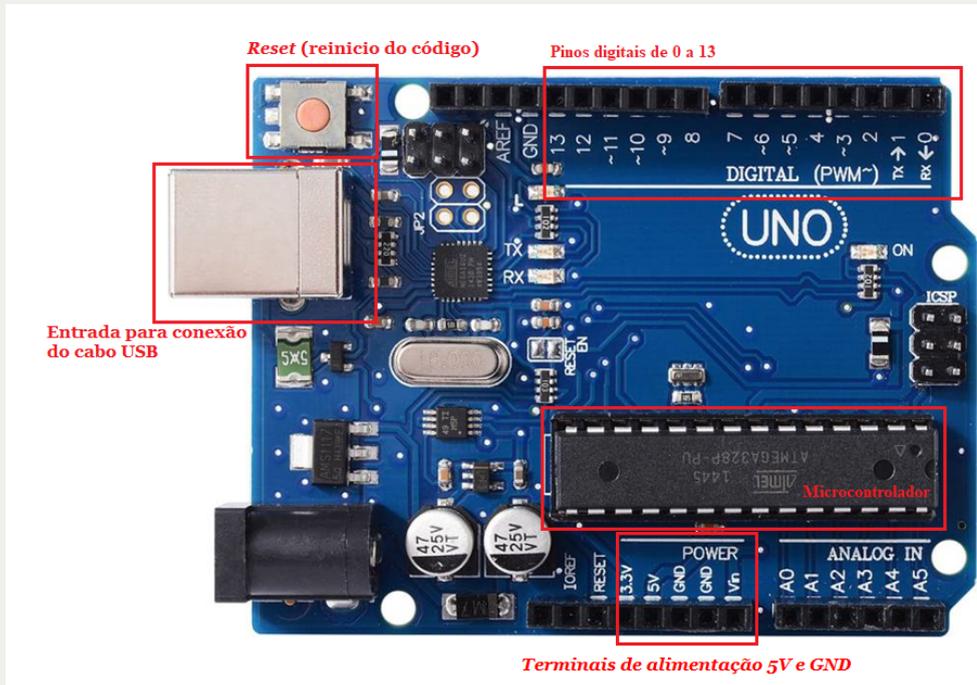


Figura 1 – Placa Arduino.
Fonte: Próprio autor (2022).

2.2 Protoboard/Matriz de contatos

O protoboard ou matriz de contatos, é um equipamento que nos permite realizar as conexões elétricas entre os componentes sem a necessidade de soldá-los, facilitando o desenvolvimento de projetos e testes.

O protoboard possui furos organizados em grupos de 5 (vertical) na parte central, ou vários grupos de 5 furos (horizontal) na parte de cima e de baixo, como mostra a Figura 2. A interligação elétrica existe somente entre os furos pertencentes ao mesmo grupo. Ou seja, na Figura 2, os furos estão interligados em grupos de 5 nas colunas, e interligados em vários grupos de 5 furos nas linhas.

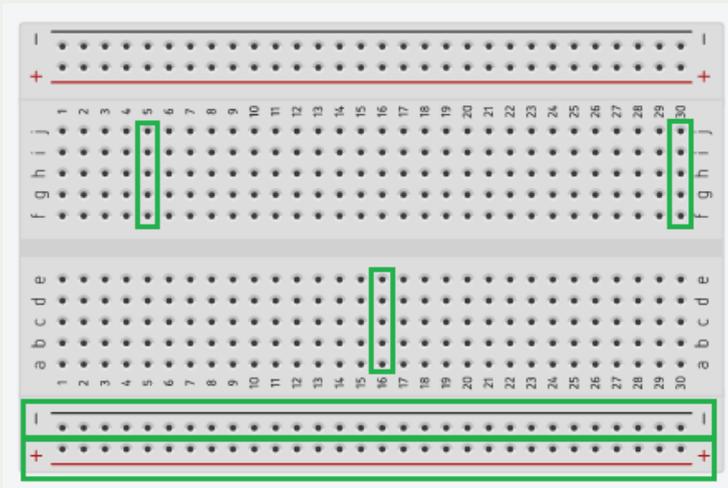


Figura 2 – Ligação interna protoboard/matriz de contatos.
Fonte: Próprio autor (2022).

Atividade proposta

O semáforo para carros será montado com a ajuda do protoboard e demais componentes, sendo as lâmpadas do semáforo representadas pelos LED's: vermelho, amarelo e verde. Sugere-se que o circuito funcione inicialmente da seguinte forma:

- Após um botão (botão 1) ser apertado:

1. O sinal verde (LED verde) acende por 3s e apaga;
2. Instantaneamente o sinal amarelo (LED amarelo) liga, fica aceso por 1s, apagando após este tempo;
3. Assim que o sinal amarelo apaga, o sinal vermelho (LED vermelho) liga instantaneamente, fica aceso por 3s e apaga;
4. Após o sinal vermelho apagar, o processo será reiniciado (item 1), ou seja, o semáforo irá ficar repetindo o ciclo representado pela Figura 3, até que um outro botão (botão 2) seja pressionado até o final do ciclo e apague todos os LED's.

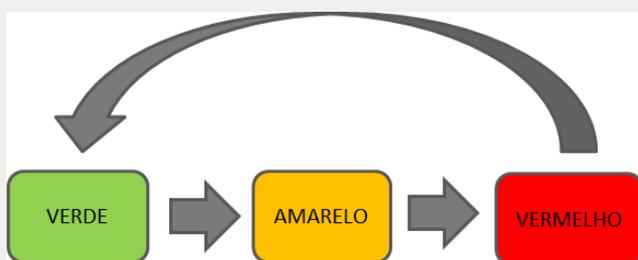


Figura 3 – Ciclo do semáforo de carros.
Fonte: Próprio autor (2022).

3.1 O circuito eletrônico (Etapa 1)

Inicialmente conecte o pino 5V da placa Arduino em uma das trilhas da parte externa do protoboard (das que são conectadas internamente na horizontal), através de um jumper vermelho e também conecte o pino GND da placa Arduino em uma das trilhas da parte externa do protoboard (das que são conectadas internamente na horizontal) diferente da trilha de ligação do 5V, através de um jumper preto. A Figura 4 ilustra a ligação.

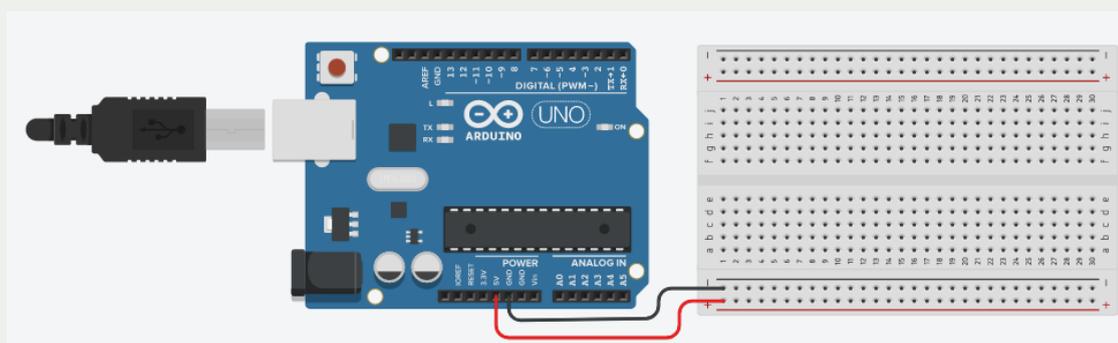


Figura 4 – Ligação dos pinos 5V e GND da placa Arduino ao protoboard.
Fonte: Próprio autor (2022).

Inicie a montagem do circuito pelos LED's (verde, amarelo e vermelho) e seus respectivos resistores. Monte os LED's no protoboard na sequência lógica do semáforo (cores verde, amarela e vermelha, nesta ordem especificamente), para ficar mais fácil verificar o funcionamento posteriormente. A Figura 5 traz um exemplo de como os LED's podem ser montados no protoboard e como podem ser conectados a placa Arduino.

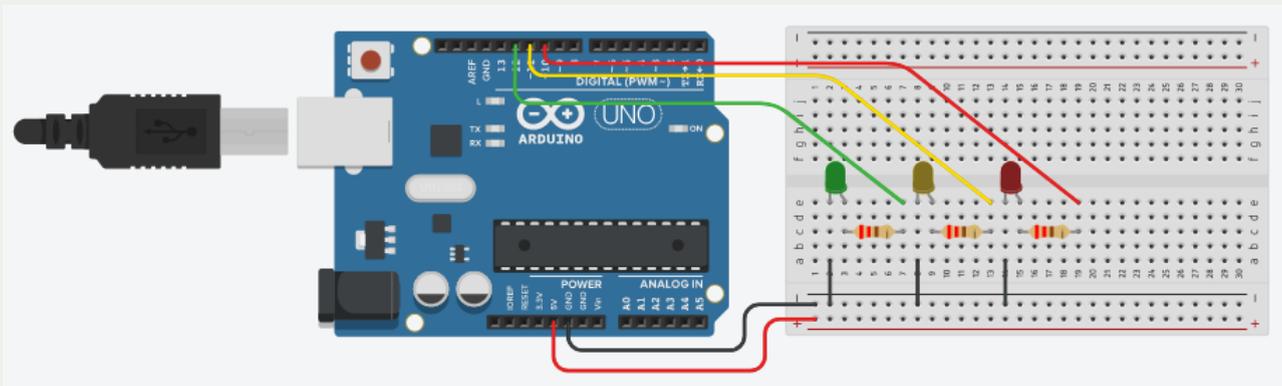


Figura 5 – Montagem dos LED's e seus respectivos resistores no protoboard.
Fonte: Próprio autor (2022).



Atenção: Observar a polaridade dos LED's.

Após a montagem dos LED's, prossiga para a montagem dos botões no protoboard, com os seus respectivos resistores. A Figura 6 demonstra como os botões podem ser montados no protoboard e como podem ser conectados a placa Arduino.

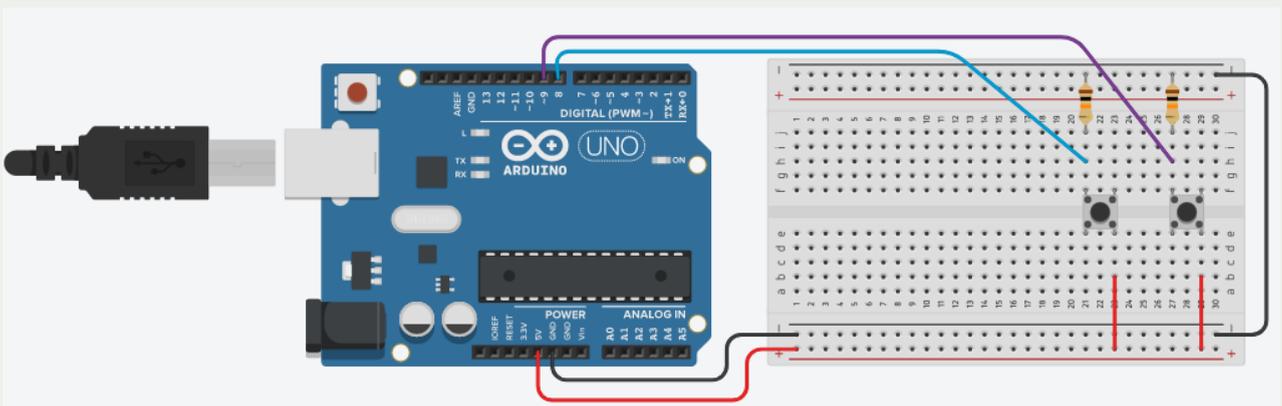


Figura 6 – Montagem dos botões e seus respectivos resistores no protoboard.
Fonte: Próprio autor (2022).

Enfim, a Figura 7 exemplifica a montagem completa do circuito do semáforo no protoboard.

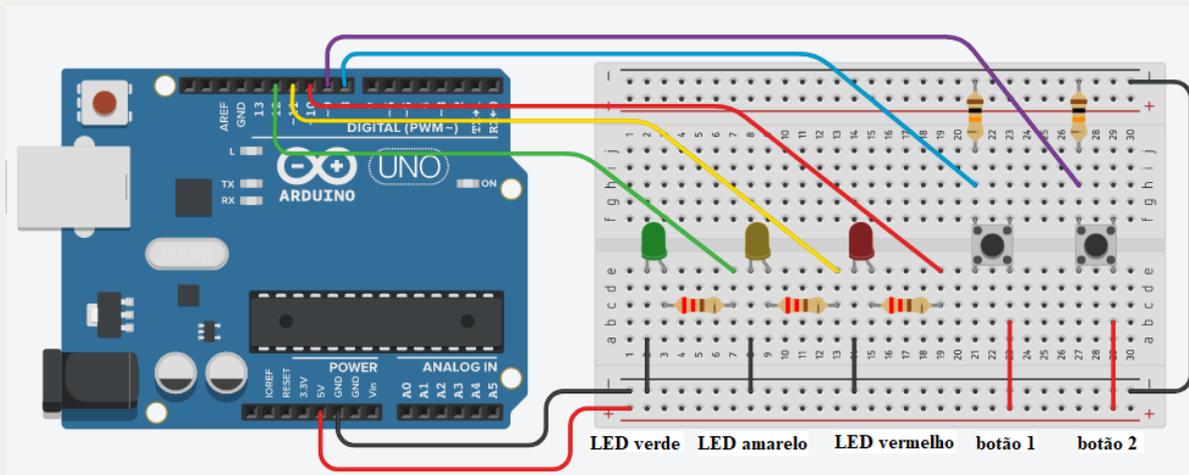


Figura 7 – Montagem completa do circuito do semáforo no protoboard.
Fonte: Próprio autor (2022).

3.2 A programação (Etapa 2)

Com o auxílio do notebook/computador, desenvolva a programação do semáforo para carros, para ser gravada no microcontrolador da placa Arduino. A Figura 8 apresenta a tela inicial do programa Arduino.

```

sketch_oct03a | Arduino 1.8.19
Arquivo Editar Sketch Ferramentas Ajuda
sketch_oct03a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
  
```

1 Arduino Uno em COM3

Figura 8 – Tela inicial do programa Arduino.
Fonte: Próprio autor (2022).

Ao abrir o programa Arduino, comece a programação pela declaração de variáveis (que deve ser feita antes do *void setup*), definindo em qual dos pinos digitais do Arduino ficará ligado cada componente (LED's e botões).



Atenção: É necessário que os pinos definidos na programação coincidam com os pinos em que foram ligados os componentes eletrônicos fisicamente na placa Arduino.

Em seguida, configure dentro das '{ }' (chaves) do *void setup* os botões como dispositivos de entrada e os LED's como dispositivos de saída.

Agora chegou o momento de implementar a parte principal do código do semáforo para carros, de acordo com a proposição feita inicialmente. Para isso, desenvolva o código dentro das '{ }' (chaves) do *void loop*. Lembre-se que não existe uma única solução.

A Figura 9 traz um exemplo de um código funcional para o semáforo proposto na atividade. O código está comentado em todas as suas linhas, facilitando o seu entendimento.

```

semaforo_LIC | Arduino 1.8.19
Arquivo Editar Sketch Ferramentas Ajuda

semaforo_LIC
1 // Declaração de variáveis e definições de pinos utilizados
2 byte botao_1 = 8; // Define o pino do Arduino que o botão 1 será conectado
3 byte botao_2 = 9; // Define o pino do Arduino que o botão 2 será conectado
4 byte LED_vermelho = 10; // Define o pino do Arduino que o LED vermelho será conectado
5 byte LED_amarelo = 11; // Define o pino do Arduino que o LED amarelo será conectado
6 byte LED_verde = 12; // Define o pino do Arduino que o LED verde será conectado
7
8 void setup () // Configuração
9 {
10 pinMode(botao_1, INPUT); // Define o pino como entrada
11 pinMode(botao_2, INPUT); // Define o pino como entrada
12 pinMode(LED_verde, OUTPUT); // Define o pino como saída
13 pinMode(LED_amarelo, OUTPUT); // Define o pino como saída
14 pinMode(LED_vermelho, OUTPUT); // Define o pino como saída
15 }
16
17 void loop () // Execução infinita
18 {
19 if(digitalRead (botao_1)==1) // Se o botão 1 for apertado torna-se verdadeiro e executa o código dentro das {}
20 {
21 while (digitalRead (botao_2)==0) // Enquanto o botão 2 não é pressionado fica executando infinitamente o semáforo
22 {
23 digitalWrite (LED_verde, 1); // Liga o led verde
24 digitalWrite (LED_amarelo, 0); // Desliga o led amarelo
25 digitalWrite (LED_vermelho, 0); // Desliga o led vermelho
26 delay (3000); // Espera 3 segundos
27 digitalWrite (LED_verde, 0); // Desliga o led verde
28 digitalWrite (LED_amarelo, 1); // Liga o led amarelo
29 digitalWrite (LED_vermelho, 0); // Desliga o led vermelho
30 delay (1000); // Espera 1 segundo
31 digitalWrite (LED_verde, 0); // Desliga o led verde
32 digitalWrite (LED_amarelo, 0); // Desliga o led amarelo
33 digitalWrite (LED_vermelho, 1); // Liga o led vermelho
34 delay (3000); // Espera 3 segundos
35 }
36 digitalWrite (LED_verde, 0); // Desliga o led verde
37 digitalWrite (LED_amarelo, 0); // Desliga o led amarelo
38 digitalWrite (LED_vermelho, 0); // Desliga o led vermelho
39 }
}

```

Figura 9 – Código do semáforo para carros no programa Arduino.
Fonte: Próprio autor (2022).

Após finalizado o código, para testar o funcionamento do semáforo é necessário enviar o código para o microcontrolador da placa Arduino. Primeiramente conecte a placa Arduino em alguma entrada USB do notebook/computador, com o cabo USB que acompanha a placa Arduino. Após feito isso, vá no menu superior do programa, em **Ferramentas (1°)**, **Porta (2°)** e selecione a porta serial - **COM (3°)**, na qual aparece o nome do Arduino entre parênteses, conforme demonstrado na Figura 10.

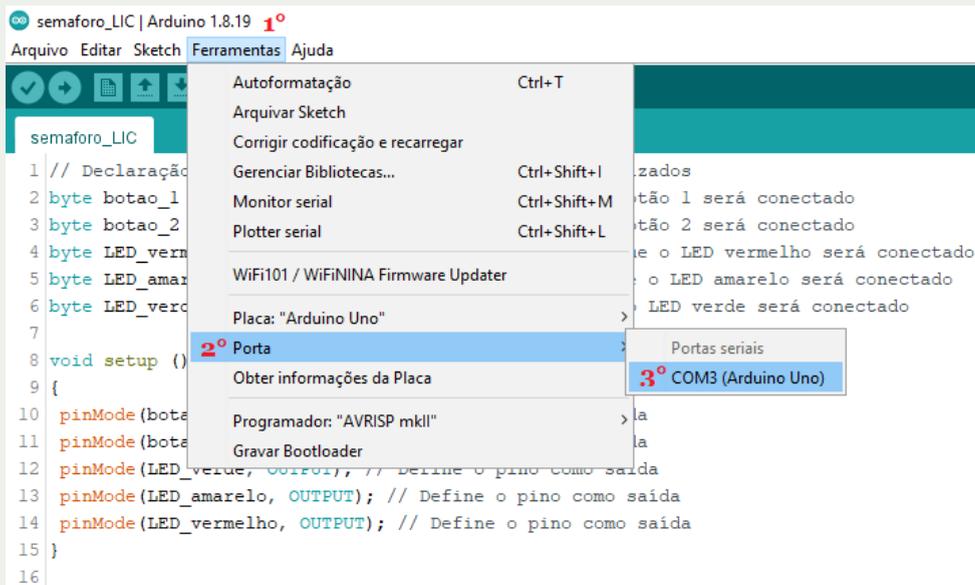


Figura 10 – Procedimento para selecionar a porta serial (USB), na qual a placa Arduino está conectada. Fonte: Próprio autor (2022).

Após selecionada a porta serial corretamente, para enviar o código, basta clicar na **seta que aponta para a direita**, no menu superior. Assim que terminar de gravar o código, o programa mostrará a mensagem **“carregado”**, no menu inferior, do lado esquerdo do programa, conforme exemplifica a Figura 11.

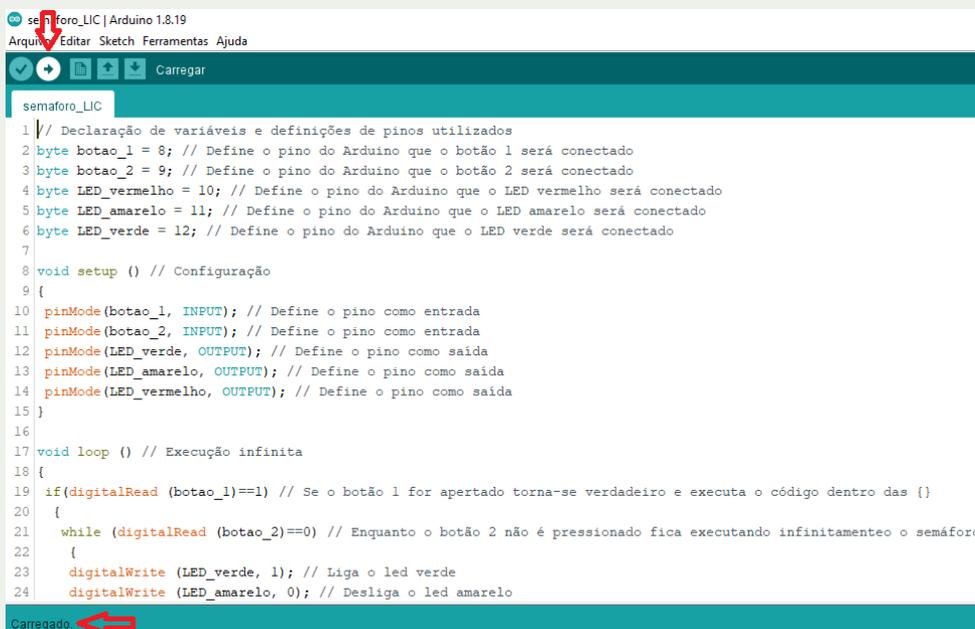


Figura 11 – Procedimento para enviar o código para o microcontrolador da placa Arduino. Fonte: Próprio autor (2022).

Assim, com a placa conectada ao notebook/computador via cabo USB e com o código já carregado, pode-se testar o semáforo, que deve funcionar de acordo com o que foi proposto.

1) Ao apertar o botão 1, o que ocorreu?

2) Preencha o quadro a seguir com o tempo que cada LED acendeu:

Pergunta	Tempo
LED verde	
LED amarelo	
LED vermelho	

3) Ao pressionar o botão 2 até o final do ciclo, o que ocorreu?

4) Qual a importância da definição correta dos pinos de entrada/saída entre o *hardware* e o *software*?

5) Qual a relevância da programação para o funcionamento correto do circuito?

6) Desafio: Aumente o tempo de sinal aberto (LED verde) e diminua o tempo de sinal fechado (LED vermelho).

Cursos do IFMG

O Quadro 2 apresenta os cursos ofertados pelo IFMG (relacionados ao conhecimento trabalhado nesta atividade), o nível de ensino do curso e o campus em que o curso é ofertado.

Campus do IFMG	Cursos relacionados	Nível de ensino
Betim	- Técnico em Automação Industrial - Bacharelado em Engenharia de Controle e Automação	- Ensino médio integrado - Ensino superior
Formiga	- Bacharelado em Engenharia Elétrica	- Ensino superior
Ibirité	- Técnico em Automação Industrial - Técnico em Mecatrônica - Bacharelado em Engenharia de Controle e Automação	- Ensino médio integrado/Concomitante - Ensino médio integrado/Concomitante - Ensino superior
Ipatinga	- Técnico em Automação Industrial - Bacharelado em Engenharia Elétrica	- Ensino médio integrado - Ensino superior
Itabirito	- Técnico em Automação Industrial - Bacharelado em Engenharia Elétrica	- Ensino médio integrado - Ensino superior
Ouro Preto	- Técnico em Automação Industrial	- Ensino médio integrado
Ribeirão das Neves	- Técnico em Eletroeletrônica	- Ensino médio integrado
Sabará	- Técnico em Eletrônica - Bacharelado em Engenharia de Controle e Automação	- Ensino médio integrado - Ensino superior

Quadro 2 – Cursos ofertados pelo IFMG, relacionados ao conhecimento trabalhado nesta atividade.
Fonte: Próprio autor (2022).

