



Introdução à Robótica Arduino

única

PRIMEIROS PASSOS



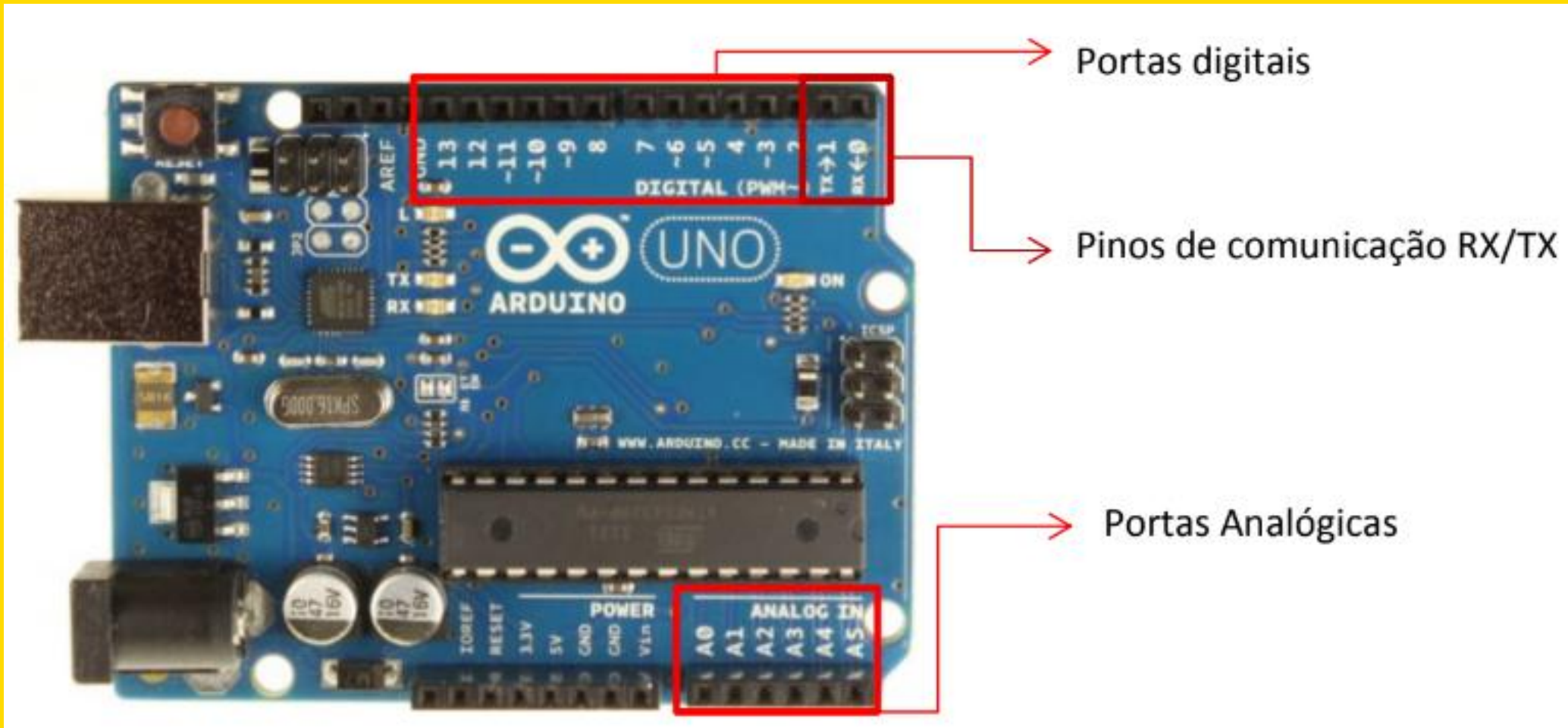
Hoje iremos apresentar os primeiros passos com arduino, ou seja, tudo o que é necessário para que você comece a desenvolver utilizando essa plataforma. Iremos mostrar o processo de instalação da IDE Arduino, materiais necessários, estrutura de um programa e programa exemplo pisca LED.

ARDUINO



O Arduino UNO é um microcontrolador em que possui 13 entradas do tipo digital 5 do tipo analógica, sendo que 6 das digitais possuem o recurso de PWM, que conta também com 2 pinos de comunicação RX/TX, e mais saídas de 3,3V e 5V, e alguns outros recursos que serão comentados durante o curso.

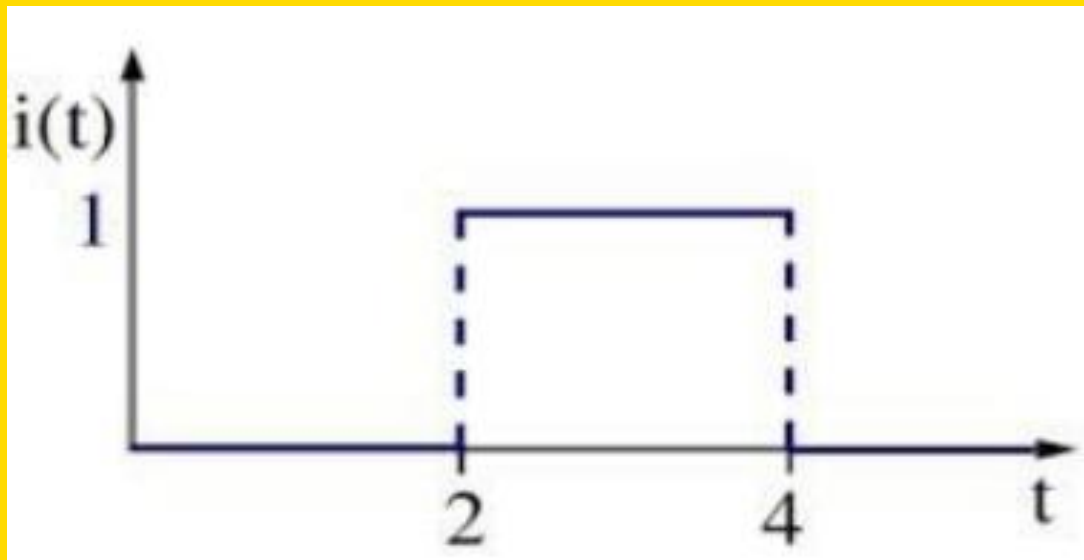
ARDUINO



ENTRADAS E SAÍDAS ANALÓGICAS E DIGITAIS

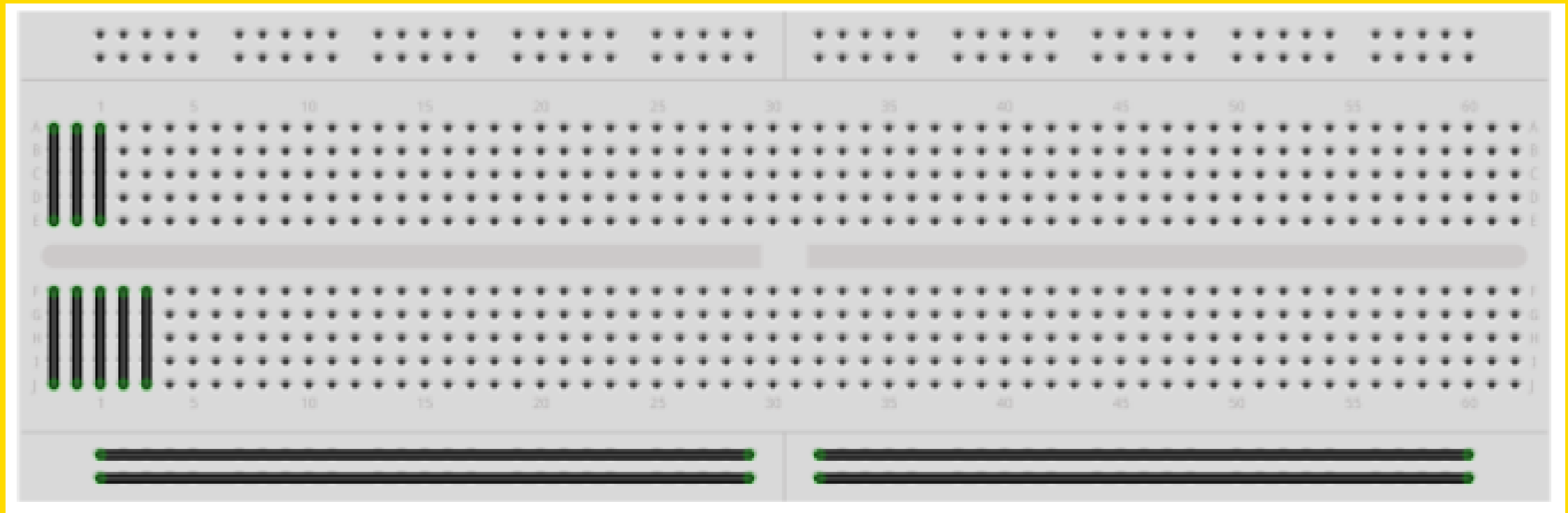


A porta digital tem o seu funcionamento de acordo com os princípios booleanos, ou seja, possui apenas dois estados: Verdadeiro ou Falso, certo ou errado, **0 ou 1**, **HIGH** ou **LOW**. No caso do arduino, ser High ou Low quer dizer que está passando corrente elétrica por aquela porta ou não.



OBS.: A portas digitais do arduino suportam até 40mA

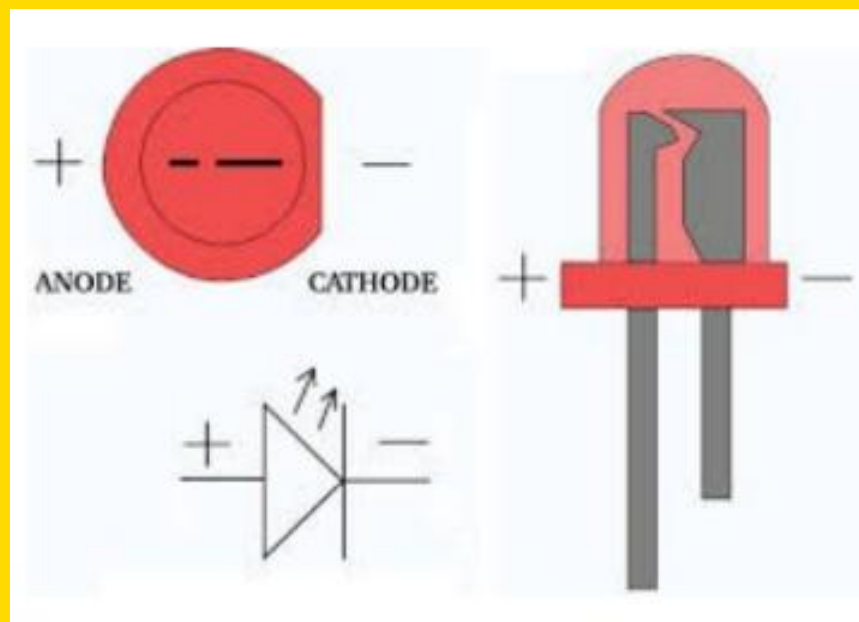
PROTOBOARD



LED



O led é um dispositivo que permite a passagem de corrente elétrica em apenas um sentido do catodo para o anodo, o LED de 5mm simples precisa de apenas 4mA para e 1,90V para ser ligado.



PROJETANDO



Em todo projeto eletrônico, seja ele o mais simples ou os mais complexos, é muito importante que todas as características de funcionamento dos componentes que serão utilizados, sejam muito bem conhecidas.

Em nosso primeiro exemplo prático, vamos fazer com que um led fique piscando, então antes de mais nada vamos conhecer algumas características deste componente.

PROJETANDO



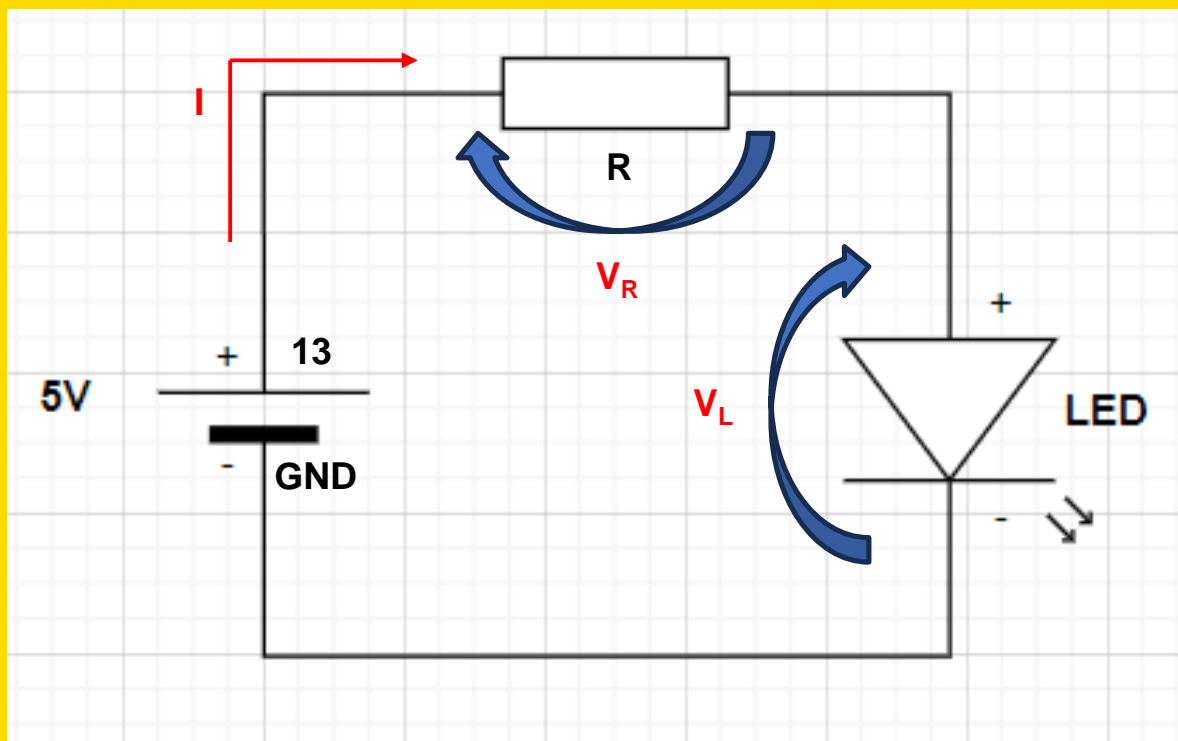
LEDs		
Cor do LED	Tensão em Volts (V)	Corrente em Miliamperes (mA)
Vermelho	1,8V – 2,0V	20 mA
Amarelo	1,8V – 2,0V	20 mA
Laranja	1,8V – 2,0V	20 mA
Verde	2,0V – 2,5V	20 mA
Azul	2,5V – 3,0V	20 mA
Branco	2,5V – 3,0V	20 mA

Em nosso primeiro projeto, vamos utilizar um Led Vermelho. Portanto sabemos que a **tensão máxima que ele suporta é de 2,0V** e que a **corrente elétrica máxima suportada por ele é de 20mA**.

PROJETANDO



Sabendo disso vamos dar uma analisada no circuito que vamos montar já pensando que precisaremos inserir um resistor que servirá como proteção para que o led ou a saída da placa do arduino não queimem.





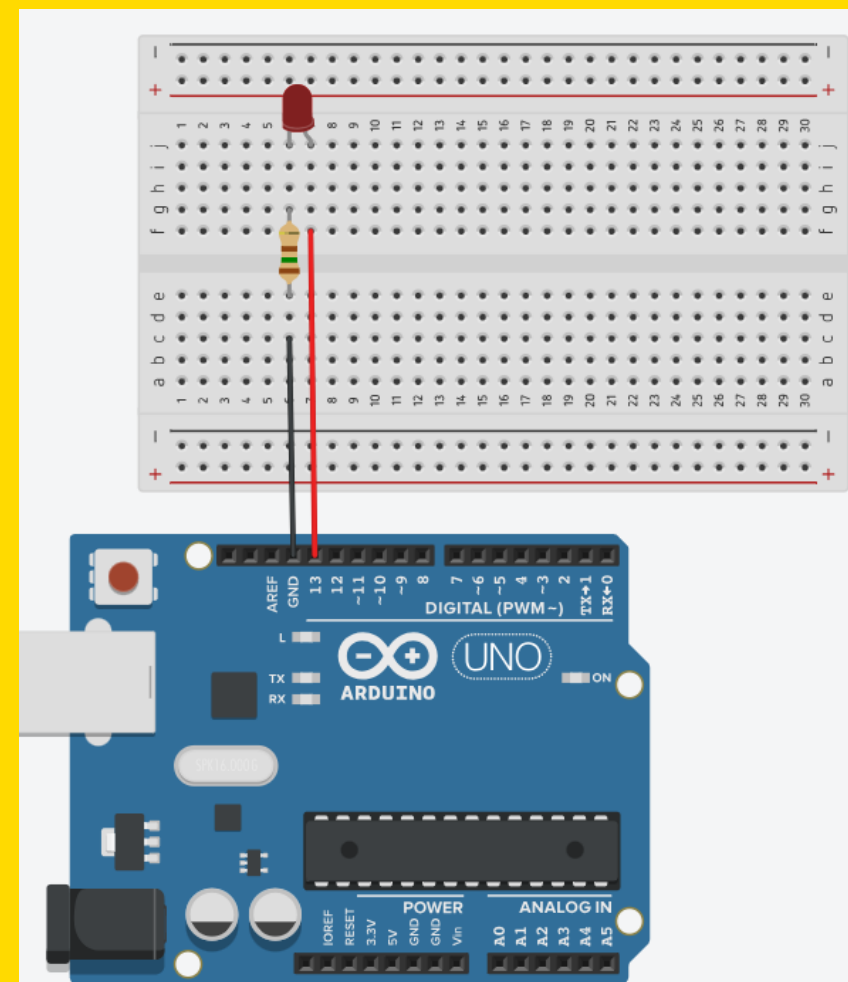
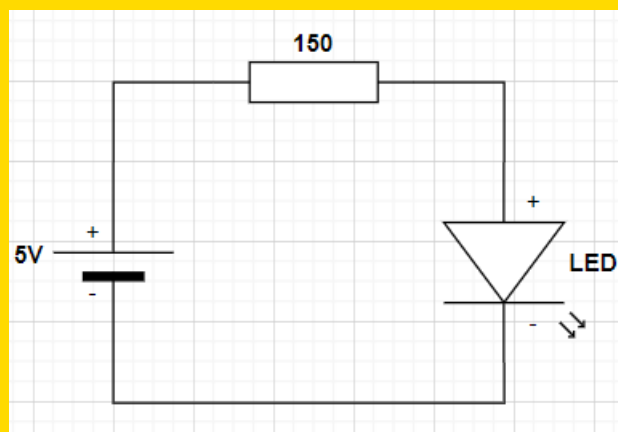
PROJETANDO

Sabendo que a tensão máxima em cima do led é de 2V e que a corrente máxima é de 20mA e que a tensão fornecida ao circuito é de 5V, sobra sobre o resistor 3V e a corrente é a mesma do led pois estão em um circuito em série. Daí para se calcular o led utilizamos a 1ª Lei de Ohm.

$$R = \frac{V}{I}$$
$$R = \frac{3V}{20mA} = \frac{3}{0,02} = 150 \Omega$$

PROJETANDO

Nosso circuito final deverá ter no mínimo a seguinte configuração para o perfeito funcionamento do Led e do Arduino:



PROJETANDO



NOTA: Feitos os cálculos, pode ser que o valor encontrado para o resistor não seja encontrado comercialmente, então neste caso, podemos escolher um resistor com um valor aproximado que seja um pouquinho maior. Exemplo: Nos cálculos encontramos o valor ideal de 312Ω , então nesse caso podemos utilizar um de 330Ω que é facilmente encontrado nas lojas de eletrônicos.



Vamos a prática!



MATERIAIS NECESSÁRIOS

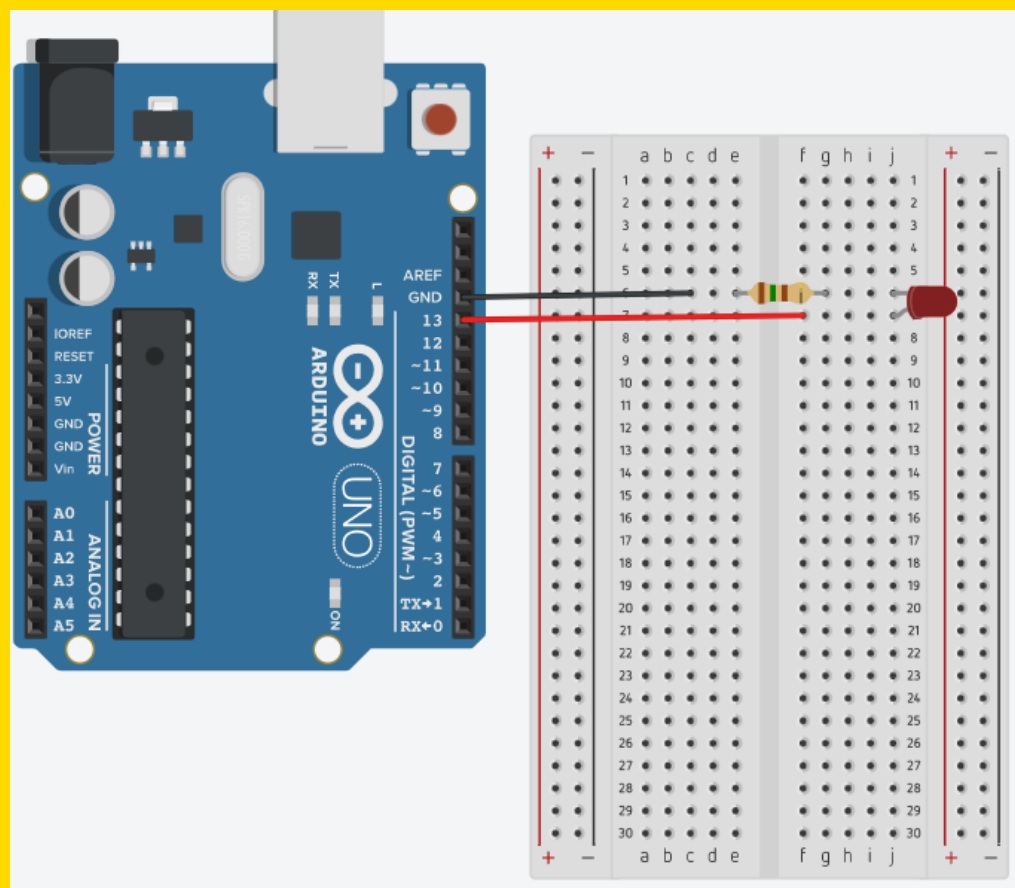


- PLACA UNO R3 COM CABO USB
- PROTOBOARD
- LED VERMELHO
- RESISTOR 150 OHM
- JUMPERS MACHO/MACHO
- COMPUTADOR

O CIRCUITO



O circuito deve ficar da seguinte maneira na protoboard:



ESTRUTURA DE UM PROGRAMA ARDUINO

A screenshot of the Arduino IDE 2.1.0 interface. The window title is "exemplo_01 | Arduino IDE 2.1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar shows icons for checking, running, and uploading code, along with a dropdown menu set to "Arduino Uno". The left sidebar contains icons for file explorer, libraries, serial monitor, and search. The main editor area shows the code for "exemplo_01.ino". The code is as follows:

```
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 }  
10
```

Two blue arrows point from the right towards the code. The first arrow, labeled with a red "1", points to the "void setup()" function. The second arrow, labeled with a red "2", points to the "void loop()" function. The status bar at the bottom indicates "Ln 10, Col 1" and "Arduino Uno [not connected]".

ESTRUTURA DE UM PROGRAMA ARDUINO



1 - setup() – É nessa parte do programa que você configura as **opções iniciais do seu programa**: os valores iniciais de uma variável, se uma porta será utilizada como entrada ou saída, mensagens para o usuário, etc. **Essa função irá executar apenas uma vez no início do programa.**

2 - loop() – Diferente da função setup(), essa parte do programa **repete uma estrutura de comandos de forma contínua** ou até que alguma comando de “parar” seja enviado ao Arduino.

EXEMPLO PISCAR UM LED



```
exemplo_01 | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Icons] Arduino Uno [Dropdown] [Icons]
exemplo_01.ino
1 void setup() {
2   //Define a porta do led como saida
3   pinMode(13, OUTPUT);
4 }
5
6 void loop() {
7   //Acende o led
8   digitalWrite(13, HIGH);
9   //Aguarda o intervalo especificado
10  delay(1000);
11  //Apaga o led
12  digitalWrite(13, LOW);
13  //Aguarda o intervalo especificado
14  delay(1000);
15 }
16
```

Ln 3, Col 23 Arduino Uno [not connected]

ESTRUTURA DE UM PROGRAMA ARDUINO

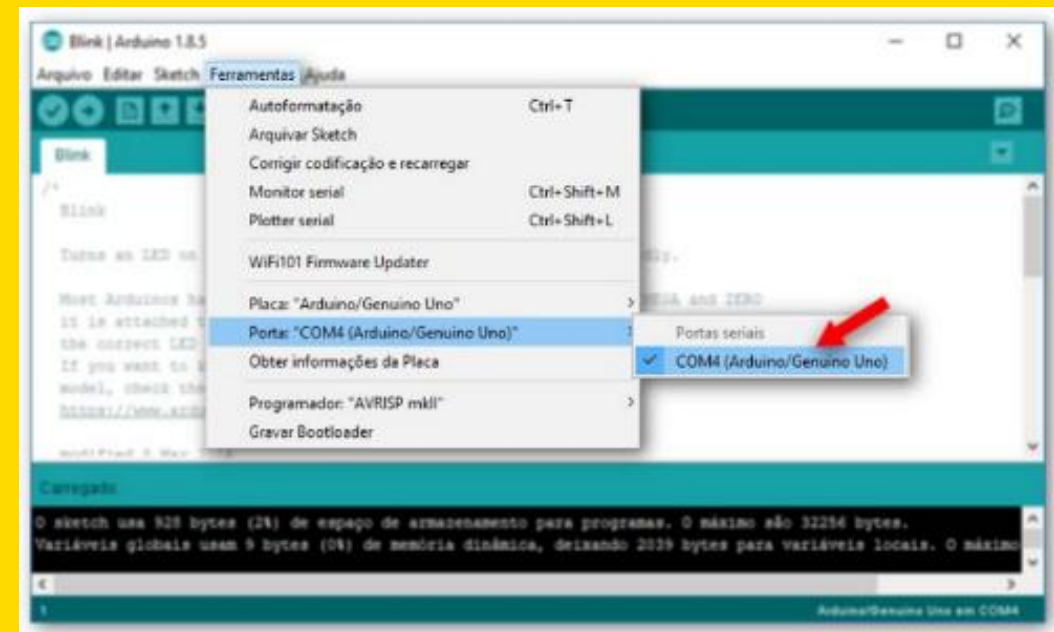
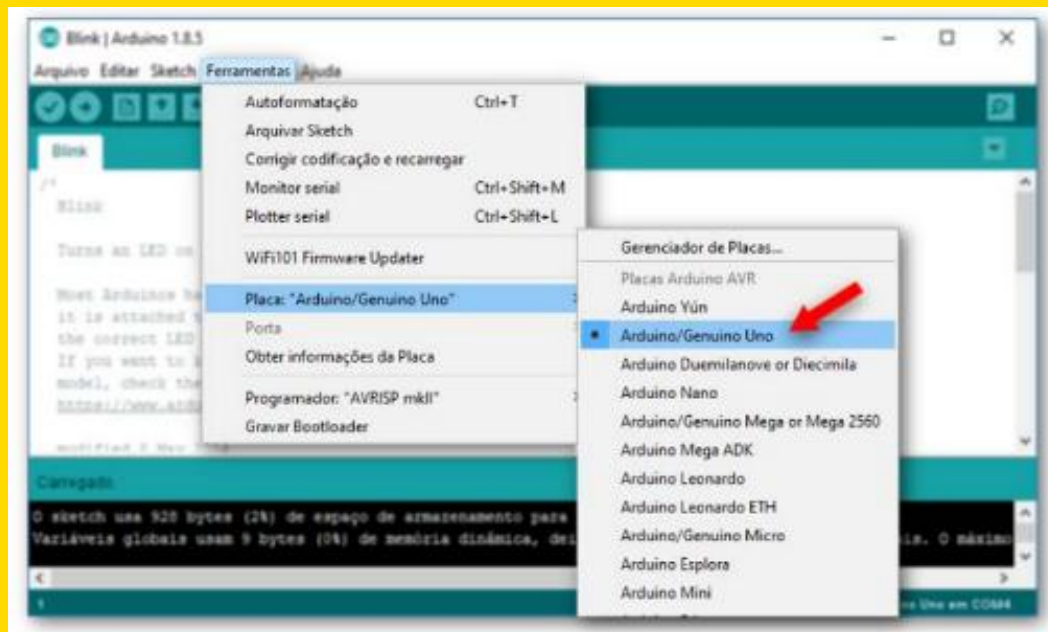


- Na estrutura do `setup()` definimos **`pinMode(13, OUTPUT)`** que faz com que o pino 13 do Arduino seja utilizado como saída.
- A linha do código contendo **`digitalWrite(13, HIGH)`** coloca a porta 13 em nível alto (HIGH, ou 1), acendendo o led embutido na placa. O comando `delay(1000)`, especifica o intervalo, em milissegundos, no qual o programa fica parado antes de avançar para a próxima linha.
- O comando **`digitalWrite(13, LOW)`**, apaga o led, colocando a porta em nível baixo (LOW, ou 0), e depois ocorre uma nova parada no programa, e o processo é então reiniciado.

VERIFICAÇÃO E ENVIO DO PROGRAMA



Quando o código estiver pronto para ser carregado na placa, conecte a placa Arduino no seu computador, entre no menu ferramentas, escolha o modelo da placa e a porta na qual a mesma está conectada:



VERIFICAÇÃO E ENVIO DO PROGRAMA



Se estiver tudo OK, clique no botão carregar. Isso irá gravar o programa na placa.



Caso não apareça nenhum erro, você deverá ver o LED piscando em um intervalo de 1 segundo.

