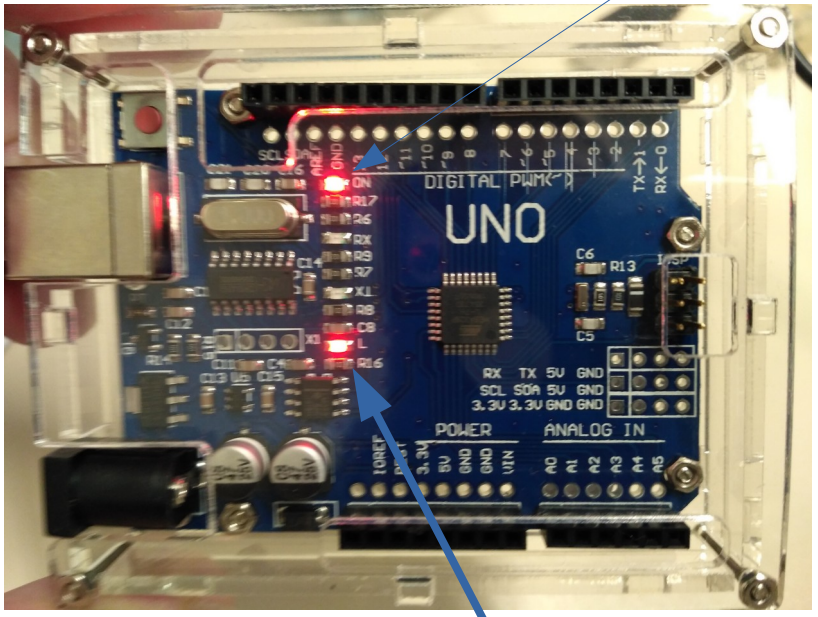
Workshop Arduino

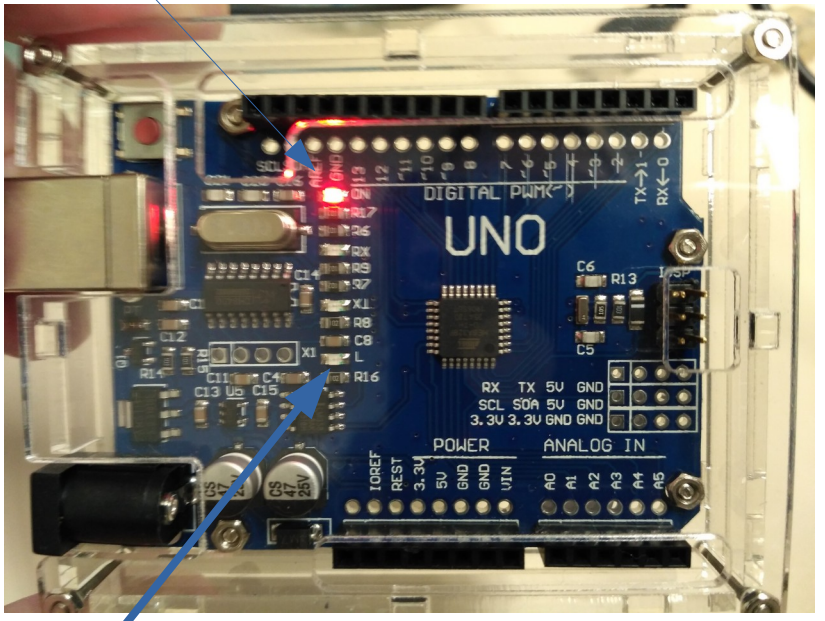
Prof. Simão

simao@ufpr.br - 2022

Já temos um led piscando...

LED de energia – sempre aceso





LED da placa (pino 13) deverá piscar

Imagens: fotos do autor

simao@ufpr.br - 2022

Vamos fazer isto fora da placa

- A placa possui pinos configuráveis como entradas ou saídas
  - Digitais: ligado e desligado
  - Analógicas: uma gama de valores entre mínimo e máximo

Vamos fazer isto fora da placa

- Vamos reusar o código já testado, mudando o número da porta que usaremos como saída

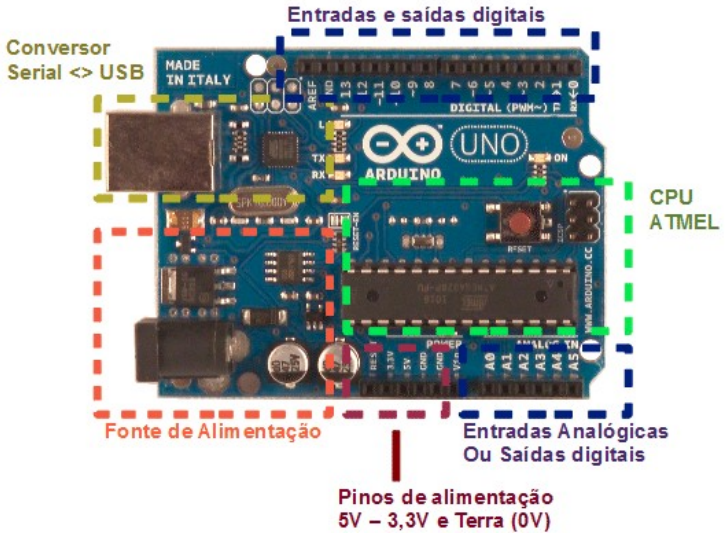


Imagem: https://www.researchgate.net/figure/Figura-1-Placa-de-prototipagem-eletronica-Arduino-UNO-Fonte-wwwrobotizandocombr_fig1_307593728

simao@ufpr.br - 2022

Breadboard / Protoboard

- Placa para montagens experimentais

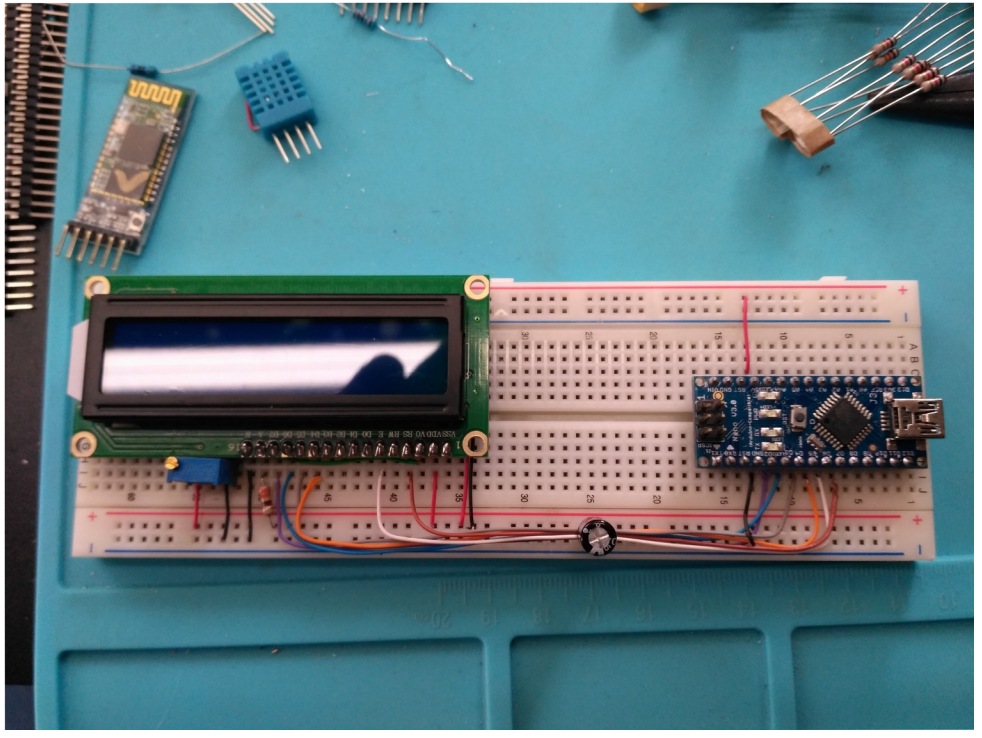


Imagem: foto do autor

simao@ufpr.br - 2022

Breadboard / Protoboard

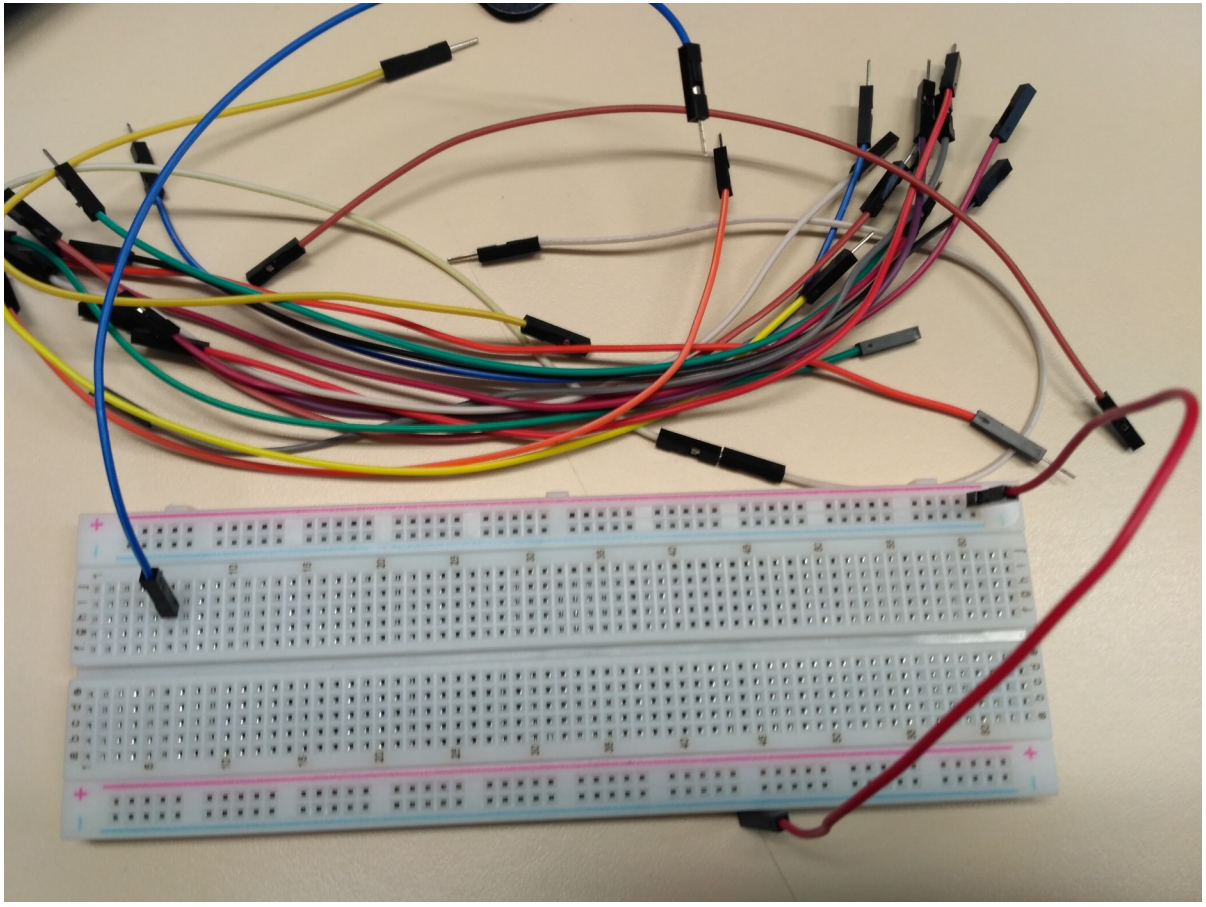
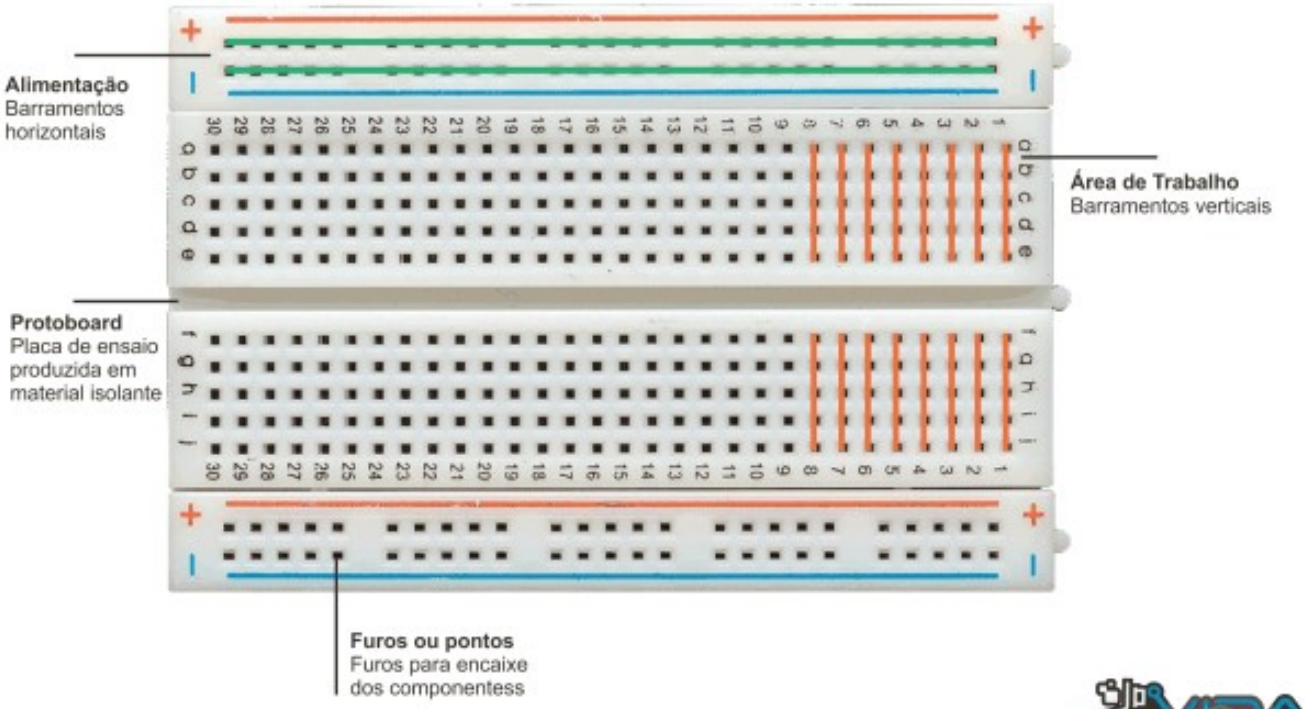


Imagem: foto do autor

simao@ufpr.br - 2022

Breadboard / Protoboard

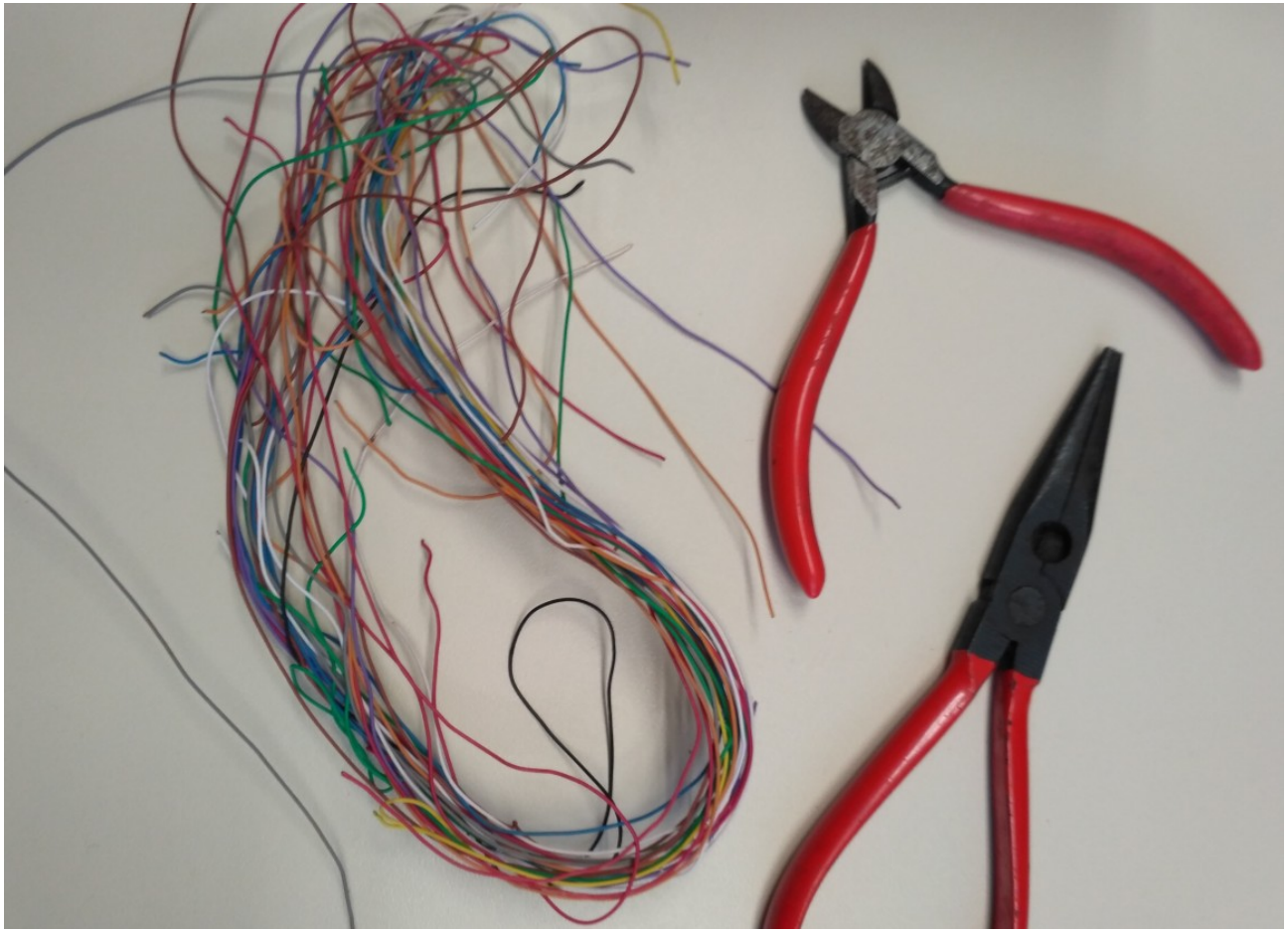


Robótica e Sistemas Digitais.

Imagem: vida de silício

simao@ufpr.br - 2022

Componentes



simao@ufpr.br - 2022

Placas padronizadas - soldagem

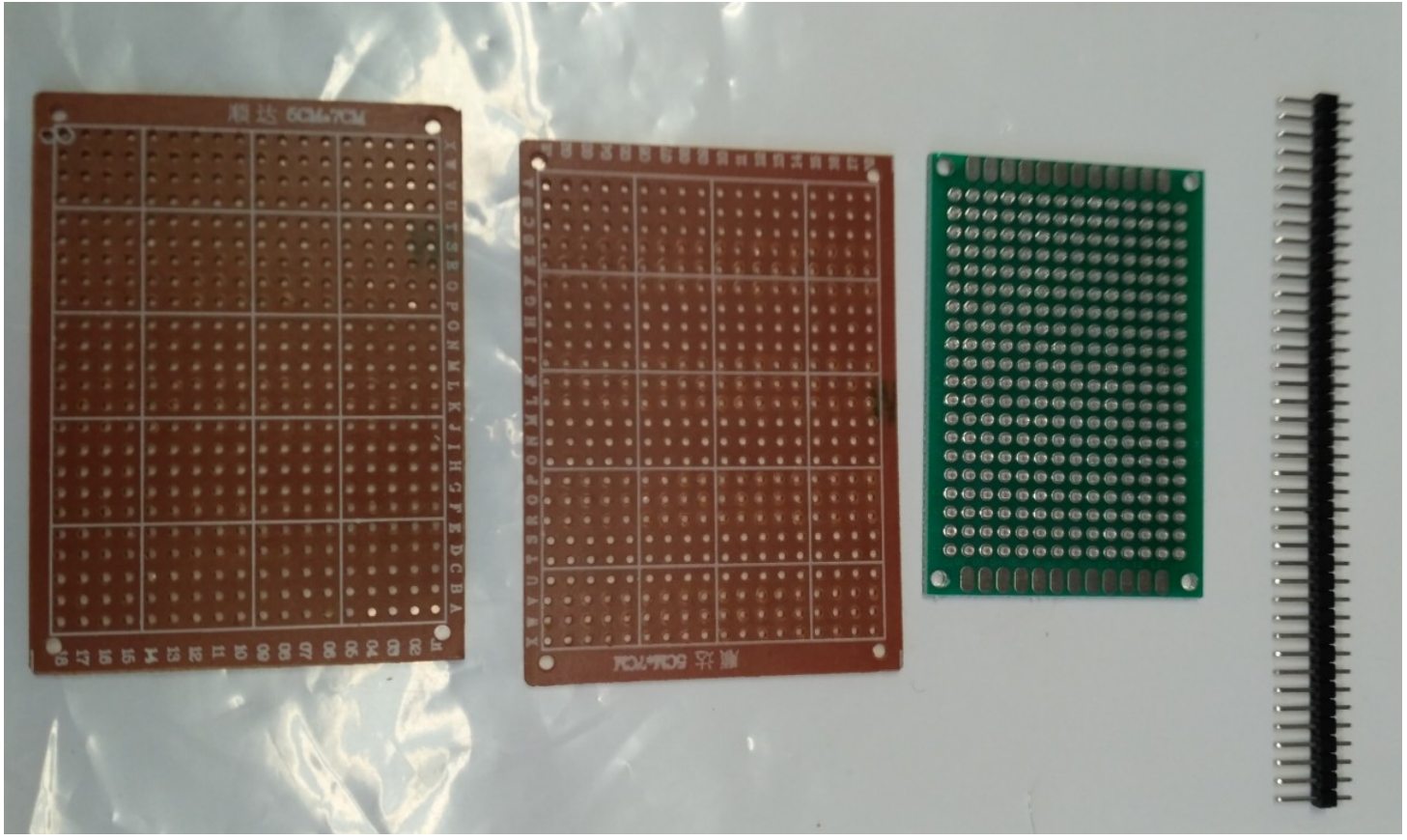


Imagem: foto do autor

simao@ufpr.br - 2022

Vamos trabalhar

O que precisamos?

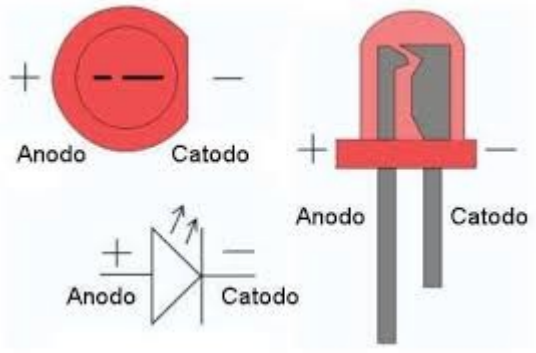
- Arduino e IDE configurados
- *Breadboard* e fios de conexão
- Um LED, um resistor
  - O resistor é necessário em função de que o LED trabalha com características elétricas diferentes do Arduino

simao@ufpr.br - 2022

Início da eletrônica teórica

simao@ufpr.br - 2022

Diodo emissor de luz - LED



É um componente que possui polaridade definida, um lado vai ligado à tensão positiva e outro à negativa da fonte.

O Arduino trabalha com 5V; algumas placas trabalham com 3,3V. Os leds vermelhos trabalham com 1,7V, o amarelo com 2V e o verde com 2,1V. Azuis e brancos beiram 3V e infravermelhos 1,1V.

A corrente dos leds dificilmente ultrapassa 20mA.

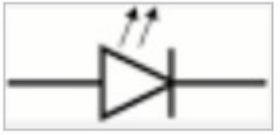
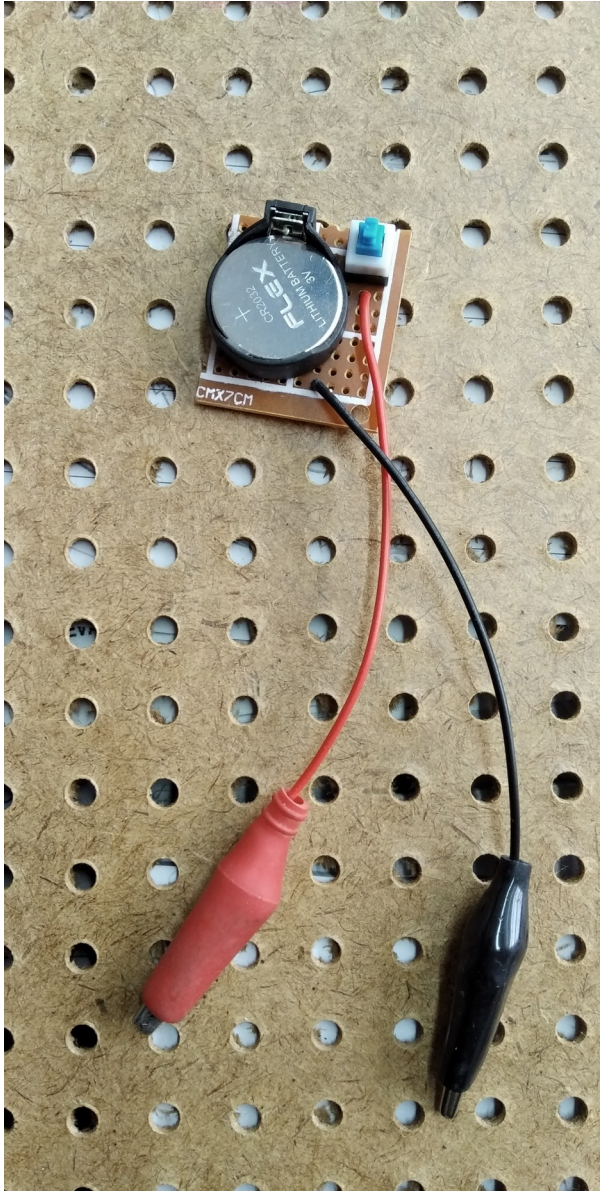
O símbolo do LED é  (ou  ou )

Imagem: Google

simao@ufpr.br - 2022

Polaridade

- Se tiver dúvidas quanto à polaridade do LED, use um multímetro ou uma pequena bateria (CR2032) para testar.



simao@ufpr.br - 2022

Há problemas em inverter?

- A inversão da polaridade em um componente polarizado, tal como o led, um transistor e alguns tipos de capacitores, poderá queimá-lo.
- Em certas condições também poderá haver dano na placa do Arduino.

Como limitar a corrente/ tensão

- De forma a adequar as grandezas elétricas disponíveis na placa do Arduino às características do led, usamos resistores.

Resistores

- Como diz o nome, são componentes que resistem à passagem da corrente elétrica.
- Em função do esforço realizado para a passagem por um resistor será gerada uma queda na tensão disponível.

simao@ufpr.br - 2022

Lei de Ohm

- Estudada na cadeira de física do ensino médio, a Lei de Ohm estabelece uma relação entre tensão, corrente e resistência.
- A queda de tensão é dada pelo produto da corrente pela resistência:
  - E = R.I

Valor do resistor

- Vamos estabelecer uma baixa corrente para nosso resistor, 10mA (0,01A), e vamos calcular para que produza uma queda de 3V na tensão:
  - E = R.I
  - 3V = R.0,01
  - R = 300 Ω (300 ohms)

Séries comerciais de resistores

	A	B	C	D
1	**Valor**	**E24**	**E12**	**E6**
2	1	5,00%	10,00%	20,00%
3	1,1	5,00%		
4	1,2	5,00%	10,00%	
5	1,3	5,00%		
6	1,5	5,00%	10,00%	20,00%
7	1,6	5,00%		
8	1,8	5,00%	10,00%	
9	2	5,00%		
10	2,2	5,00%	10,00%	20,00%
11	2,4	5,00%		
12	2,7	5,00%	10,00%	
13	3	5,00%		
14	3,3	5,00%	10,00%	20,00%
15	3,6	5,00%		
16	3,9	5,00%	10,00%	
17	4,3	5,00%		
18	4,7	5,00%	10,00%	20,00%
19	5,1	5,00%		
20	5,6	5,00%	10,00%	
21	6,2	5,00%		
22	6,8	5,00%	10,00%	20,00%
23	7,5	5,00%		
24	8,2	5,00%	10,00%	
25	9,1	5,00%		

Existem várias séries de valores padronizados para os resistores.

Nos circuitos, escolhe-se o componente comercial de valor mais próximo do desejado.

Os resistores são fabricados/ vendidos baseando-se em um valor central, ao longo do qual temos variações para mais ou para menos (tolerância).

Os valores indicados ao lado terão multiplicadores dados por potências de dez (ex.: x 10, x10.000, etc).

Imagem: cópia de tela de planilha LibreOffice efetuada pelo autor

simao@ufpr.br - 2022

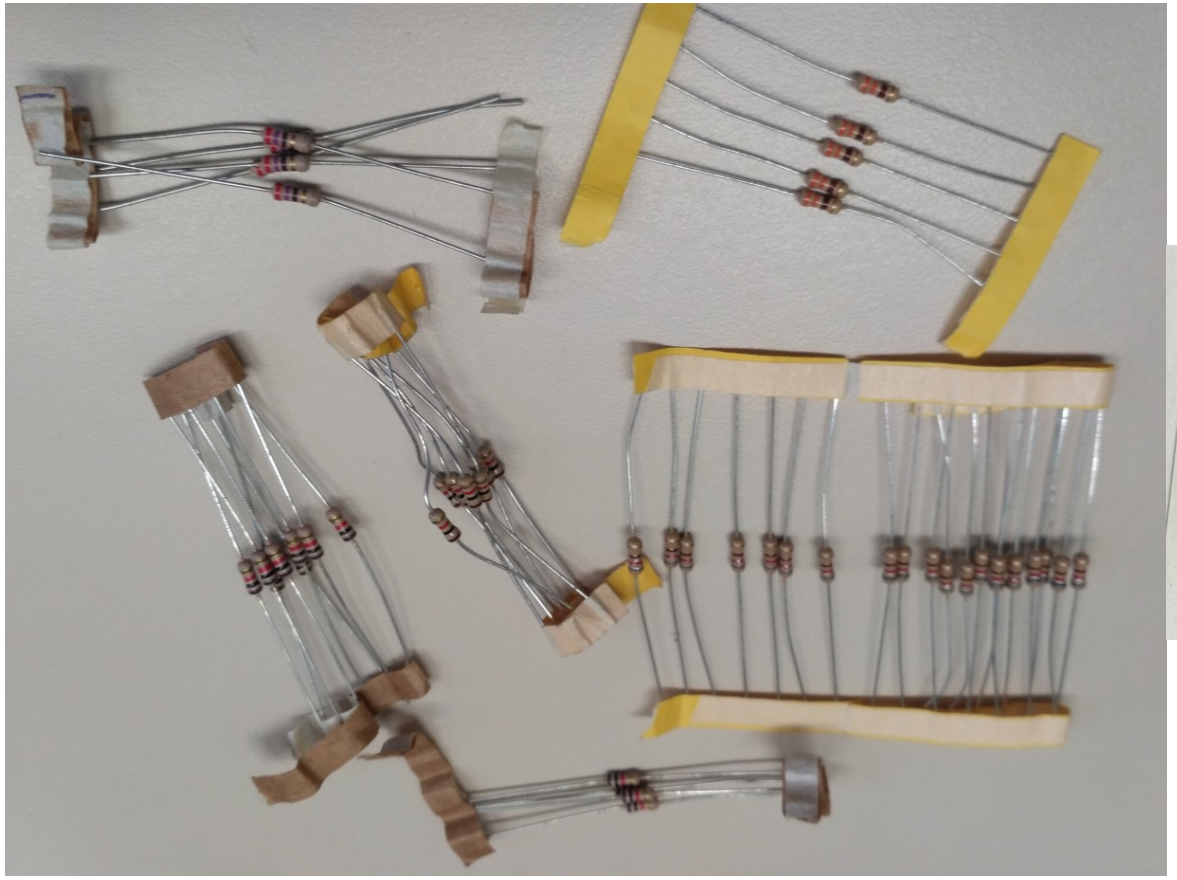
E o nosso resistor de 300Ω ?

- Os valores mais próximos seriam 2,7; 3 e 3,3; todos multiplicados por 100.
- A escolha óbvia seria 3 x 100 = 300, porém nem sempre temos à mão o valor desejado.

12	2,7	5,00%	10,00%	
13	3	5,00%		
14	3,3	5,00%	10,00%	20,00%

Imagem: cópia de tela de planilha LibreOffice efetuada pelo autor

Aspecto de resistores



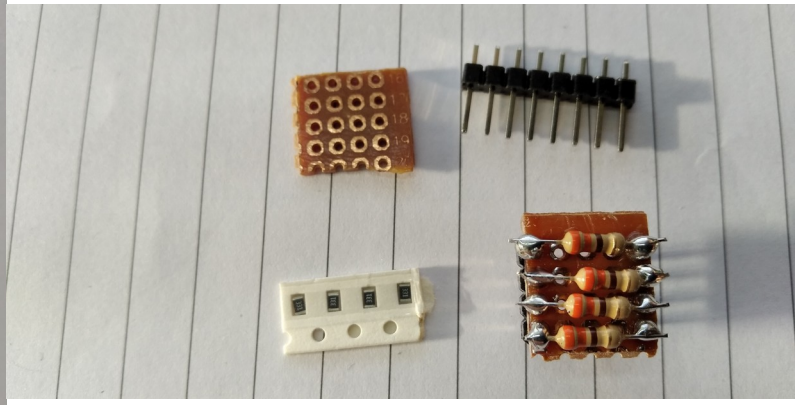


Imagem: foto do autor

simao@ufpr.br - 2022

Código de cores

Cor	1ª Faixa	2ª Faixa	N° de zeros/multiplicador	Tolerância
Preto	0	0	0	
Marrom	1	1	1	
Vermelho	2	2	2	
Laranja	3	3	3	
Amarelo	4	4	4	
Verde	5	5	5	
Azul	6	6	6	
Violeta	7	7	7	
Cinza	8	8	8	
Branco	9	9	9	
Dourado			x0,1	
Prata			x0,01	
Sem cor				± 20%

Resistores são identificados por códigos de cores, conforme mostrado ao lado.

O resistor que aparece abaixo possui (marrom=)1 (preto=) 0 (vermelho=) 00 ohms, ou 1000 ohms, ou 1000 Ω (normalmente identificado como sendo de 1kΩ).

Imagem: https://www.mundodaeletrica.com.br/i/766/3-faixas-600.webp

simao@ufpr.br - 2022

Finalmente, nossas opções

Vermelho (2), violeta (7), marrom (1 zero, x10) = 270 Ω

Laranja (3), Preto (0), marrom (1 zero, x10) = 300 Ω

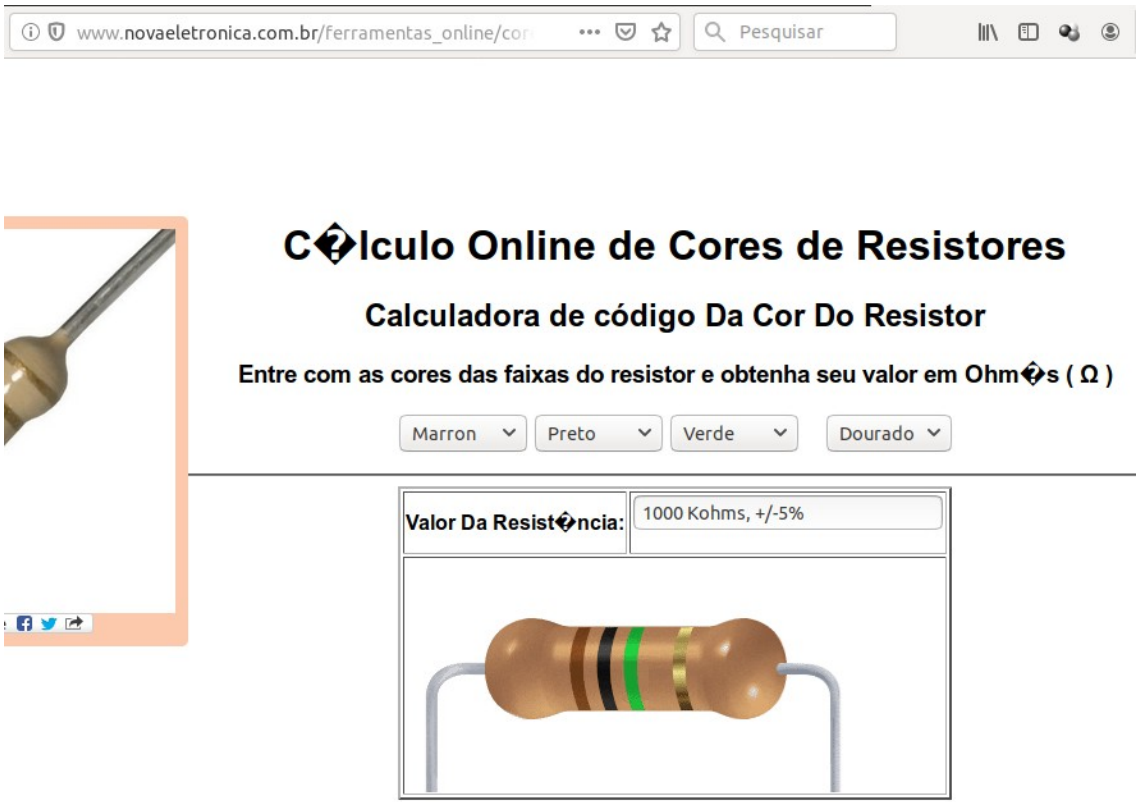
Laranja (3), Laranja (3), marrom (1 zero, x10) = 330 Ω

Imagens: Google

simao@ufpr.br - 2022

Calculadora *online*

http://www.novaeletronica.com.br/ferramentas_online/cores-de-resistor-online.php



Fim da eletrônica teórica

simao@ufpr.br - 2022

Ligar led externo

- Primeiramente vamos levar a energia para a *breadboard* e testar
- Depois, iremos mudar o código que faz piscar o led da placa do Arduino para que pisque o led externo

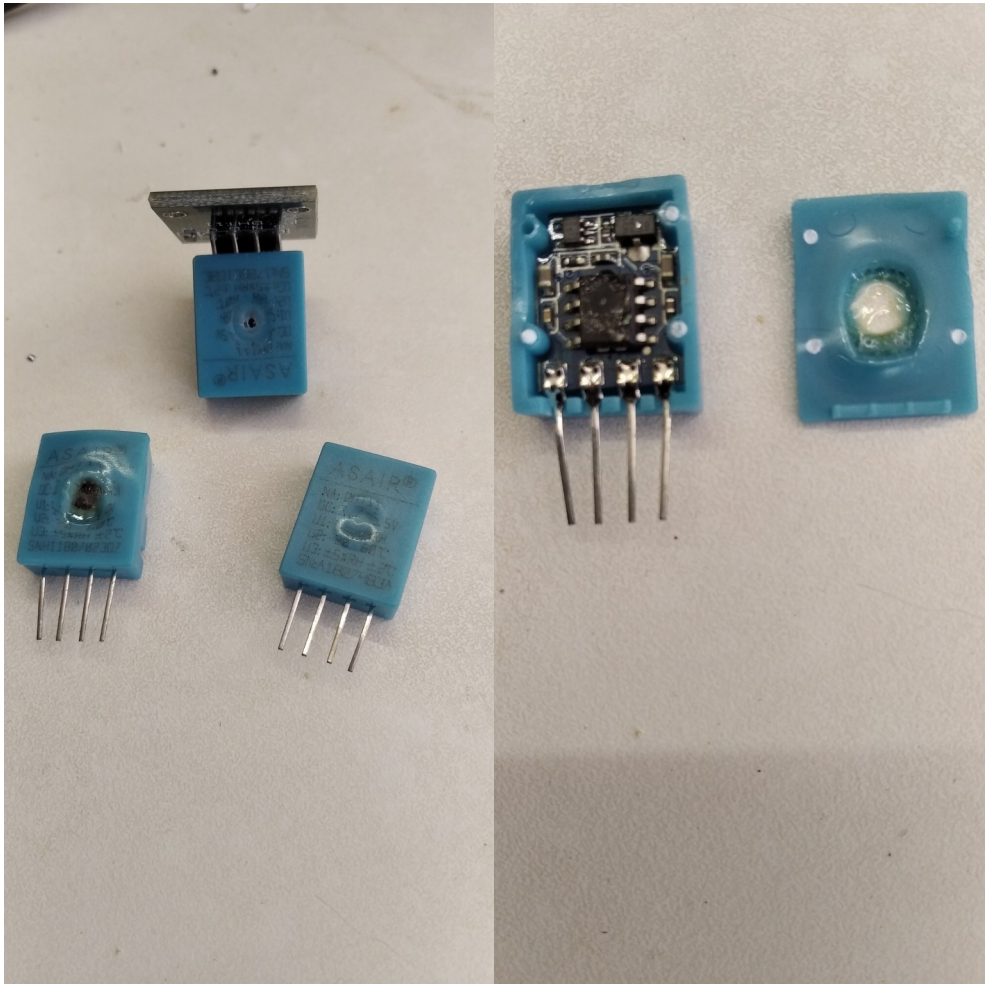
simao@ufpr.br - 2022

Evite problemas.

Trabalhe com a energia desligada
(desconecte o cabo USB).

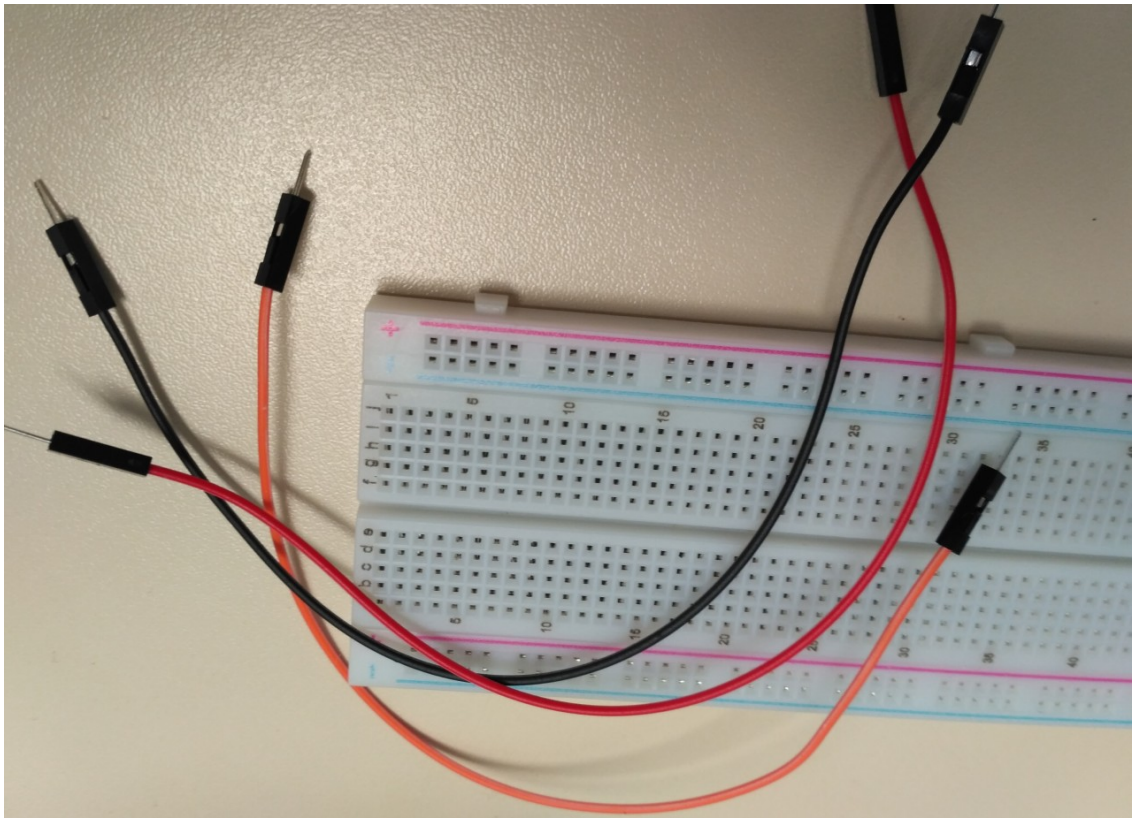
Evite problemas.

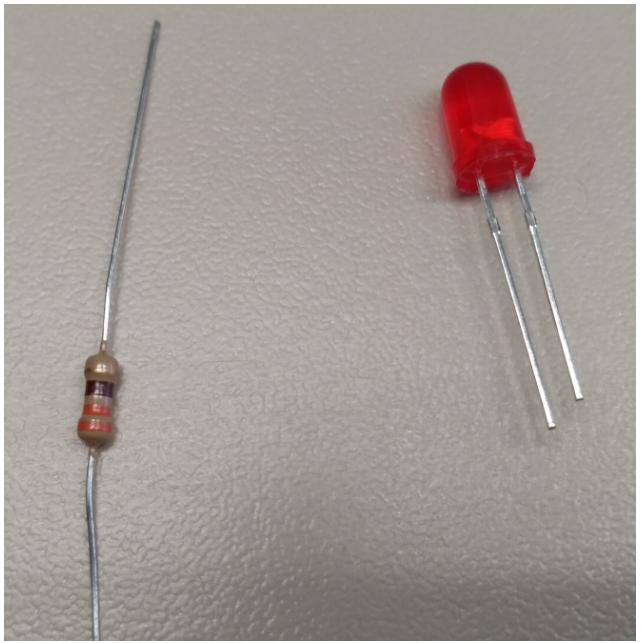
Atenção à polaridade dos componentes. Se você inverter (o positivo com o negativo)...



simao@ufpr.br - 2022

Componentes necessários

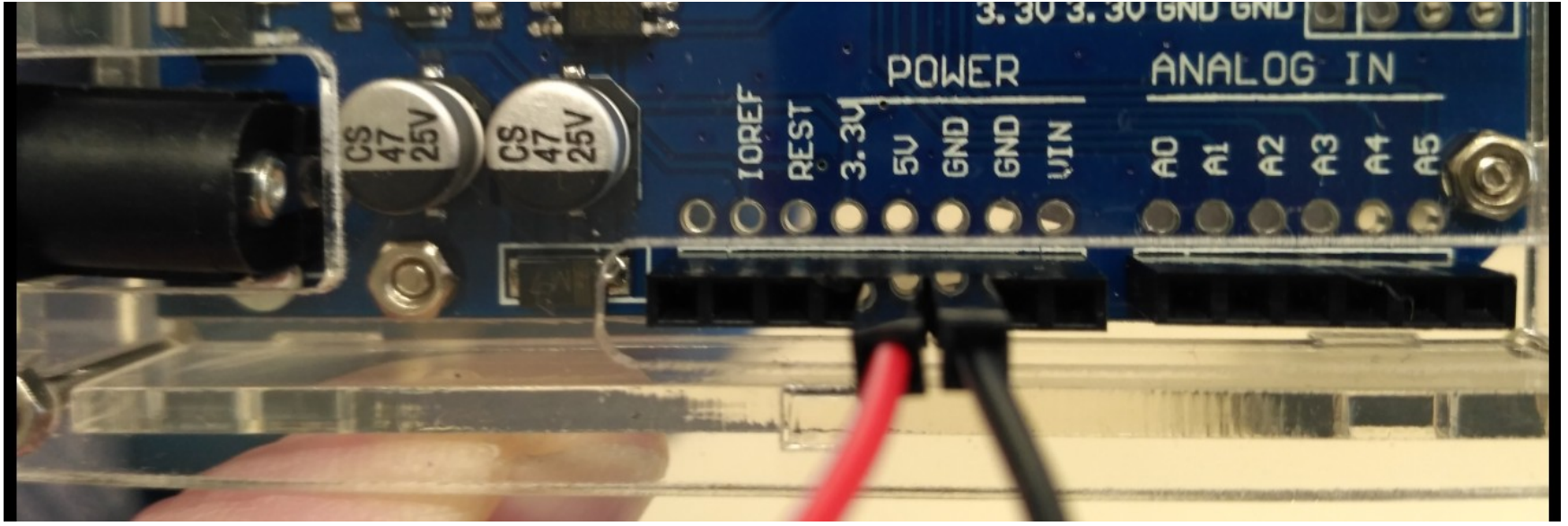




Imagens: foto do autor

simao@ufpr.br - 2022

1. Ligação de energia

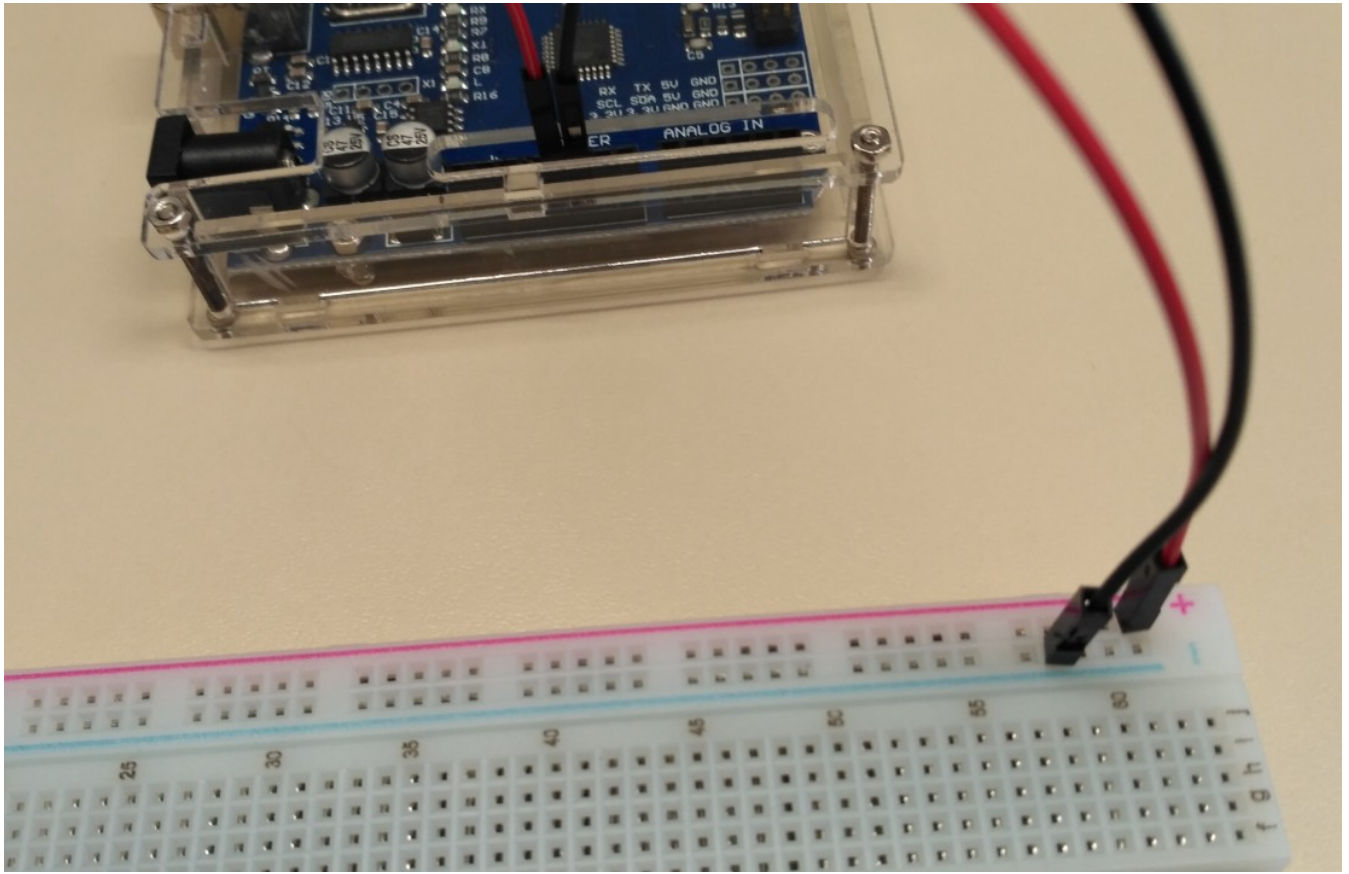


Separe dois condutores (no exemplo **vermelho** e **preto**) e ligue-os aos pontos **5V** (ou VCC) e **GND** (*ground*, 0V) de sua placa Arduino. Em geral, o condutor **vermelho é o positivo** e o **preto é o negativo**.

Imagens: foto do autor

simao@ufpr.br - 2022

1. Ligação de energia



Leve os pontos de energia para placa de experimentação.

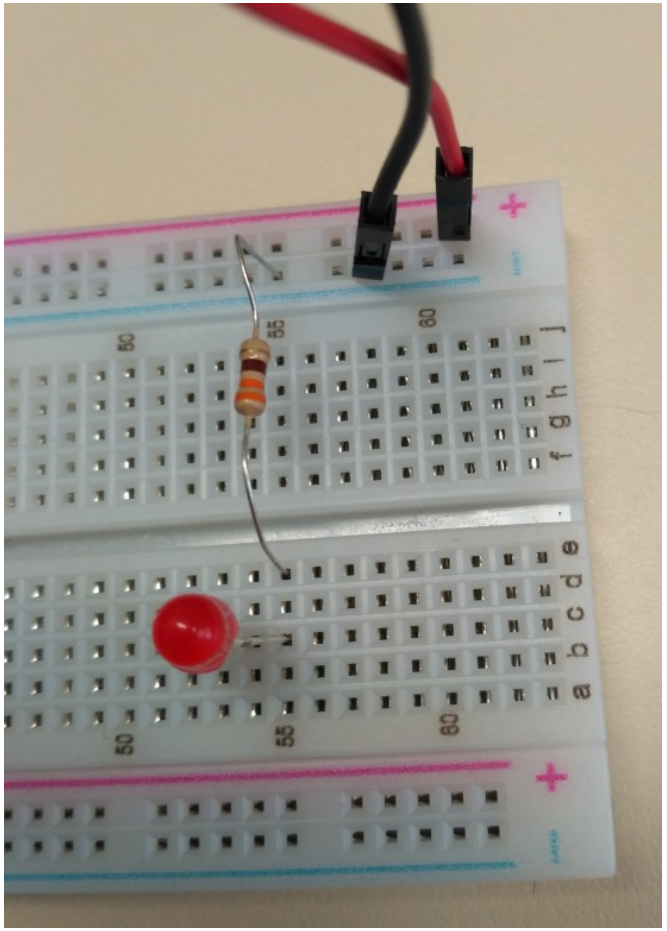
As linhas coloridas no exemplo indicam que os pontos horizontais estão interconectados em uma barra.

Teremos uma linha de alimentação **positiva** (+5V) e outra com o **0V**.

Imagens: foto do autor

simao@ufpr.br - 2022

1. Ligação de energia



O resistor não é polarizado, o led é.

Ligue um terminal do resistor na barra de 0V e o outro terminal em uma coluna qualquer (no exemplo, na 55).

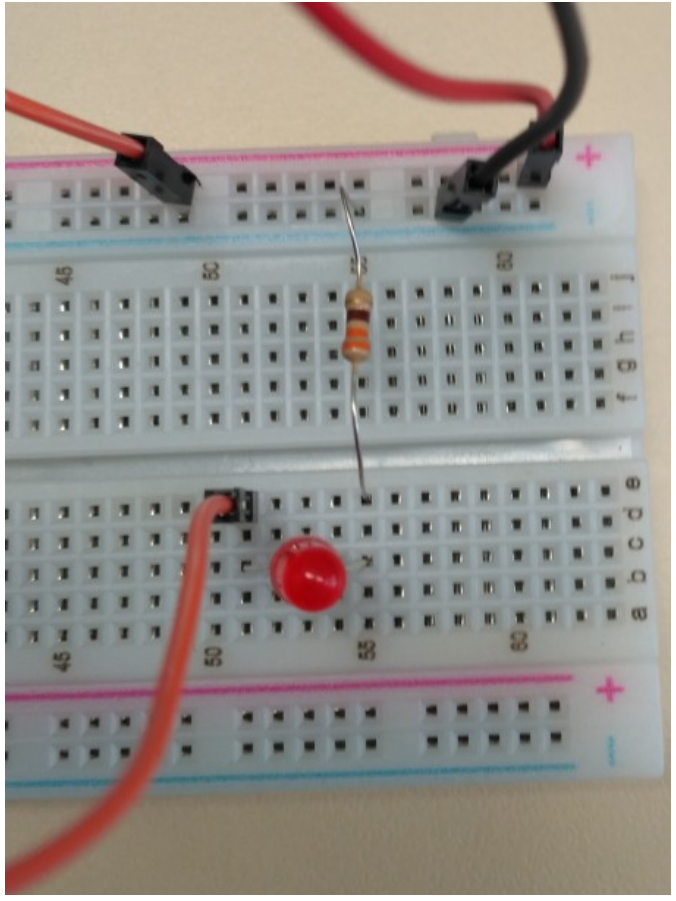
Ligue o lado negativo do led (o cátodo, o terminal menor – lado chanfrado) na mesma coluna em que foi ligado o resistor (55 no exemplo ao lado).

Se os terminais não tiverem sido cortados, o lado maior é o positivo.

Imagens: foto do autor

simao@ufpr.br - 2022

1. Ligação de energia

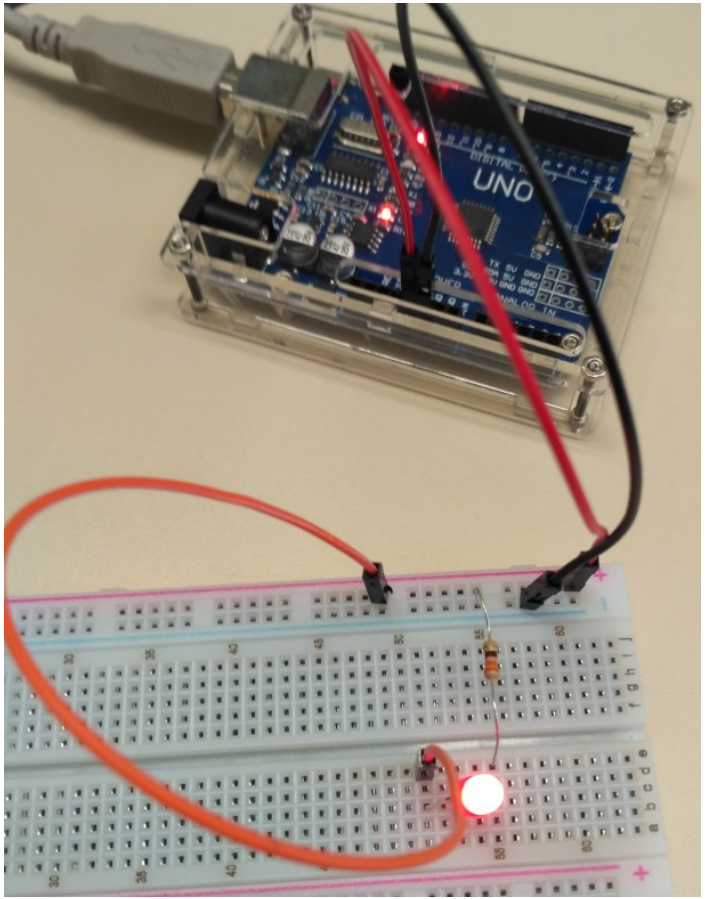


Utilize um fio de ligação para conectar o outro terminal do led (o positivo ou ânodo, terminal mais longo – na figura ao lado ficou na coluna nº 49) até a barra de alimentação positiva (a linha na qual foi conectado o 5V vindo da placa do Arduino).

Imagens: foto do autor

simao@ufpr.br - 2022

1. Ligação de energia



Conecte o cabo USB ao computador.

Não precisa entrar no Ide, somente estamos testando as ligações (e praticando…).

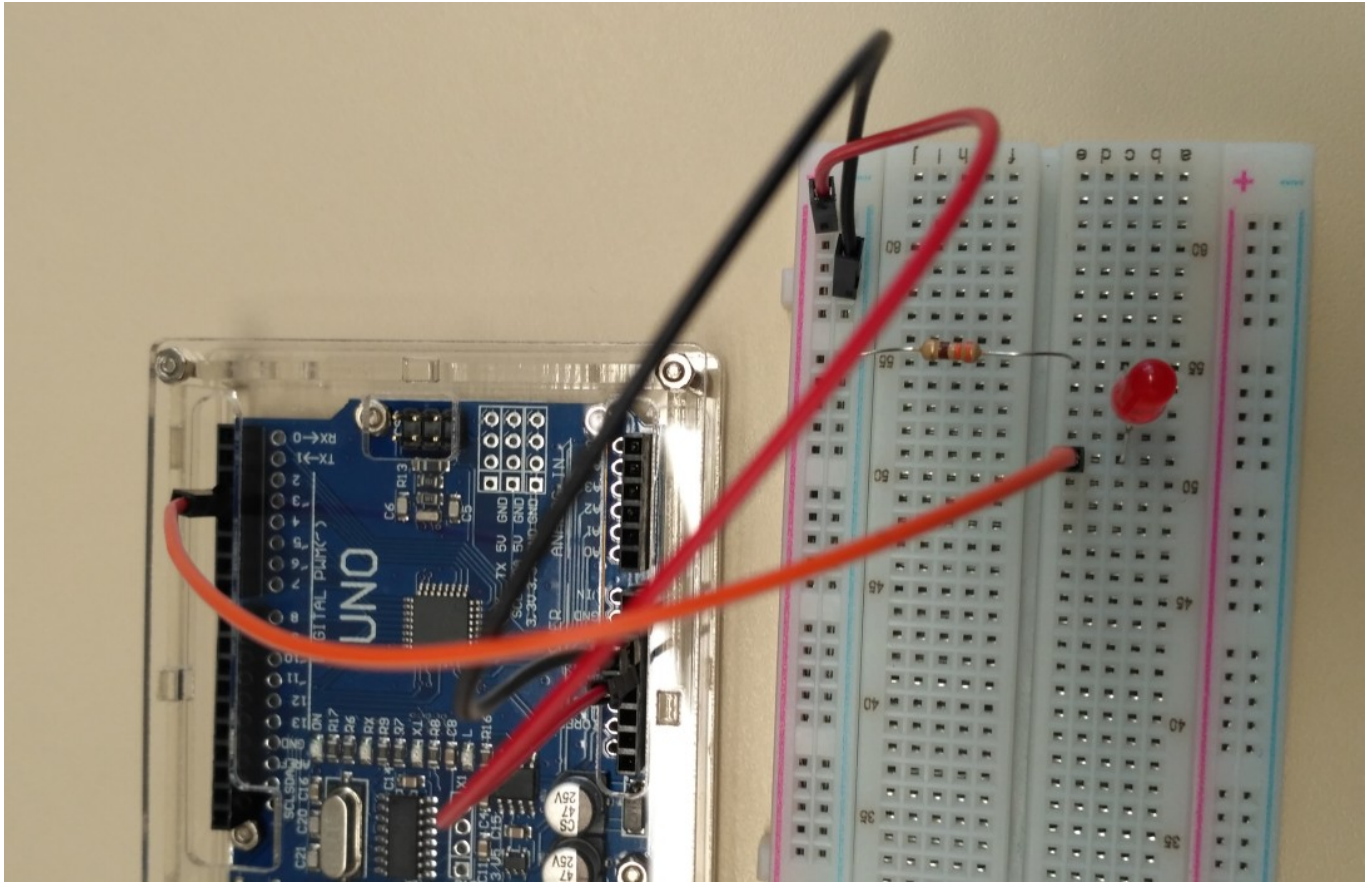
Se não deu certo, desligue o cabo USB e revise as ligações, em especial a do led; troque os fios, eles costumam dar problemas.

Se tudo deu certo, desligue o cabo USB e vá para próxima etapa.

Imagens: foto do autor

simao@ufpr.br - 2022

2. Ligação de comando



Desconecte o cabo USB.

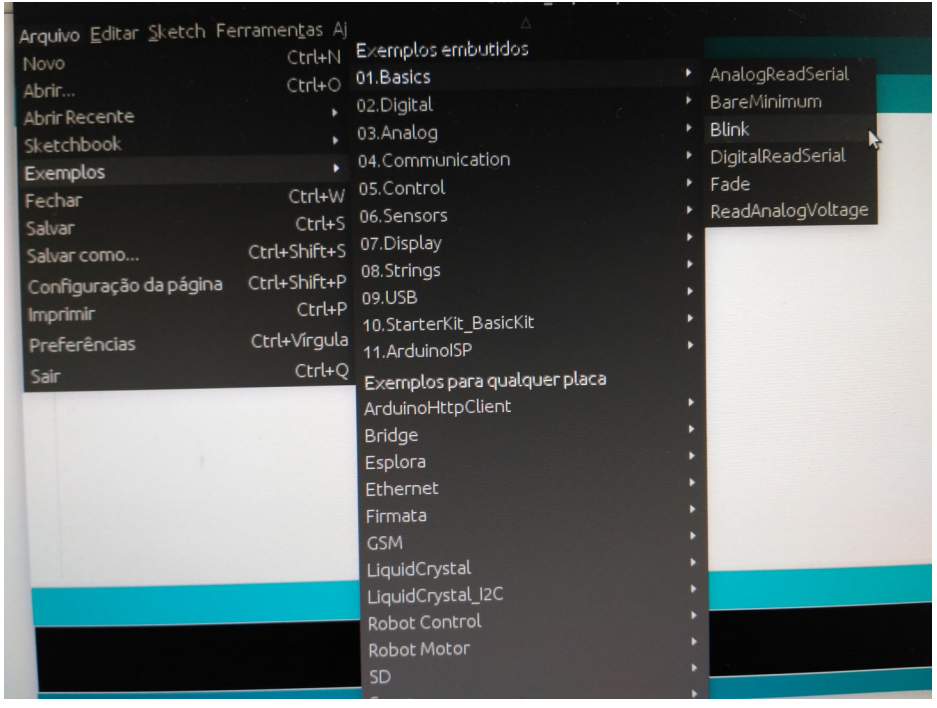
Ligue a conexão led que estava na barra de 5V ao pino de saída digital 3 de seu Arduino.

Imagens: foto do autor

simao@ufpr.br - 2022

3. Codificação

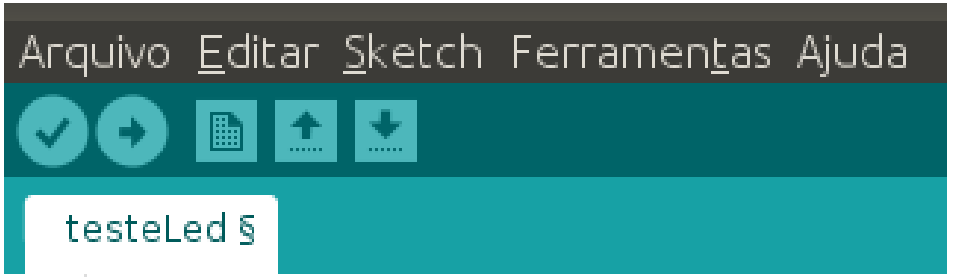
Inicie o IDE e carregue o seu primeiro exemplo, de piscar o led da placa (Arquivo/ Exemplos/ 01.Basics/ Blink)



Imagens: foto do autor

simao@ufpr.br - 2022

3. Codificação



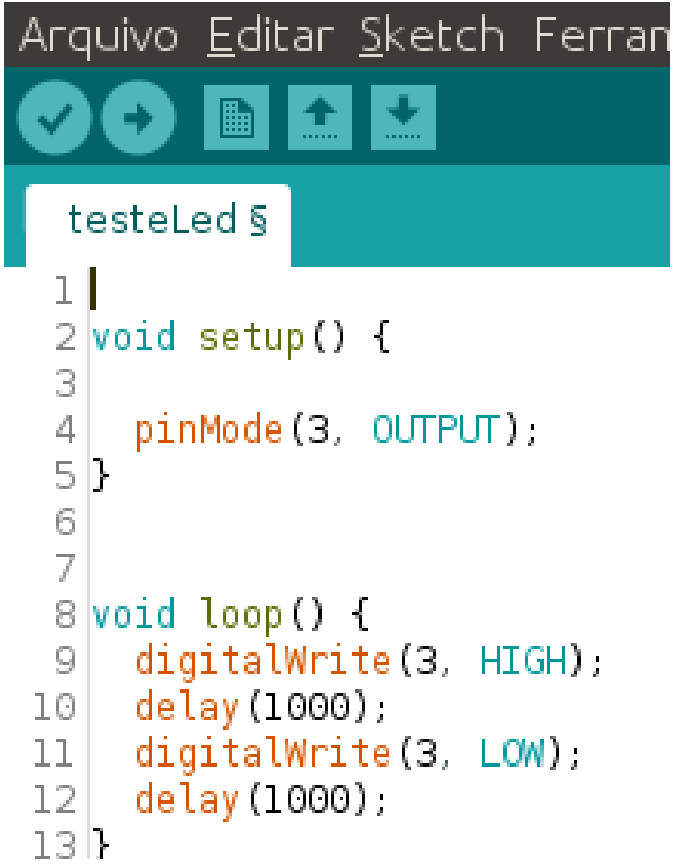

```
1
2  void setup() {
3
4    pinMode(LED_BUILTIN, OUTPUT);
5  }
6
7
8  void loop() {
9    digitalWrite(LED_BUILTIN, HIGH);
10   delay(1000);
11   digitalWrite(LED_BUILTIN, LOW);
12   delay(1000);
13 }
```

Salve o arquivo com outro nome (Arquivo/ Salvar Como…).

No exemplo ao lado todas as linhas de comentários forma removidas para podermos nos concentrar no código.

Imagens: foto do autor

3. Codificação



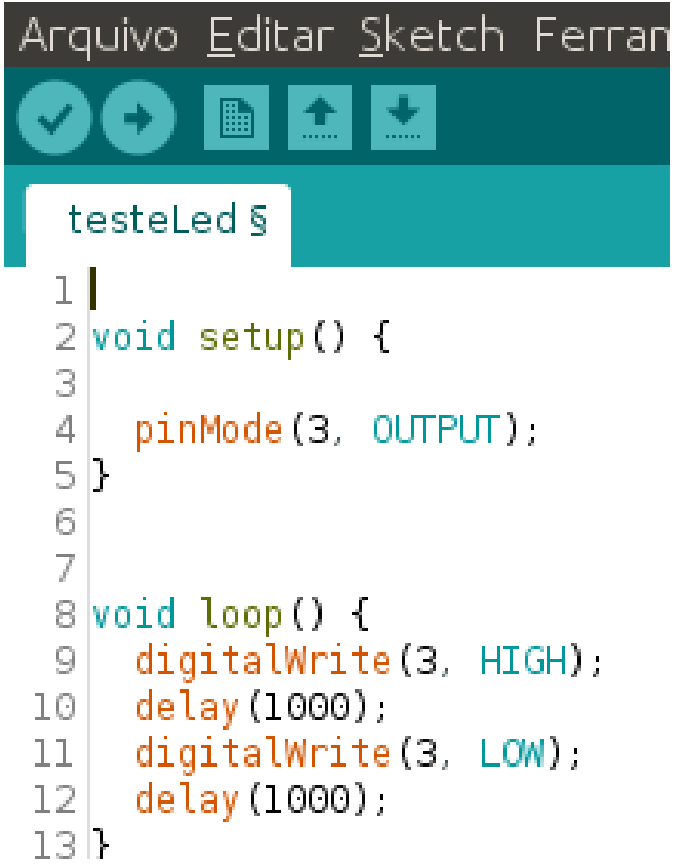
Defina o pino 3 como saída, compile e envie para o Arduino, com a ligação da placa experimental já efetuada e conferida.

Imagens: foto do autor

Parabéns!

Bem vindo ao mundo *maker* do Arduino.

Melhorando o código



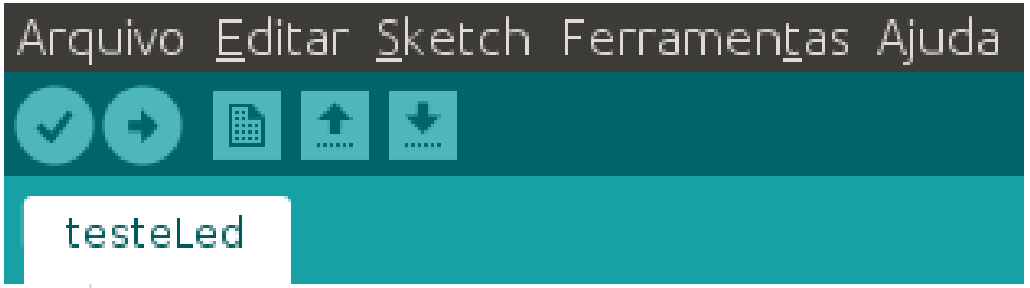
No código original havia uma palavra reservada, LED_BUILTIN, a qual corresponde à porta 13 (um led na placa).

Nós, explicitamente, utilizamos o número da porta que escolhemos, 3.

Mas, podemos também dar um ‘nome’ para nossa porta.

Imagens: foto do autor

Melhorando o código




```
 1 #define nossoled 3
 2 void setup() {
 3
 4   pinMode(nossoled, OUTPUT);
 5 }
 6
 7
 8 void loop() {
 9   digitalWrite(nossoled, HIGH);
10   delay(1000);
11   digitalWrite(nossoled, LOW);
12   delay(1000);
13 }
```

Foi criada uma constante nossoled, e atribuída para ela o valor 3.

Isto foi feito com a declaração *#define*.

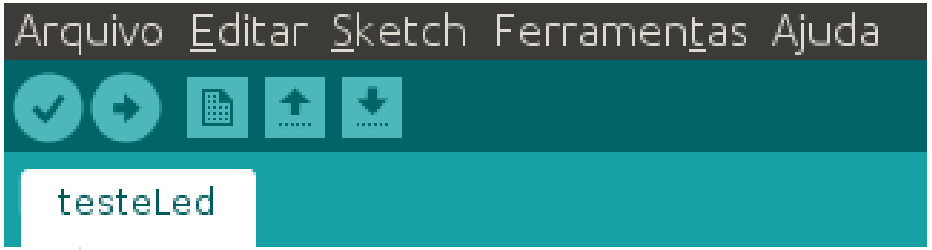
Veja ao lado como ficou.

Teste!

Imagens: foto do autor

simao@ufpr.br - 2022

Agora com dois leds




```
1  #define nossoled 3
2  void setup() {
3
4    pinMode(nossoled, OUTPUT);
5  }
6
7
8  void loop() {
9    digitalWrite(nossoled, HIGH);
10   digitalWrite(LED_BUILTIN, LOW);
11   delay(1000);
12   digitalWrite(nossoled, LOW);
13   digitalWrite(LED_BUILTIN, HIGH);
14   delay(1000);
15 }
```

Com o código ao lado, queremos um led apagando (LOW) e outro acendendo (HIGH), alternadamente.

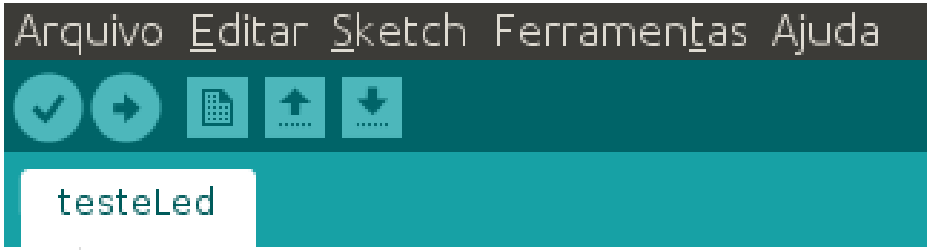
Teste!

Imagens: foto do autor

simao@ufpr.br - 2022

???
Por que não funcionou
???

Agora com dois leds




```
1  #define nossoled 3
2  void setup() {
3
4    pinMode(nossoled, OUTPUT);
5  }
6
7
8  void loop() {
9    digitalWrite(nossoled, HIGH);
10   digitalWrite(LED_BUILTIN, LOW);
11   delay(1000);
12   digitalWrite(nossoled, LOW);
13   digitalWrite(LED_BUILTIN, HIGH);
14   delay(1000);
15 }
```

Definimos que haverá um comando de escrita na saída, na função *loop()*.

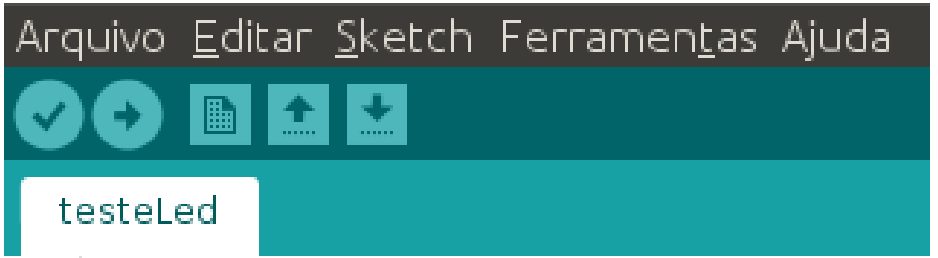
Porém, não definimos na função *setup()* que a saída da placa estaria ligada…

As duas funções funcionam em conjunto...

Imagens: foto do autor

simao@ufpr.br - 2022

Agora com dois leds




```
1  #define nossoled 3
2  void setup() {
3    pinMode(LED_BUILTIN, OUTPUT);
4    pinMode(nossoled, OUTPUT);
5  }
6
7
8  void loop() {
9    digitalWrite(nossoled, HIGH);
10   digitalWrite(LED_BUILTIN, LOW);
11   delay(1000);
12   digitalWrite(nossoled, LOW);
13   digitalWrite(LED_BUILTIN, HIGH);
14   delay(1000);
15 }
```

Acerte o código conforme ao lado.

Teste!

Imagens: foto do autor

HIGH ou LOW ?

- SE você ligar um lado do LED no GND, que é equivalente ao 0V, para que o LED acenda o outro lado deve estar positivo.
  - Portanto, acionamos o LED com o comando HIGH (que corresponde a colocar +5V na saída).
  - … mas, ...

HIGH ou LOW ?

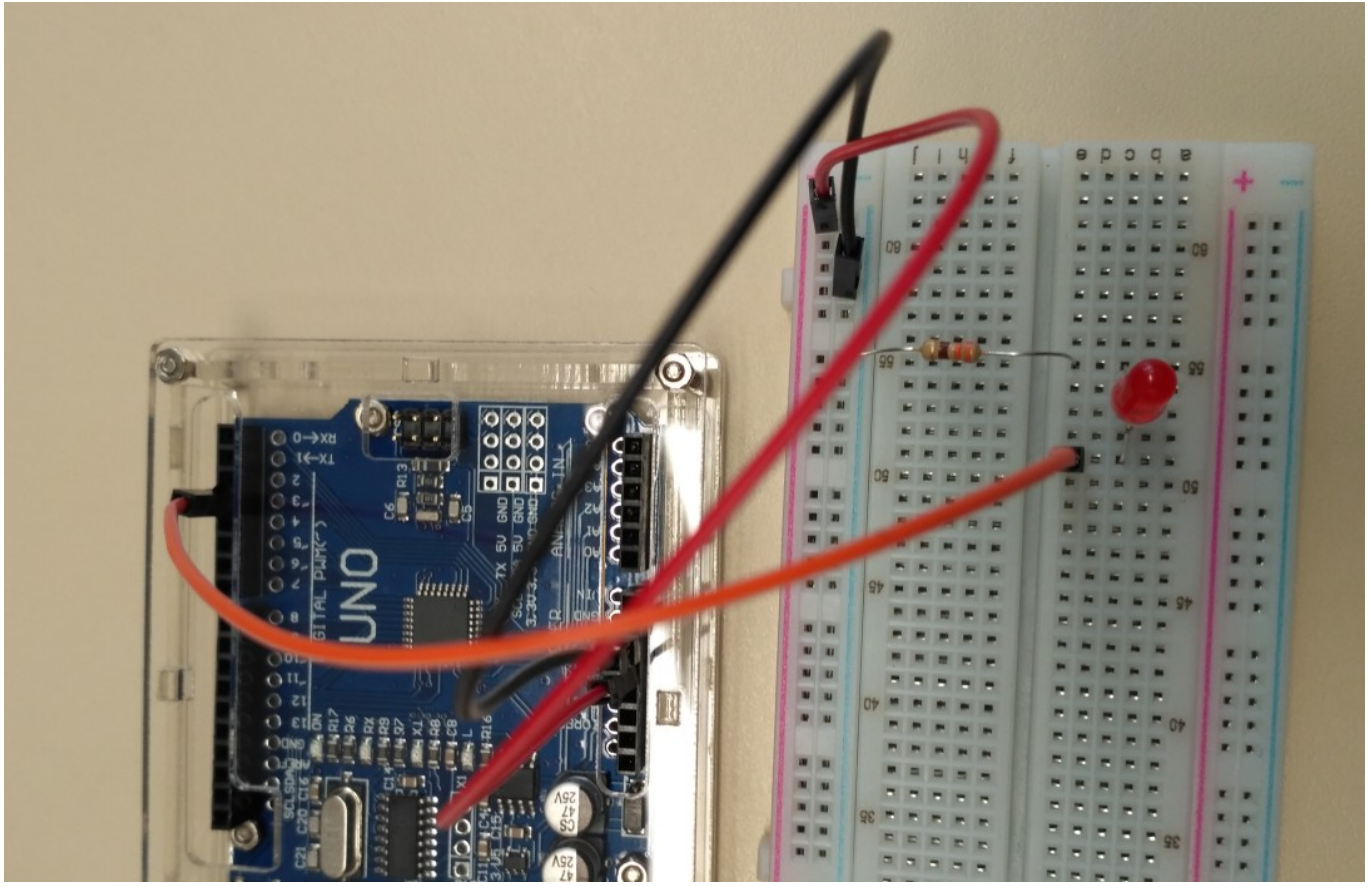
- Também podemos deixar um lado do LED ligado no +5V e ligar o outro lado no Arduino, usando uma lógica negativa:
  - Quando colocarmos um sinal LOW no pino em que foi ligado o LED ele acenderá.
- Usar a lógica positiva ou a negativa é indiferente, depende de com qual você se sente mais à vontade.

Escrita PWM

PWM

- PWM significa "*Pulse Width Modulation*"
- A Modulação por Largura de Pulso possibilita que controlemos quanto de energia (0 a 100%) será fornecida dentro de um pulso (qual será sua largura).
- Vamos aproveitar nosso protótipo para testar esta funcionalidade.

PWM



Usaremos a mesma conexão.

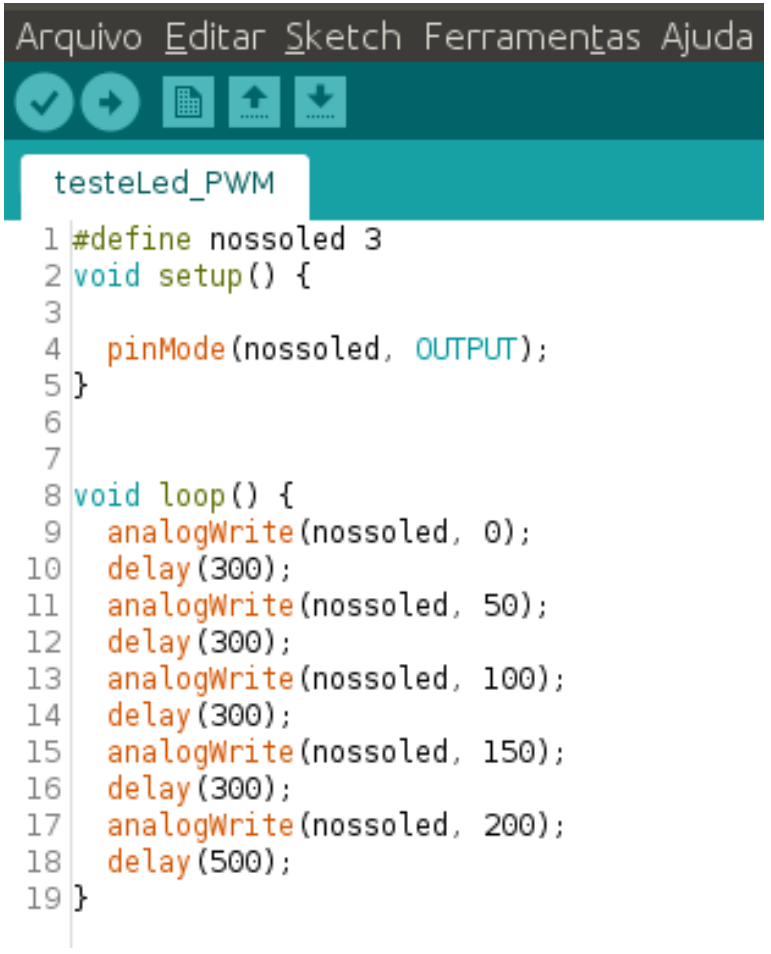
Não precisa desconectar.

Salve seu programa com outro nome.

Imagens: foto do autor

simao@ufpr.br - 2022

PWM



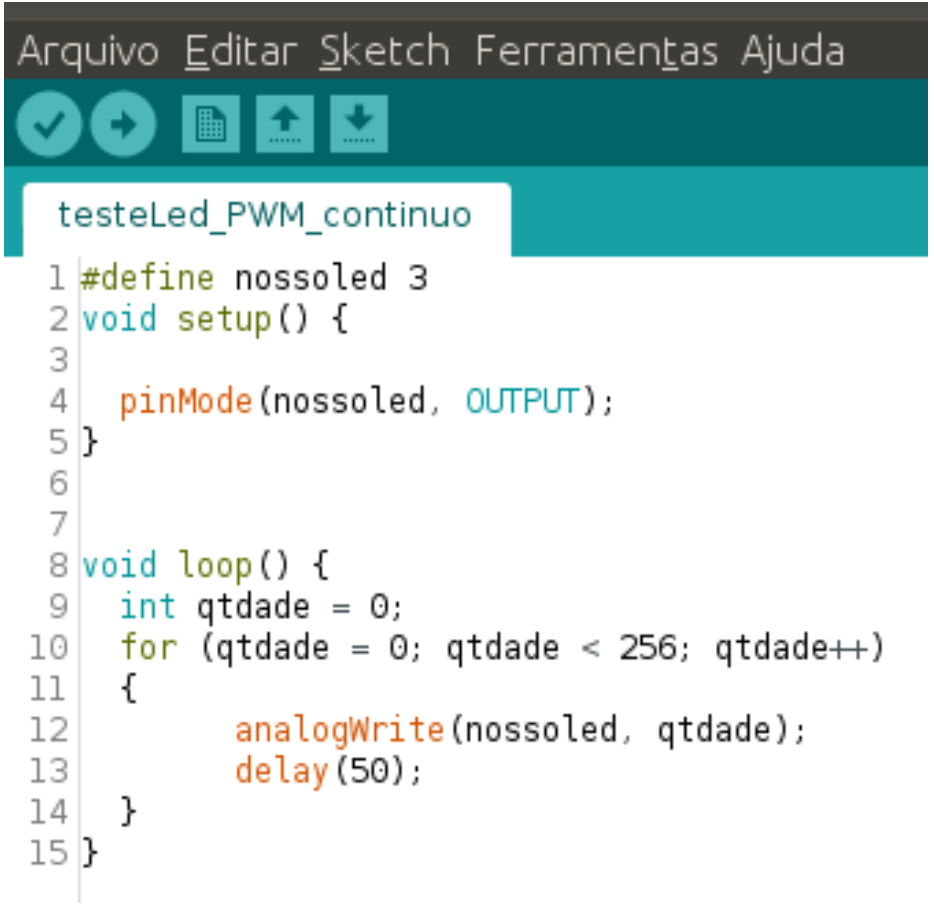
A função de escrita agora é a *analogWrite(porta, valor)*.

O valor é a quantidade do ciclo de trabalho PWM que ficará ativo (em nível HIGH).

Teste!

Imagens: foto do autor

PWM



Agora transformamos o valor PWM em uma variável inteira, chamada ‘qtdade’.

E colocamos um loop controlado que varia o valor armazenado em ‘qtdade’ de 0 a 255, continuamente.

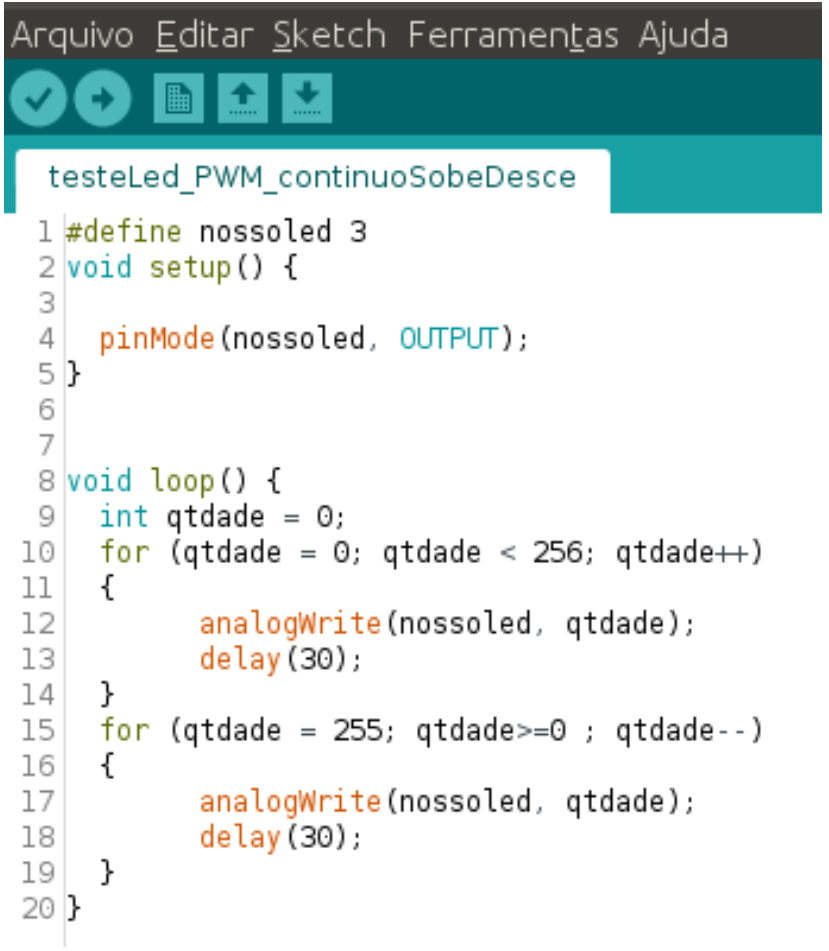
Teste!

(obs.: qtdade++ significa some um ao valor atual da variável qtdade).

Imagens: foto do autor

simao@ufpr.br - 2022

PWM



Agora o código aumentará o valor até o máximo e depois diminuirá.

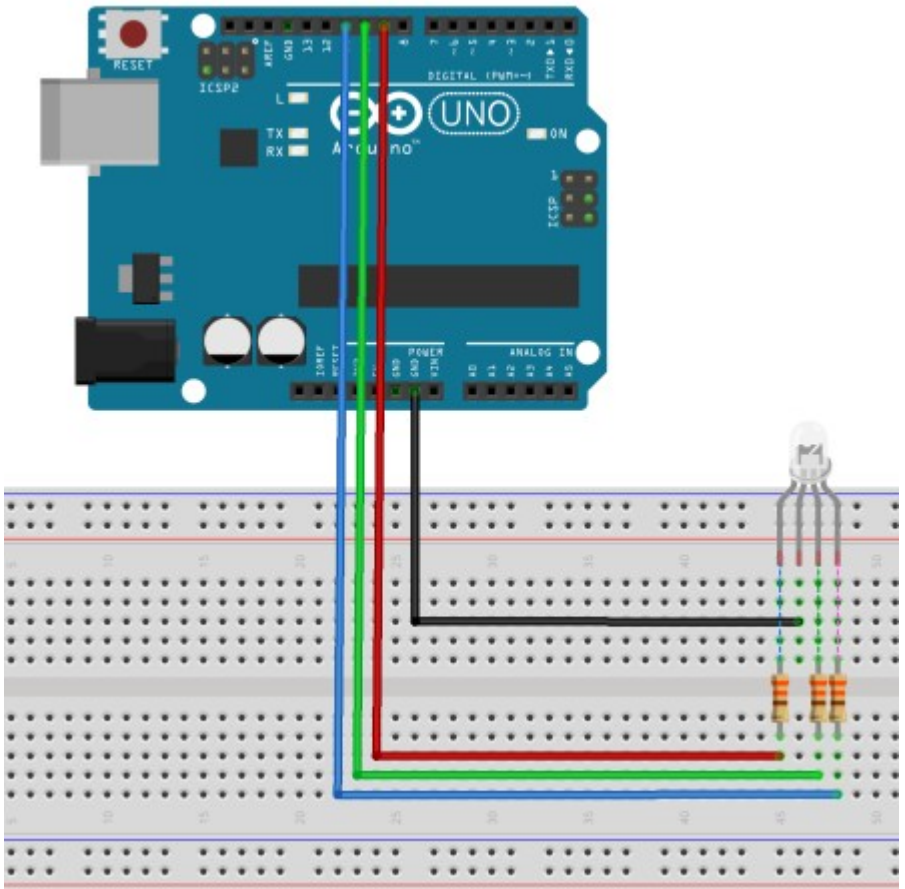
Teste!

Experimente comentar as linhas nas quais ocorre o delay(30) e/ ou alterar seu valor e veja o resultado. Ajuste da forma que ficar melhor para você.

(Para comentar, coloque um '//' antes do comando, ele ficará cinzento e passará a ser um comentário – ou seja, não será funcional).

Imagens: foto do autor

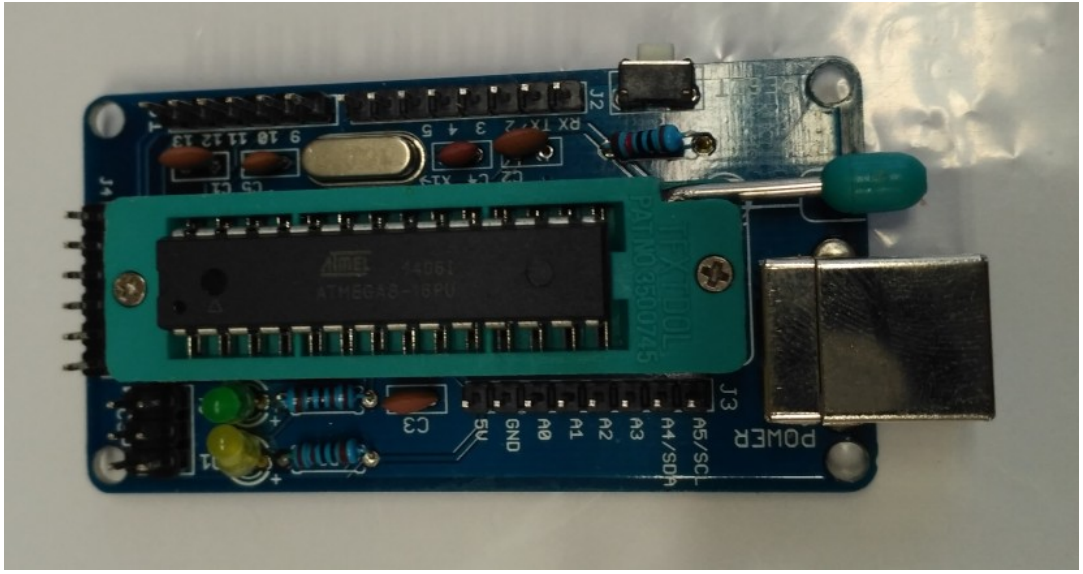
LED RGB



```
#define vermelho 9
#define verde 10
#define azul 11

int aleatorio;

void setup() {
  Serial.begin(9600);
  pinMode(vermelho, OUTPUT);
  pinMode(verde, OUTPUT);
  pinMode(azul, OUTPUT);
  randomSeed(analogRead(0));
}

void loop() {
  aleatorio = random(256);
  analogWrite(vermelho, aleatorio);
  delay(100);
  aleatorio = random(256);
  analogWrite(verde, aleatorio);
  delay(100);
  aleatorio = random(256);
  analogWrite(azul, aleatorio);
  delay(100);
}
```

Terminou seu protótipo?

Gravação do código

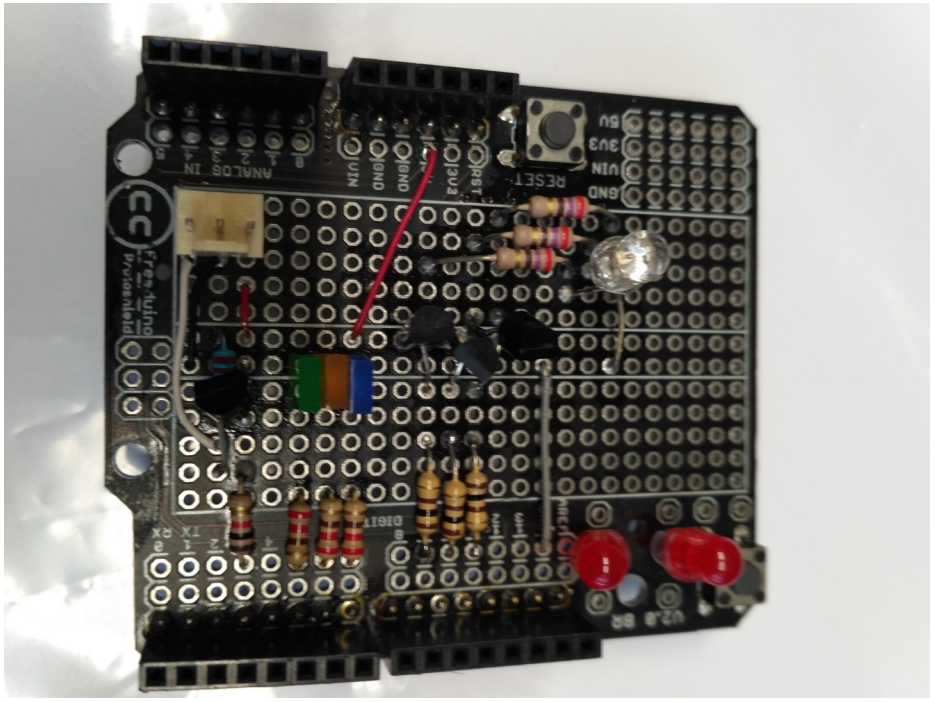


Pode gravá-lo em um microcontrolador compatível, que será levado para sua placa definitiva, ou seu MVP!

Imagem: foto do autor

simao@ufpr.br - 2022

Componentes



Também pode produzir uma versão em placas com solda, de forma que não ficará com fios soltos e será mais prático de demonstrar.

Imagem: foto do autor

simao@ufpr.br - 2022

O que você aprendeu, que será necessário para os próximos passos:

1. Ligar componentes externos à placa do Arduino
2. Utilizar uma saída digital com escrita digital
3. Utilizar uma saída digital com escrita analógica

simao@ufpr.br - 2022

Parabéns!

simao@ufpr.br - 2022