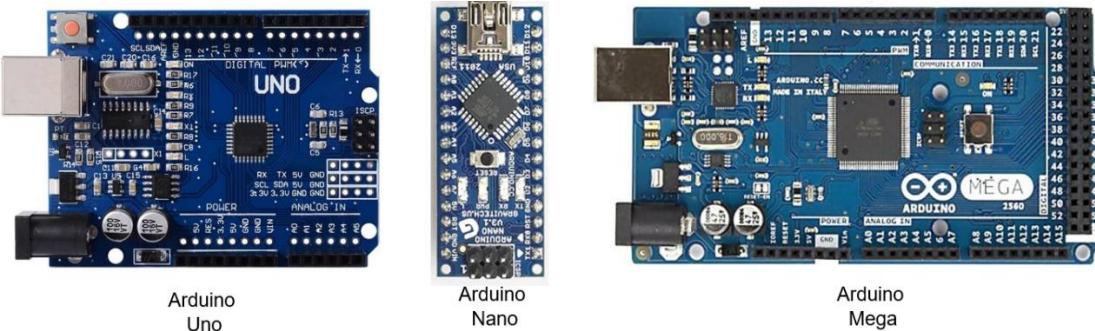


### Objetivo

- Utilizar as portas digitais de entrada e saída do Arduino

### Conteúdo

O Arduino é uma plataforma de prototipagem eletrônica baseada em hardware que integra um microcontrolador e portas de entrada e saída. Seu funcionamento pode ser configurado e implementado por meio de uma linguagem de programação semelhante a C/C++. A Figura 01 abaixo apresenta alguns dos modelos de placas disponíveis no mercado:



**Figura 01** – Modelos de placas do Arduíno

A placa Arduino UNO possui 14 pinos que podem ser configurados como entradas ou saídas digitais, de acordo com as necessidades do seu projeto. Esses pinos são numerados de 0 a 13 (Figura 02).



**Figura 02** – Entradas e saídas digitais

O Arduino disponibiliza bibliotecas que simplificam o acesso aos pinos digitais, permitindo configura-los como entrada ou saída por meio das seguintes funções:

- **void pinMode():** Esta função é usada para configurar um pino como entrada ou saída digital e, geralmente, é empregada dentro da função `setup()`. Sua sintaxe é a seguinte: `pinMode(pino, modo)`.

Parâmetros:

- **pino:** O número correspondente ao pino que deseja configurar, que pode variar de 0 a 13 no caso da placa Arduino UNO.
- **modo:** Indica o modo desejado para a configuração do pino, podendo ser INPUT ou OUTPUT.

- **int digitalRead():** Essa função realiza a leitura do valor em um pino digital. Esse valor pode ser um (HIGH = 5V) ou zero (LOW = 0V). Sua sintaxe é a seguinte: `digitalRead(pino)`.

Parâmetro:

- **pino:** valor correspondente ao pino que será lido

- **void digitalWrite():** Esta função estabelece um nível lógico alto (HIGH, 5V) ou baixo (LOW, 0V) em um pino configurado como saída digital. Sua sintaxe é a seguinte: `digitalWrite(pino, valor)`.

Parâmetros:

- **pino:** número correspondente ao pino
- **valor:** HIGH ou LOW

## Exemplo - Semáforo

O circuito (Figura 03) e código abaixo ilustram o funcionamento de um semáforo com três estados. O simulador [Tinkercad](https://www.tinkercad.com) (<https://www.tinkercad.com>) foi utilizado na implementação do projeto.

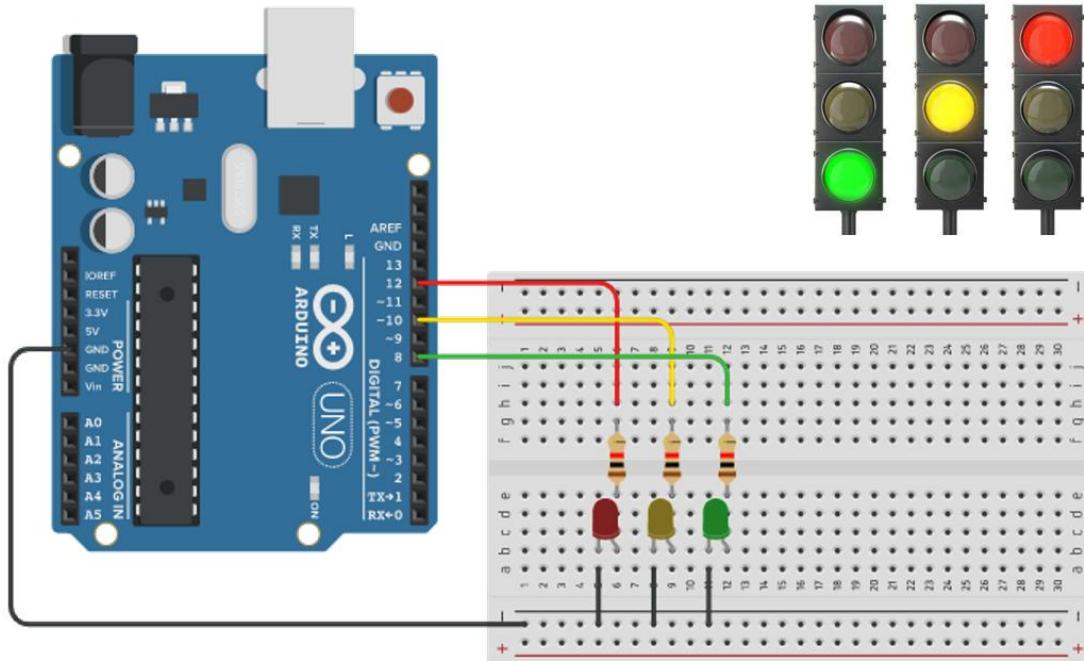


Figura 03 – Circuito da montagem do semáforo

Código:

```
const int ledVermelho = 12;
const int ledAmarelo = 10;
const int ledVerde = 8;

void setup(){
    pinMode(ledVermelho, OUTPUT);
    pinMode(ledAmarelo, OUTPUT);
    pinMode(ledVerde, OUTPUT);
}

void loop(){
    digitalWrite(ledVermelho, LOW);
    digitalWrite(ledVerde, HIGH);
    delay(3000);

    digitalWrite(ledVerde, LOW);
    digitalWrite(ledAmarelo, HIGH);
    delay(1500);

    digitalWrite(ledAmarelo, LOW);
    digitalWrite(ledVermelho, HIGH);
    delay(4000);
}
```

## Exemplo - Botão e piscar LED

O circuito (Figura 04) e código abaixo ilustra o funcionamento do controle de um LED através do botão. O simulador Tinkercad (<https://www.tinkercad.com>) foi utilizado na implementação do projeto.

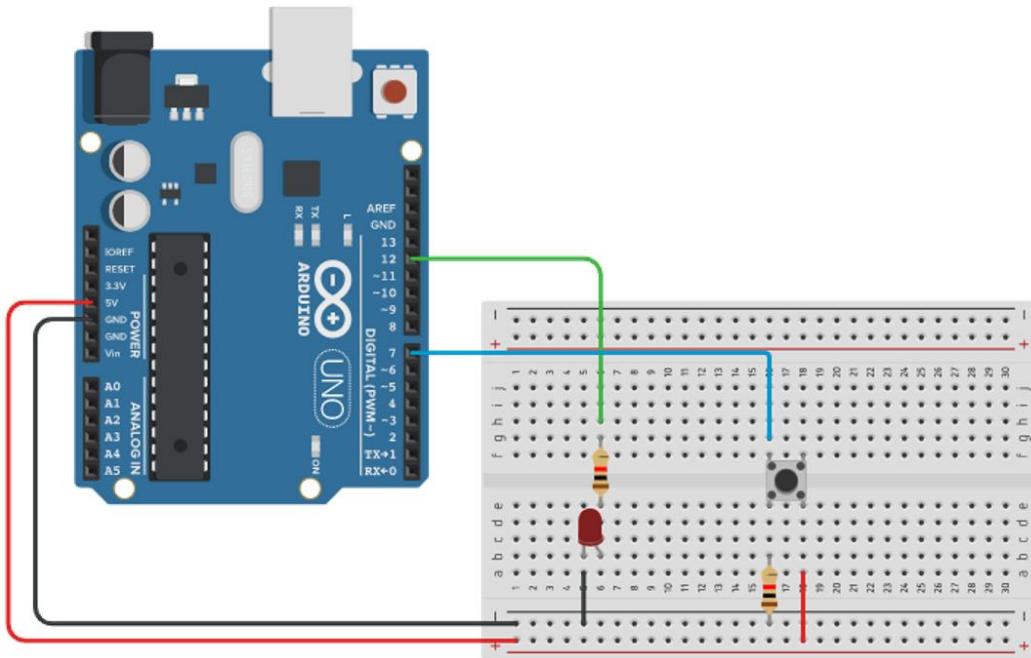


Figura 04 – Circuito da montagem do controle do LED com um botão

Código:

```
const int ledVermelho = 12;
const int botao = 7;
int estadoBotao = 0;

void setup(){
    pinMode(ledVermelho, OUTPUT);
    pinMode(botao, INPUT);
    Serial.begin(9600);
}

void loop(){
    estadoBotao = digitalRead(botao);
    Serial.println(estadoBotao);
    if(estadoBotao == HIGH ){
        piscarLed(ledVermelho);
    }
}

void piscarLed(int pino){
    digitalWrite(pino, HIGH);
    delay(500);
    digitalWrite(pino, LOW);
    delay(500);
    digitalWrite(pino, HIGH);
    delay(500);
    digitalWrite(pino, LOW);
    delay(500);
    digitalWrite(pino, HIGH);
    delay(500);
    digitalWrite(pino, LOW);
    delay(500);
}
```

## Exercícios

- Realize a montagem do circuito (Figura 05) a seguir e desenvolva um código para um sistema de semáforo de trânsito destinado a carros e pedestres

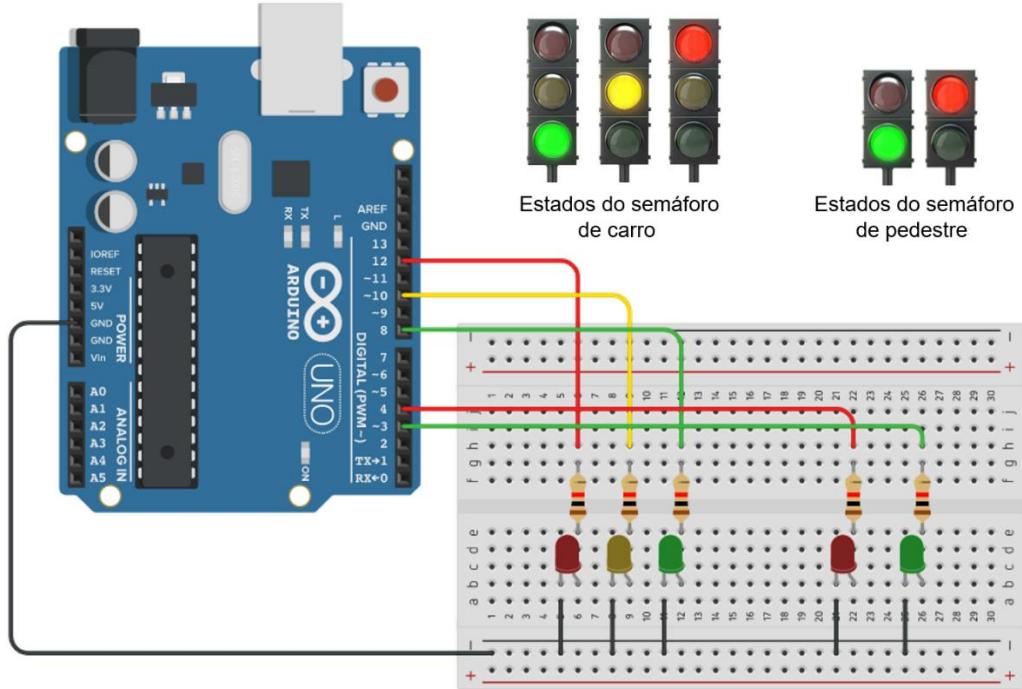


Figura 05 – Circuito da montagem do semáforo de carro e pedestre

- Realize as adaptações necessárias tanto no circuito (Figura 06) quanto no código do exercício anterior para cumprir as seguintes especificações:

- Inicialmente, o semáforo de carro permanece na fase verde e o semáforo de pedestres na fase vermelha, até que o botão seja pressionado.
- Após o botão ser pressionado, o semáforo de carro muda para a fase amarela e, em seguida, para a fase vermelha. Concomitantemente, o semáforo de pedestres muda para a fase verde.
- Os dois semáforos (carro e pedestres) mantêm essas fases por um período de três segundos. Após isso, o semáforo de pedestre realiza duas piscadas (desliga e liga) no LED verde e, em seguida, liga o LED vermelho.
- Por fim, o semáforo de carro faz a transição do estado vermelho de volta para o estado verde.

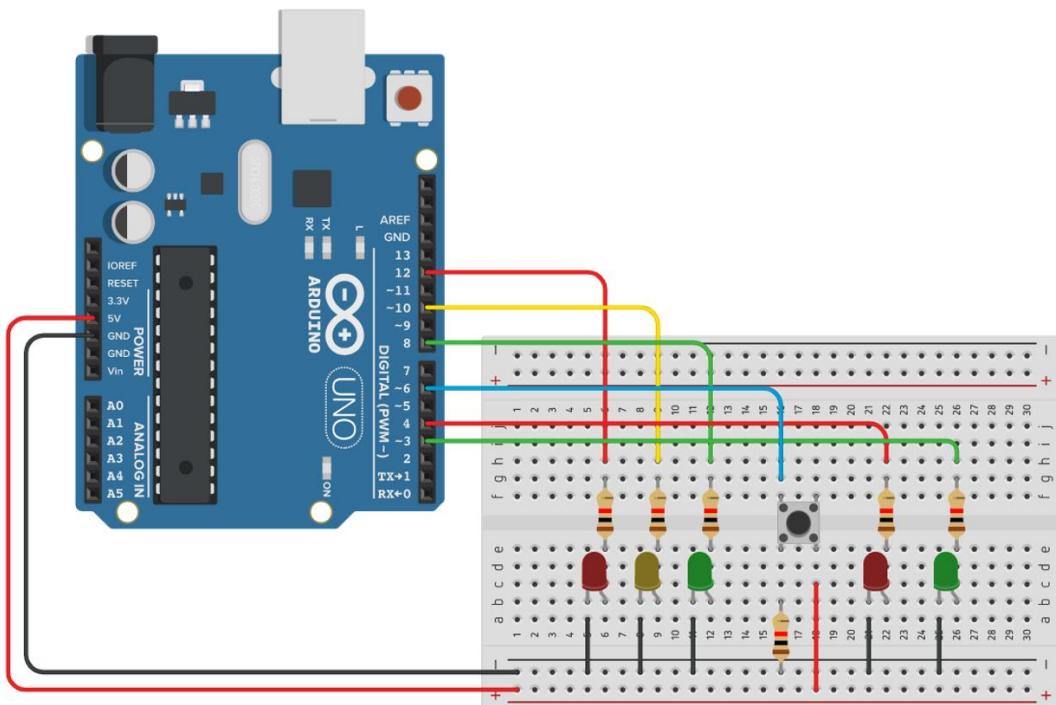


Figura 06 – Circuito da montagem do semáforo de carro e pedestre com botão