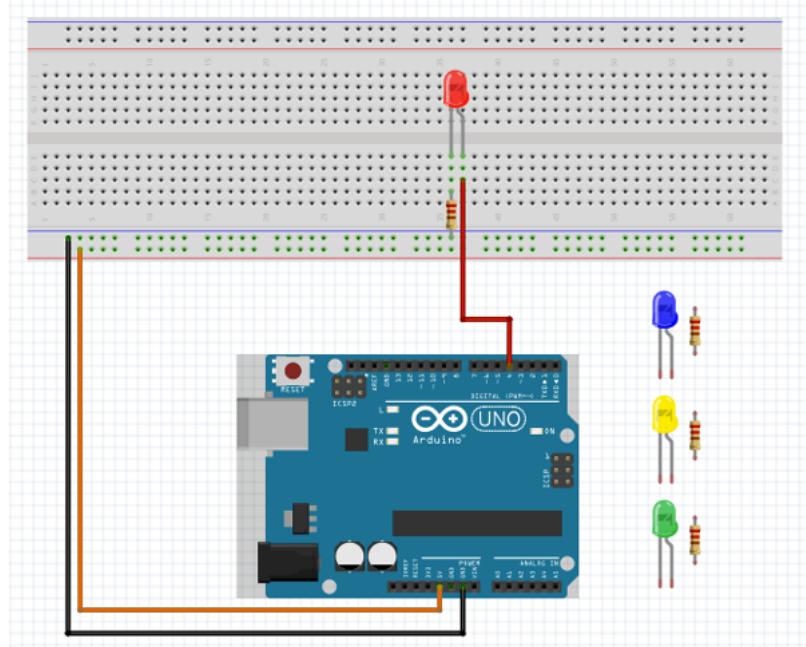


01  
**Controlando LED's**

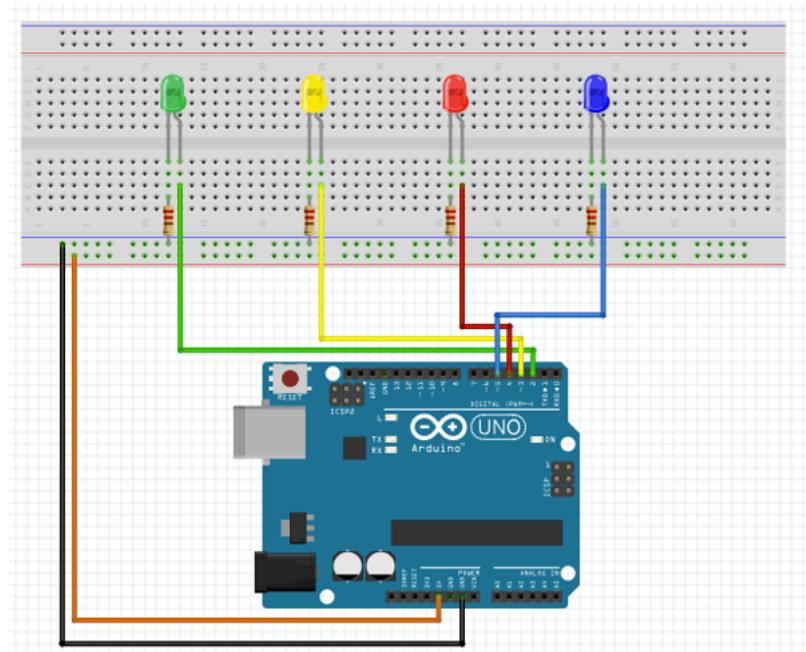
### Transcrição

No capítulo passado ajeitamos o LED Vermelho e o programamos para que ficasse piscando. No caso, nosso circuito ficou da seguinte maneira:



### Inserindo outros LEDs

Agora vamos organizar nosso circuito (*no Fritzing!*) para que possamos colocar os **4 LEDs** no jogo. Para tanto, utilizaremos a mesma ideia do primeiro, sempre deixando um espaço igual entre cada um dos LEDs, para que o circuito não fique bagunçado. Colocaremos na ordem: *verde, amarelo, vermelho, azul*. Teremos:



Perceba as **conexões**:

- LED Verde na porta 2;
- LED Amarelo na porta 3;
- LED Vermelho na porta 4;
- LED Azul na porta 5;

É importante que o circuito esteja nessa mesma ordem, pois isso interfere na programação do **Arduino**!

Após a adição desses outros LEDs podemos testar o código e ver como ele funciona. Perceba que na verdade quem acende é o LED Verde, pois é ele que está mapeado na porta 2. Os outros LEDs não realizam ação, uma vez que não foram programados para isso.

## Programando os Novos LEDs

Vamos partir para a ação? Devemos indicar ao nosso programa que temos 4 novos LEDs. Portanto, é preciso definir mais 3 constantes no nosso programa, além disso, não se esqueça de ajustar o número do LED Vermelho.

```
#define LED_VERDE 2
#define LED_AMARELO 3
#define LED_VERMELHO 4
#define LED_AZUL 5
```

Agora, repetimos o processo que fizemos com o LED Vermelho. Ou seja, precisamos falar para o **Arduino** que os nossos LEDs são uma saída e não entrada de dados. Por isso, dentro da nossa função de `setup()` teremos o seguinte:

```
void setup(){
    Serial.begin(9600);
    pinMode(LED_VERDE, OUTPUT);
    pinMode(LED_AMARELO, OUTPUT);
    pinMode(LED_VERMELHO, OUTPUT);
    pinMode(LED_AZUL, OUTPUT);
}
```

Depois de inserir o `setup` podemos utilizar a função `piscaLed()` que indicará para cada LED que ele deve piscar, em `loop()`. Vamos adicionar também um `delay()` no final, assim, poderemos ver claramente como o código está acontecendo. Ficará assim:

```
void loop(){
    piscaLed(LED_VERDE);
    piscaLed(LED_AMARELO);
    piscaLed(LED_VERMELHO);
    piscaLed(LED_AZUL);
    delay(MEIO_SEGUNDO);
}
```

Observe que devido as nossas definições o código ficou bastante expressivo. Até mesmo uma pessoa que não saiba programar, consegue compreender que ele pisca cada LED em sequência e depois dá uma pausa de `MEIO_SEGUNDO`.

