



RESISTORES

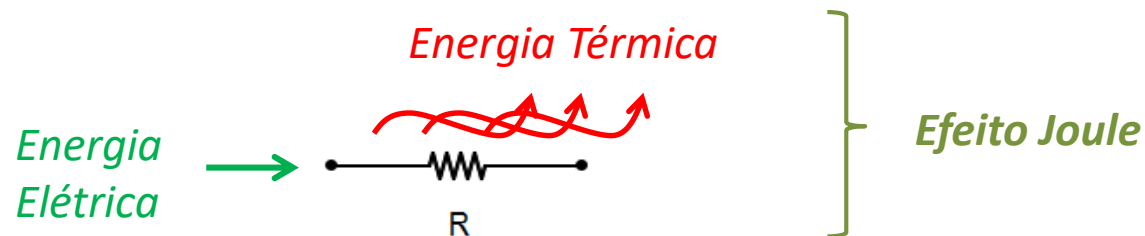


Resistores são dispositivos que oferecem uma resistência a passagem da corrente elétrica. A função do resistor é converter a energia elétrica em energia térmica, este efeito de conversão de energia é chamado de efeito Joule.

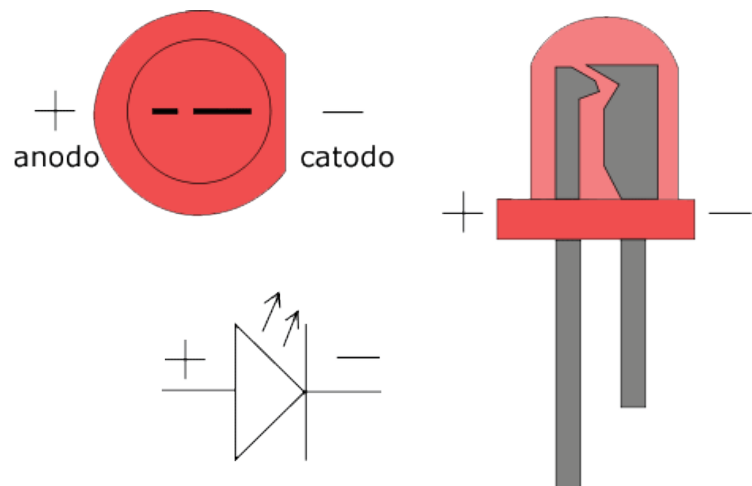
<http://nerdeletrico.blogspot.com/2011/03/resistores-uma-visao-basica.html>



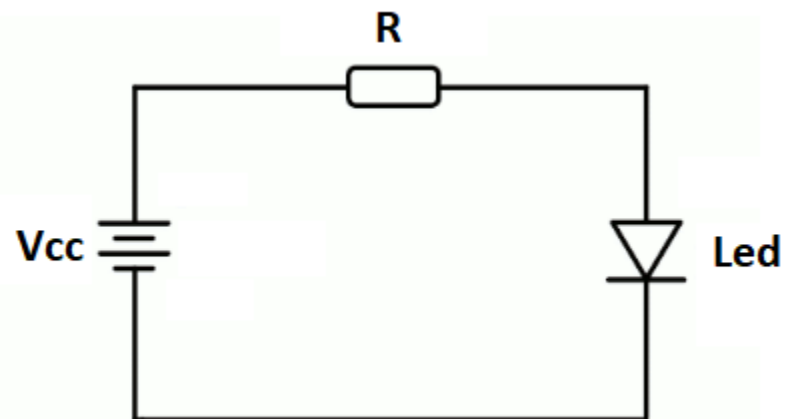
Efeito Joule



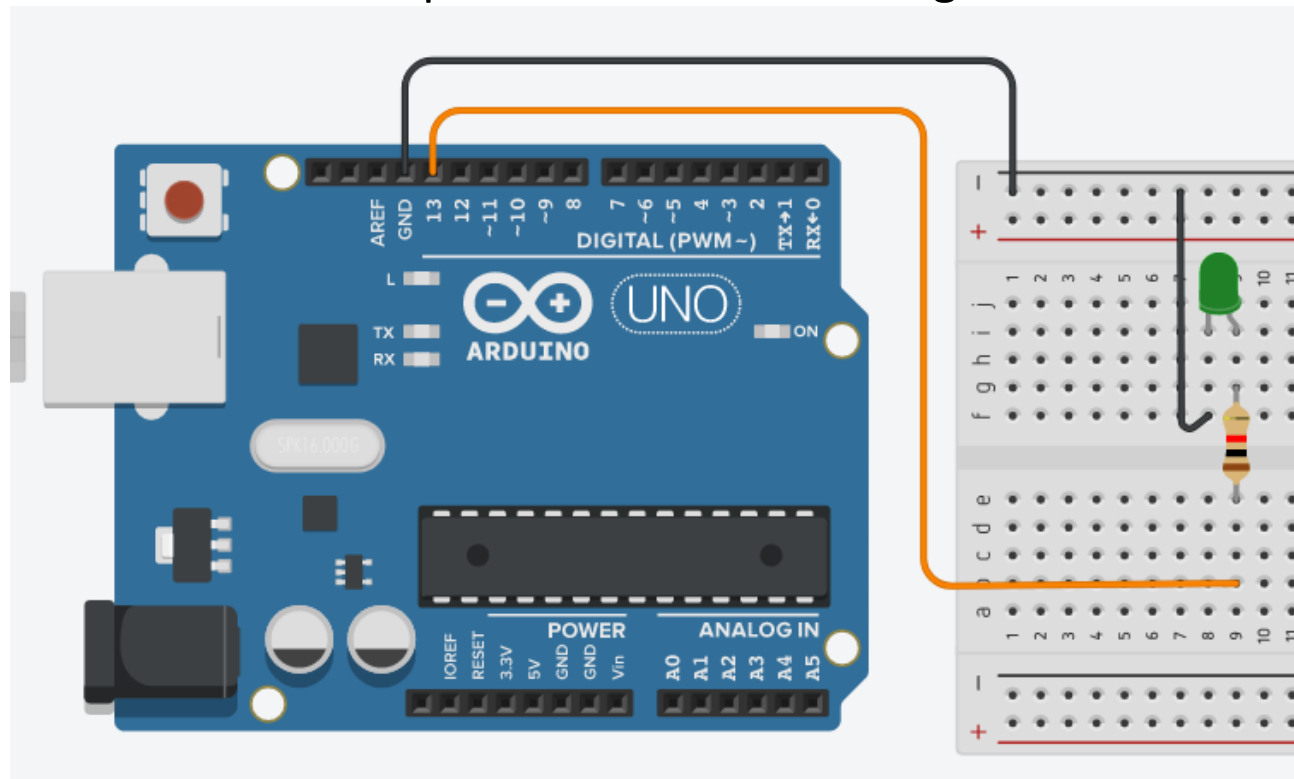
LED - Light Emitting Diode



<https://blog.fazedores.com/guia-fazedores-led/projeto-semaforo-led/>

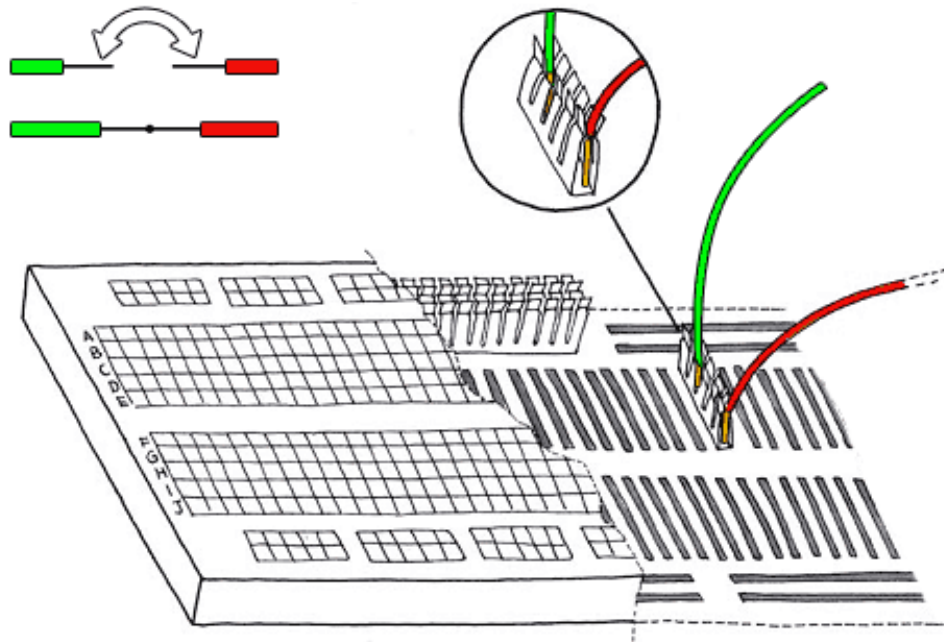


Leds são semicondutores que transformam a energia elétrica em luz.

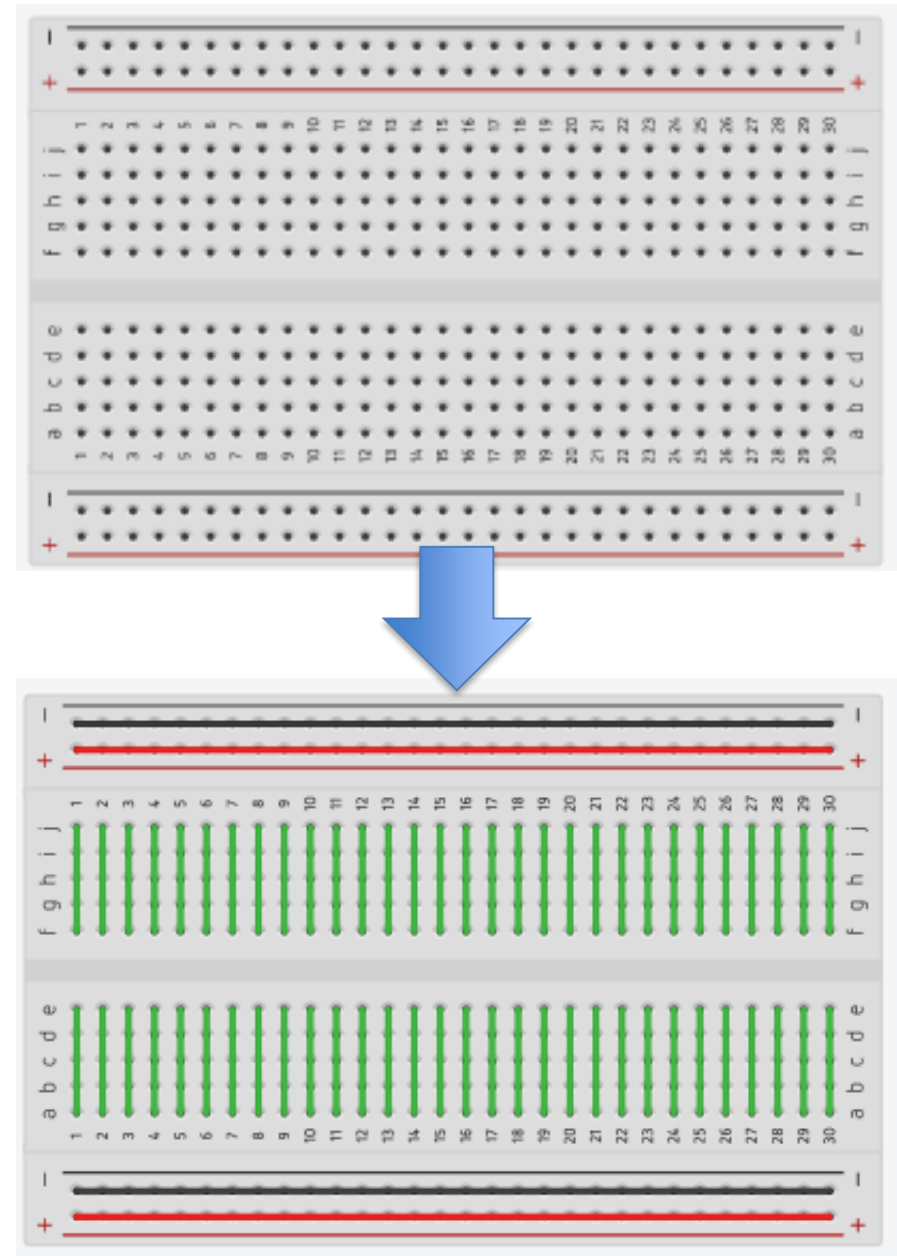


PROTOBOARD

É um dispositivo utilizado para realizar a conexão de dispositivos elétricos e consequentemente a elaboração de circuitos elétricos para testes.

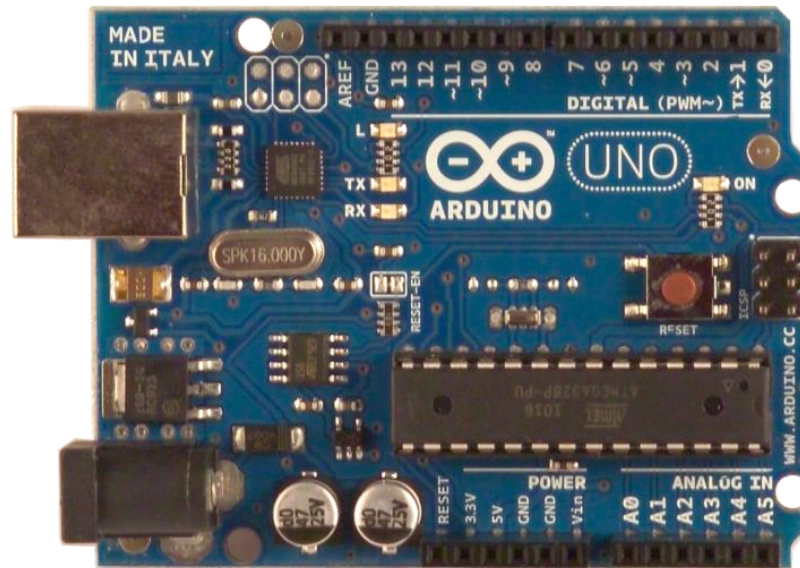


<http://www.boscojr.com/arquitetura/lab1.html>



OBJETIVO

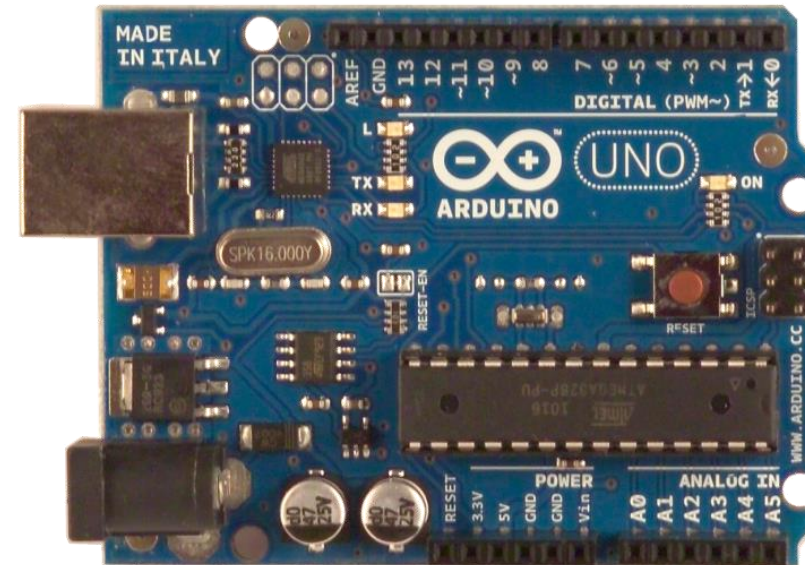
- Definição de pinos digitais como entrada e saída;
- Manusear os pinos digitais do arduino;
- Função delay.



ARDUINO

O Arduino UNO é uma placa baseada no microcontrolador ATmega168 ou ATmega328 e possui:

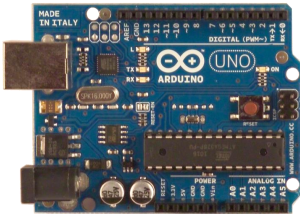
- 14 pinos de entrada ou saída digital;
 - ❑ 6 podem ser utilizados como saídas PWM
- 6 entradas analógicas;
- USB;
- Alimentação (7V-12V);
- Botão de reset.



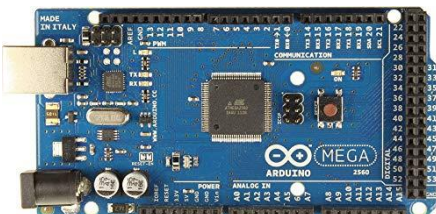
TIPOS DE ARDUINO

	UNO	MEGA 2560	LEONARDO	DUE	ADK	NANO	PRO MINI	ESPLORA
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32K (0,5K usado pelo bootloader)	256K (8K usado pelo bootloader)	32K (4K usado pelo bootloader)	512K disponível para aplicações	256K (8K usado pelo bootloader)	16K (ATmega168) ou 32K (ATmega328) (bootloader: 2K)	16K (2K usado pelo bootloader)	32K (4K usado pelo bootloader)
Clock	16Mhz	16Mhz	16Mhz	84Mhz	16Mhz	16Mhz	8Mhz (modelo 3.3v) ou 16Mhz (modelo 5v)	16Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial/Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5V	5V	5V	3.3V	5V	5V	3.3 ou 5V, dependendo do modelo	5V
Corrente máxima portas E/S	40mA	40mA	40mA	130mA	40mA	40mA	40mA	-
Alimentação	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	3.3-12V (modelo 3.3v) ou 5-12V (modelo 5v)	5V

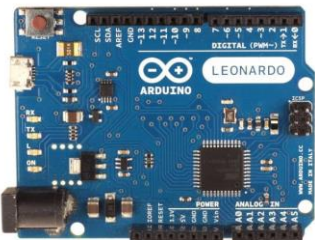
UNO



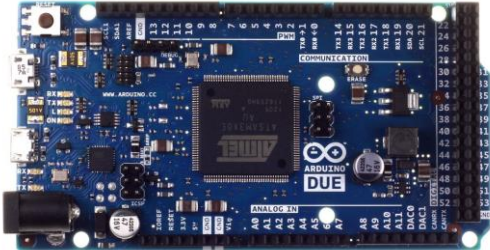
MEGA



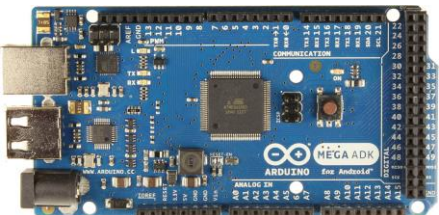
LEONARDO



DUE



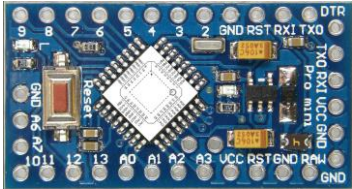
ADK



NANO



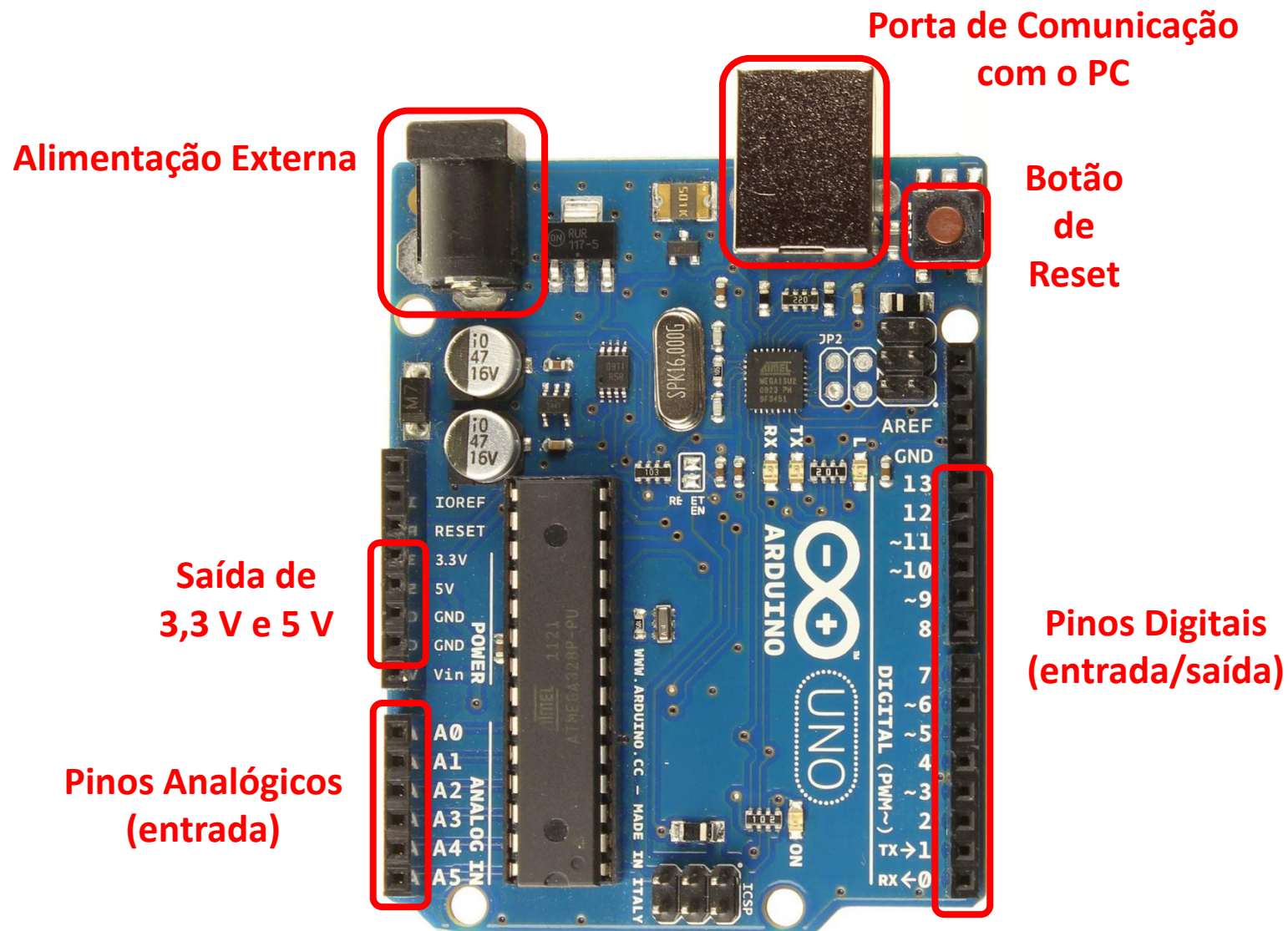
PRO MINI



ESPLORA



ESTRUTURA DO ARDUINO



SHIELDS

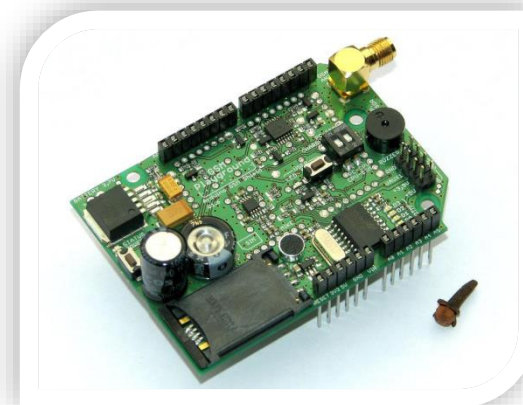
Ethernet



LCD



GPRS / GSM



Wi-Fi



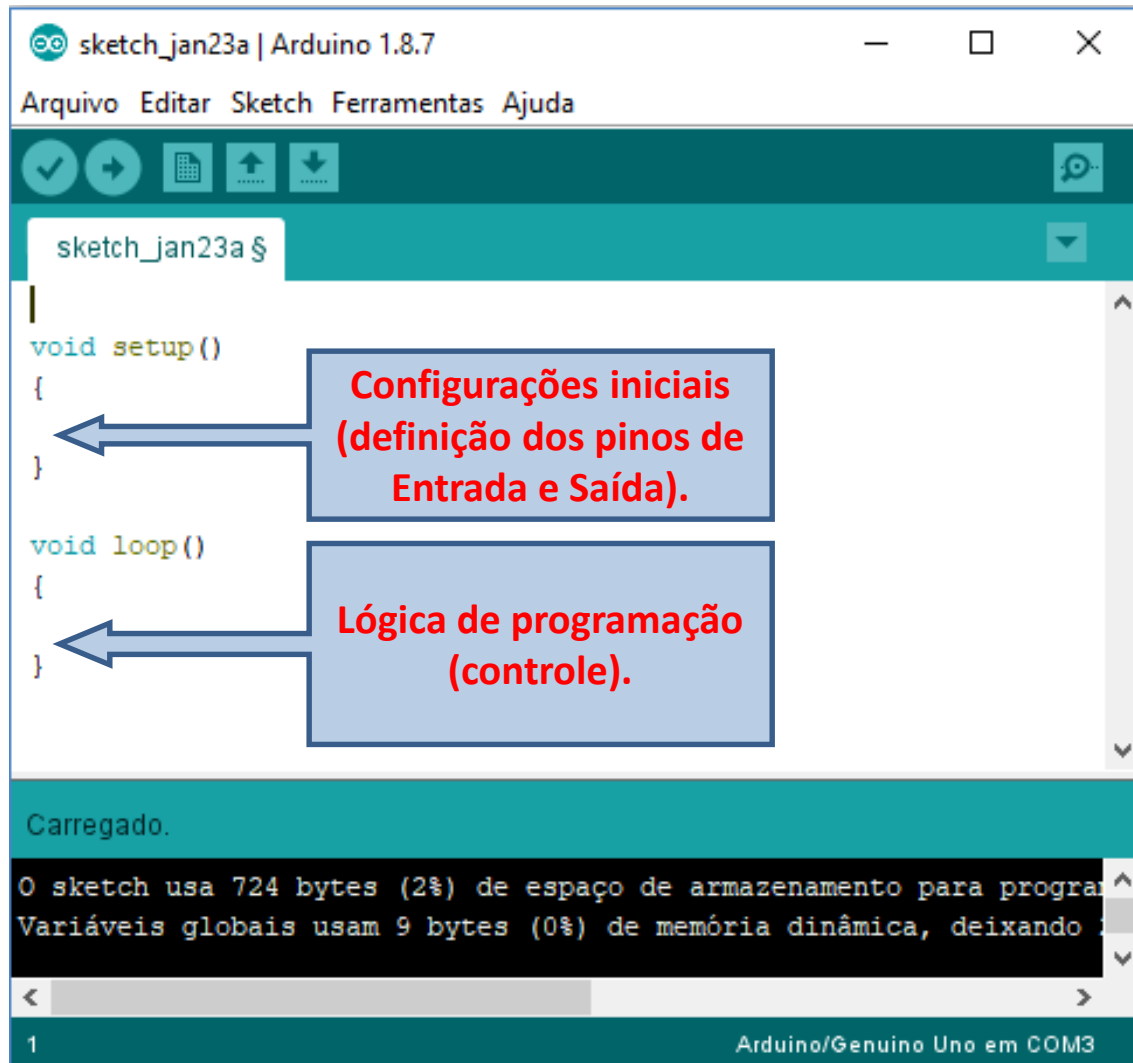
Wave



Motor



ESTRUTURA DA PROGRAMAÇÃO



CONSTANTES:

- **INPUT**: define o status do pino como entrada.
- **OUTPUT**: define o status do pino como saída.
- **LOW**: define a saída do pino como nível lógico “zero” (0 V).
- **HIGH**: define a saída do pino como nível lógico “um” (5 V).

SÍMBOLOS:

- **;**: deve ser inserido em todo final de linha de código.
- **{ }**: são blocos de comandos;
- **// ou /* */**: servem para inserir comentários.

SAÍDAS DIGITAIS

Configuração da porta como saída digital

`pinMode (Y, K);`

Pino de saída

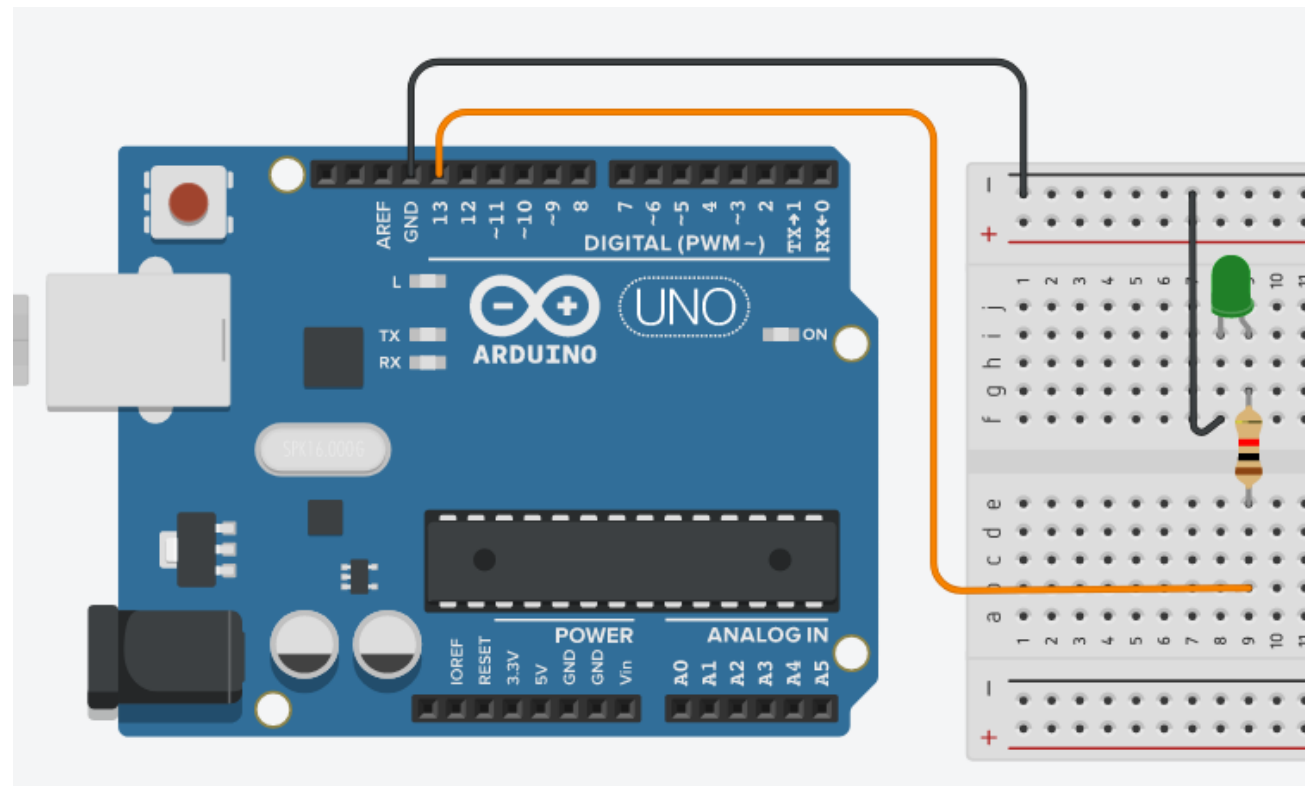
Saída (OUTPUT)

Acionamento da saída digital

`digitalWrite (Y, N);`

Pino de saída

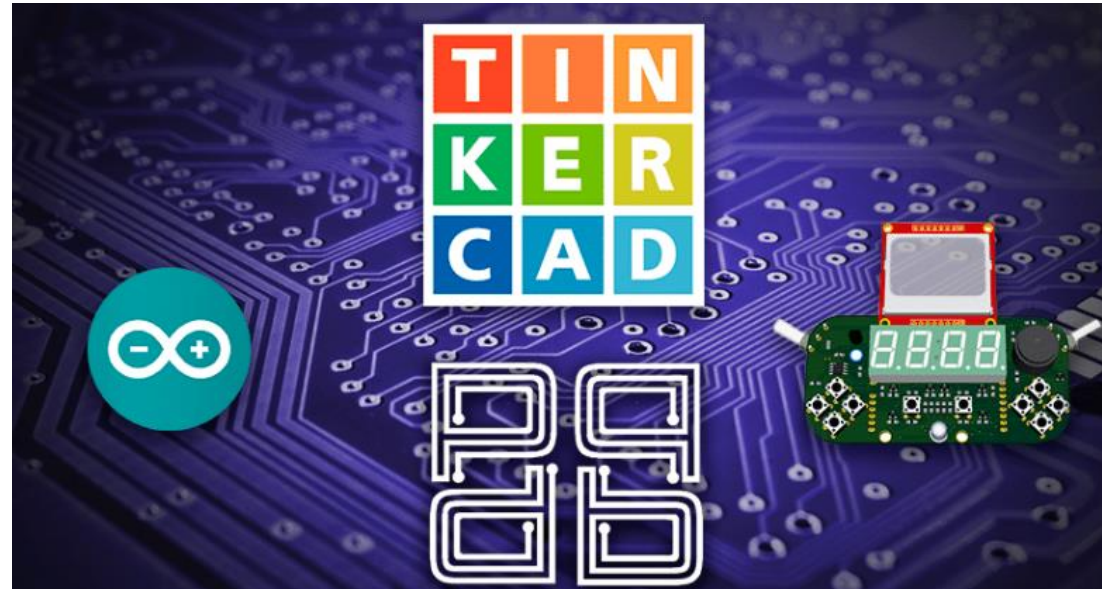
Nível da saída



MATERIAL

- (1) Protoboard;
- (1) Arduino;
- (1) Cabo de Comunicação Arduino;
- (3) Resistor de 1 k Ω de 1/4 W;
- (3) Led.

Tinkercad: ferramenta online e gratuita de simulação de circuitos elétricos



O Tinkercad é uma ferramenta online de design de modelos 3D em CAD e também de simulação de circuitos elétricos analógicos e digitais, desenvolvida pela Autodesk. Por ser gratuito e fácil de usar, encontramos nele uma oportunidade de ensino de Programação Embarcada e seus derivados.

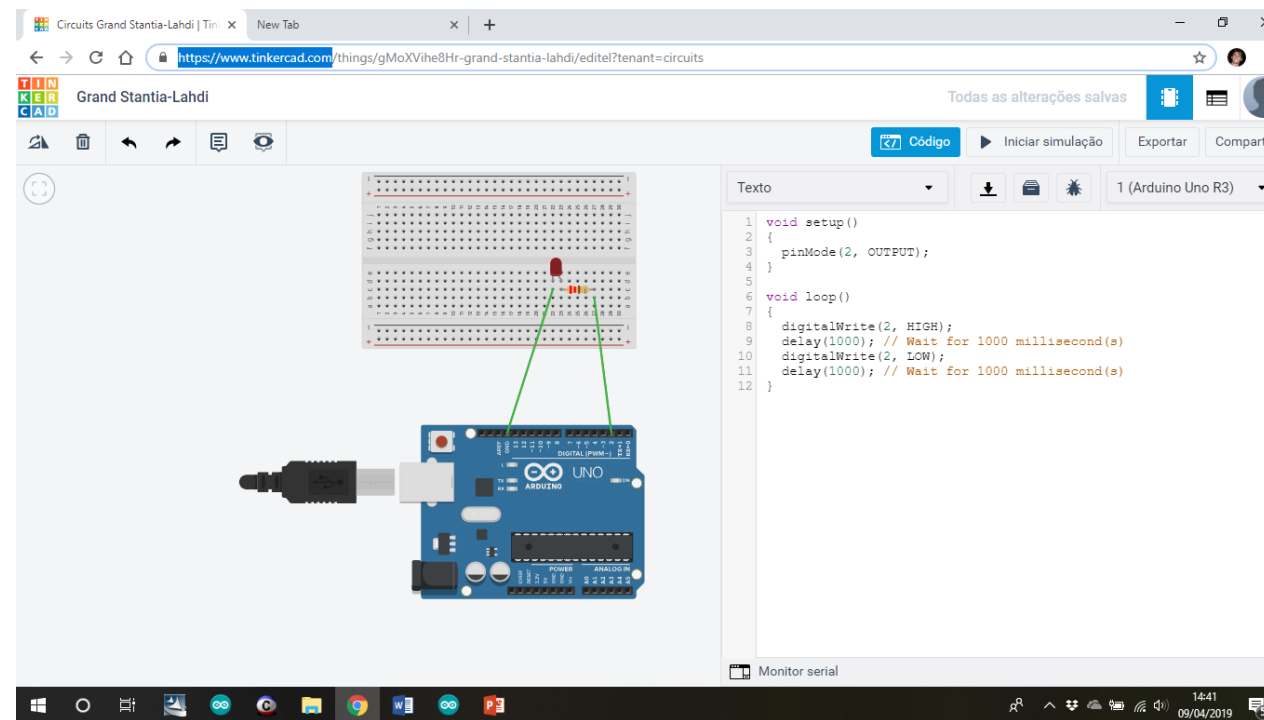
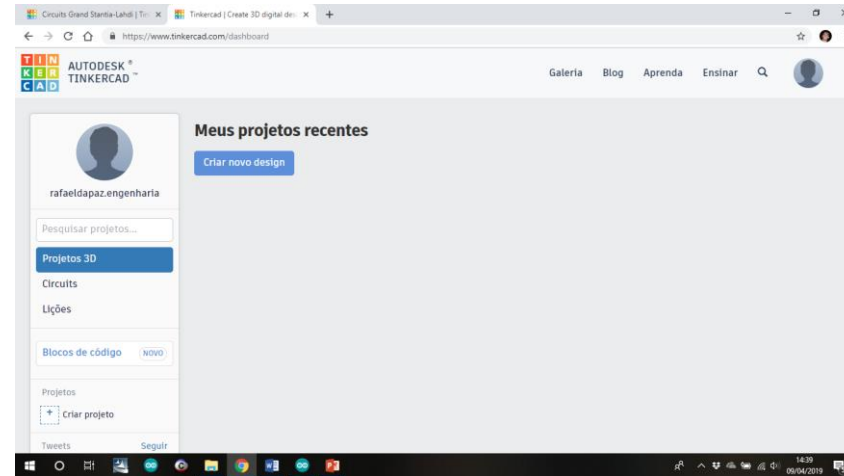
<https://www.tinkercad.com>

INSCRIÇÃO NO TINKERCAD

The image shows the Tinkercad website interface with several elements and navigation arrows:

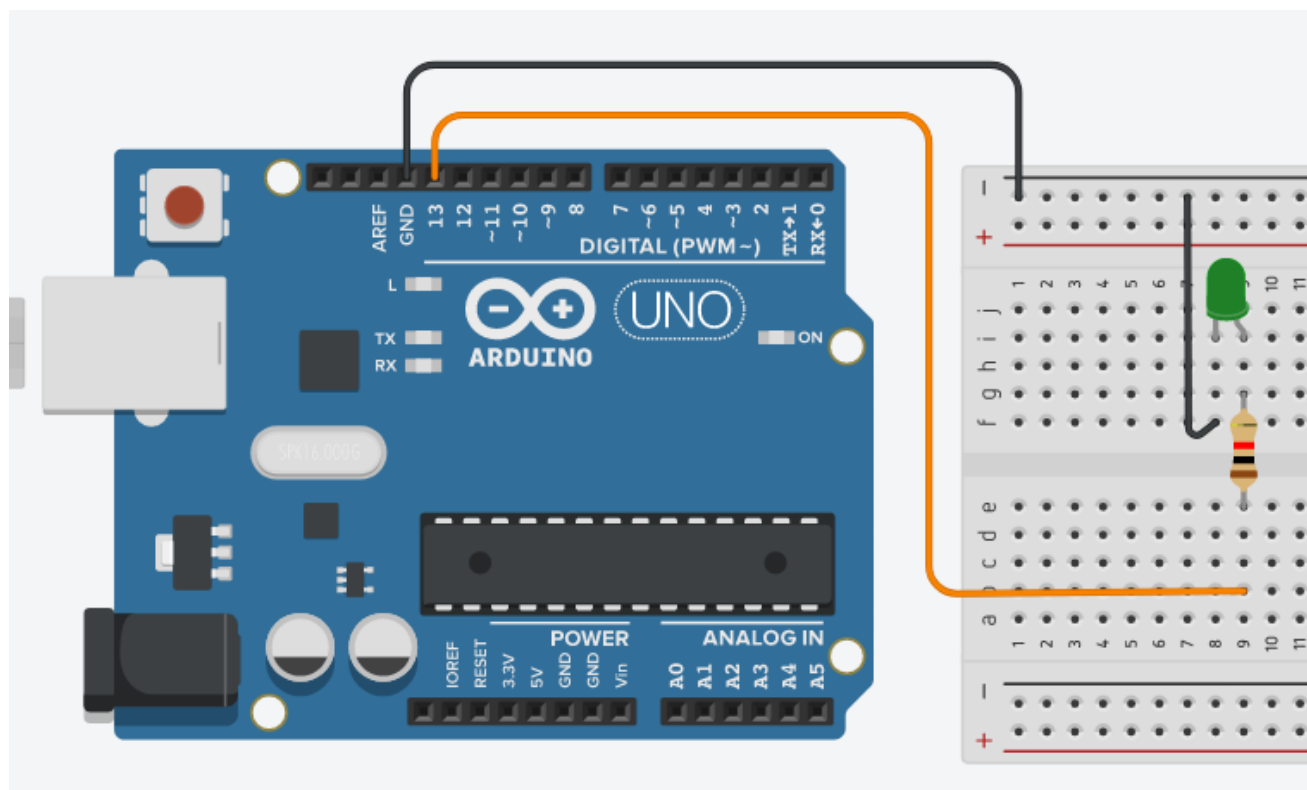
- Top Navigation:** Includes the Tinkercad logo (TIN KER CAD), the text "AUTODESK TINKERCAD", and links for "Galeria", "Blog", "Aprenda", "Ensinar", "Entrar", and a button "INSCREVA-SE AGORA" (highlighted by a red arrow).
- URL:** The address bar shows <https://www.tinkercad.com>.
- Left Sidebar:** Contains a search bar "Pesquisar projetos...", a list of categories ("Projetos 3D", "Circuitos" - highlighted by a red arrow), "Blocos de código" (marked "NOVO"), and "Lições".
- Registration Flow (Indicated by Red Arrows):**
 - The "INSCREVA-SE AGORA" button leads to the "Iniciar edição" screen.
 - The "Iniciar edição" screen asks "Como você criará sua conta?" and offers options: "Entrar com e-mail", "Entrar com o Google", "Fazer login com a Apple", and "Mais opções de login...".
 - From the "Iniciar edição" screen, a red arrow points to the "Circuitos" category in the sidebar.
 - Another red arrow points from the "Iniciar edição" screen to a second "Iniciar edição" screen.
 - The second "Iniciar edição" screen asks "Como você usará o Tinkercad?" and offers options: "Os educadores começam aqui", "Alunos, entrem em uma turma", and "Criar uma conta pessoal" (highlighted by a red arrow).
 - Below these options, it asks "Já tem uma conta?" with an "Entrar" button.
- Bottom Right:** Features the "Facens" logo.

<https://www.tinkercad.com>

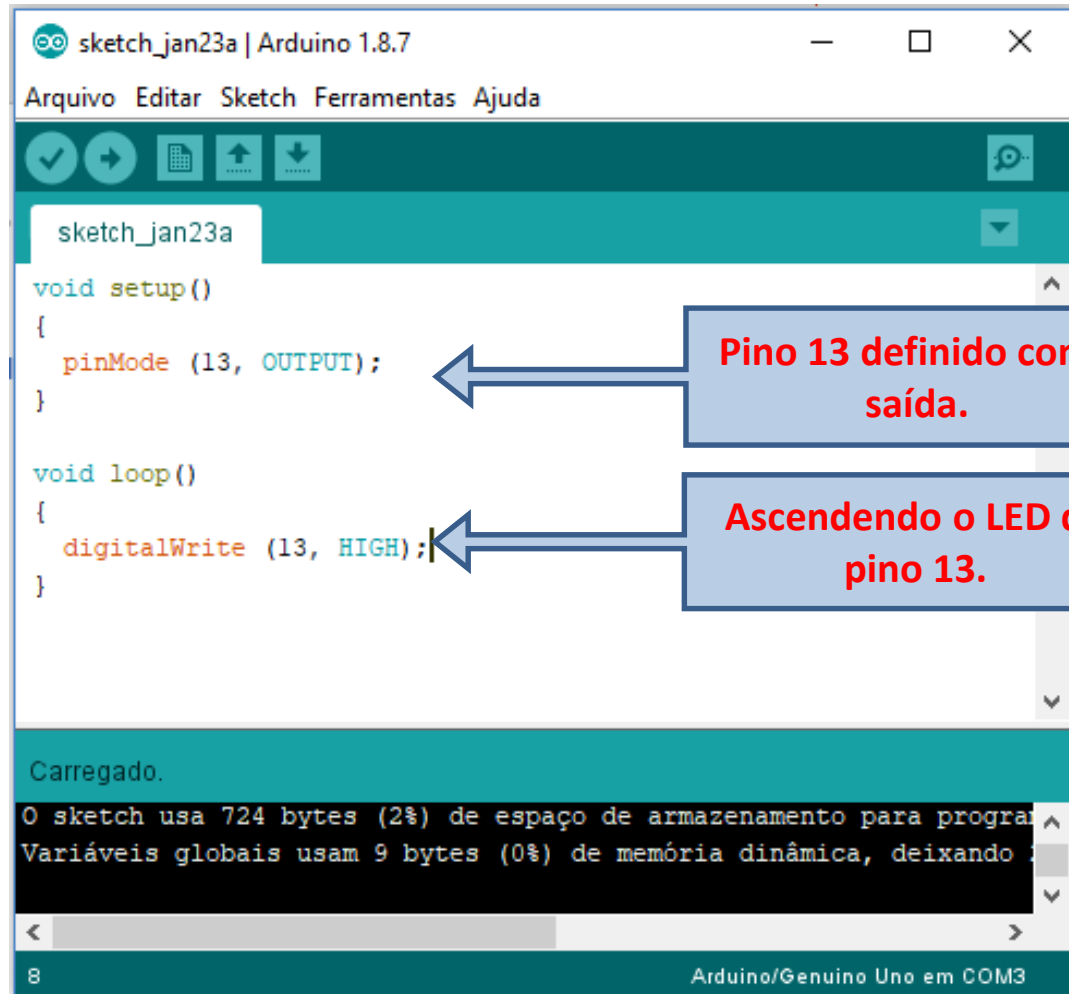


PRIMEIRO PROGRAMA

Ligando o LED que está no pino 13 do arduino



PRIMEIRO PROGRAMA



```
sketch_jan23a | Arduino 1.8.7
Arquivo  Editar  Sketch  Ferramentas  Ajuda

sketch_jan23a
void setup()
{
  pinMode (13, OUTPUT);

void loop()
{
  digitalWrite (13, HIGH);
}
```

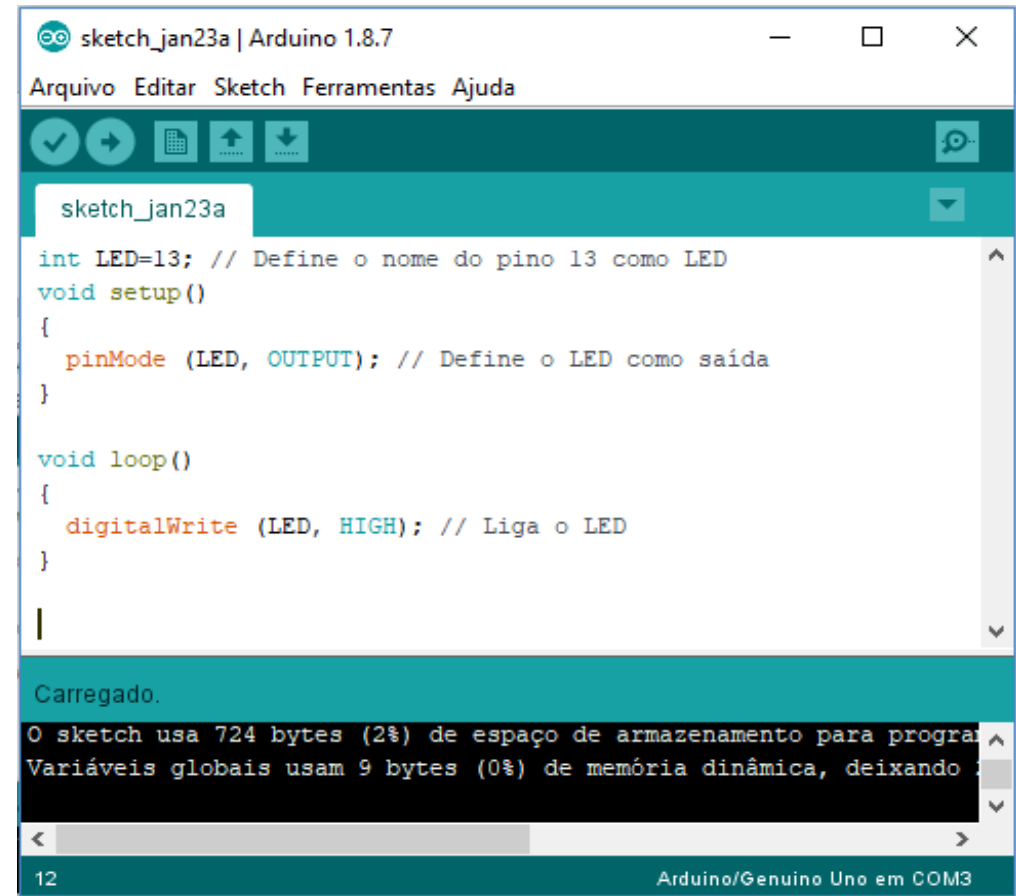
Pino 13 definido como saída.

Ascendendo o LED do pino 13.

Carregado.

O sketch usa 724 bytes (2%) de espaço de armazenamento para programar.
Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 1023 bytes disponíveis.

8 Arduino/Genuino Uno em COM3



```
sketch_jan23a | Arduino 1.8.7
Arquivo  Editar  Sketch  Ferramentas  Ajuda

sketch_jan23a
int LED=13; // Define o nome do pino 13 como LED
void setup()
{
  pinMode (LED, OUTPUT); // Define o LED como saída
}

void loop()
{
  digitalWrite (LED, HIGH); // Liga o LED
}

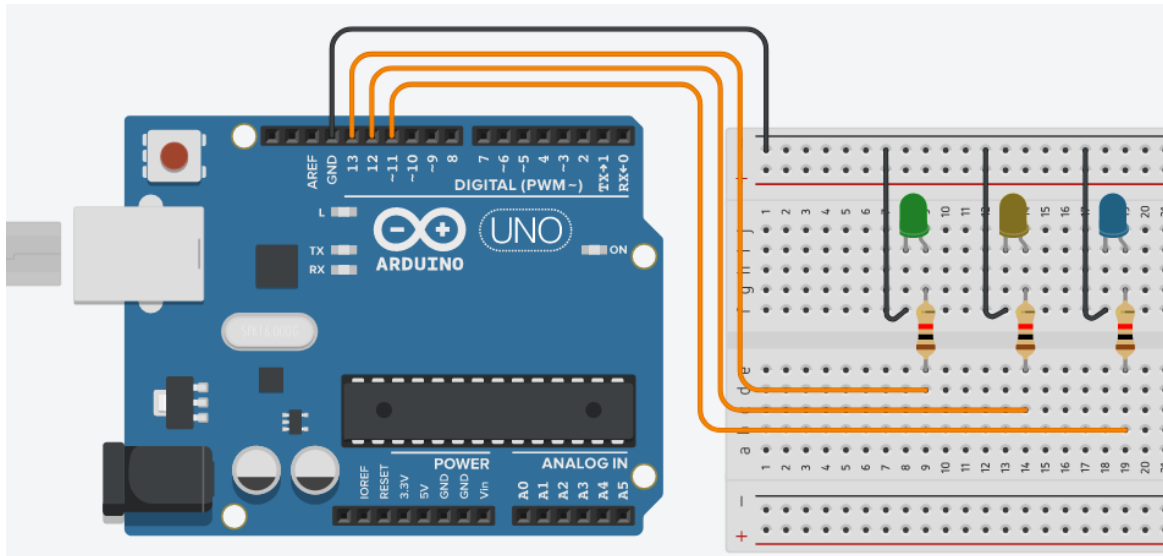
Carregado.
```

O sketch usa 724 bytes (2%) de espaço de armazenamento para programar.
Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 1023 bytes disponíveis.

12 Arduino/Genuino Uno em COM3

SEGUNDO PROGRAMA

Configurar os pinos 11, 12 e 13 como saídas e mantê-los acionados.

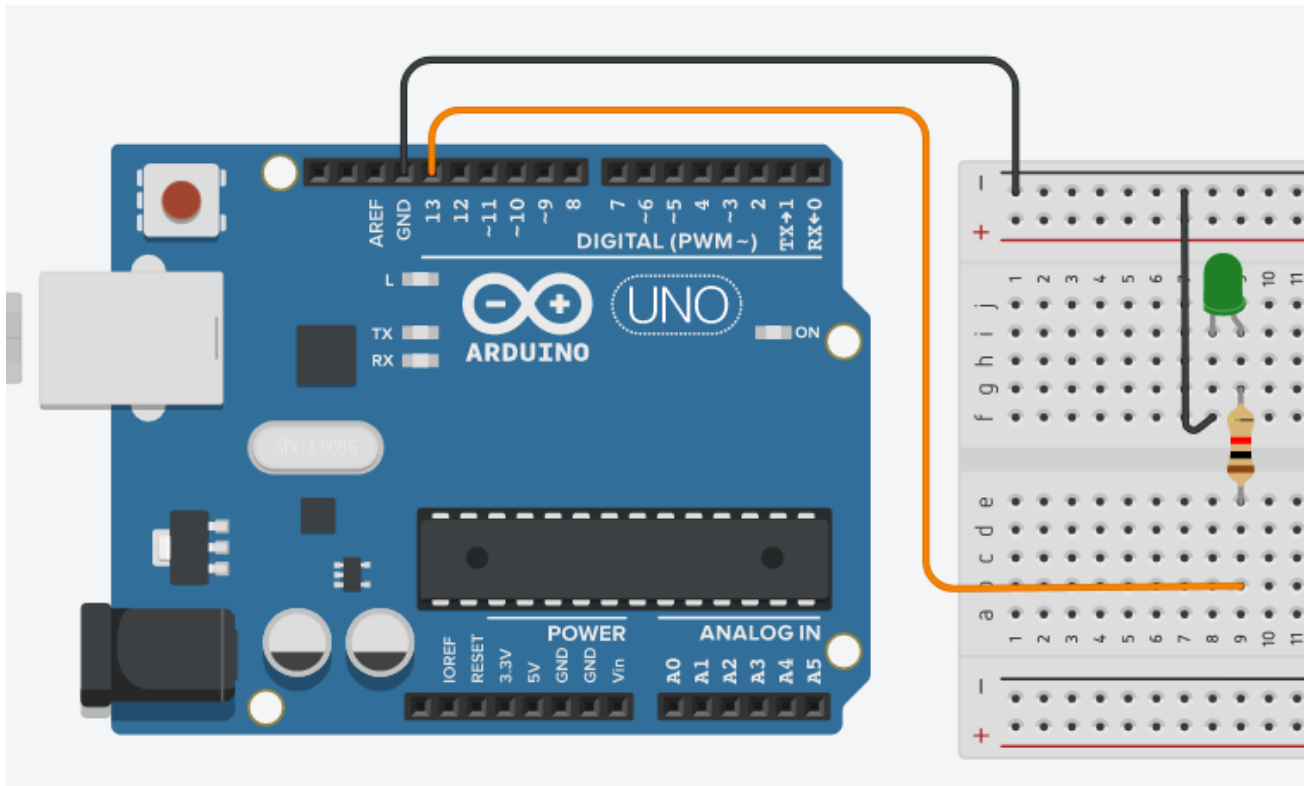


```
void setup ()  
{  
    pinMode (11, OUTPUT);  
    pinMode (12, OUTPUT);  
    pinMode (13, OUTPUT);  
}
```

```
void loop ()  
{  
    digitalWrite (11, HIGH);  
    digitalWrite (12, HIGH);  
    digitalWrite (13, HIGH);  
}
```


TERCEIRO PROGRAMA

Configure o pino 13 como saída e o programe para ele ficar 1 segundo ligado e 1 segundo desligado.

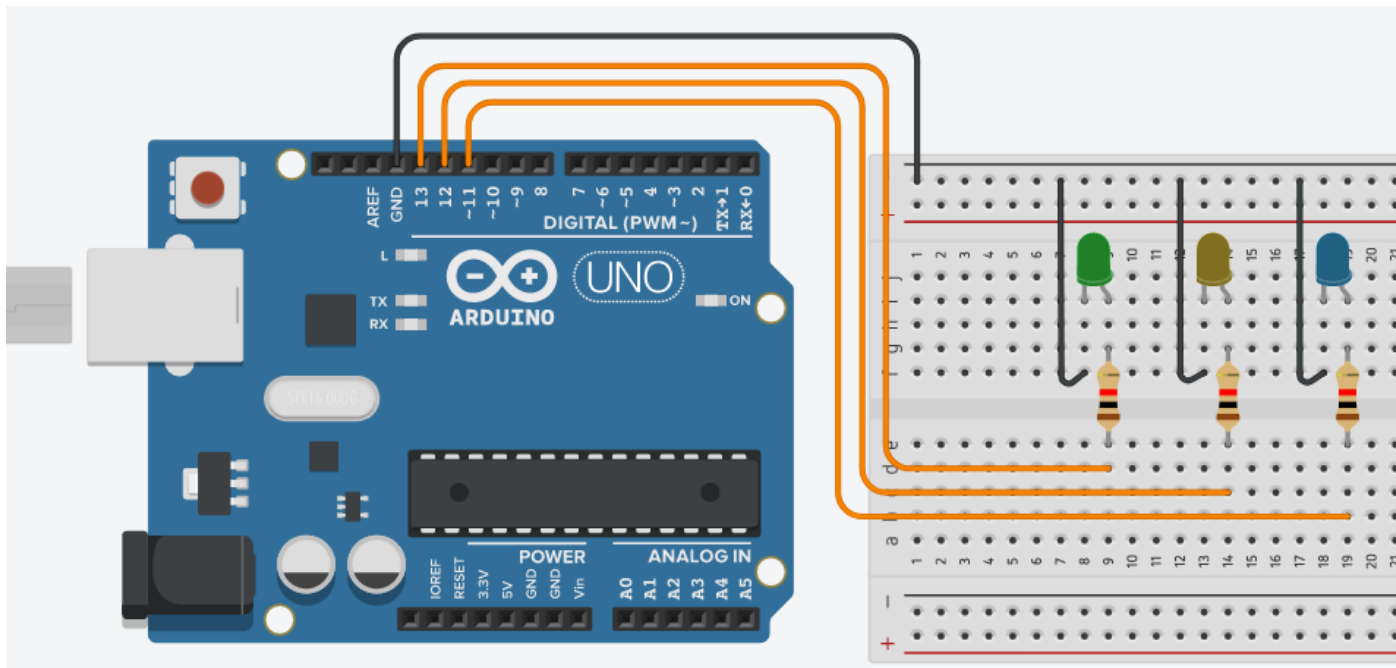


```
void setup ()  
{  
    pinMode (13, OUTPUT);  
}  
  
void loop ()  
{  
    digitalWrite (13, HIGH);  
    delay (1000);  
    digitalWrite (13, LOW);  
    delay (1000);  
}
```

QUARTO PROGRAMA

Configurar os pinos 11, 12 e 13 como saídas e siga a sequência descrita abaixo:

- **Pino 11:** 0,7 segundos ligado e 0,7 segundos apagado;
- **Pino 12:** 0,5 segundo ligado e 0,5 segundos apagado;
- **Pino 13:** 1 segundo ligado e 1 segundo desligado.

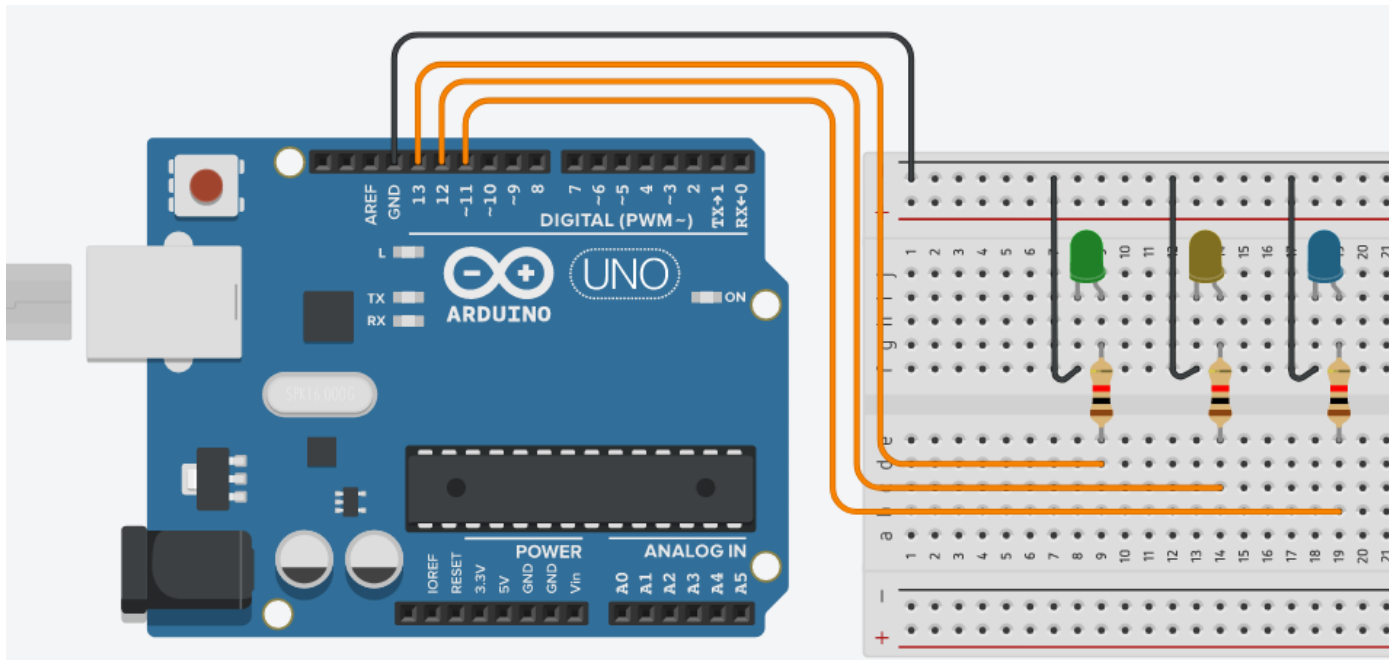


```
void setup ()  
{  
    pinMode (11, OUTPUT);  
    pinMode (12, OUTPUT);  
    pinMode (13, OUTPUT);  
}
```

```
void loop ()  
{  
    digitalWrite (11, HIGH);  
    delay (700);  
    digitalWrite (11, LOW);  
    delay (700);  
    digitalWrite (12, HIGH);  
    delay (500);  
    digitalWrite (12, LOW);  
    delay (500);  
    digitalWrite (13, HIGH);  
    delay (1000);  
    digitalWrite (13, LOW);  
    delay (1000);  
}
```

QUINTO PROGRAMA

Configurar os pinos 11, 12 e 13 e pisca-los de forma sequencial da esquerda para direita e da direita para esquerda de um em um segundo.



```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(11, LOW);  
  digitalWrite(12, LOW);  
  digitalWrite(13, HIGH);  
  delay(1000);
```

```
  digitalWrite(11, LOW);  
  digitalWrite(12, HIGH);  
  digitalWrite(13, LOW);  
  delay(1000);
```

```
  digitalWrite(11, HIGH);  
  digitalWrite(12, LOW);  
  digitalWrite(13, LOW);  
  delay(1000);
```

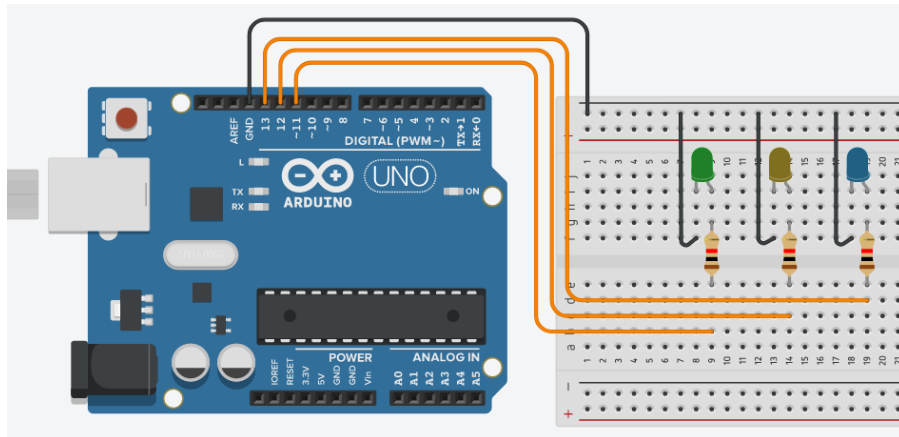
```
  digitalWrite(11, LOW);  
  digitalWrite(12, HIGH);  
  digitalWrite(13, LOW);  
  delay(1000);
```

```
}
```

SEXTO PROGRAMA

Faça a sequência abaixo:

- Liga Led 1, espera 1 s;
- Liga Led 2, espera 1 s;
- Pisca Led 1 e Led 2 em 0,5 s por 6 s;
- Desliga Led 1 e Led 2, espera 1 s;
- Liga Led 1, Led 2 e Led 3, espera 4 s;
- Pisca Led 1 e Led 3 em 0,5 s por 4 s;
- Recomeça o ciclo.



```
void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
```

```
void loop() {
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(1000);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(1000);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(1000);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  delay(4000);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(500);
```

```
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay(500);
```

```
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(500);
```