

Curso Introdutório de Arduino

SEMANA DAS ENGENHARIAS UFABC 2025
DIA 24 DE SETEMBRO DAS 8H ÀS 12H
LABORATÓRIO L406-2

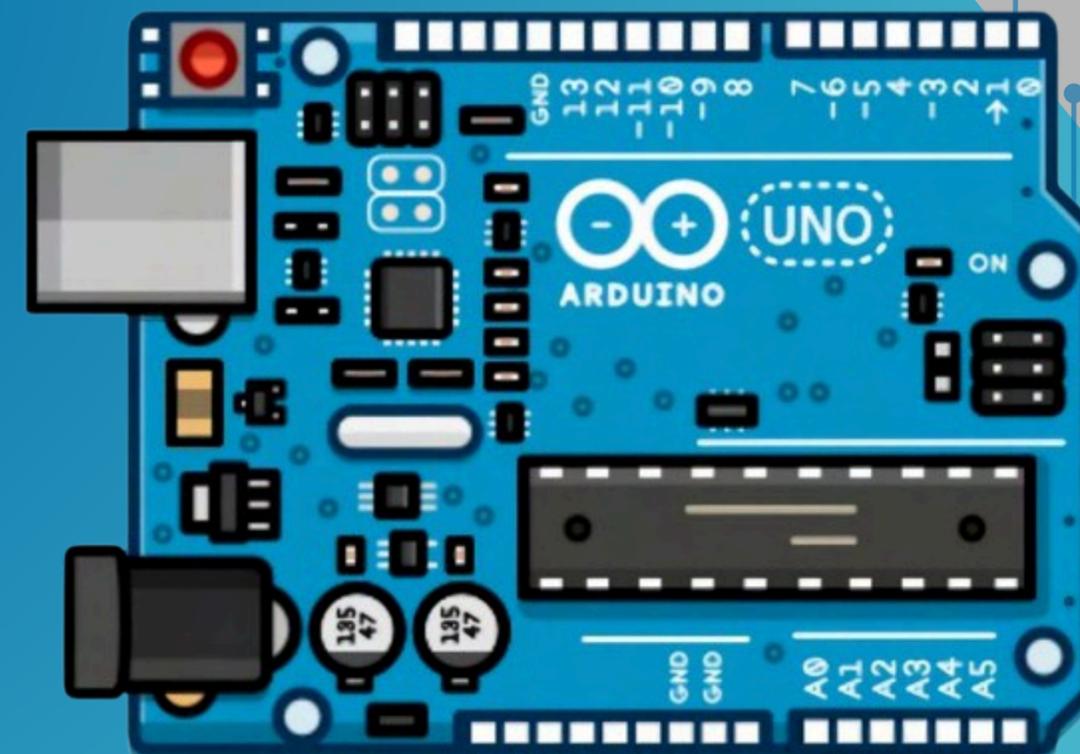
Enrico Zangirolami Raimundo
Gabriel Pereira de Araújo
Lucas Jácome Cabral
Otávio Augusto Tonetto
Pedro Henrique Campos
Ricardo Sandro Siqueira Lobato
Walter Silva Oliveira



UFABC

PROEC

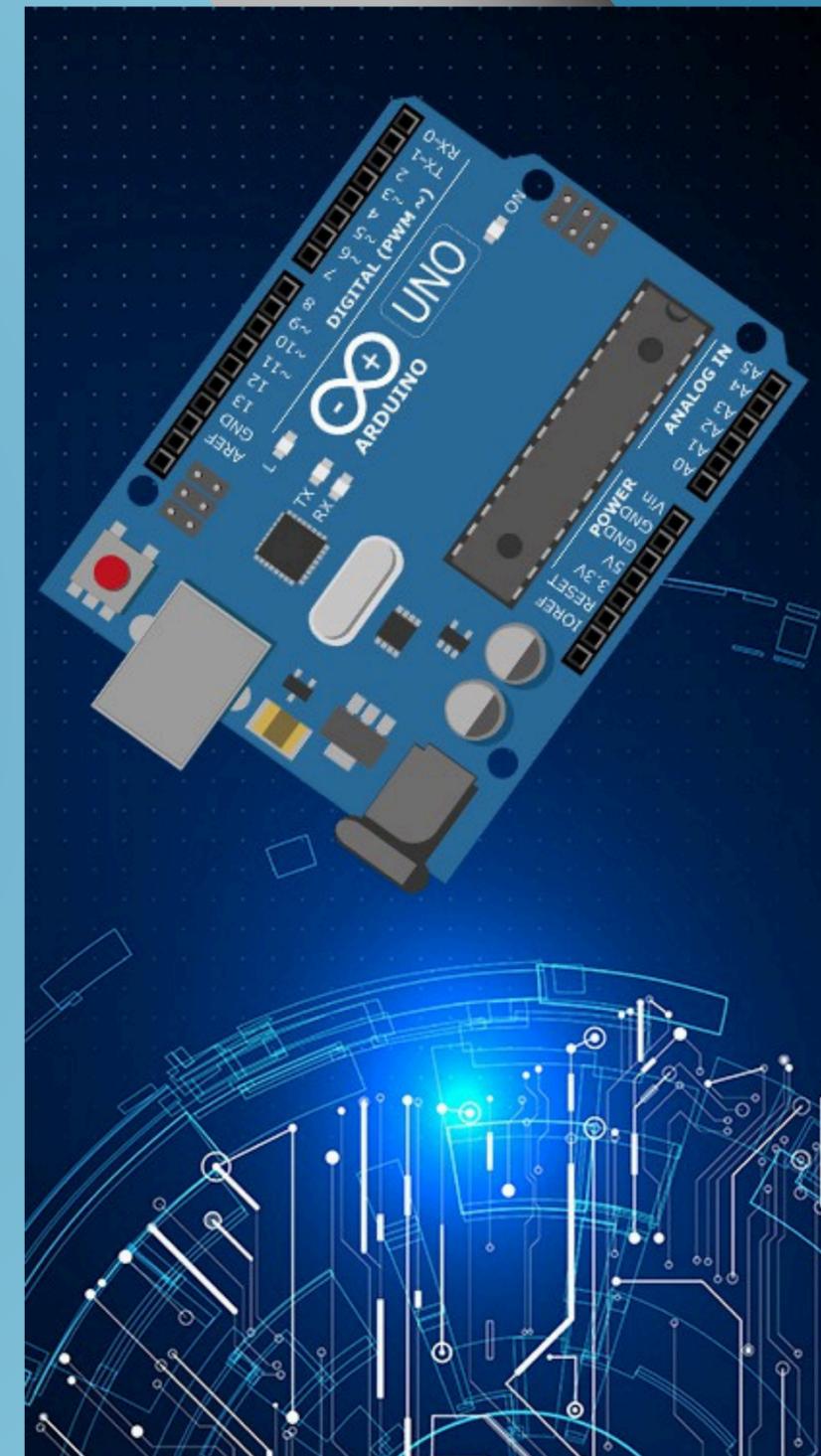
PRÓ-REITORIA DE
EXTENSÃO E CULTURA
Universidade Federal do ABC



O que é Arduino

O Arduino é uma plataforma de prototipagem eletrônica de código aberto, simples e acessível. Funciona como um pequeno computador programável, que permite criar projetos capazes de interagir com o ambiente. Sua linguagem de programação é fácil de aprender, tornando-o ideal tanto para iniciantes quanto para usuários avançados.

“Arduino is an open-source electronics platform based on easy-to-use hardware and software.”



Aplicações e Vantagens

Com o Arduino é possível ler dados de sensores de temperatura, luz, som, gases, vibração, umidade e até reconhecimento de cores.

Essas informações podem ser usadas para controlar motores, luzes, alarmes, displays e outros atuadores. As aplicações vão desde robôs educacionais, drones e automação residencial até máquinas CNC e equipamentos médicos.

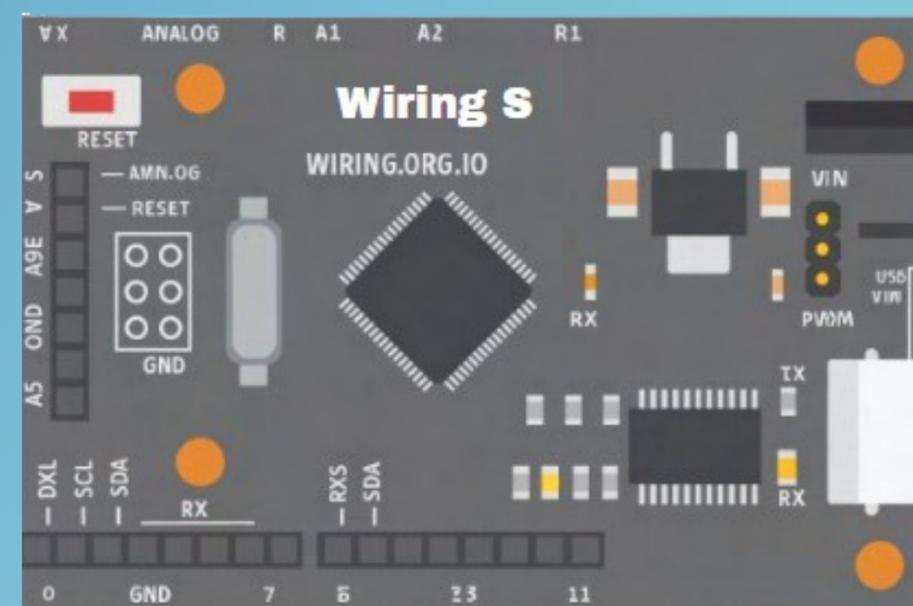
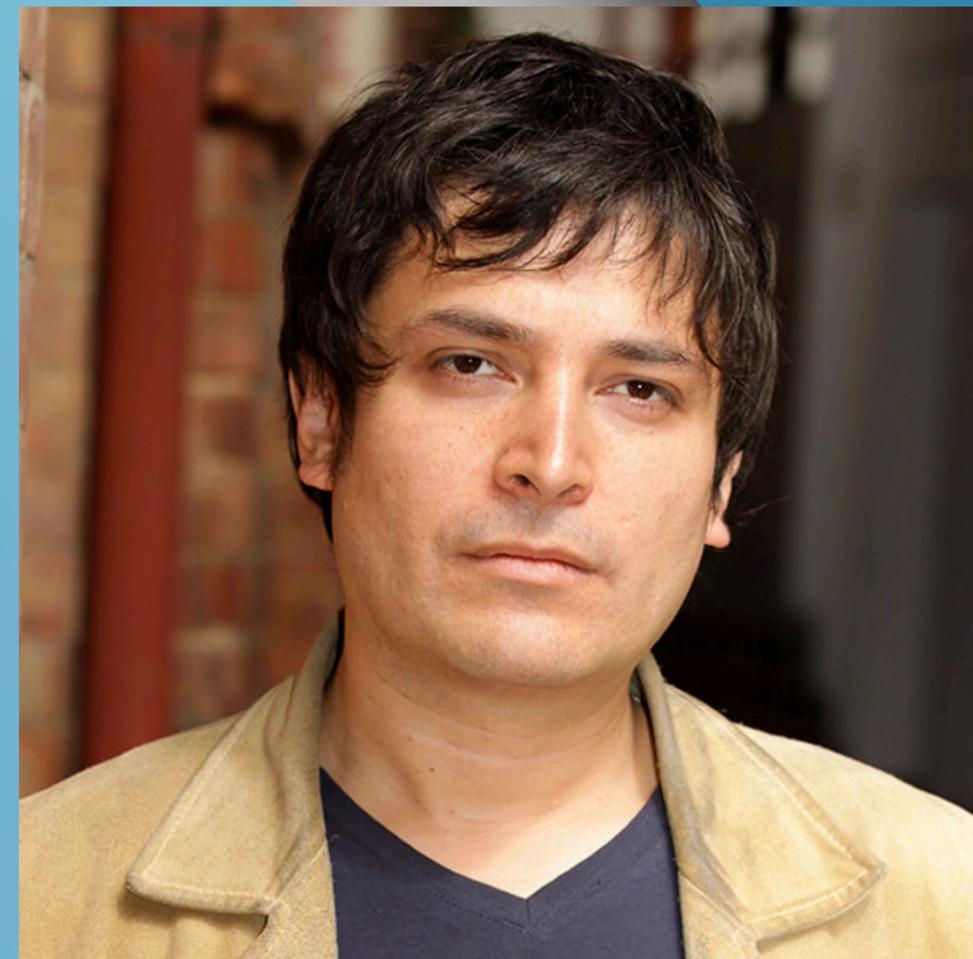
Além de versátil, o Arduino é acessível e não exige muitos componentes externos para iniciar o aprendizado.



História

A história do Arduino começou em 2002, quando o colombiano Hernando Barragán desenvolveu o Wiring, sua tese de graduação na Itália.

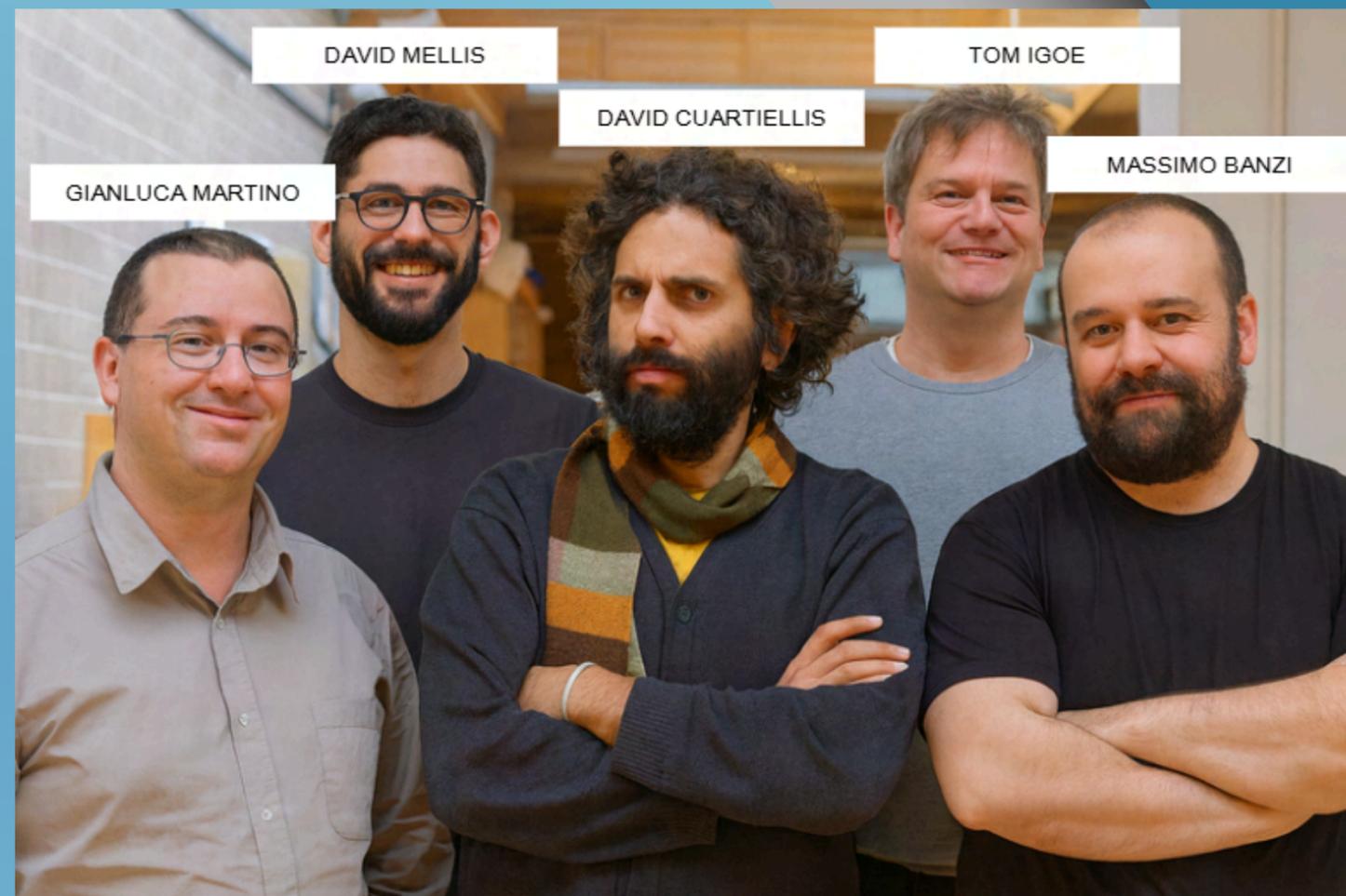
O projeto reunia uma placa com microcontrolador e uma linguagem de programação simples, permitindo que artistas e designers criassem projetos sem precisar dominar eletrônica.



Em 2005, Massimo Banzi, David Mellis e David Cuartielles usaram o Wiring como base para criar o Arduino, mais barato, acessível e fácil de usar.

A proposta era manter a simplicidade, mas ampliar o acesso para estudantes e entusiastas.

A partir daí, surgiram vários modelos de placas, sempre com a mesma filosofia: baixo custo e facilidade de aprendizado. Hoje, o Arduino é utilizado em todo o mundo em escolas, universidades, empresas, laboratórios e projetos criativos, apoiado por uma comunidade global que compartilha conhecimento e inovação.



Modelos de Placas

Arduino Uno

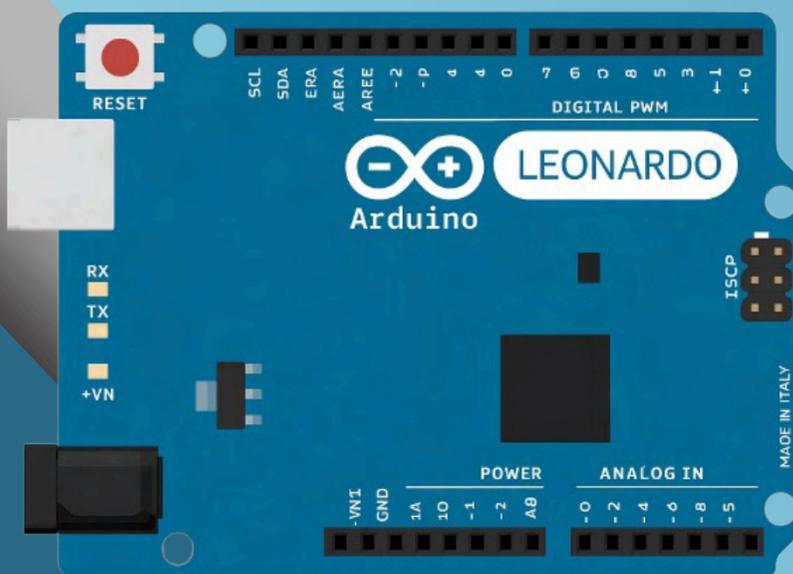
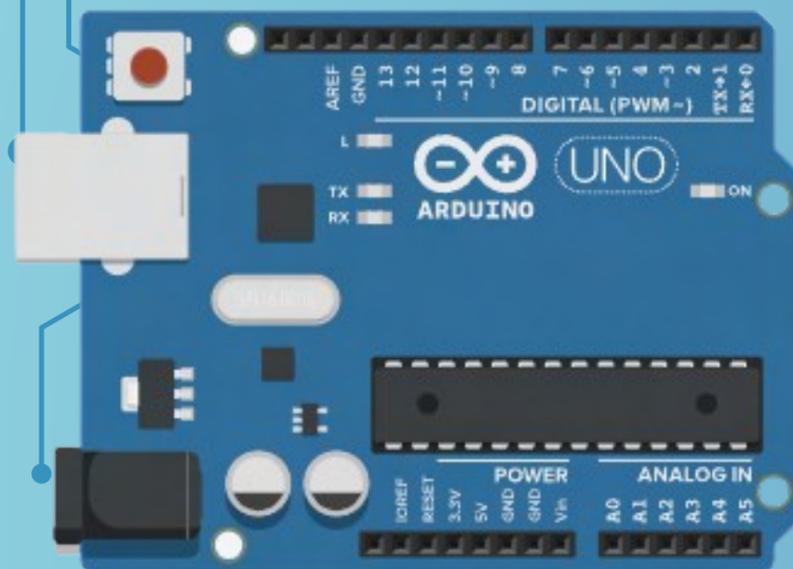
Dentre todos os modelos, o Arduino Uno é o mais popular, com medidas aproximadas de 53mm x 68mm

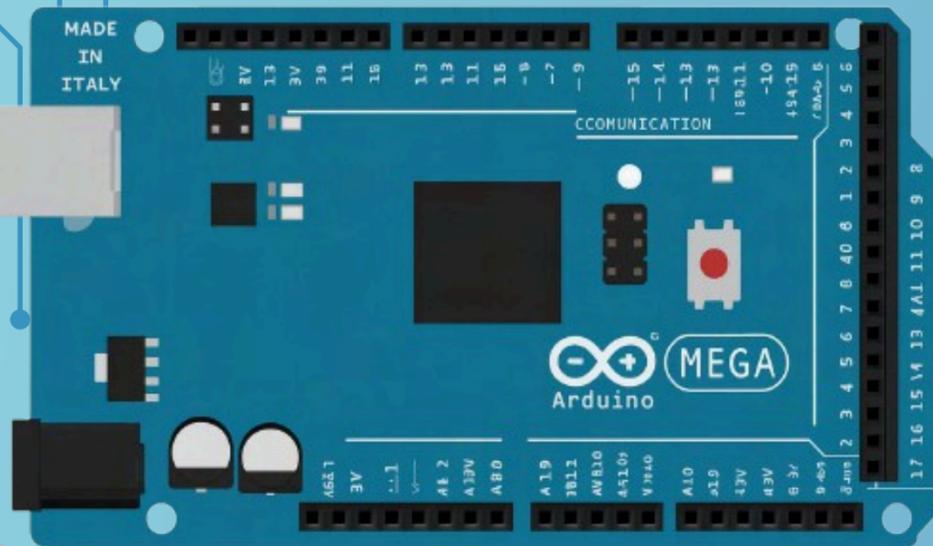
Microcontrolador	ATmega328P	Portas Digitais	14	Portas Analógica	6	Portas PWM	6
------------------	------------	-----------------	----	------------------	---	------------	---

Arduino Leonardo

Apesar de fisicamente muito parecido com o Arduino UNO, apresenta grande diferencial em seu processador, que é um ATmega32u4, o que permite que o Leonardo seja reconhecido com um mouse ou teclado por um computador.

Microcontrolador	ATmega32u4	Portas Digitais	20	Portas Analógica	12	Portas PWM	7
------------------	------------	-----------------	----	------------------	----	------------	---

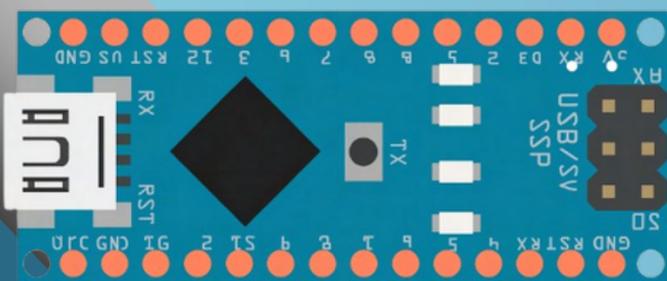




Arduino Mega

Trata-se de uma placa mais robusta, com cerca de 101mm x 53mm, é indicado para trabalhos mais complexo, que exigem maior capacidade de processamento e mais portas.

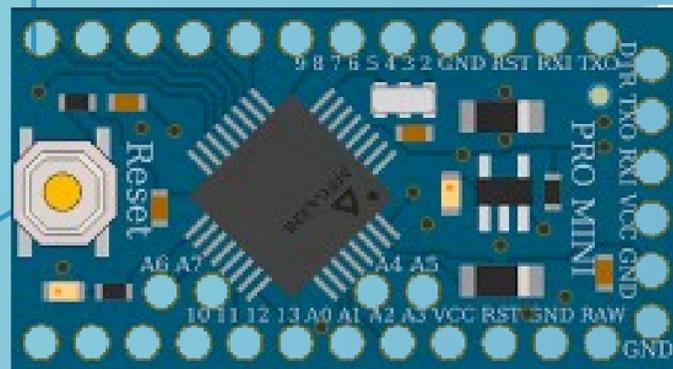
Microcontrolador	ATmega2560	Portas Digitais	54	Portas Analógica	16	Portas PWM	15
------------------	------------	-----------------	----	------------------	----	------------	----



Arduino Nano

Uma placa mais compacta com cerca de 43mm x 18mm porém mantendo sua porta USB C, facilitando seu uso.

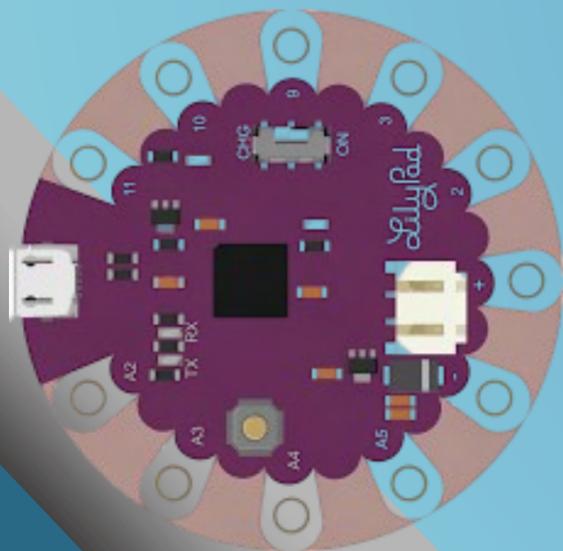
Microcontrolador	ATmega328P AU	Portas Digitais	14	Portas Analógica	8	Portas PWM	6
------------------	---------------	-----------------	----	------------------	---	------------	---



Arduino Pro Mini

Ainda mais compacto que o Nano, com cerca de 33mm x 18mm, com o mesmo processador que o Nano, mas sem a entrada USB, o que diminui suas dimensões, mas dificulta o uso, sendo necessário um conversor.

Microcontrolador	ATmega238P AU	Portas Digitais	14	Portas Analógica	8	Portas PWM	6
------------------	---------------	-----------------	----	------------------	---	------------	---



Arduino LilyPad

Desenvolvida para vestuários e tecidos inteligentes, com 50mm de diâmetro e coloração roxa, possui o mesmo processador que o Nano e Pro Mini, mas não apresenta USB, assim como o Pro Mini

Microcontrolador	ATmega238P AU	Portas Digitais	14	Portas Analógica	6	Portas PWM	6
------------------	---------------	-----------------	----	------------------	---	------------	---

Modelo Esquemático

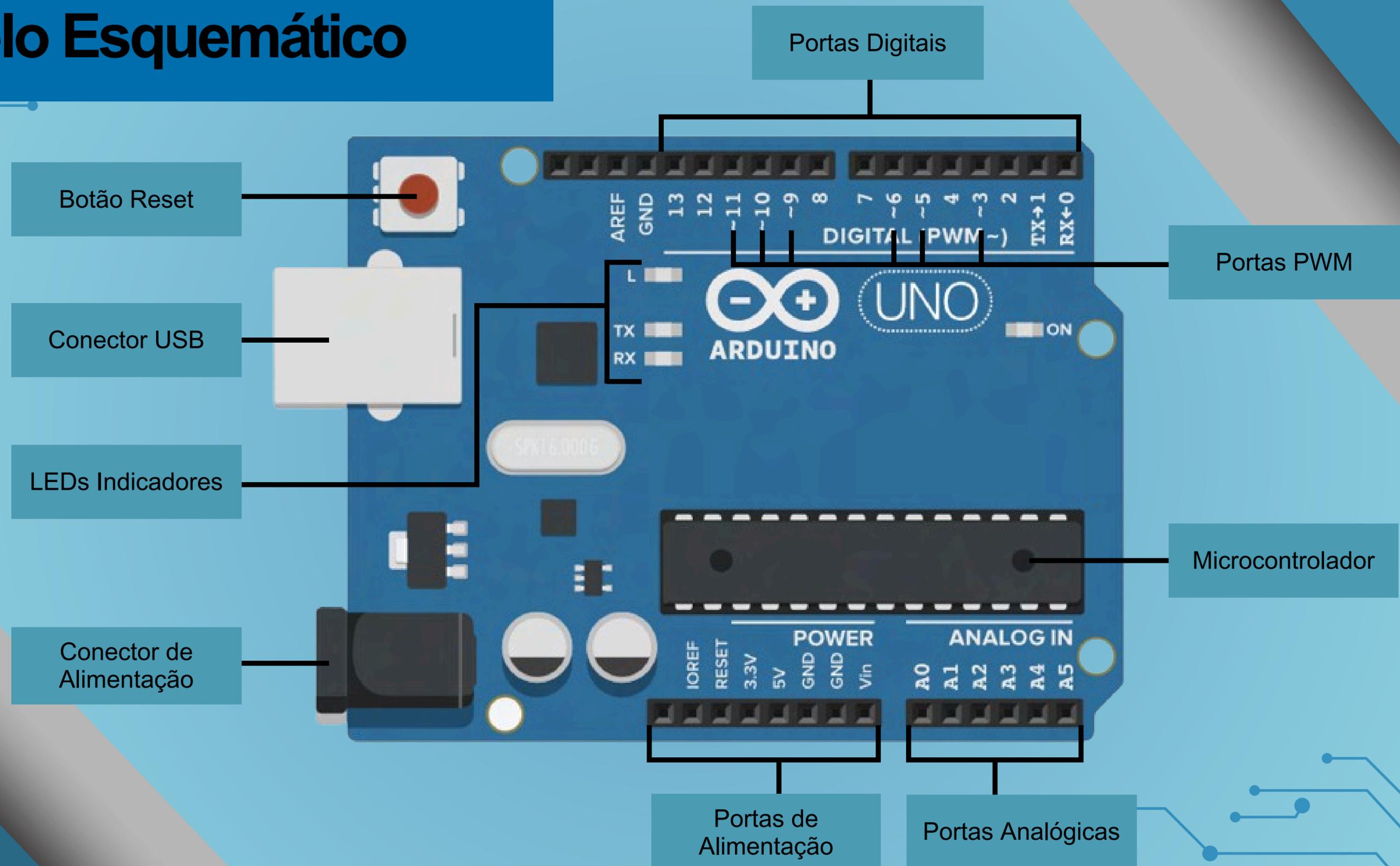
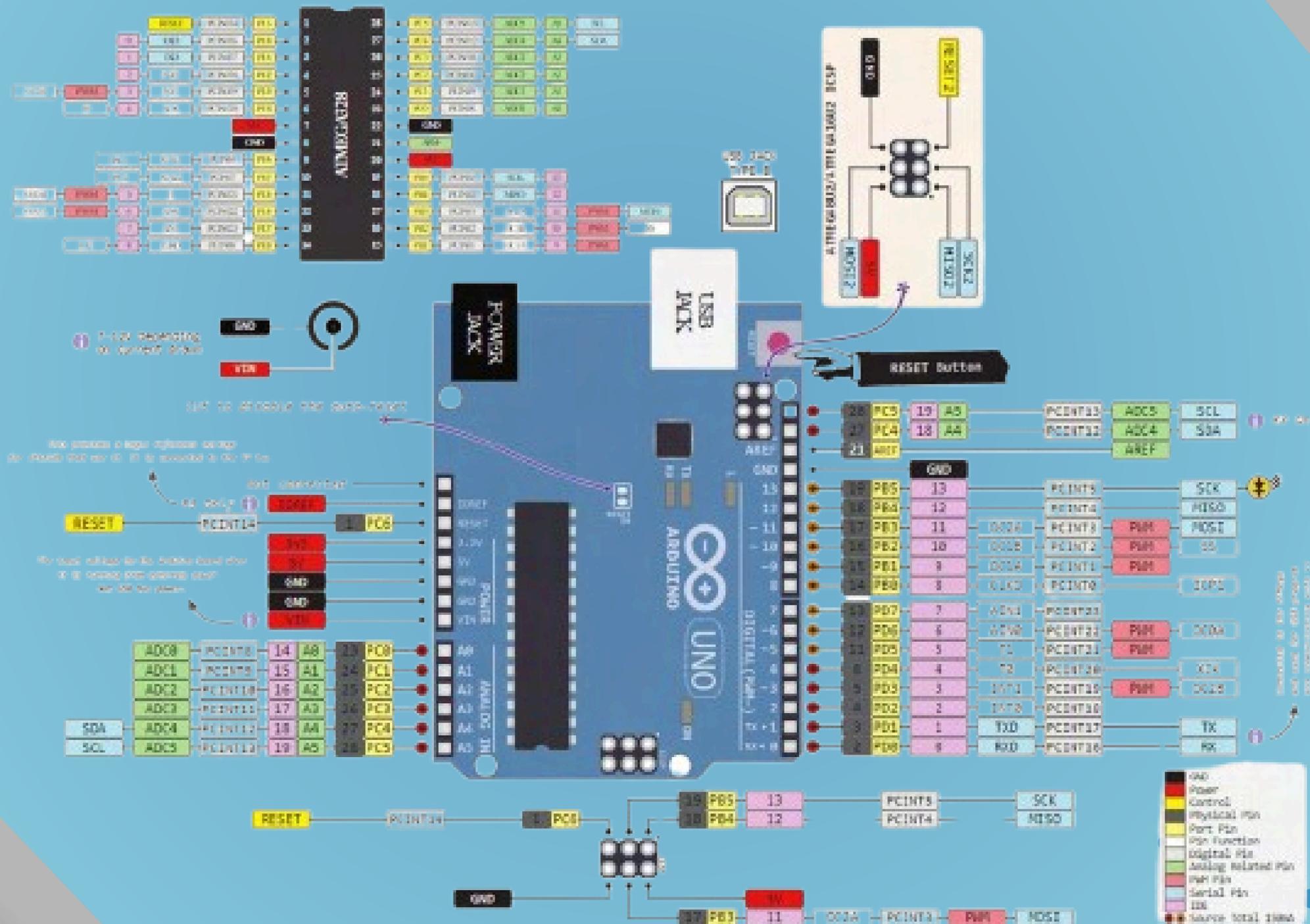


Diagrama dos Pinos

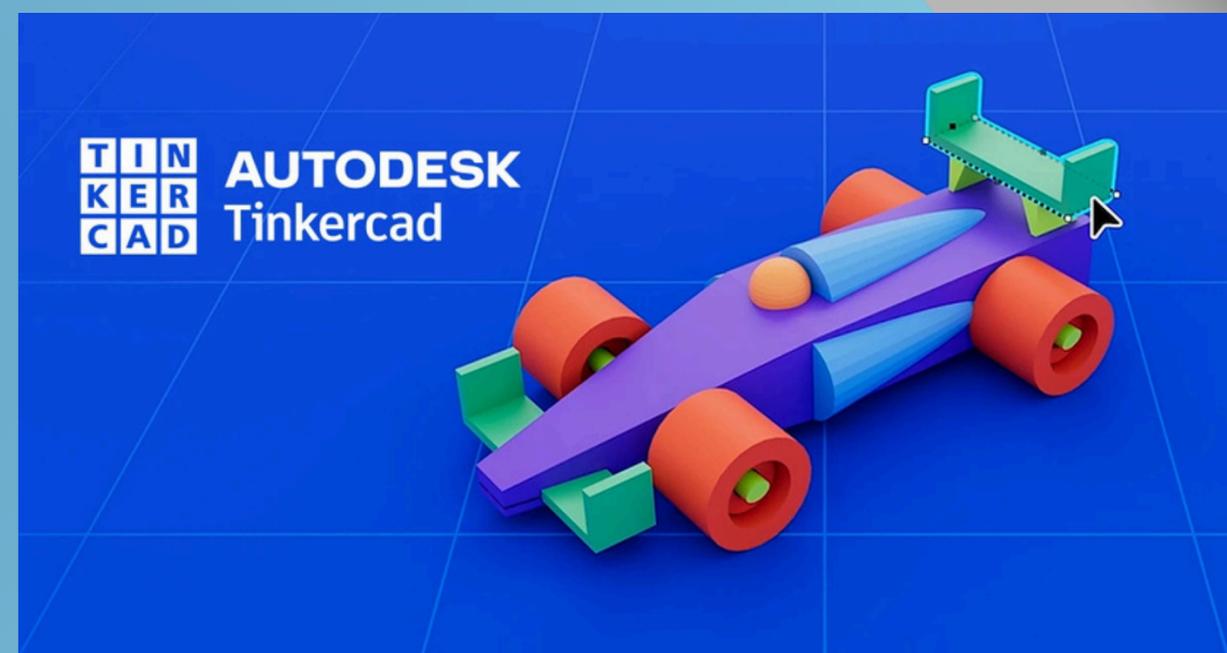


- **Portas Digitais:** O Arduino Uno possui 14 portas digitais (0 a 13), que podem ser configuradas por código como entradas ou saídas de sinais. Elas operam em nível ALTO (5V) ou BAIXO (0V), permitindo interação com dispositivos externos. Os pinos 3, 5, 6, 9, 10 e 11 têm função especial PWM (~), usada para simular sinais analógicos, como no controle do brilho de LEDs ou da velocidade de motores.
- **Portas Analógicas:** Os pinos analógicos do Arduino detectam ou geram tensões entre 0V e 5V, convertidas em valores de 0 a 1023 (0V = 0, 5V = 1023). Há 6 pinos (A0 a A5), que também podem funcionar como pinos digitais.
- **Portas PWM:** Portas com a capacidade de se comportarem como analógicas e digitais por meio da técnica PWM
- **Botão Reset:** Utilizado para encerrar o Arduino, fazendo com que o código interno reinicie.

- **Chip conversor de Comunicação:** O Arduino possui um chip conversor que transforma o sinal TTL em USB, permitindo tanto a gravação de códigos quanto a comunicação com a placa.
- **Conector de Alimentação:** O conector de alimentação do Arduino permite usar fonte AC/DC externa ou até painel solar, desde que o plug seja P4 com polo positivo no centro. A tensão recomendada para a placa Uno é de 7 a 12V.
- **Conector USB:** O conector USB do Arduino permite a comunicação com o computador, sendo usado para gravar códigos, realizar testes de interação ou apenas para alimentação elétrica.
- **LEDs indicadores:** O Arduino possui LEDs de status: o “ON” indica que a placa está ligada; os “TX” e “RX” piscam durante a transmissão e recepção de dados; e o “L”, ligado a um resistor interno, permite realizar testes simples sem componentes adicionais.
- **Pinos de Alimentação:** Os pinos de alimentação distribuem energia a componentes externos. Os GND funcionam como aterramento, os 5V e 3,3V fornecem tensão para circuitos adicionais, e o Vin permite alimentar diretamente a placa.

Tinkercad

O Autodesk Tinkercad é uma plataforma online gratuita, ideal para iniciantes em eletrônica, programação e design 3D. Permite simular circuitos com componentes e placas como o Arduino sem precisar de kit físico, além de modelar objetos em 3D de forma prática. Muito usado em educação, robótica e prototipagem rápida, destaca-se por ser acessível e interativo.



Interface do Tinkercad

Primeiro passo é entrar em “Criar”, e selecionar “Circuitos”.

01 - Nome do projeto.

02 - Área de trabalho.

03 - Código / Programação.

04 - Inicia o código e a passagem de corrente pelo circuito.

05 - Vista padrão do projeto, “Vista do Circuito”.

06 - Vista esquemática do projeto.

07 - Lista de componentes utilizados no projeto.

08 - Menu dos componentes.

09 - Buscador de componentes.

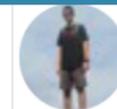
10 - Componentes.

1



Nome

Salvo



Código

Iniciar simulação

Enviar para

3

4

5

6

7

2

Componentes Básico

Pesquisar

8

9

10



Resistor



LED



Botão



Potenciômetro



Capacitor



Interruptor deslizante



Bateria 9V



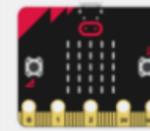
Bateria 3V do tipo moeda



Bateria 1,5V



Placa de



micro:bit



Arduino Uno

Programando em Blocos

 Saída	 Controlar
 Entrada	 Matemática
 Notação	 Variáveis

- **Saída:** Executam ações, como acender LEDs, mover motores ou enviar sinais.
- **Entrada:** Recebem dados de sensores ou botões.
- **Notação:** Comentários e marcações para organizar o código.
- **Controlar:** Estruturas de fluxo, como loops, condições e temporizadores.
- **Matemática:** Operações e cálculos lógicos para processar dados.
- **Variáveis:** Espaços de memória que armazenam valores temporários reutilizáveis.

Arduino IDE

O Arduino IDE é o programa usado para escrever, compilar e enviar códigos para placas Arduino. Gratuito e de código aberto, funciona em diferentes sistemas operacionais e traz exemplos prontos e bibliotecas que facilitam a criação de projetos. A versão mais recente possui interface moderna, suporte ao Arduino Cloud e ferramentas como depurador integrado, tornando o desenvolvimento mais rápido e intuitivo.



Interface do Arduino IDE

- 01 - Placa e porta: Mostra a placa conectada e sua porta.
- 02 - Verificar/Compilar : Compila e envia o código para a placa.
- 03 - Sketchbook: Acesso rápido aos projetos salvos.
- 04 - Gerenciador de Placas: Adiciona suporte a diferentes microcontroladores.
- 05 - Library Manager: Instala bibliotecas da comunidade.
- 06 - Depurador: Testa e corrige o código em tempo real.
- 07 - Pesquisar: Localiza palavras no código.
- 08 - Arduino Cloud: Acesso à conta online.
- 09 - Monitor Serial: Exibe dados recebidos/enviados pela porta serial.
- 10 - Serial Plotter: Mostra dados da porta serial em gráficos.
- 11 - Editor de Código: Área de escrita e edição do programa.
- 12 - Status de Resultados: Mostra erros e resultados da compilação.
- 13 - Número da Linha: Exibe posição do cursor, placa e porta usadas.

File Edit Sketch Tools Help

Select Board

sketch_sep7a.ino

```
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
```

9
10

11

12

13

1
2
3
4
5
6
7

8

Bibliotecas

As bibliotecas do Arduino são coleções de códigos prontos que facilitam a programação, permitindo usar funções específicas sem escrever tudo do zero. Elas organizam os projetos e reduzem a quantidade de comandos. Existem bibliotecas para motores, sensores, telas LCD e muitos outros módulos. O Arduino IDE já vem com algumas instaladas, mas a comunidade disponibiliza várias gratuitamente. Para usá-las, basta instalá-las pelo Library Manager e incluí-las no início do código, liberando suas funções no projeto.

Adicionando Bibliotecas

- 01 - Selecionar “Sketchbook”;
- 02 - Clicar em “Include Library” (adicionar biblioteca);
- 03 - Caso seja necessário, será preciso baixar um arquivo .zip contendo a biblioteca a ser adicionada, após baixado, basta clicar em “add ZIP lybrary...”, que a bibliotera será adicionada ao programa.

Observação: Geralmente as bibliotecas já apresentam códigos inteiros, totalmente acessíveis.

01

```
File Edit Sketch Tools Help  
✓ → ⚙ Select Board  
sketch_sep7a.ino  
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
```

02

The image shows the Arduino IDE interface with the 'Sketch' menu open. The menu items are as follows:

- Verify/Compile (Ctrl+R)
- Upload (Ctrl+U)
- Configure and Upload
- Upload Using Programmer (Ctrl+Shift+U)
- Export Compiled Binary (Alt+Ctrl+S)
- Optimize for Debugging
- Show Sketch Folder (Alt+Ctrl+K)
- Include Library** (highlighted with a red box)
- Add File...

The background shows a code editor with the text "ere, to run once:" and "e, to run repeatedly:".

03

The image shows the Arduino IDE interface with the 'Sketch' menu open. The menu items are as follows:

- Verify/Compile (Ctrl+R)
- Upload (Ctrl+U)
- Configure and Upload
- Upload Using Programmer (Ctrl+Shift+U)
- Export Compiled Binary (Alt+Ctrl+S)
- Optimize for Debugging
- Show Sketch Folder (Alt+Ctrl+K)
- Include Library >
- Add File...

The 'Include Library' submenu is open, showing the following options:

- Manage Libraries... (Ctrl+Shift+I)
- Add .ZIP Library...** (highlighted with a red box)
- Contributed libraries
- Adafruit MFRC630 RFID
- MFRC522
- MFRC522-spi-i2c-uart-async
- Servo
- Ultrasonic

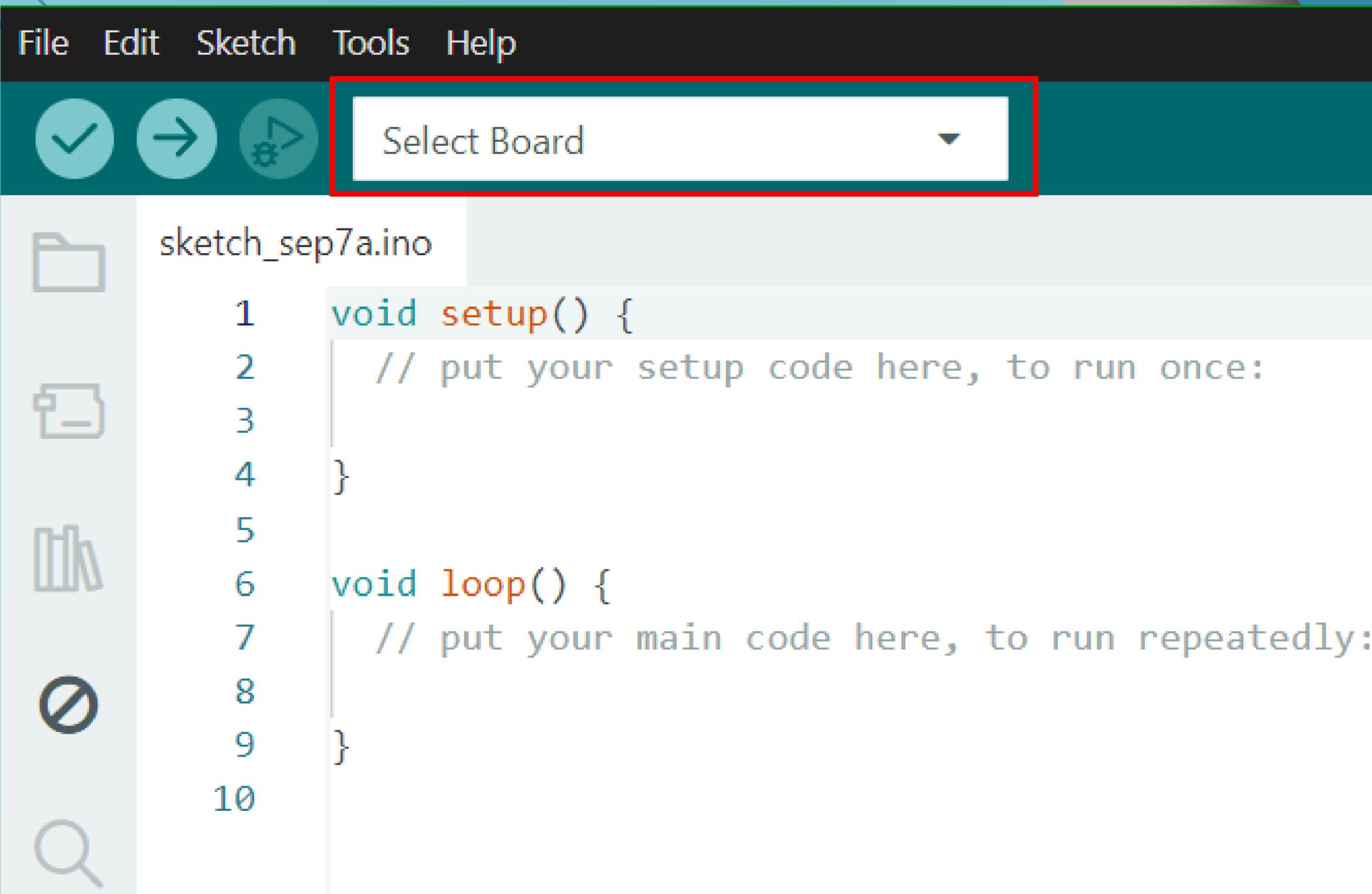
The background shows a code editor with the following text:

```
ere, to run once:  
  
e, to run repeatedly:
```

Enviando Códigos

- 01 - Selecionar a placa (“Select Board”);
- 02 - Escolher qual placa (board) e qual porta (port/COM) deseja usar para carregar o código (“Select other board and port...”);
- 03 - Caso não reconheça a placa, apenas digite o nome do modelo dela, não precisa ser o nome completo. Por exemplo com a placa de Arduino UNO R3, basta apenas escrever UNO, que aparecerá um modelo compatível;
- 04 - Código (“Sketch”);
- 05 - Enviar o código para a placa (“Upload”).

01



The image shows the Arduino IDE interface. At the top, there is a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with several icons: a checkmark, a right arrow, a gear, and a dropdown menu. The dropdown menu is highlighted with a red box and contains the text 'Select Board'. Below the toolbar is a file explorer on the left with icons for folders, files, libraries, and a search icon. The main area is a code editor showing a sketch named 'sketch_sep7a.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
```

02

File Edit Sketch Tools Help

Select Board

Unknown
COM4

Select other board and port...

```
1 v
2
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

to run once:

Select Other Board and Port



Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS



Adafruit Circuit Playground

Adafruit Circuit Playground Express

Analog ADI

Arduino 101

Arduino BT

PORTS

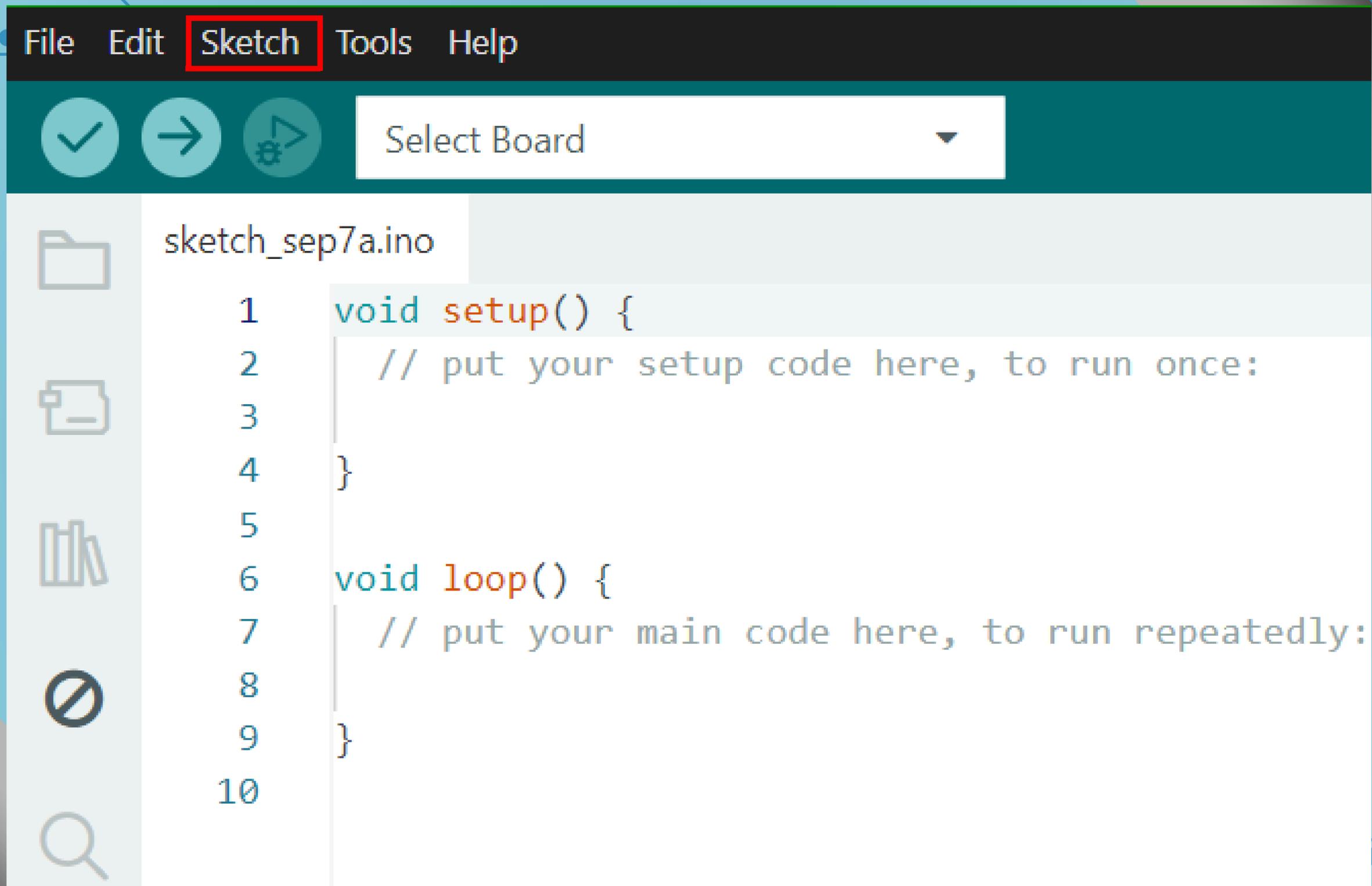
COM4 Serial Port (USB)

Show all ports

CANCEL

OK

04



File Edit **Sketch** Tools Help

✓ → ⚙ Select Board ▼

sketch_sep7a.ino

```
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
```

05

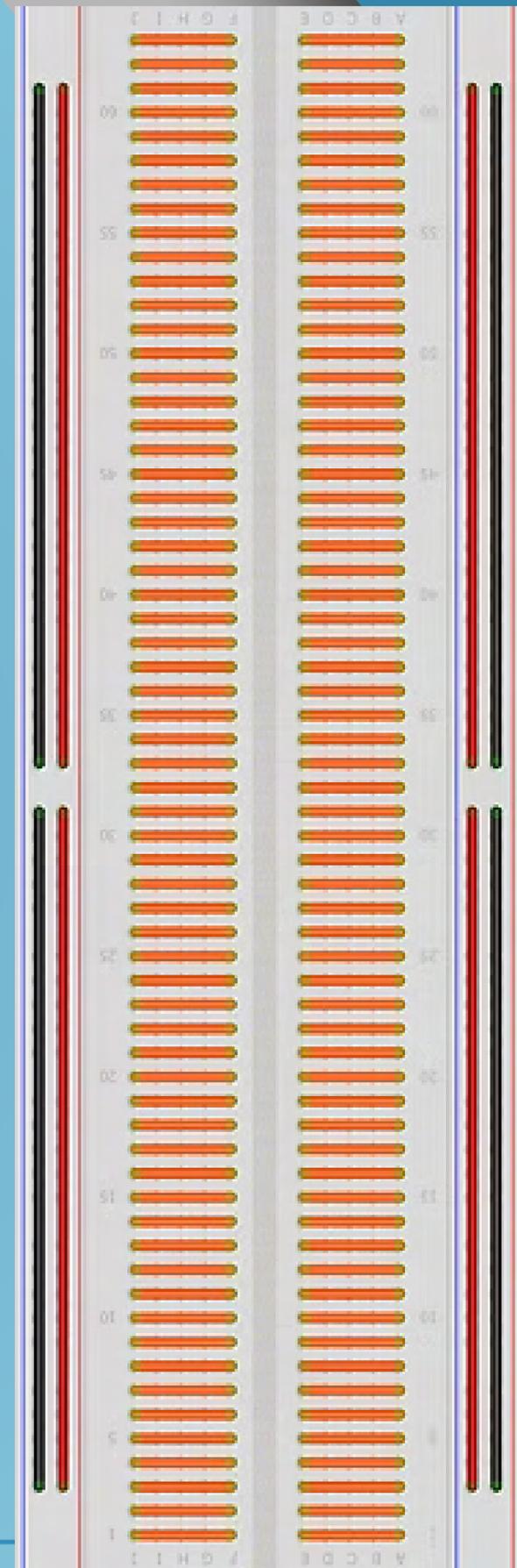
The image shows a screenshot of an IDE's menu system. The 'Sketch' menu is open, and the 'Upload' option is highlighted with a red rectangle. The menu items and their keyboard shortcuts are as follows:

- Verify/Compile (Ctrl+R)
- Upload (Ctrl+U)**
- Configure and Upload
- Upload Using Programmer (Ctrl+Shift+U)
- Export Compiled Binary (Alt+Ctrl+S)
- Optimize for Debugging
- Show Sketch Folder (Alt+Ctrl+K)
- Include Library >
- Add File...

The background shows a code editor with text that is partially obscured by the menu. Visible text includes "ere, to run once:" and "re, to run repeatedly:". The IDE interface includes a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with a checkmark icon and a minus icon. On the left side, there is a sidebar with icons for a folder, a document, a book, a prohibition sign, and a magnifying glass.

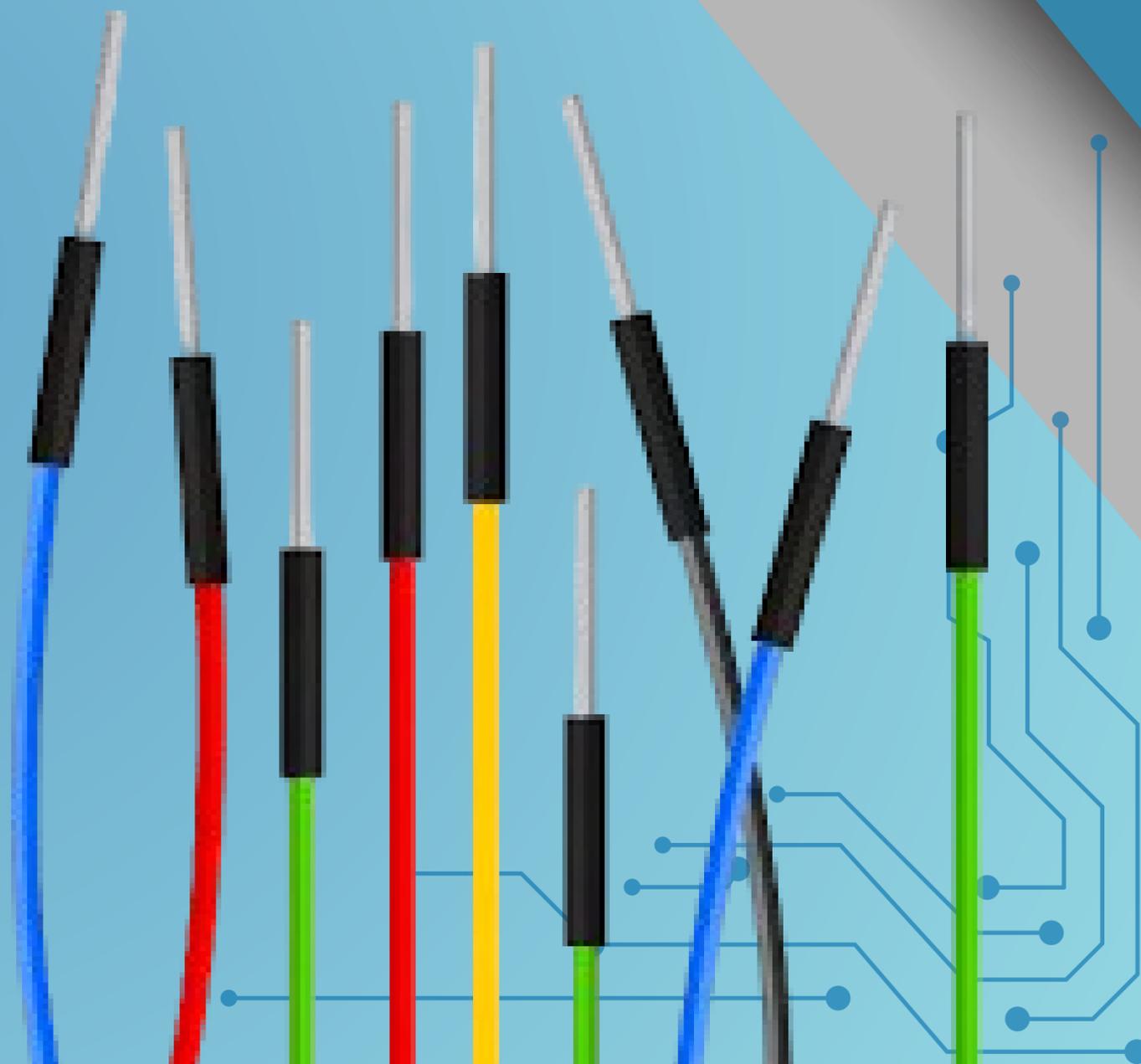
Protoboard

A Protoboard ou Breadboard é uma placa com furos utilizada para construção de circuitos temporários. Essa possibilita que o usuário insira e remova fios e componentes de forma rápida e prática sem a necessidade de solda, tornando possível a alteração do esquema do projeto a qualquer momento. Dessa forma, é possível que se obtenha a prova do funcionamento de um projeto antes de sua montagem final em uma placa de circuito impresso.



Jumper

Os jumpers utilizados nas protoboards são fios flexíveis que contêm terminais metálicos em suas extremidades, facilitando muito o processo de união temporária dos terminais dos componentes.



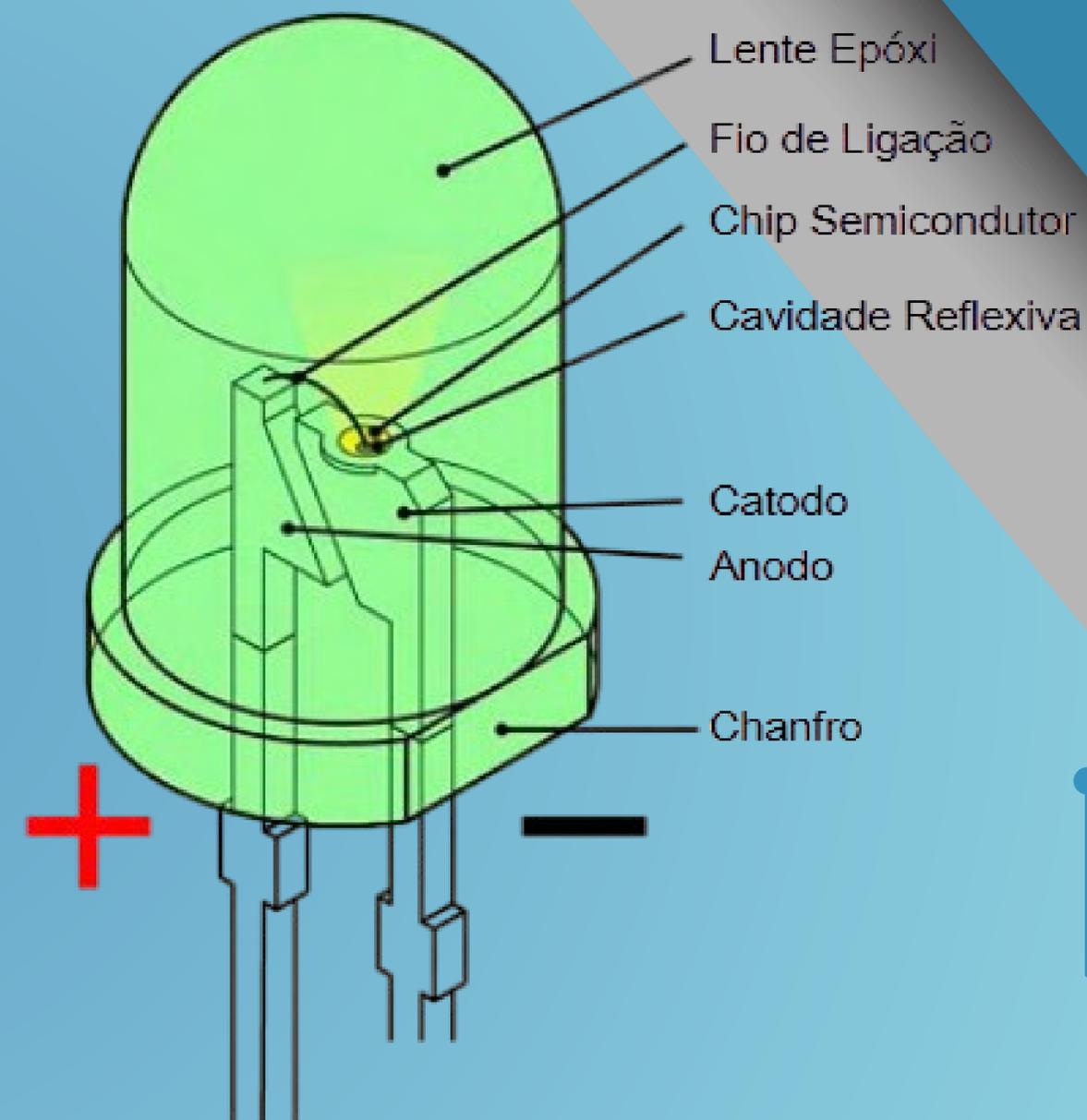
LED

A palavra LED é uma abreviatura de "Light Emitter Diodo" ou "Diodo Emissor de Luz".

Este componente eletrônico é capaz de transformar energia elétrica em luz.

Sendo um componente bipolar, possui dois terminais: ânodo (terminal maior) e cátodo (terminal menor).

A forma pela qual estes estão polarizados determina a passagem da corrente elétrica, resultando na emissão de luz.



Resistor

O resistor é um componente eletrônico capaz de limitar a corrente elétrica em um determinado fluxo de energia.

Pelo fato de possuir uma resistência maior do que a dos cabos e das trilhas de um circuito elétrico, o resistor força a redução da corrente elétrica que passa por ele, provocando assim uma queda de tensão.



Primeiro Dígito
Segundo Dígito
Multiplicador

Tolerância

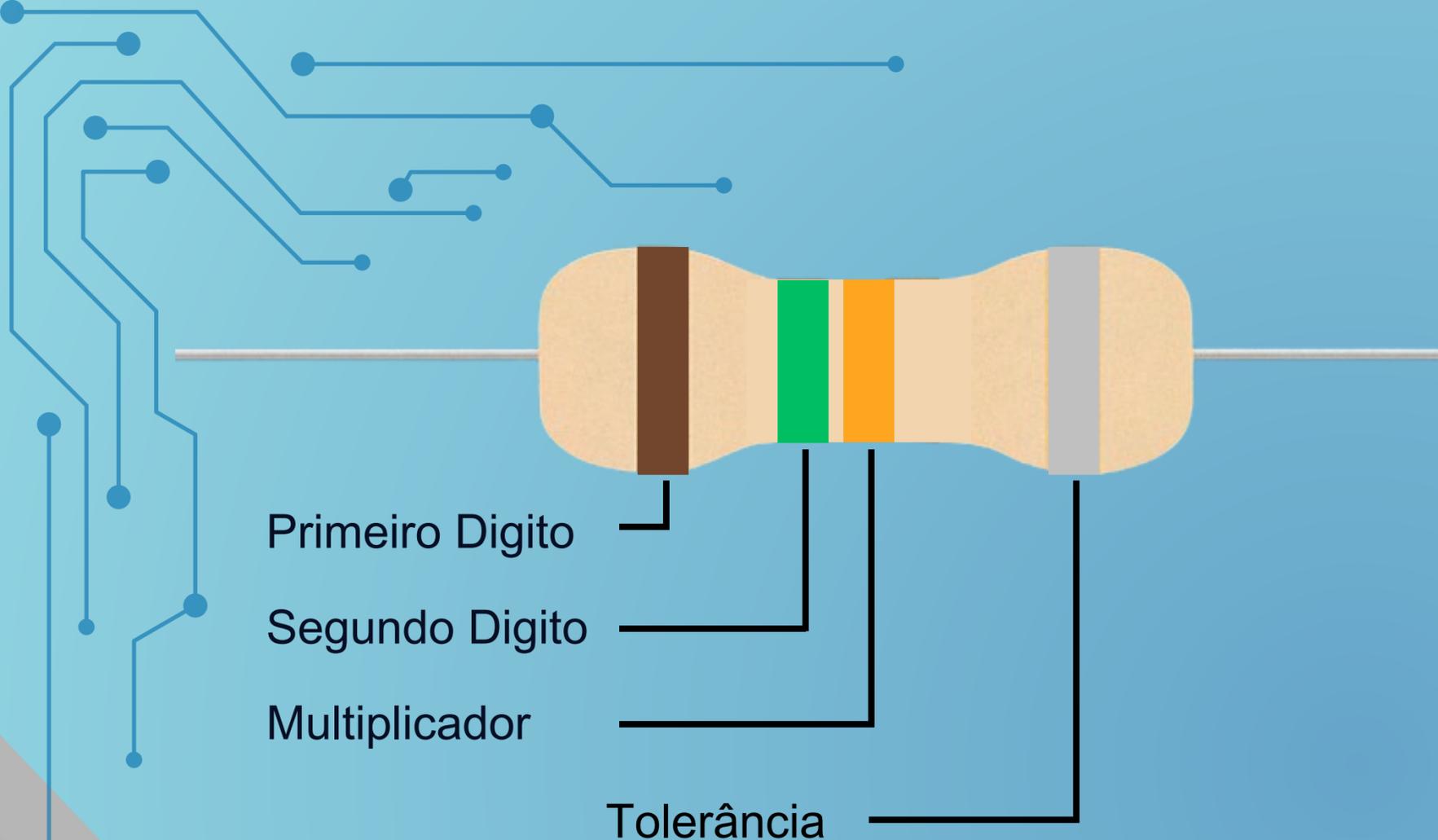
- MARROM 1%
- VERMELHO 2%
- DOURADO 5%
- PRATA 10%

No exemplo a cima temos:

1 5 000

15000Ω ou 15KΩ com tolerância de 10%

COR	VALOR
PRETO	0
MARROM	1
VERMELHO	2
LARANJA	3
AMARELO	4
VERDE	5
AZUL	6
VIOLETA	7
CINZA	8
BRANCO	9



Primeiro Dígito

Segundo Dígito

Multiplicador

Tolerância

- MARROM 1%
- VERMELHO 2%
- DOURADO 5 %
- PRATA 10%

No exemplo a cima temos:

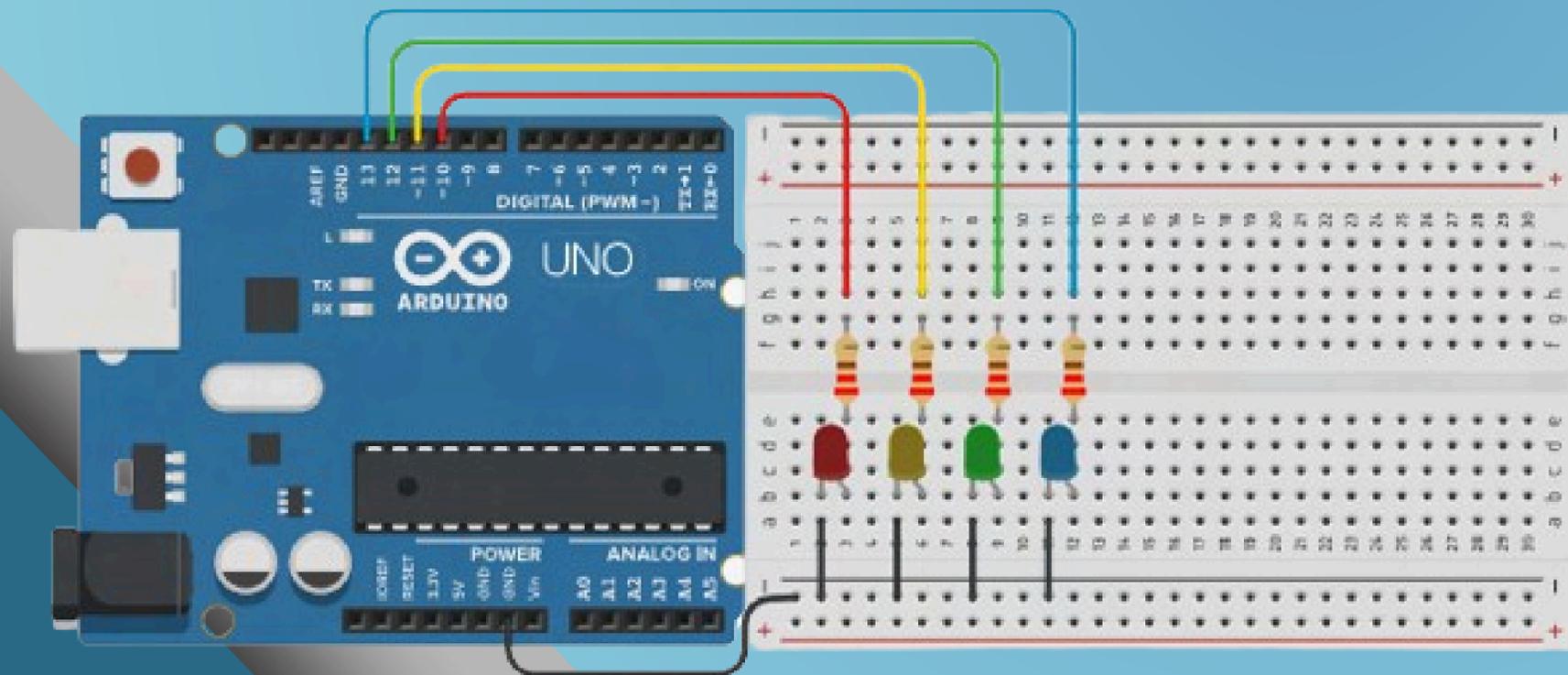


15000Ω ou 15KΩ com tolerância de 10%

COR	VALOR
PRETO	0
MARROM	1
VERMELHO	2
LARANJA	3
AMARELO	4
VERDE	5
AZUL	6
VIOLETA	7
CINZA	8
BRANCO	9

Exemplo

Exemplo de circuito com 4 LEDs piscando de maneira sequencial (ou “chaser”).



```
para sempre
  definir pino 10 como ALTO
  definir pino 11 como BAIXO
  definir pino 12 como BAIXO
  definir pino 13 como BAIXO
  aguardar 1 s
  definir pino 10 como BAIXO
  definir pino 11 como ALTO
  definir pino 12 como BAIXO
  definir pino 13 como BAIXO
  aguardar 1 s
  definir pino 10 como BAIXO
  definir pino 11 como BAIXO
  definir pino 12 como ALTO
  definir pino 13 como BAIXO
  aguardar 1 s
  definir pino 10 como BAIXO
  definir pino 11 como BAIXO
  definir pino 12 como BAIXO
  definir pino 13 como ALTO
  aguardar 1 s
```

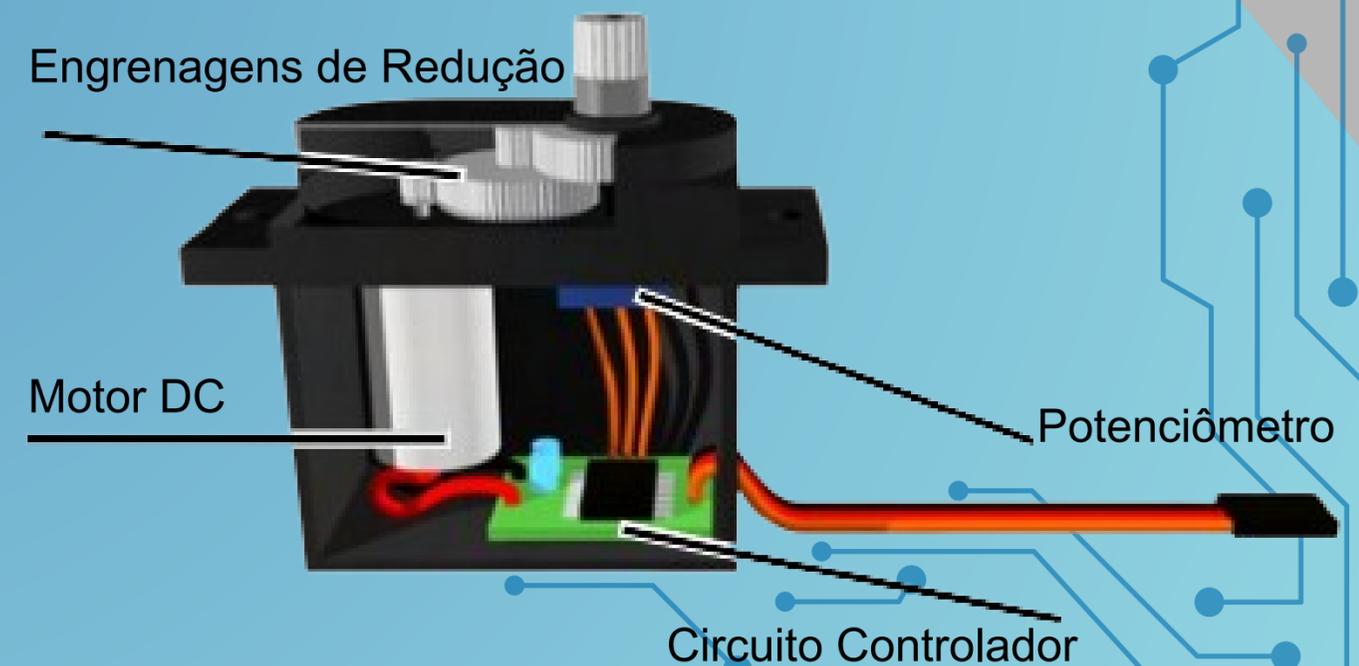
Para transformar a programação de blocos para C++, basta fazer o download do arquivo que já estará em formato .ino, o que permite que seja colocado direto no Arduino IDE

The screenshot shows the TinkerCAD web interface. At the top, the title bar reads "Exemplo Curso Introdutório - Semana das Engenharias" and "Todas as alterações salvas". The main workspace displays a 3D model of an Arduino Uno R3 board connected to a breadboard with several resistors and jumper wires. On the right side, there is a "Blocos" (Blocks) palette with a red box around the download icon. Below the palette, a list of blocks is visible, including "definir LED incorporado como ALTO", "definir pino 0 como ALTO", "definir pino 3 como 0", "gitar servo no pino 0 em 0 graus", "reproduzir alto-falante no pino 0 com", "desativar alto-falante no pino 0", and "imprimir no monitor serial hello world". The main workspace area contains a "para sempre" (loop) block containing several "definir pino" (define pin) blocks and "aguardar" (wait) blocks. The top right of the workspace has buttons for "Código" (Code), "Iniciar simulação" (Start simulation), and "Enviar para" (Send to). The bottom left corner shows a "Monitor serial" window.

Servo Motor SG90

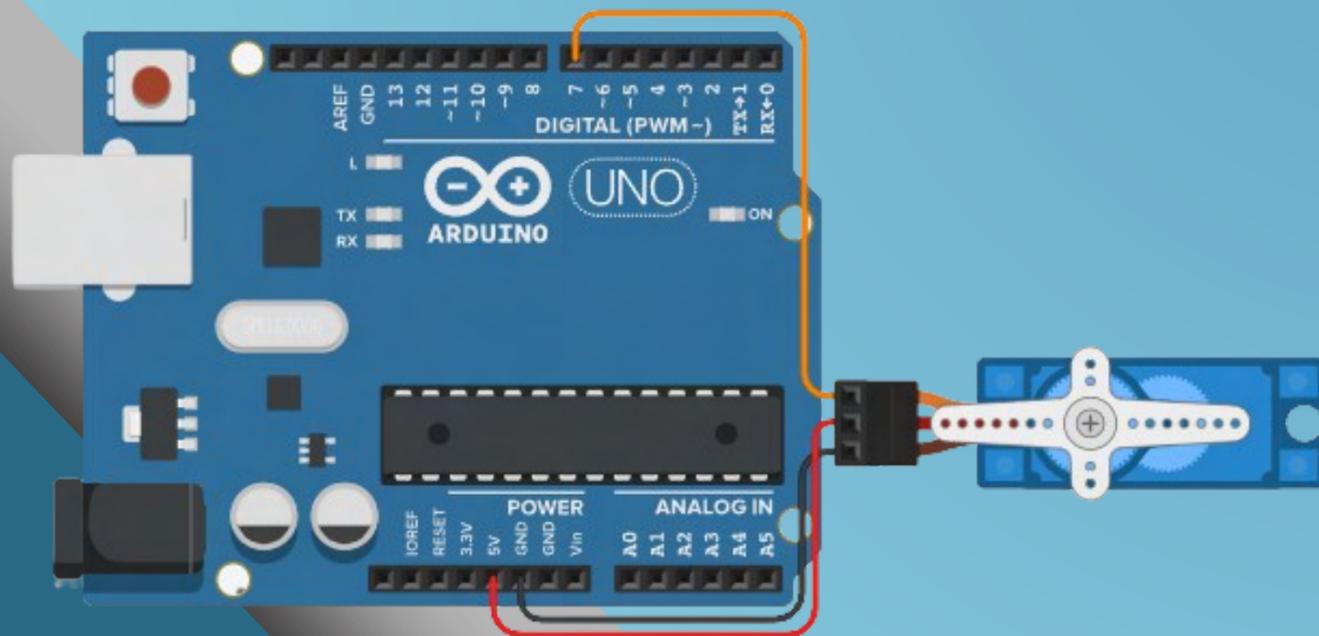
O servo motor SG90 é controlado por sinal PWM do Arduino. Possui um motor DC com engrenagens de redução, que diminui a velocidade e aumenta o torque. Ele não gira livremente, mas apenas em ângulos específicos, chegando a cerca de 180° (90° para cada lado). Conta com circuito controlador e potenciômetro interno, sendo muito usado para dar movimento a robôs e máquinas.

<https://www.arduino-libraries.info/libraries/servo>



Exemplo

Exemplo de circuito com um servo motor controlado por dados transmitidos pelo monitor serial.



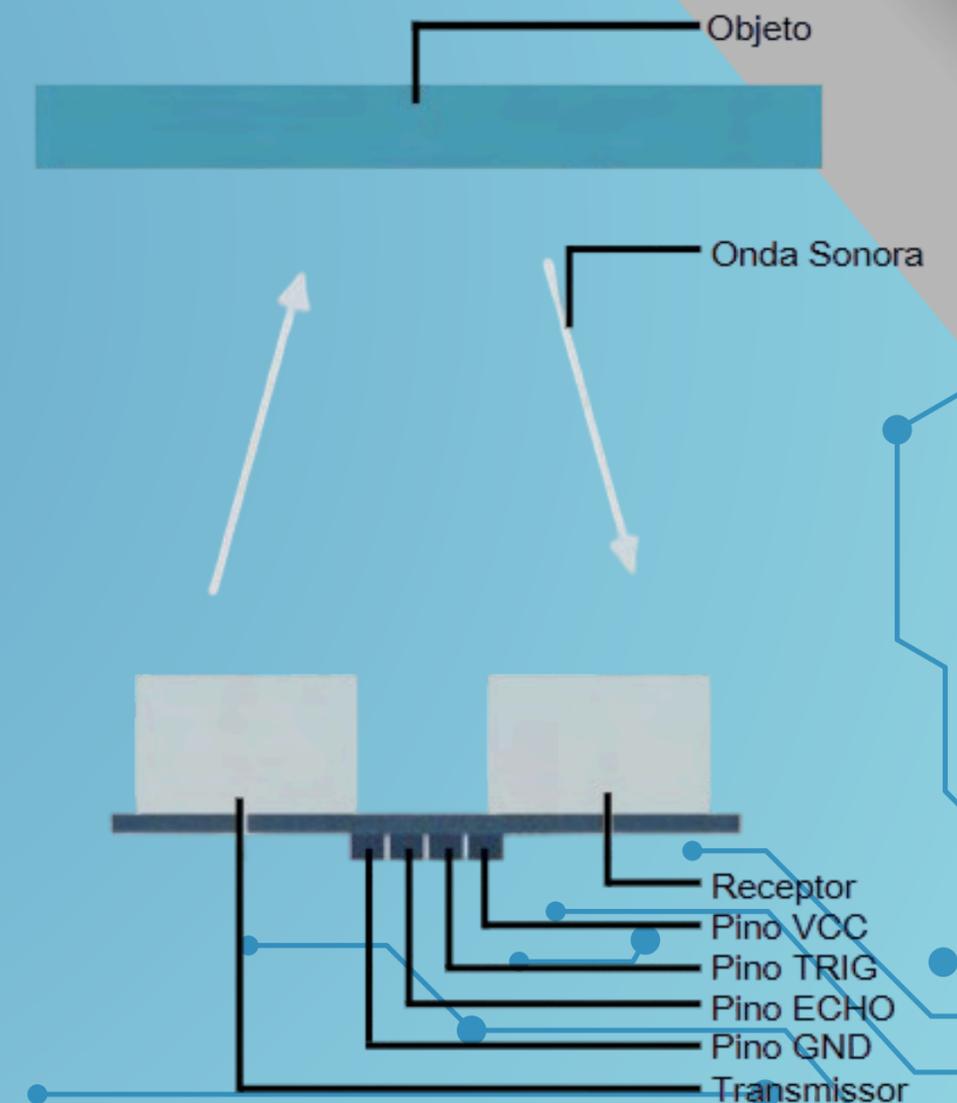
Texto

```
1 #include <Servo.h>
2
3 #define pinServo 7
4
5 Servo serv0;
6 int grau = 0;
7 int ultimoGrau = -1;
8
9 void setup() {
10   serv0.attach(pinServo);
11   Serial.begin(9600);
12 }
13
14 void loop() {
15   if (Serial.available() > 0) {
16     String entrada = Serial.readStringUntil('\n');
17     entrada.trim();
18
19     if (entrada.length() > 0) {
20       int novoGrau = entrada.toInt();
21       if (novoGrau >= 0 && novoGrau <= 180 && novoGrau != ultimoGrau) {
22         grau = novoGrau;
23         serv0.write(grau);
24         Serial.println(grau);
25         ultimoGrau = grau;
26       }
27     }
28   }
29 }
```

Ultrassônico HC-SR04

O sensor ultrassônico detecta objetos e mede distâncias usando um emissor e receptor de ondas sonoras em 40 kHz (inaudíveis ao ouvido humano). O tempo entre a emissão e o retorno do sinal refletido permite calcular a distância, dividindo-o por dois, já que o pulso percorre ida e volta. O sensor possui quatro pinos: VCC e GND (alimentação), TRIG (emissão) e ECHO (retorno). Sua faixa de medição vai de 2 cm a 4 m, podendo variar conforme o modelo.

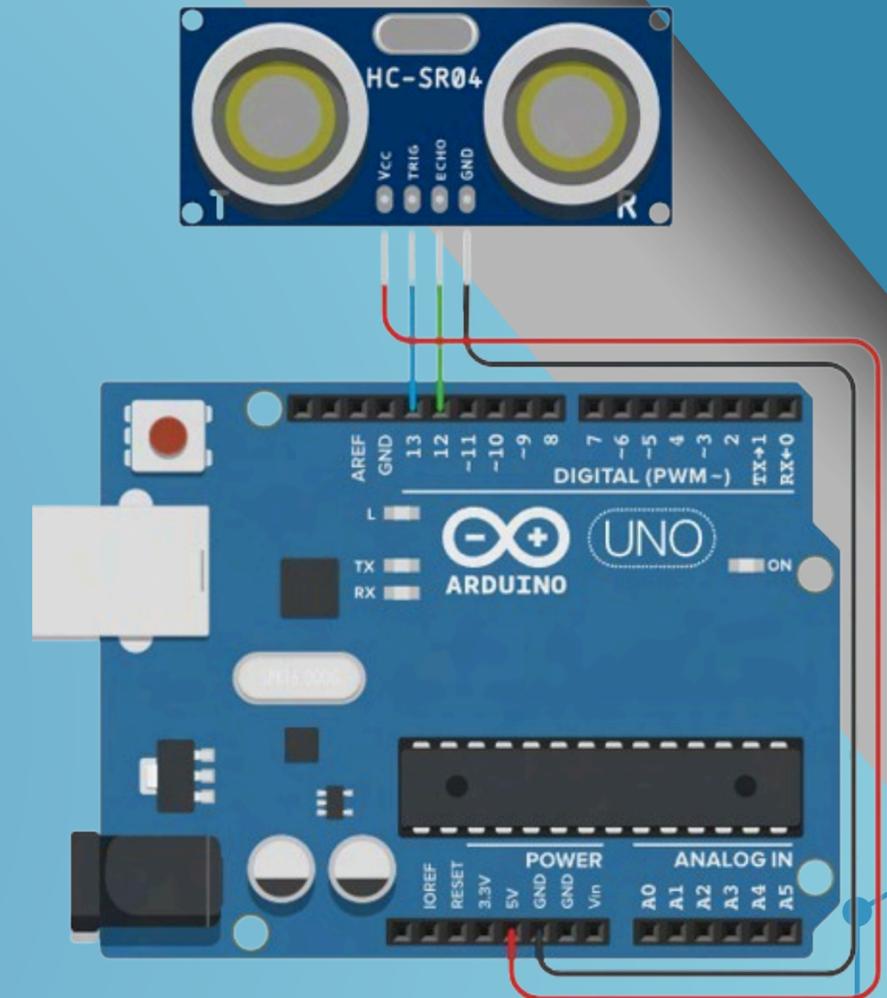
<https://www.arduino-libraries.info/libraries/ultrasonic>



Exemplo

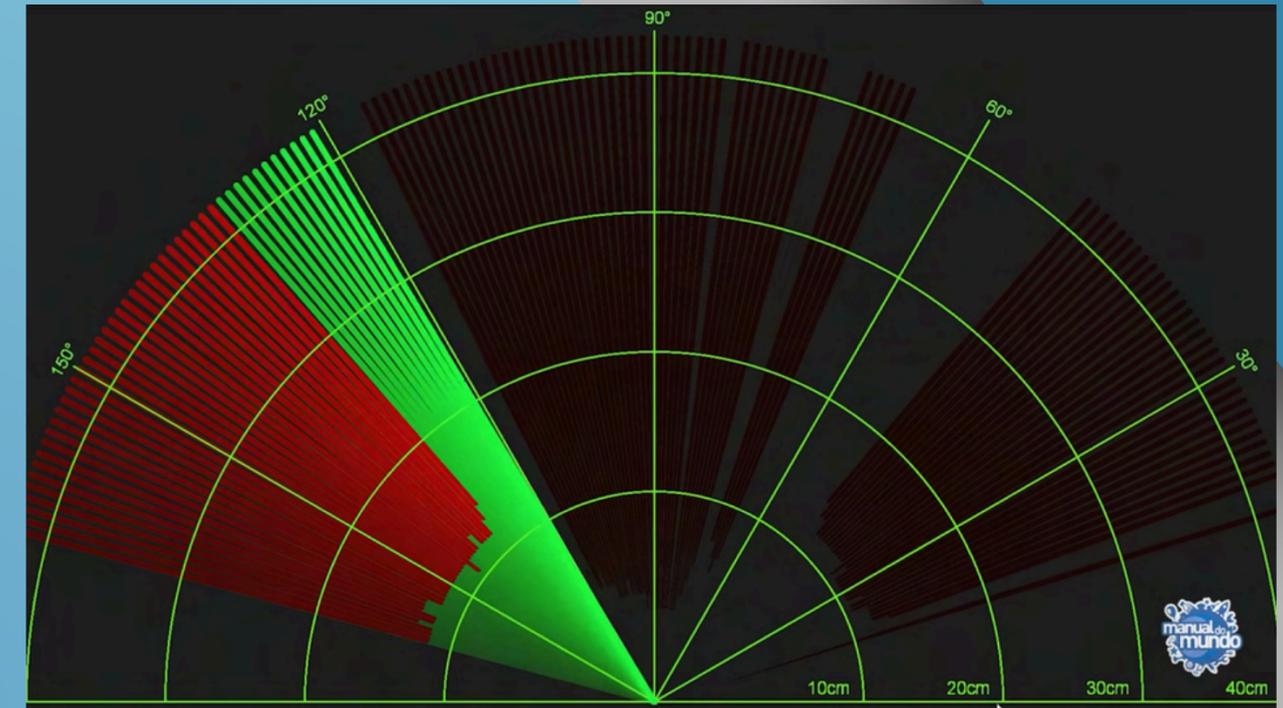
Exemplo de circuito com um sensor ultrassônico HC-SR04 que detecta a que distância está um objeto e fornece essa no monitor serial.

```
1 // baixar biblioteca 'Ultrasonic'
2
3 #include "Ultrasonic.h" //inclusão da biblioteca necessária
4
5 const int echoPin = 13; //pino digital utilizado pelo echo(recebe)
6 const int trigPin = 12; //pino digital utilizado pelo trig(envia)
7
8 Ultrasonic ultrasonic(trigPin,echoPin); //inicializando os pino do arduino
9
10 int distancia; //variável inteira
11 String result; //variável string
12
13 void setup(){
14   pinMode(echoPin, INPUT); //define o echoPin como entrada
15   pinMode(trigPin, OUTPUT); //define o trigPin como saída
16   Serial.begin(9600); //inicia a transmissão para monitor serial
17 }
18
19 void loop(){
20
21   hcsr04(); // faz a chamada do método "hcsr04()"
22
23   //imprime a distancia no monitor serial
24   Serial.print("Distancia= ");
25   Serial.print(result);
26   Serial.println("cm");
27 }
28
29 //método responsável por calcular a distância
30 void hcsr04(){
31   digitalWrite(trigPin, LOW); //seta o pino 12 com um pulso baixo "low"
32   delayMicroseconds(2); //intervalo de 2 microssegundos
33   digitalWrite(trigPin, HIGH); //seta o pino 12 com pulso alto "high"
34   delayMicroseconds(10); //intervalo de 10 microssegundos
35   digitalWrite(trigPin, LOW); //seta o pino 12 com pulso baixo "low" novamente
36
37   distancia = (ultrasonic.Ranging(CM)); //função ranging, faz a conversão para centímetros, e armazena na variável "distancia"
38
39   result = String(distancia); //variável result recebe a distância convertido de inteiro para string
40   delay(500); //intervalo de 500 milissegundos
41 }
```



Exemplo de Aplicação

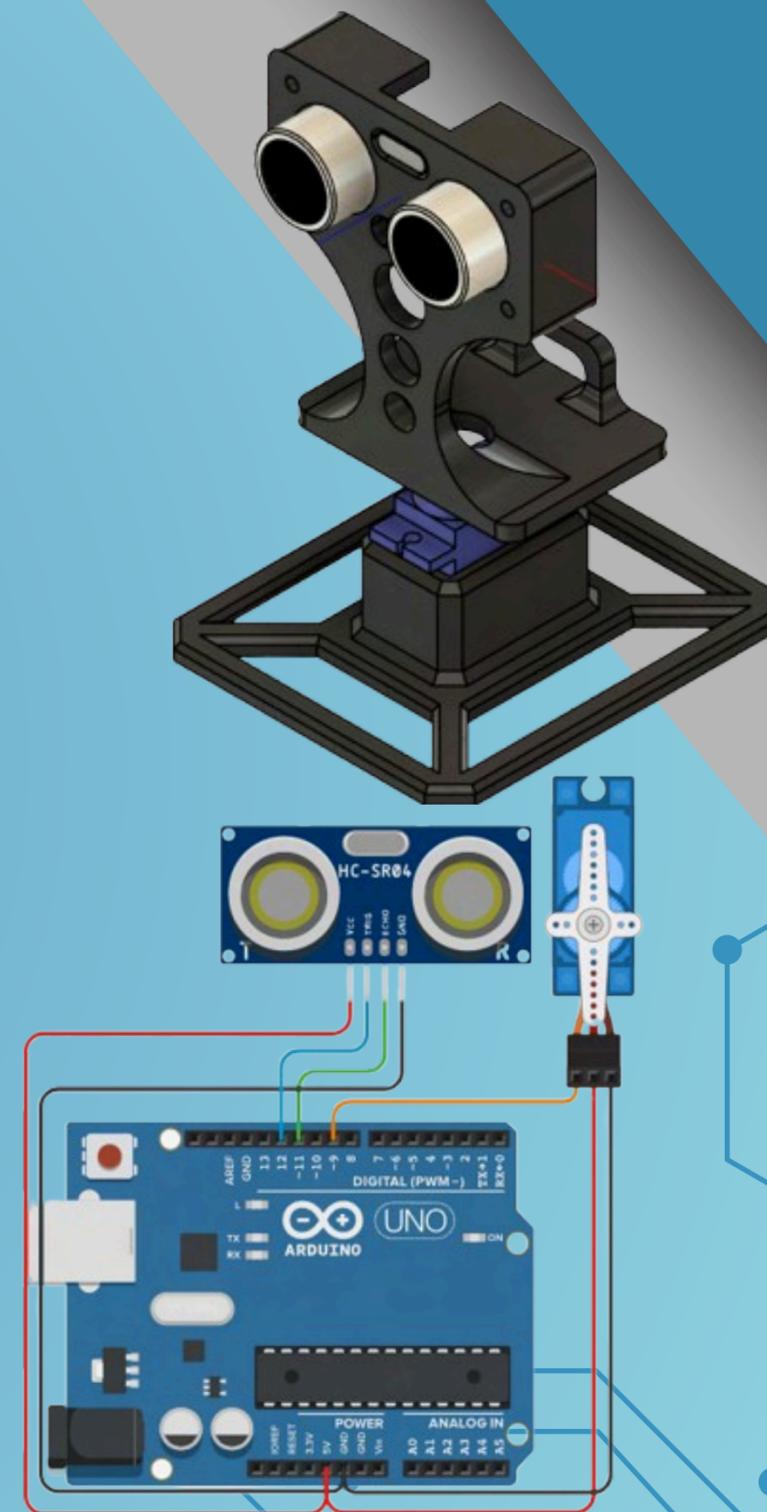
Sonar: um sistema que usa ondas sonoras para detectar e localizar objetos debaixo d'água. Ele pode ser ativo (emite sons e capta os ecos) ou passivo (escuta sons do ambiente), sendo usado em aplicações militares, pesquisas, pesca e navegação.



- HowToMechatronics, “Arduino Radar Project”, HowToMechatronics.com, [Online]. Disponível: <https://howtomechatronics.com/projects/arduino-radar-project>.
- Manual do Mundo, “Como ENXERGAR com SOM: SONAR caseiro”, YouTube, 15 jun. 2021. [Online]. Disponível: <https://www.youtube.com/watch?v=2z8xKo8yD14>.

Sonar com Arduino

Para construir um sonar usando Arduino será necessário um sensor ultrassônico, um servo motor e uma forma de integrar o sensor com o servo. Além disso, para que seja possível visualizar os dados coletados pelo dispositivo na forma de um radar, será necessário baixar o software Processing (<https://processing.org/download>), que é semelhante com o Arduino IDE, mas com mais possibilidades gráficas.



Como buscar no Tinkercad

- 01 - Clique na abade busca;
- 02 - Procure por “enrzan” e selecione a opção, pessoas.
- 03 - Por fim habilite a opção, circuitos, e navegue até o projeto desejado.

01

The image shows the Autodesk Tinkercad website interface. At the top left is the logo 'TINKERCAD AUTODESK Tinkercad'. The top navigation bar includes 'Tinker', 'Galeria', 'Aprenda', 'Professores', and 'Recursos'. On the right side of the navigation bar, there is a search icon (magnifying glass) and a user profile icon, both highlighted with red boxes. A left sidebar contains navigation options: 'Inicio', 'Classes', 'Projetos', 'Coleções', 'Tutoriais', 'Desafios', and 'Centro de ajuda...'. The main content area features a 'New Tinkercad challenges await you!' banner with a 'See what's next' button. Below this is an 'Upcoming Challenges' section with various icons. A '+ Criar' button is located on the right. The 'Projetos 3D' section includes a 'Crie seu primeiro projeto 3D' button and three tutorial cards: 'Place It', 'View It', and 'Move It'.

02



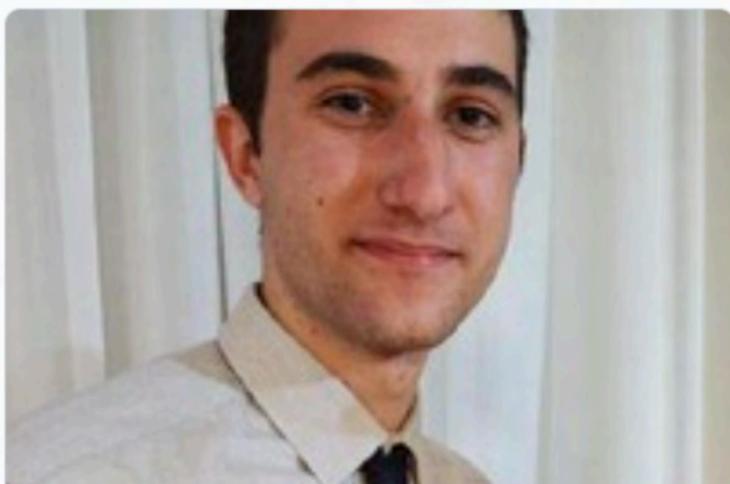
enrzan

Projetos 3D

Circuitos

Blocos de código

Pessoas



enrzan

03



Sinalizar usuário



enrzan

Ativo desde March 22, 2024

Sou Enrico Zangirolami Raimundo, discente da Universidade Federal do ABC no Bacharelado em Ciência e Tecnologia, com curso específico em Engenharia de Instrumentação, Automação e Robótica. Fui bolsista de Iniciação Científica em

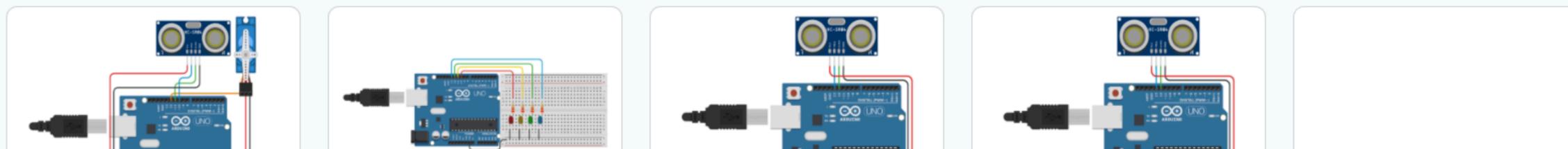
[Mostrar mais](#)

Projetos 3D **Circuitos** Blocos de código

Última edição ▾



Seleção de equipes Copiável Todos os usos ▾ Todos os tipos ▾



Para mais

Informações do autor, acesse:

- Site: https://enrzan.github.io/Site_enrzan/
- Instagram: [enrico.zangirolami](https://www.instagram.com/enrico.zangirolami)
- Lattes: <http://lattes.cnpq.br/0406779293160585>
- LinkedIn: <https://www.linkedin.com/in/enrico-zangirolami-raimundo-a85987359>

Projetos e conteúdos relacionados:

- Github: github.com/enrzan
- Tinkercad: <https://www.tinkercad.com/users/5zIXzHxFWNF?type=circuits>

Meus site, que
contem todos os
demais links.



**Obrigado pela Atenção.
Espero que tenham gostado.**

