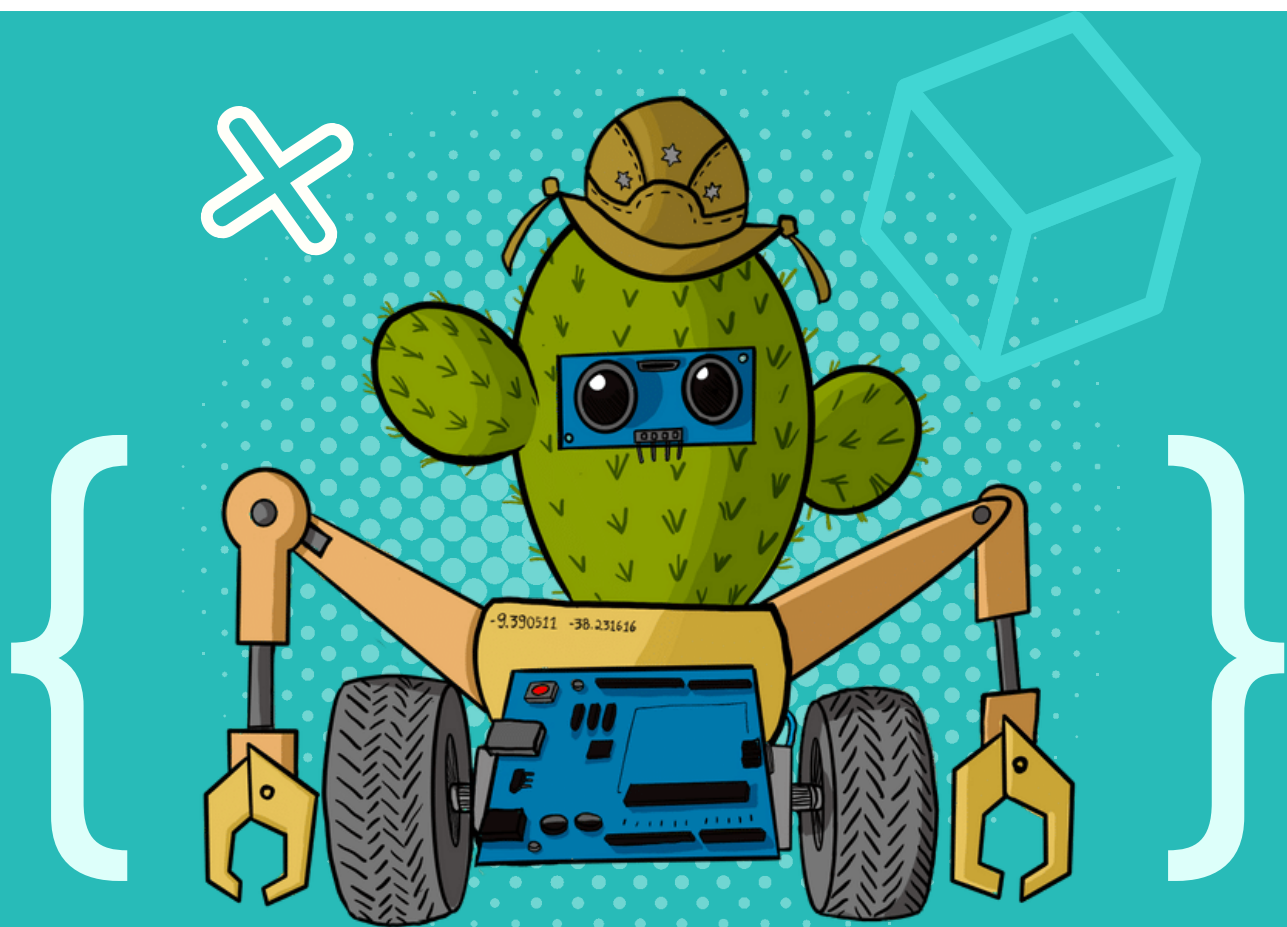


# ROBÔCACTUS

Sua apostila para o estudo de Arduino, lógica de programação e robótica;



## EXPEDIENTE TÉCNICO

**Bruno Amorim Ramos**

**Prof. Dr. Eduardo Cardoso Moraes**



**Dados Internacionais de Catalogação na Publicação**  
**Instituto Federal de Alagoas**  
**Campus Avançado Benedito Bentes**  
**Biblioteca**

R175r

Ramos, Bruno Amorim.

Robôcactus: sua apostila para o estudo de Arduino, lógica de programação robótica / Bruno Amorim Ramos; Eduardo Cardoso Moraes. – 2021.

46 f. : il.

1 CD-ROM: il.

Produto Educacional da Dissertação Robôcactus: uma proposta para a prática de lógica de programação por meio da robótica educacional e da plataforma Arduino na educação profissional e tecnológica (Mestrado em Educação Profissional e Tecnológica) Instituto Federal de Alagoas, Campus Avançado Benedito Bentes, Maceió, 2021.

1. Educação. 2. Metodologia Ativas. 3. Construcionismo. 4. Plataforma Arduino.  
5. COVID 19. I. Título. II. Moraes, Eduardo Cardoso.

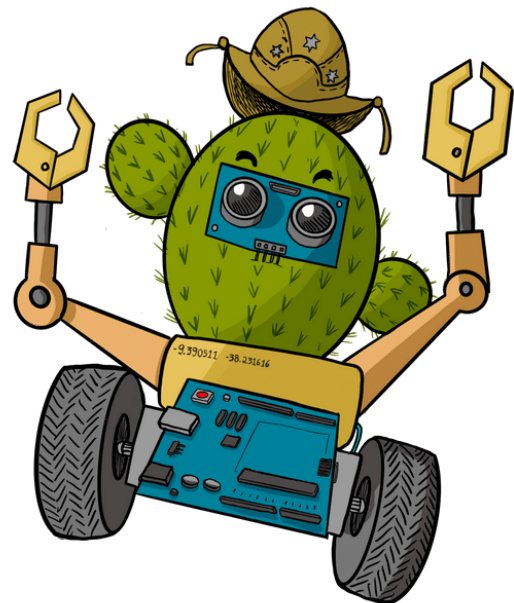
CDD: 370

**Fernanda Isis Correia da Silva**  
**Bibliotecária - CRB-4/1796**

## SAUDAÇÕES!

Começaremos nossa jornada nas áreas de robótica, programação e desenvolvimento de tecnologias. Para tanto, utilizaremos a plataforma Arduino como ferramenta para introduzir os conhecimentos necessários ao seu desenvolvimento. Também aprenderemos os princípios da robótica, programação de robôs, montagem, conexões elétricas, sensores, etc., para que, ao final da jornada, possamos construir nosso primeiro projeto robótico juntos.

O Robôcactus trata-se de um projeto criado no sertão brasileiro com o intuito de aproximar a robótica educacional às escolas de todo o país, hoje ele será seu companheiro de equipe, divirtam-se!



Atenciosamente,

**ROBÔCACTUS**

*Para mais informações:*

[brunno00amorim@gmail.com](mailto:brunno00amorim@gmail.com)

# SUMÁRIO

## Módulo I – Vamos falar sobre robótica?

<b>1. Robótica</b> .....	1
1.1 Robótica Industrial .....	1
1.2 Robótica Educacional .....	2
1.3 Inteligência artificial .....	3
1.5 Atividade M.I.....	5

## Módulo II – E o Arduino, que tal?

<b>2. A plataforma Arduino</b> .....	8
2.1 Hardware Arduino Uno.....	9
2.2 Software Arduino .....	12
2.3 Atividade M.II .....	15

## Módulo III – A lógica das máquinas

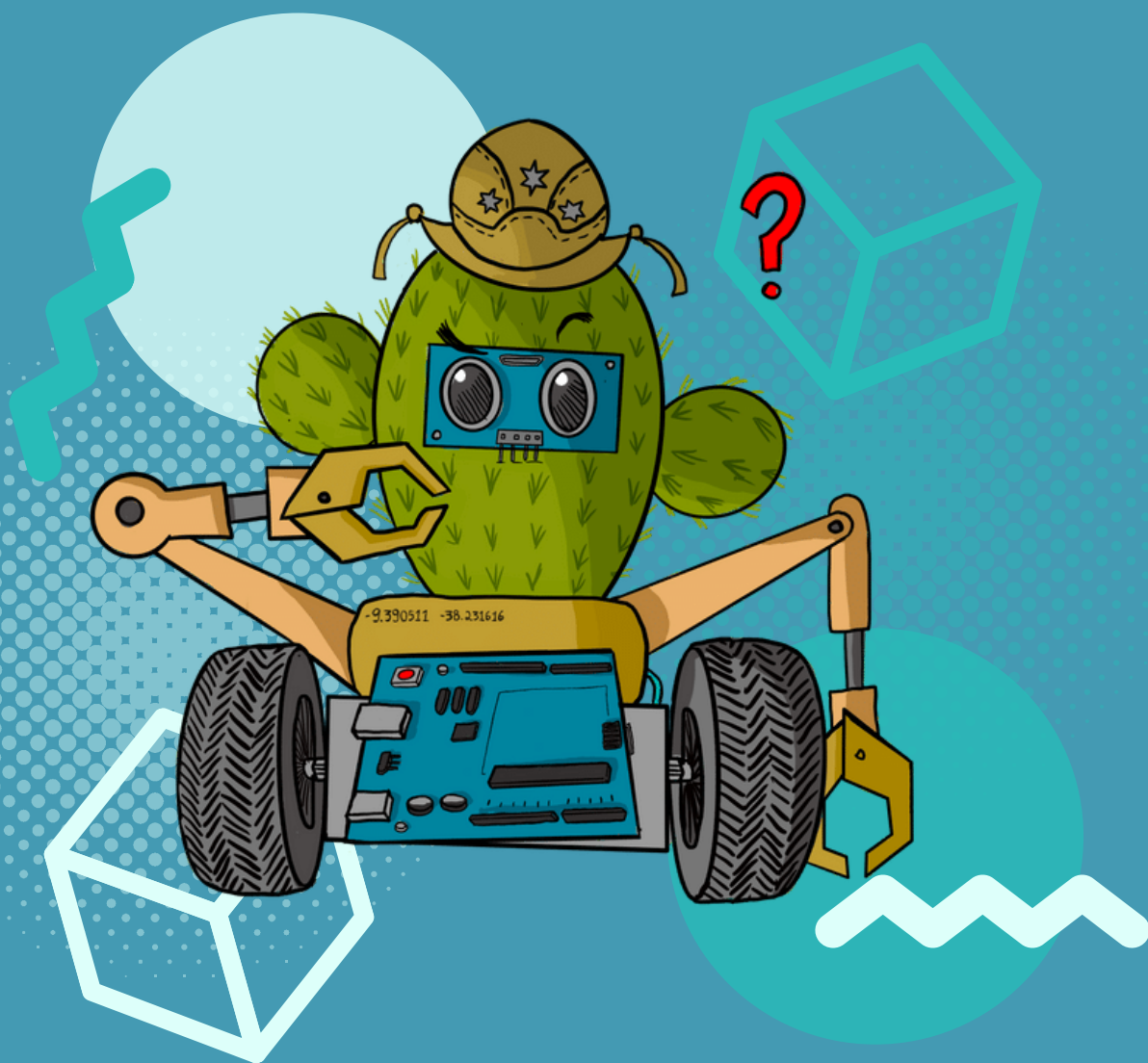
<b>3. Lógica e linguagem de programação</b> .....	19
3.1 Programando e testando sua placa.....	20
3.2 Régua eletrônica com Arduino.....	23
3.2.1 Sensor - HC-SR04.....	23
3.2.2 Conectando o sensor HC-SR04 ao Arduino.....	24
3.2.3 Codificando nosso projeto.....	25
3.3. Atividade M.III.....	27

## Módulo IV– projetando o futuro

<b>4. Conhecendo seu kit robótico</b> .....	29
4.1 Protoboard .....	29
4.2 LED .....	30
4.3 Resistores .....	30
4.2 Projeto sistema de alerta .....	31
4.4.1 Esquema elétrico.....	31
4.4.2 Programando o sistema de alerta.....	33
5. Atividade M.IV.....	36
6. DESAFIO .....	37

# MÓDULO I

Vamos falar sobre robótica?



<b>1. Robótica</b> .....	1
1.1 Robótica Industrial .....	1
1.2 Robótica Educacional .....	2
1.3 Inteligência artificial .....	3
1.5 Atividade M.I .....	6

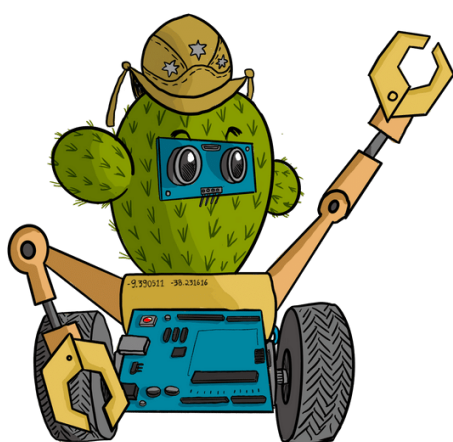
# Vamos falar sobre robótica ?

---

## 1. Robótica

Robótica é uma ciência que combina mecânica, eletrônica e ciência da computação. Seus estudos permitem construir máquinas programadas para realizar tarefas, robôs. Existem vários tipos de robôs, e sua classificação, e aplicação dependem de seus sensores, cadeia cinemática de movimento, autonomia, etc. Várias são as áreas nas quais o robô pode desempenhar um papel, seja para indústria, medicina, segurança, entre outras.

Para entendermos como o robô funciona, precisaremos identificar suas partes físicas e lógicas. O **hardware** (parte física) é feito de peças como Arduino, sensores, motores, fios e todos os componentes que podemos tocar, e o **software** (parte lógica), é um programa, um ambiente de desenvolvimento de códigos de computador (IDE) que vão controlar o hardware.



### Estrutura de um robô

Funciona como uma imitação do corpo humano. Considere que nosso corpo é a junção das peças físicas e a mente o programa controlador.

---

## 1.1 Robótica Industrial

O ramo denominado robótica industrial visa automatizar a criação de produtos, como é o caso da montagem de automóveis em grandes empresas como Chevrolet, Toyota e Fiat. A maior parte da pintura, montagem, manutenção e acabamento dos carros é feita por máquinas, pois um robô bem programado repete seus movimentos com precisão, sem interrupção e inúmeras vezes mais rápido que um humano. Um robô pode realizar uma tarefa perigosa, portanto, evita danos a uma pessoa assim como os robôs usados para desarmar bombas. A robótica é, portanto, uma ciência essencial para a evolução industrial e garante eficiência, segurança e agilidade nos processos de produção.

---

# Vamos falar sobre robótica?

---

## 1.3 Robótica Educacional

A associação da robótica com a educação deu origem a robótica educacional, ou ainda, robótica pedagógica; um ambiente dinâmico que utiliza a robótica para o estudo de diversas disciplinas, desenvolvimento de habilidades e tecnologias.

---

## 1.4 Introdução à inteligência artificial

São muitos os conhecimentos que participam do conceito de Inteligência Artificial (IA), não iremos aqui definir exatamente o que é, mas conhecer e refletir a partir das seguintes definições:

“IA é o estudo de como fazer os computadores realizarem tarefas as quais, até o momento, os homens fazem melhor” (Rich, 1994).

“A Inteligência Artificial é uma disciplina científica que utiliza as capacidades de processamento de símbolos da computação com o fim de encontrar métodos genéricos para automatizar atividades perceptivas, cognitivas e manipulativas, por via do computador.” (Pereira, 1988)

“Durante milhares de anos, procuramos entender como pensamos, isto é, como um mero punhado de matéria pode perceber, compreender, prever e manipular um mundo muito maior e mais complicado que ela própria. O campo da inteligência artificial, ou IA, vai ainda mais além: ele tenta não apenas compreender, mas também construir entidades inteligentes.” (Russel, 2004)

Criar máquinas que possam agir, pensar e aprender como os seres humanos é um desejo antigo, quantos filmes, seriados e histórias você conhece com essa temática? muitos não!?

Muitas são as áreas que a IA está interligada, hoje, está presente principalmente nos sistemas especializados; programas de computador que utilizam uma base de dados e um sistema de algoritmos para processar dados e gerar informações. As definições da IA classificadas abaixo, são ainda, ramificações ou caminhos de estudo<sup>3</sup>.

---

# Vamos falar sobre robótica?

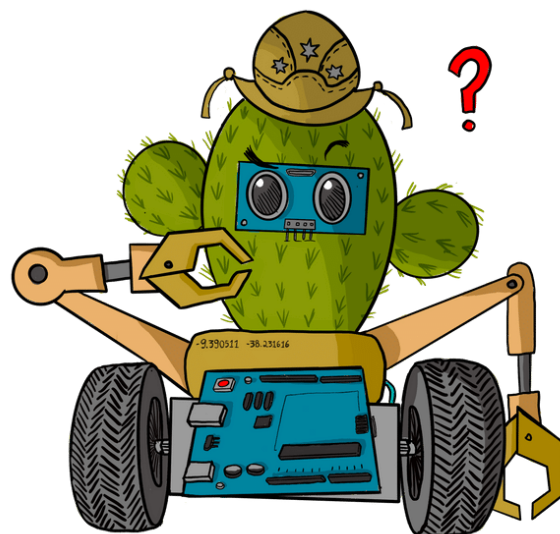
---

Categorias de definição da IA:

- (a) Sistemas que pensam como humanos
- (b) Sistemas que agem como humanos
- (c) Sistemas que pensam logicamente
- (d) Sistemas que agem logicamente

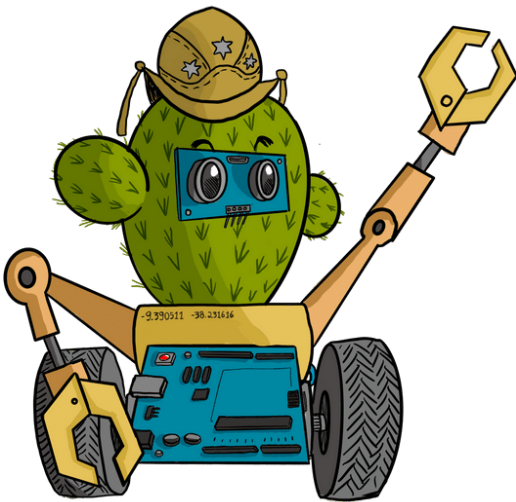
Curiosamente, o computador pode realizar muitos cálculos complicados em milissegundos, mas não consegue distinguir uma cadeira de metal de uma de madeira, o que até uma criança de 3 anos faz. <sup>1</sup> Para que um robô aprenda e compreenda as informações, sua programação e suas partes físicas devem ser capazes de interagir com a realidade, para isso a inteligência artificial utiliza conhecimentos de diversas áreas, como psicanálise, filosofia, psicologia, matemática, informática, medicina, entre outros. Não é uma tarefa fácil, mas estamos avançando e em breve iremos descobrir máquinas com essa capacidade.

**TUDO BEM, MAS O  
QUE GARANTE QUE AS  
MÁQUINAS NÃO VÃO  
NOS DOMINAR?**



# Vamos falar sobre robótica?

Com isso em mente, Isaac Asimov em seu livro de 1950 "I Robot" criou a lógica da proteção humana que deve ser aplicada na robótica e na inteligência artificial, ou seja, as leis das 3 leis da robótica. São elas:



**01 Primeira Lei:** Um robô não pode ferir um ser humano ou, por inação, permitir que um ser humano sofra algum mal.

**02 Segunda Lei:** Um robô deve obedecer às ordens dadas por seres humanos, exceto nos casos em que tais ordens entrem em conflito com a Primeira Lei.

**03 Terceira Lei:** Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira ou a Segunda Lei.

# Vamos falar sobre robótica?

---

## 1.5 Atividade M.I

1) Assista os seguintes vídeos na plataforma do YouTube para completar nosso conteúdo.



Título: **A origem dos robôs**

Plataforma: YouTube

Canal: A origem das coisas

link: <https://youtu.be/5lh3OtqkLMU>



Título: **A história da inteligência artificial**

Plataforma: YouTube

Canal: Tecmundo

link: <https://youtu.be/Lhu8bdmkMCM>

2) Escreva sobre robótica, seu significado, tipos e a importância dessa ciência para humanidade. Cite exemplos.

3) O que você entende sobre inteligência artificial? Cite alguns exemplos de como essa tecnologia pode ajudar a humanidade.

4) Qual a diferença entre robótica, robótica industrial e educacional?

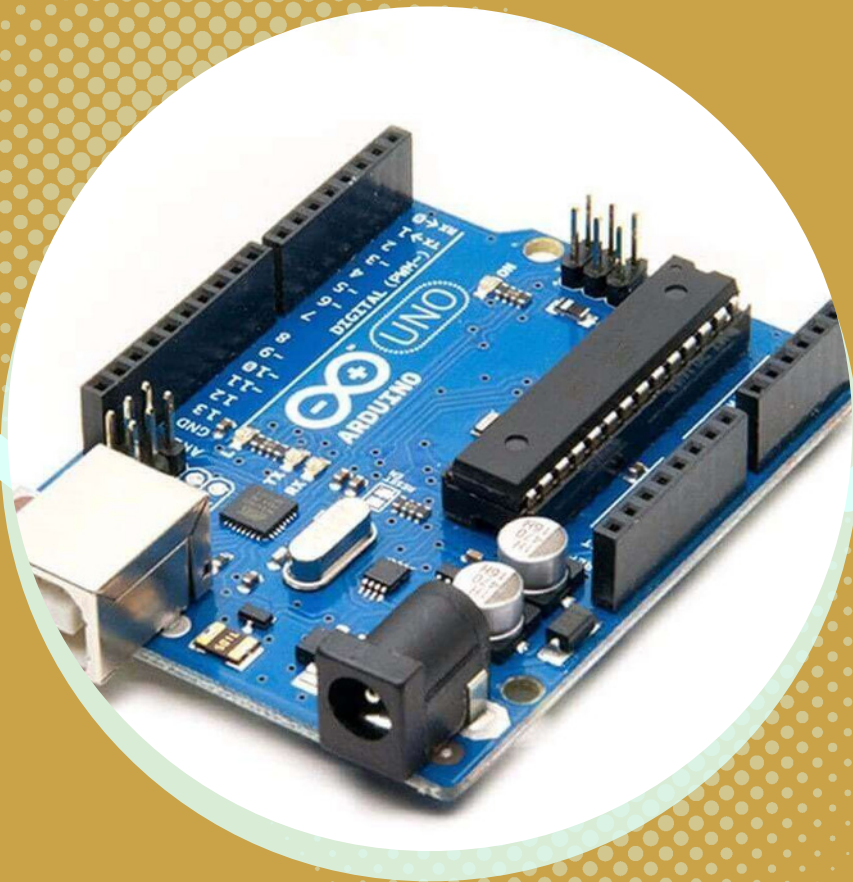
5) Imagine que você vai construir um robô que vai resolver um problema em sua casa, bairro, trabalho, escola ou outro lugar que você visita constantemente, descreva o problema e como sua ideia funcionará.

Obs: A imaginação é livre, tente identificar o problema que o robô poderia resolver.

---

# MÓDULO II

E o Arduino, que tal?



<b>2. A plataforma Arduino</b> .....	8
2.1 Hardware Arduino Uno.....	9
2.2 Software Arduino .....	12
2.3 Atividade M.II .....	15

# E o Arduino, que tal?

---

## 2. A plataforma Arduino

Criado por Massimo Banzi em 2005 na Itália, o Arduino é uma placa de teste eletrônica na qual você pode prototipar robôs, testar, modificar, estender e reutilizar peças sempre que julgar necessário. Seu ambiente é dividido em duas partes, hardware que consiste em uma placa física e software; Ambiente de desenvolvimento Arduino.

É um sistema embarcado, ou seja, seu conjunto físico e lógico é dedicado a realizar uma tarefa específica, assim como os dispositivos de micro-ondas que possuem um programa embutido para controlar o temporizador e seus sensores, um programa projetado apenas para esta função, impossível por exemplo, acessar a Internet ou tocar uma música enquanto se faz pipoca (por enquanto).

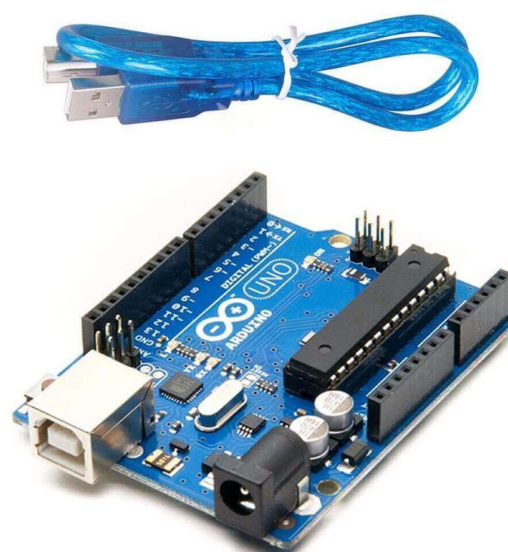
Funciona programando o microprocessador, que por sua vez controla o armazenamento, a memória, os circuitos de entrada e saída (portas). Podemos também adicionar outras peças para estender sua funcionalidade, são chamadas de **Shields** (escudos) e podem ser sensores de presença, sensores de temperatura, receptores GPS, atuadores, display de cristal líquido (LCD), módulos de controle de motor, ethernet, wireless, bluetooth e outros componentes de que seu projeto precisar.

Existem diversos modelos de Arduino no mercado, cada um com funções específicas, seja para construção de drones, controles de videogame, microplacas para pequenos sistemas, ou mesmo placas com mais portas para interagir com mais componentes. Entrando no site: [www.arduino.cc/en/Main/Products](http://www.arduino.cc/en/Main/Products) você poderá ver alguns modelos e suas funcionalidades.

A versão escolhida para nossos projetos é o **Arduino Uno**, disponível na Figura 1, uma placa versátil e acessível capaz de atender a projetos simples e robustos.

---

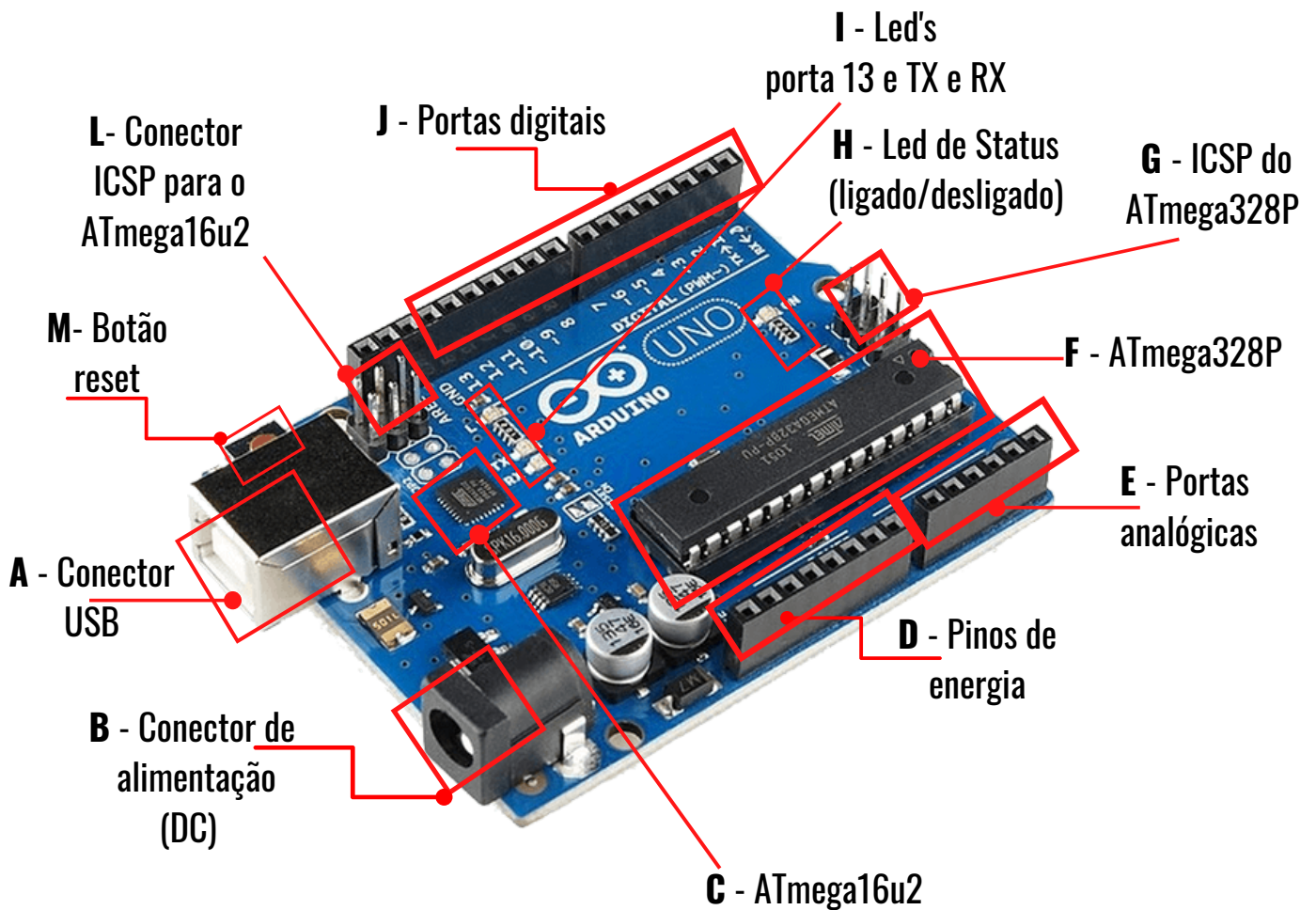
Figura 1 - Arduino Uno



# E o Arduino, que tal?

## 2.1 Hardware Arduino Uno

A placa Arduino Uno tem três versões disponíveis até agora, seu código e design são abertos e podem ser facilmente duplicados, motivo pelo qual existem várias placas de outros fabricantes no mercado. Vamos conhecer algumas funções e características do seu hardware.



# A

**Conector USB** - conecta o Arduino ao computador e funciona como fonte de alimentação (5v).

# C

**ATmega16u2** - Microcontrolador responsável por estabelecer a comunicação e tradução dos sinais USB / Serial.

# B

**Conector de alimentação(DC)**- Jack de alimentação de energia externa. Caso precise utilizar uma fonte externa , ou ainda, não utilizar a porta USB como fonte de energia. A tensão de entrada é de 7 a 20 Volts, porém, tensões maiores que 12v podem ocasionar danos a placa.

# E o Arduino, que tal?

**D** **Pinos de energia** - esses conectores são utilizados para fornecer saídas de tensão de 5v e 3,3v .

**IOREF** – Esse pino é uma de referência de entrada/saída. Fornece a referência de tensão na qual o microcontrolador opera, podendo ser adaptada para 3,3V ou 5V.

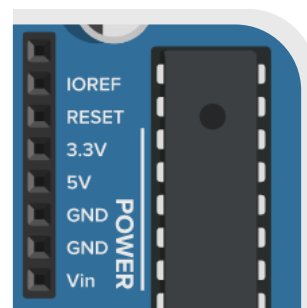
**RESET** – Pino conectado ao botão RESET, pode ser utilizado para resetar a placa de forma externa.

**3,3V** – Fornece tensão de 3,3V para alimentação de shields com corrente máxima de 50 mA.

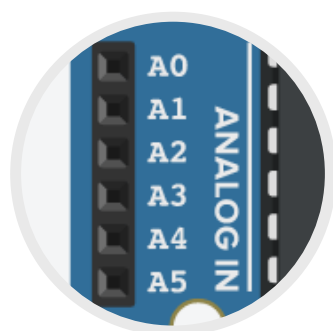
**5V** – Fornece tensão de 5V com corrente de 50 mA.

**GND** – Filtro de densidade neutra, terra.

**VIN** – Pino para alimentar a placa de uma fonte externa, seja um shield ou uma bateria. Possui um regulador de tensão que estabiliza o valor de entrada para 5V.



**D** - Pinos de energia



**E** - Portas analógicas

**E** **Portas analógicas** - o microcontrolador ATmega328 possui um conversor (A/D) de grandezas de 10 bits de resolução, com a capacidade de converter sinais digitais para analógicos. Essas portas são destinadas para uso de componentes que trabalham com medidas que variam a tensão entre 0V e 5V, como por exemplo, um potenciômetro que controla a luminosidade um led, sendo que, quanto menor valor, mais fraca será a luz do led e quanto maior o valor, maior será a luminosidade deste.

**F** **ATmega328P** - É o microcontrolador do Arduino Uno, apesar de pequeno ele contém circuitos de memória, um processador e interfaces para o processamento das informações e a gerencia das portas de entrada e saída. É no microcontrolador que fica salvo o código que programamos.



**F** - ATmega328P

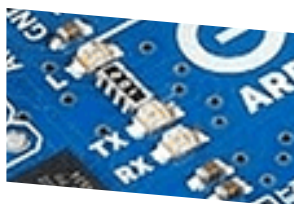
# E o Arduino, que tal?



## G - ICSP do ATmega328P

G

**ICSP do ATmega328P** – Os pinos ICSP (*in circuit system programmer*) são a porta para programar diretamente os microcontroladores da placa usando o protocolo serial SPI (*serial peripheral interface*) caso seja necessário modificar o firmware ou bootloader do microcontrolador.



## I - Led's porta 13 e TX e RX

I

**Led's porta 13 e TX e RX** – Ao lado da porta 13 tem um led, esse pode ser utilizado para testes, já os leds TX e RX são responsáveis por sinalizar a transmissão e recepção de dados entre a placa e o computador ou outra placa.



## L - ICSP do ATmega16u2

L

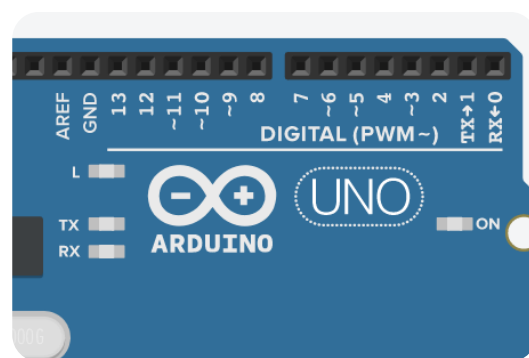
**ICSP do ATmega16u2** – porta para programar diretamente o microcontrolador ATmega16u2.



## H - Led de Status (ligado/desligado)

H

**Led de Status** – Led que informa se seu Arduino está ligado.

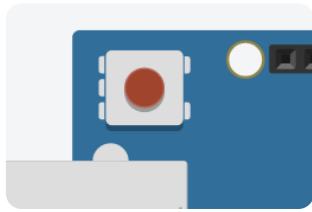


## J - Portas digitais

J

**Portas digitais** – O Arduino Uno possui 14 portas digitais (0 a 13) que trabalham enviando ou recebendo impulsos elétricos com valores lógicos de **ligado** ou **desligado**. Seu valor lógico alto, **HIGH**, é de 5V e seu nível lógico baixo, **LOW**, é de 0V. São Através dessas portas que controlaremos as peças, atuadores, sensores, motores dos nossos projetos. Observe que algumas portas estão marcadas com um “~”, essas são as portas **PWM** (Pulse Width Modulation) capazes de variar a largura de um pulso digital **ligado** ou **desligado** para valores entre **0 e 255**, possibilitando o controle de velocidade, intensidade de brilho de LEDs, posicionamento de motores, entre outros.

# E o Arduino, que tal?

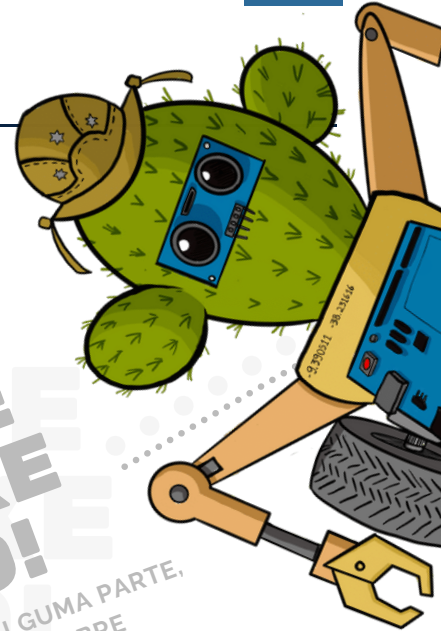


## M - Botão de Reset

M

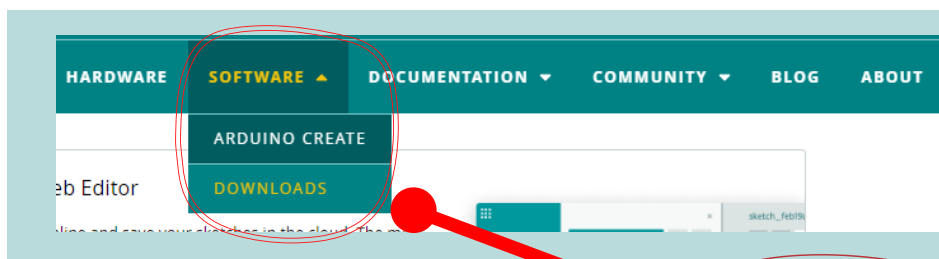
**Botão de Reset**– Serve para reiniciar o Arduino, tem o mesmo efeito de desligar e ligar a placa.

EU, SE  
AVEXE  
NÃO!  
SE VOCÊ ESQUECER ALGUMA PARTE,  
VOLTE AQUI E RELEMBRE



## 2.2 Software Arduino

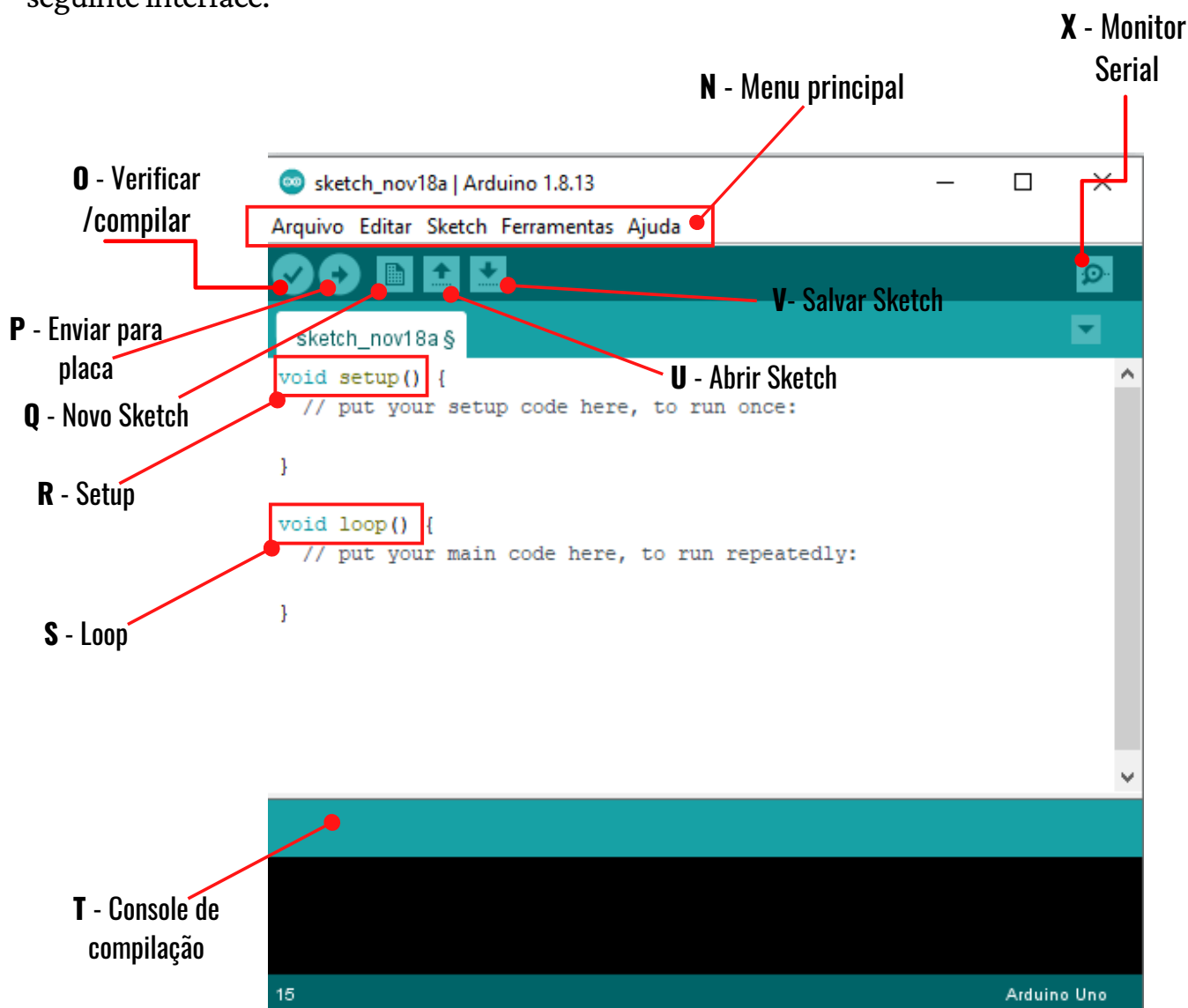
O programa que iremos utilizar para construir nossos códigos trata-se de um ambiente de desenvolvimento integrado (IDE) projetado para compilar e enviar códigos para a placa Arduino, e pode ser encontrado gratuitamente no site da plataforma <https://www.arduino.cc/en/software>, basta selecionar a versão compatível com o sistema operacional para instalar e fazer download.



<https://www.arduino.cc/en/software>

# E o Arduino, que tal?

A linguagem de programação da plataforma Arduino é C++ com a adição de bibliotecas e funções específicas da plataforma, os projetos criados no programa são chamados de Sketch. Vamos conhecer esse ambiente. Logo após instalar o programa, você encontrará a seguinte interface:



**N** **Menu principal** - A barra de menu da IDE conta com uma série de funções importantes como importar códigos de exemplo na aba Arquivos, ou ainda configurar a porta COM de conexão Arduino/computador na aba Ferramentas, entre outras.

**O** **Verificar/compilar** - Essa função realiza a compilação do código. Verifica erros de sintaxe e semântica, se o código está estruturado corretamente e a falta de sinais, e ainda aponta erros no seu projeto.

# E o Arduino, que tal?

**P** - Enviar para placa



**P**

**Enviar para placa** - Após verificar seu *Sketch* é com esse botão que embarcamos o código na placa.

```
void setup() {
  // put your setup code here, to run once:
}
```

**R** - Setup

**R**

**Setup** - Assim que é iniciado o funcionamento da plataforma a primeira função chamada (lida) pelo Arduino é o **setup()**, nesta, configuramos as portas, bibliotecas, variáveis e tudo que precisa ser informado antes de começar. Essa função é chamada apenas uma vez assim que ligada ou resetada a placa.



**T** - Console de compilação

**T**

**Console de compilação** - Aqui são apresentados os erros de compilação, comunicação, falhas de carregamento, erros de sintaxe no código, numeração e identificação dos erros para que você possa pesquisar as possíveis soluções.

**V** - Salvar



**V**

**Salvar**- Botão para salvar as alterações do seu código, é importante não esquecer de salvar seus *Sketchs* em um local seguro e acessível.

**Q** - Novo Sketch



**Q**

**Novo Sketch**- Cria um novo projeto

```
void loop() {
  // put your main code here, to run repeatedly:
}
```

**R** - Loop

**S**

**Loop** - depois do `setup()` é chamada a função `loop()`, trata-se da função principal do código, o local de programação dos nossos projetos robóticos. Após programado e embarcado o código essa função repetirá os comandos consecutivamente enquanto a placa estiver ligada.

**U** - Abrir Sketch



**U**

**Abrir Sketch**- Caso você precise abrir um *Sketch* no seu computador ou dispositivo de armazenamento externo.

**X**- Monitor Serial



**X**

**Monitor Serial** - é uma tela para troca de dados com o Arduino via USB. Através desta, é possível visualizar os dados obtidos de um sensor, ou ainda, enviar comandos para a placa.

# Vamos falar sobre robótica

## 2.3 Atividade M.II

1) Assista o vídeo na plataforma YouTube para completar nosso conteúdo.



Título: **O que é Arduino, afinal de contas? #ManualMaker**

**Aula 4, Vídeo 1**

Plataforma: YouTube

Canal: **Manual do Mundo**

link: [https://www.youtube.com/watch?](https://www.youtube.com/watch?v=sv9dDtYnE1g&t=96s&ab_channel=ManualdoMundo)

[v=sv9dDtYnE1g&t=96s&ab\\_channel=ManualdoMundo](https://www.youtube.com/watch?v=sv9dDtYnE1g&t=96s&ab_channel=ManualdoMundo)

2) Qual a diferença entre as duas partes da plataforma Arduino?

3) O que significa afirmar que o Arduino é um sistema embarcado? cite exemplos de sistemas embarcados.

4) O que são Shields? pesquise e cite exemplos.

5) Sobre o hardware do Arduino, relacione as colunas abaixo:

( ) Entrada da placa para tensão externa.

( ) Reinicia a placa.

( ) Envia e recebe impulsos elétricos HIGH e LOW para ligado e desligado.

( ) Armazena o código e controla as portas de entrada e saída.

( ) Sinaliza o recebimento ou envio de dados na placa.

( ) Informa se a placa está ligada;

( ) Nessas são conectados os sensores, atuadores, componentes que utilizam sinais que variam a tensão entre 0V a 5V.

1 – Conector de alimentação (DC)

2 – Microcontrolador ATmega328P

3 – Portas digitais

4 – LEDs TX e RX

5 – Botão Reset

6 – Led on/off

7 – Portas analógicas

6) Estamos começando a desenvolver um robô limpador. Qual o local no código em que iremos configurar as portas e bibliotecas ?

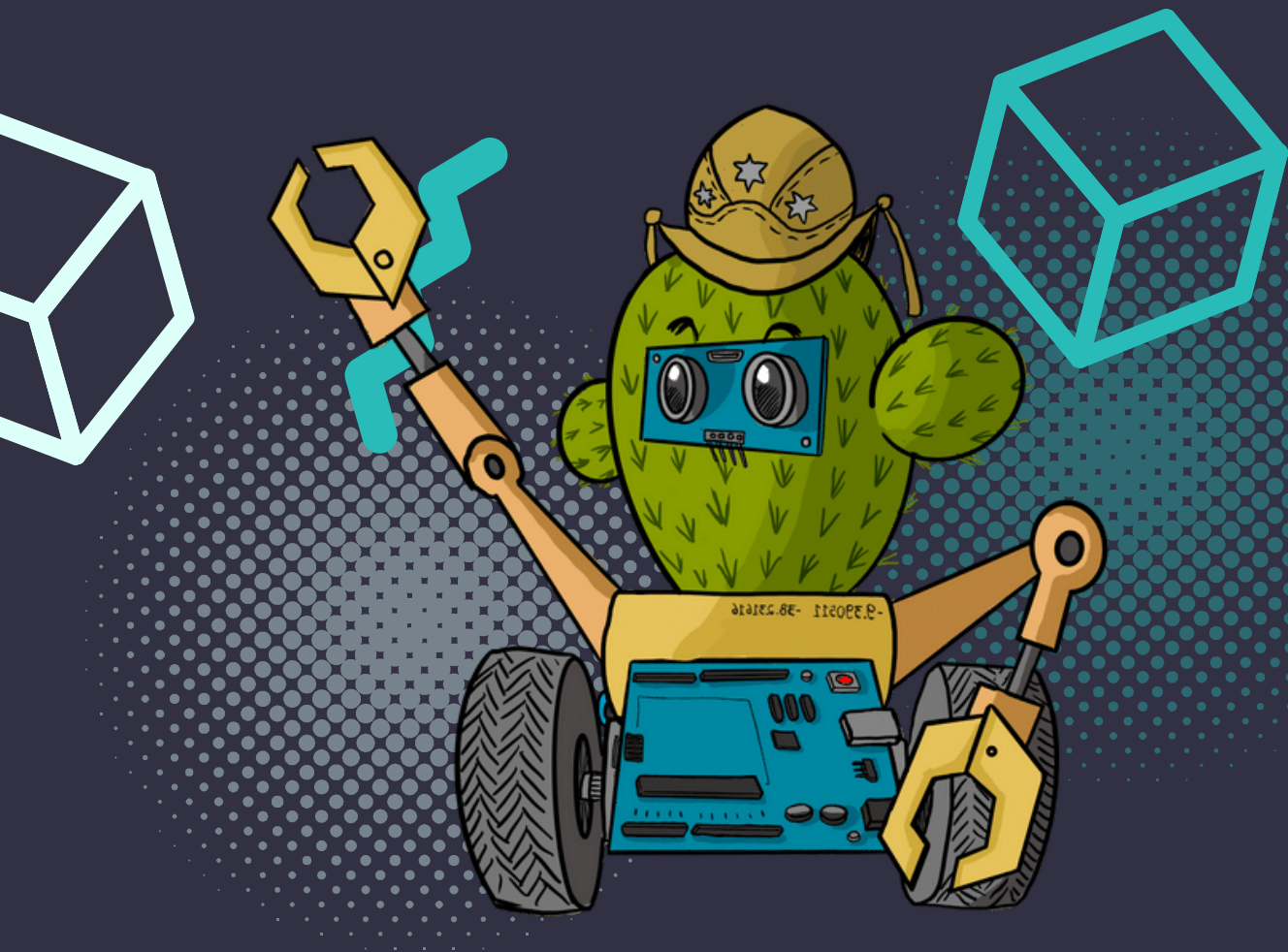
# E o Arduino, que tal?

---

- 7) Nosso robô tem um sensor de distância ultrassônico que calcula a distância entre o sensor e um objeto. Qual a janela na IDE do Arduino que apresenta os dados no computador?
  - 8) Qual a opção na IDE do Arduino em que configuramos a porta COM de conexão Arduino/computador?
  - 9) Qual a diferença entre as funções `loop()` e `setup()`?
-

# MÓDULO III

## A lógica das máquinas



<b>3. Lógica e linguagem de programação.....</b>	<b>19</b>
3.1 Programando e testando sua placa.....	20
3.2 Régua eletrônica com Arduino.....	23
3.2.1 Sensor - HC-SR04.....	23
3.2.2 Conectando o sensor HC-SR04 ao Arduino.....	24
3.2.3 Codificando nosso projeto.....	25
<b>4. Atividade M.III.....</b>	<b>27</b>

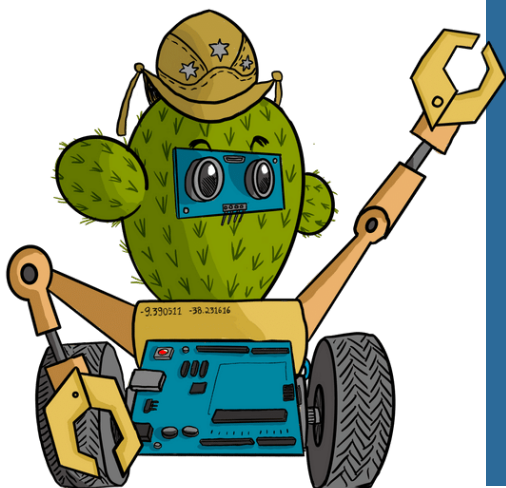
# A lógica das máquinas

## 3. Lógica e linguagem de programação

Designar comandos e rotinas de um robô depende inteiramente de sua lógica codificada. Lógica é uma palavra para a ciência do raciocínio, ou mesmo "o estudo dos métodos e princípios usados para distinguir o raciocínio correto do incorreto".

As tarefas que realizamos em nossa vida diária dependem da lógica. Para resolver um problema, nosso cérebro constrói uma sequência de etapas para seguir corretamente até que tudo seja resolvido, uma espécie de receita.

Cozinhar ovos, por exemplo:



- 1 – Encontre uma panela adequada
- 2 – Coloque água na panela (cerca de 1/3)
- 3 – Adicione os ovos na panela com água
- 4 – Leve ao fogo, assim que a água começar a ferver, marque de 8 a 10 minutos
- 5 – Retire a panela do fogo
- 6 – Escorra a água da panela
- 7 – Retire os ovos da panela
- 8 – Descasque os ovos e sirva-se

Alguns detalhes podem mudar dependendo da receita, mas a lógica será semelhante à apresentada. Este é um procedimento lógico e, se a ordem de qualquer uma dessas etapas for alterada, a receita provavelmente falhará.

# A lógica das máquinas

Nesse sentido, a lógica de programação funciona como uma receita, **um algoritmo**, um conjunto de passos finitos e organizados que, quando executados resolvem um determinado problema.

Para que um robô seja capaz de realizar suas funções precisamos programa-lo, criar rotinas, passos, algoritmos com lógica e linguagem de programação. Existem uma série de linguagens de programação e com elas escrevemos os programas, aplicativos, sites, sistemas web, entre outros.

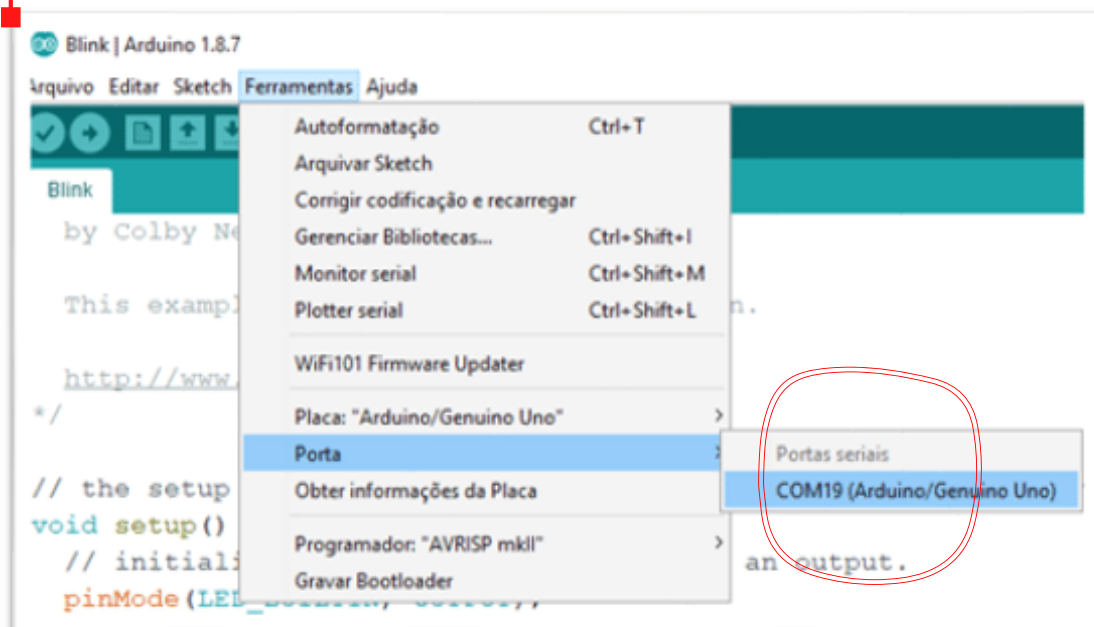
## 3.1 Programando e testando a placa

Para iniciar nossos estudos em lógica e linguagem de programação iremos utilizar um programa para testar nosso Arduino. Siga os seguintes passos:

**1**

**Conecte seu Arduino na porta USB e configure a porta de conexão COM.**

Obs: o número da porta com pode ser diferente da imagem (COM4, COM3...)



# A lógica das máquinas

## 2 Vamos embarcar o seguinte Sketch:

Código para ligar e desligar por 1 segundo o LED da placa

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

## 3 Salve seu código e envie para placa



## 4 Observe o resultado

Verifique se o LED da placa está ascendendo e desligando por 1 segundo.

*Vamos ao código!*

## pinMode()

Essa função configura como o pino irá se comportar, se será para saída (OUTPUT) ou entrada de (INPUT)

Nos parentes estão os parâmetros da função: “LED\_BUILTIN” é uma constante, criada para **representar** o LED que vem integrado à placa. No caso do Arduino Uno esse LED fica ao lado da porta 13. Logo, quando codificamos:

```
pinMode(LED_BUILTIN, OUTPUT);
```

Estamos informando que o LED da placa está configurado apenas como saída de energia, assim, não poderemos receber informações nesse pino, apenas ligar ou desligar.

# A lógica das máquinas

## digitalWrite():

Escreve um comando de HIGH (ligado) ou LOW (desligado) para a porta sinalizada na função

```
digitalWrite(LED_BUILTIN,HIGH);
```

Quando escrevemos esse comando estamos dizendo que a porta “LED\_BUILTIN” está recebendo o valor de **HIGH**, ou seja, enviando energia para ligar a porta, conseqüente o LED da placa.

```
digitalWrite(LED_BUILTIN,LOW);
```

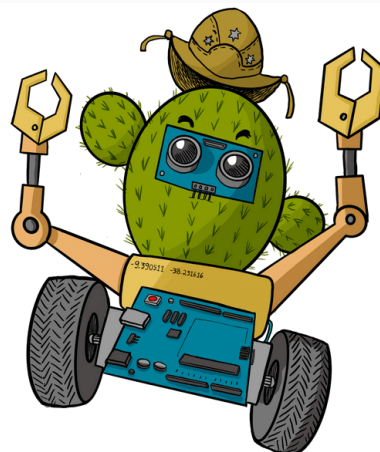
Quando usamos esse comando estamos codificando a porta “LED\_BUILTIN” para receber o valor de LOW, ou seja, desligando o led da placa.

## delay():

**determina um tempo de espera, contado em milissegundos**

Assim, quando codificamos a função **delay(1000);** significa que o Arduino irá esperar 1 segundo para seguir para próxima instrução.

TUDO  
FUNCIONANDO?  
VAMOS PARA O  
PRIMEIRO DESAFIO?



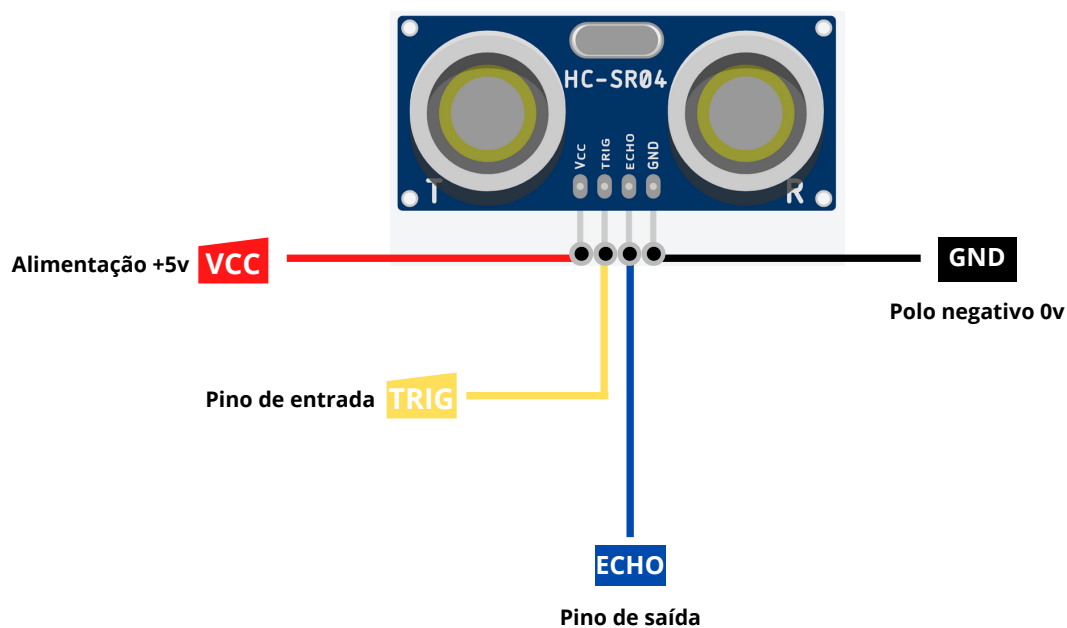
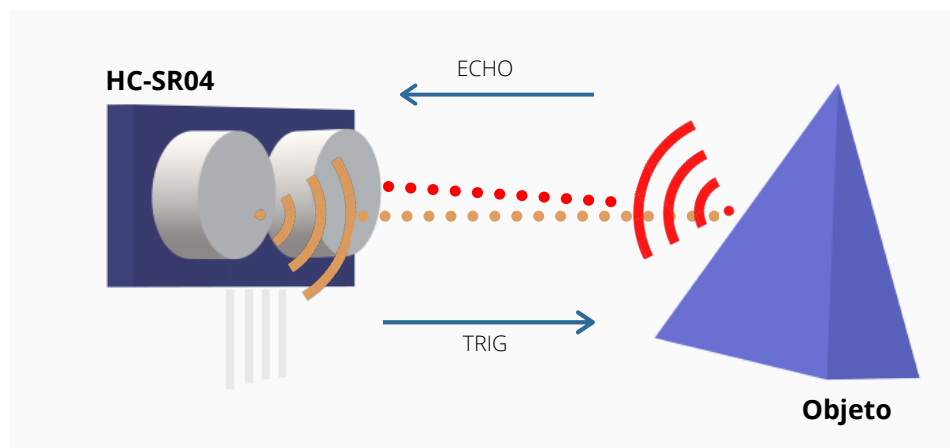
# A lógica das máquinas

## 3.2 Régua eletrônica com Arduino

Agora que nosso ambiente está pronto, vamos programar uma régua eletrônica capaz de apresentar a distância em centímetros entre um sensor ultrassônico e um objeto.

### 3.2.1 Sensor - HC-SR04

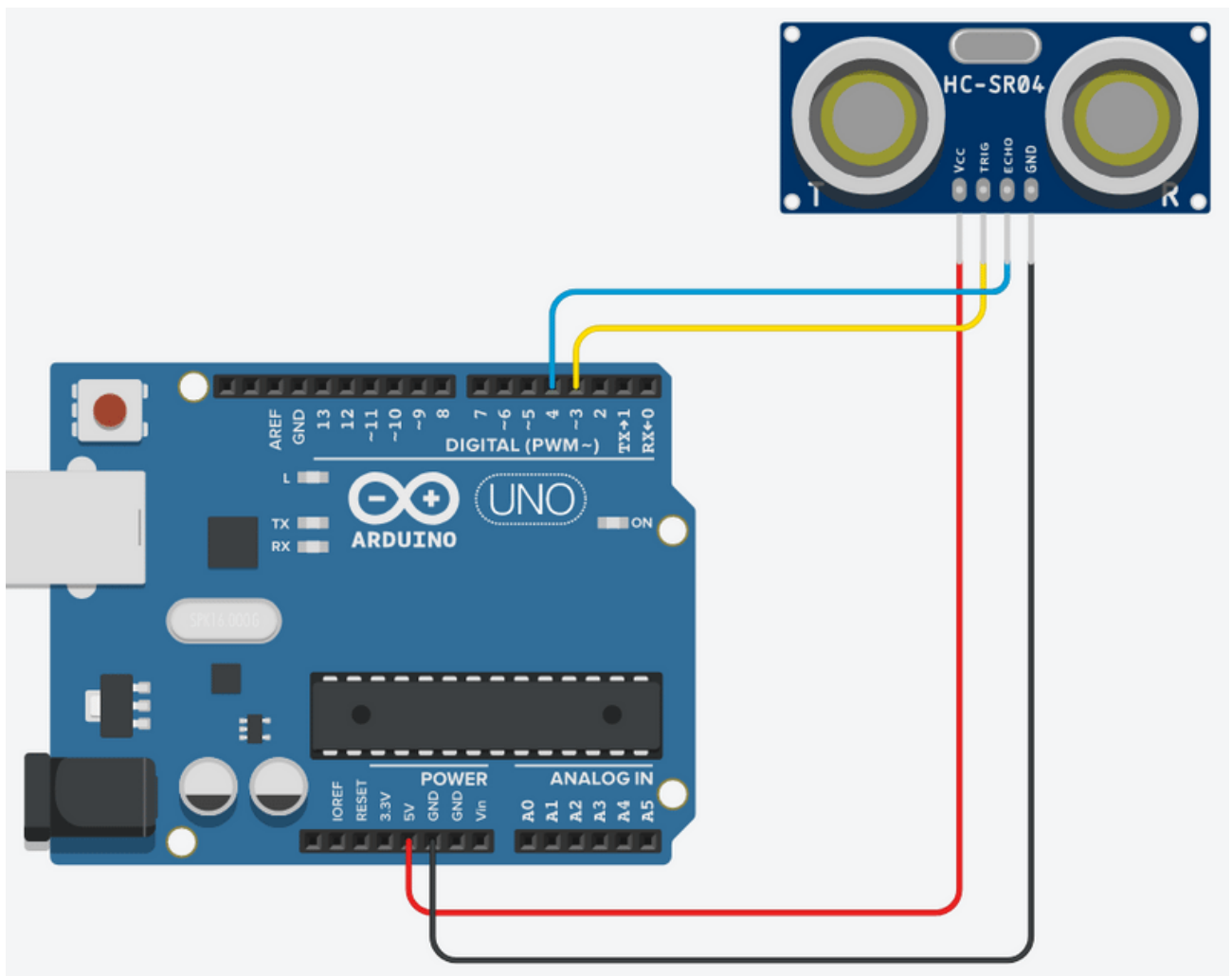
O sensor HC-SR04 emite pulsos sonoros em uma frequência de 40KHz que se propaga na velocidade do som. Quando o pulso encontra um objeto, o sinal é refletido e retorna ao sensor como um eco. A distância entre o sensor e o objeto é calculada com base no volume e no tempo de recebimento do pulso, parecido com o sistema de localização de objetos dos morcegos.



# A lógica das máquinas

## 3.2.2 Conectando o sensor ao Arduino

Para conectar o sensor HC-SR04 ao Arduino iremos utilizar pequenos cabos coloridos intitulados de "**jumpers**", cabos com um fio de cobre dentro para transmitir sinais elétricos de uma ponta a outra. Uma dica, caso não tenha os jumpers, você pode utilizar os cabos internos do cabo de internet (RJ 45) de qualquer cor. Observe o esquema do nosso projeto e siga os passos da montagem.



- 1 Com a placa desligada conecte:
  - Saída **5v** do Arduino ao **VCC** do sensor
  - O **GND** do Arduino ao **GND** do sensor
  - A **porta 4** da placa ao pino **ECHO** do sensor
  - A **porta 3** da placa ao pino **TRIG** do sensor

# A lógica das máquinas

## 3.2.3 Codificando nosso projeto

Após montar a parte física do projeto, iremos agora desenvolver seu programa.

### 2 Conecte o Arduino ao computador e abra a IDE de programação.

Construa e salve o código abaixo:

```
#include <Ultrasonic.h>

Ultrasonic sensor(3, 4);

void setup() {
  Serial.begin(9600);
  Serial.println("Lendo dados do sensor...");
}

void loop() {
  float centimetros;
  long microsec = sensor.timing();

  centimetros = sensor.convert(microsec, Ultrasonic::CM);

  Serial.print("Distancia em cm: ");
  Serial.print(centimetros);
  delay(1000);
}
```

### 3 Salve, embarque o código e visualize os resultados no monitor serial



```
COM3 (Arduino Uno)
Lendo dados do sensor...
Distancia em cm: 6.61
Distancia em cm: 6.84
Distancia em cm: 13.14
Distancia em cm: 18.64
Distancia em cm: 6.90
Distancia em cm: 13.12
Distancia em cm: 16.71
Distancia em cm: 19.26
Distancia em cm: 21.68
```

# A lógica das máquinas

---

*Vamos ao código!*

```
#include <Ultrasonic.h>
```

Aqui estamos adicionando a biblioteca "Ultrasonic" no nosso programa, com isso as funções da biblioteca ficam disponíveis para o uso.

```
Ultrasonic sensor(3, 4);
```

Nessa linha está a configuração da variável "sensor", indicando que as portas 3 e 4 estão sendo utilizadas como TRIG e ECHO.

```
Serial.begin(9600);
```

Essa função configura a taxa de comunicação em bits por segundo com o monitor serial, nesse caso 9600.

```
Serial.println("Lendo dados do sensor...");
```

Com função Serial.println podemos imprimir valores, variáveis, textos e afins. Tal como "printf" em C, ou ainda "System.out.println" em java.

---

# A lógica das máquinas

---

## `float` centímetros;

Aqui declaramos a variável "centímetros" (sem acento) do tipo **float**, isso significa que ela pode armazenar valores com ponto flutuante, exemplo: "3,4".

## `long` microsec = `sensor.timing()`;

Nessa linha estamos criando a variável "microsec" do tipo **long**, com duplo ponto flutuante para armazenar a leitura do sensor através da chamada da função `sensor.timing()`;

## `centímetros = sensor.convert(microsec, Ultrasonic::CM);`

A função `sensor.convert (microsec, Ultrasonic::CM)` converte o valor da variável "microsec" para centímetros e salva o resultado na variável "centímetros".

## `Serial.print("Distancia em cm: ");`

Aqui imprimimos a frase "Distancia em cm:" no monitor serial

## `Serial.print(centímetros);`

E por fim, imprimimos no monitor serial o valor da variável "centímetros", ou seja, a distância atual entre o sensor e objeto.

---

# A lógica das máquinas

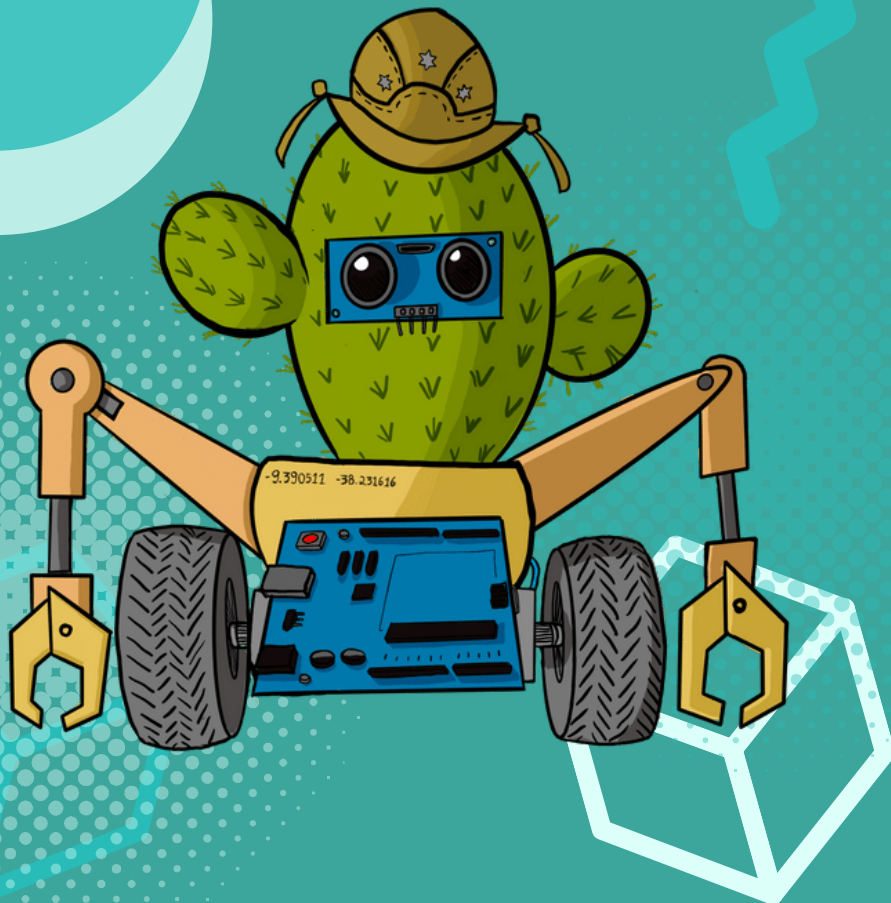
---

## 4. Atividade M.III

- 1) Desenvolva um código que faça o Arduino ligar o led da placa por 5 segundos e desligalo por 3 segundos.
  - 2) Crie um código que imprima seu nome completo no monitor serial.
  - 3) Faça um programa que imprima "ligado" no monitor serial quando o led da placa estiver ligado, quando o led estiver desligado imprima "desligado" no monitor serial.
  - 4) Qual a diferença entre a lógica e a lógica de programação ?
  - 5) Qual a importância da lógica de programação para o desenvolvimento de robôs?
  - 6) Escreva como funciona, e como conectar o sensor ultrassônico HC-SR04 ao Arduino.
  - 7) Explique as funções abaixo:
    - a) `Serial.begin()`;
    - b) `Serial.println()`;
  - 8) Descreva um projeto em que você poderia utilizar o sensor HC-SR04. (que já exista ou não)
-

# MÓDULO IV

Projetando o futuro



<b>5. Conhecendo seu kit robótico .....</b>	<b>29</b>
5.1 Protoboard .....	29
5.2 LED .....	30
5.3 Resistores .....	30
5.2 Projeto sistema de alerta .....	31
5.4.1 Esquema elétrico.....	31
5.4.2 Programando o sistema de alerta.....	33
<b>6. Atividade M.IV.....</b>	<b>36</b>
<b>7. DESAFIO .....</b>	<b>37</b>

# Projetando o futuro

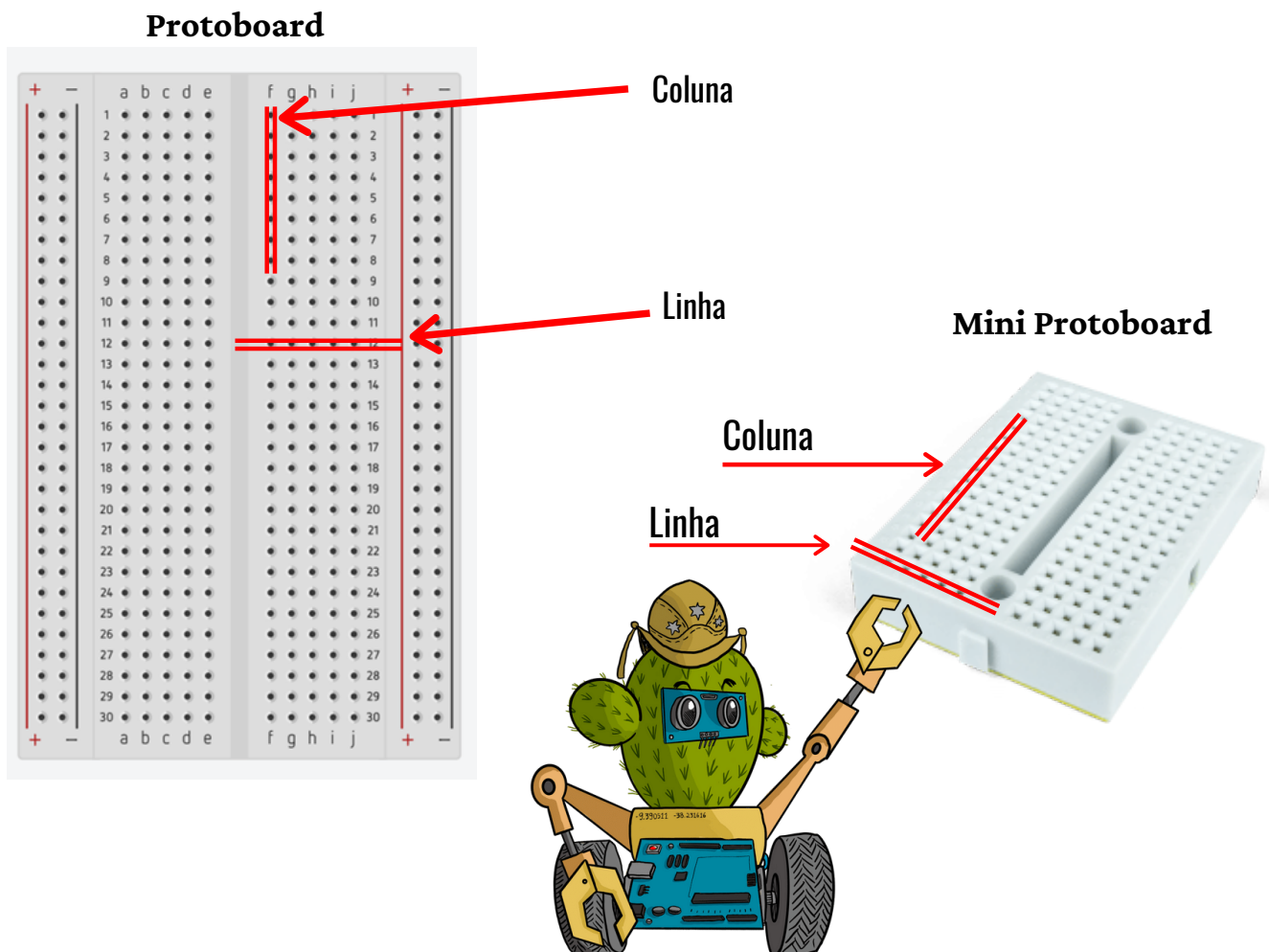
## 5. Conhecendo seu kit robótico

Agora que vamos montar nosso segundo projeto precisamos conhecer os componentes do nosso kit robótico.

### 5.1 Protoboard

A protoboard, comumente chamada de placa de ensaios, é uma placa que permite a criação de circuitos sem a necessidade de soldar os componentes para que possamos conectar componentes eletrônicos simplesmente colando-os em seus trilhos, com a possibilidade de removê-los e reutilizá-los em outros projetos

As trilhas da protoboard funcionam como uma extensão de pulsos elétricos e é composta por linhas e colunas. Logo, quando utilizamos uma entrada em uma determinada trilha, estamos conduzindo a energia para as outras saídas da mesma trilha.

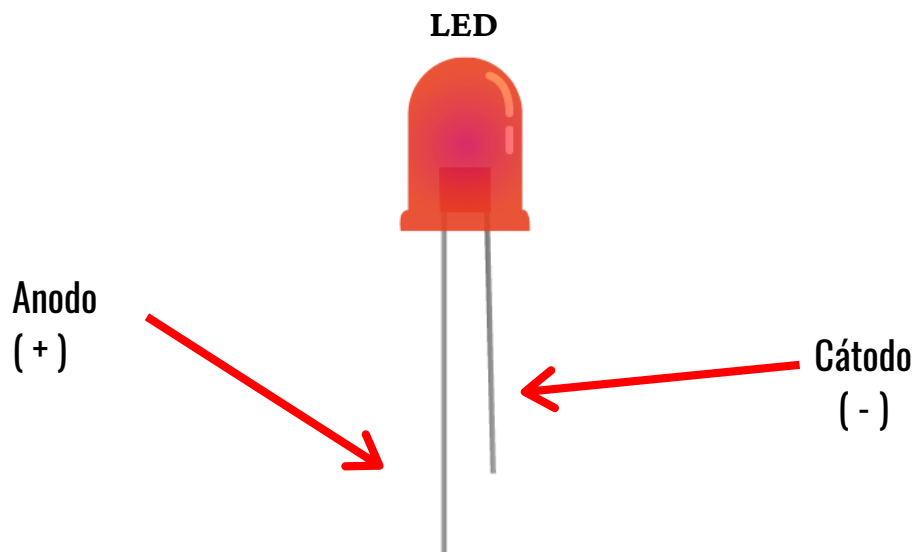


# Projetando o futuro

---

## 5.2 LED

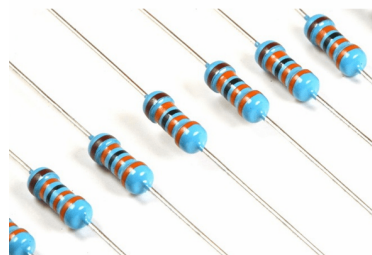
LED (Light Emitting Diode) é um diodo emissor de luz muito utilizado nos aparelhos eletrônicos para sinalizar estados de ligado ou desligado. O LED possui dois elétrodos, duas “pernas”, o anodo (positivo) e o cátodo (negativo), geralmente funcionam a uma tensão de 1.6v a 3,3v.



## 5.3 Resistores

Os resistores são componentes eletrônicos que resistem à passagem de uma corrente elétrica, diminuindo a intensidade da corrente e seu potencial elétrico. Assim, ao utilizar um resistor estamos diminuindo a corrente que chega ao componente a ele conectado. As cores do resistor determinam sua resistência, o múltiplo, pureza e tolerância. Em nosso projeto, usaremos resistores de 330 Ohm.

### Resistores



# Projetando o futuro

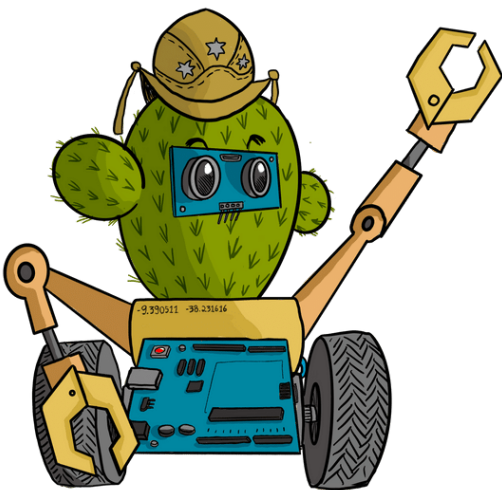
## 5.4 Projeto sistema de alerta

*A diretoria da escola instalou um sistema de painel solar e começou a produzir energia limpa para o prédio da escola. O centro de distribuição de energia fica em uma área protegida, mas como garantia, os coordenadores querem implantar um dispositivo que acenda uma luz quando algo se aproximar da central.*

*Desenvolva um projeto que acenda uma luz de alerta quando a distância entre um objeto e a central for menor que 10 centímetros.*

### Compreendendo a questão

Vamos projetar um sistema que utilizará um sensor ultrassônico para ler a distância entre o objeto e a entrada central, usaremos um LED para simular a luz de alerta e um resistor para garantir a tensão correta para o LED.



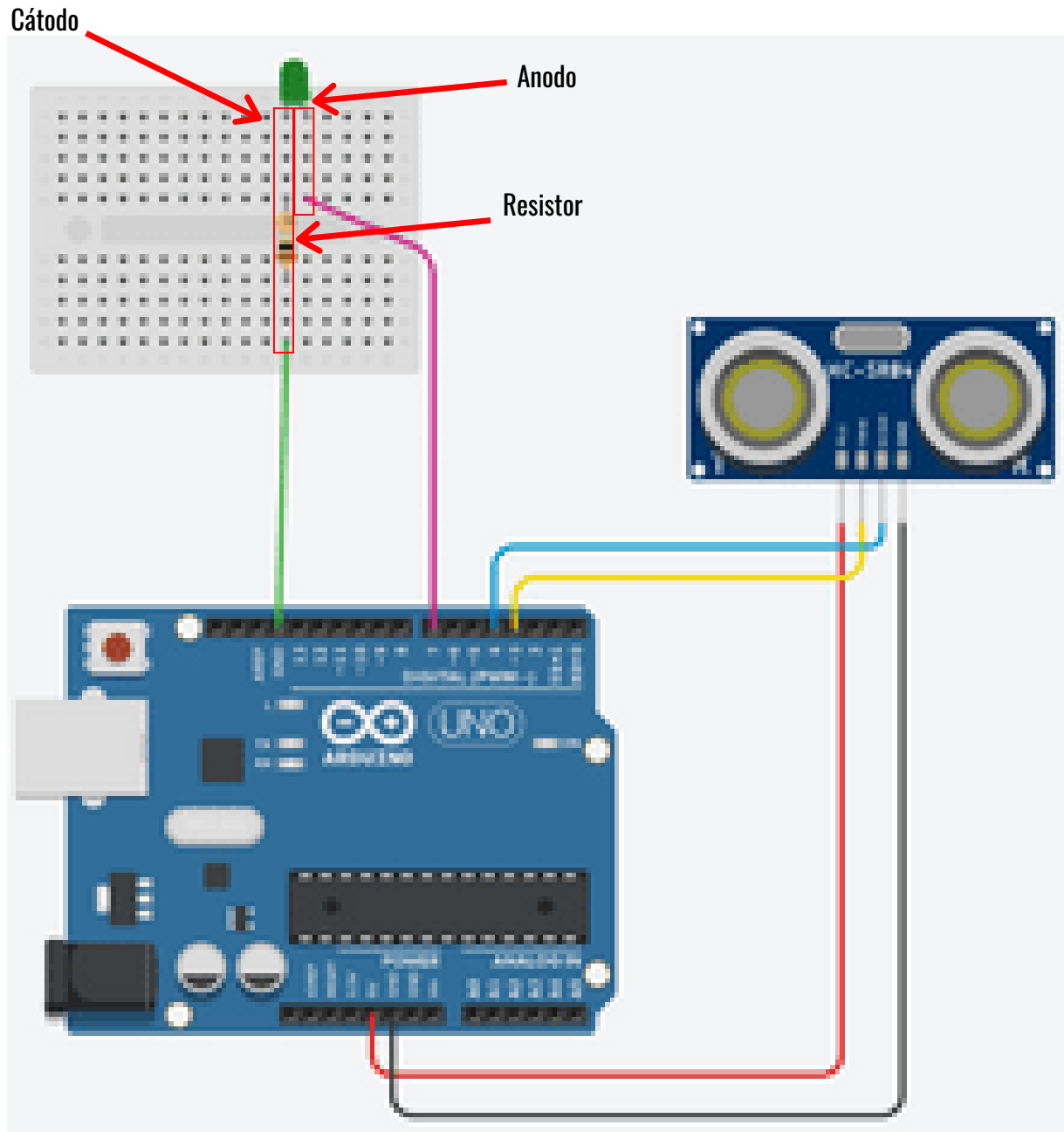
Agora que conhecemos as partes e a finalidade do sistema robótico, vamos configurar a conexão elétrica e entender como funciona a programação do sistema. Vamos prosseguir para o projeto guiado.

### 5.4.1 Esquema elétrico

Com calma, siga o tutorial de montagem do esquema elétrico. Observe na imagem as conexões e cabos utilizados para montar o projeto.

# Projetando o futuro

Figura x : Esquema elétrico do sistema de alertas



- 1** Com um jumper, conecte a **porta 7** a uma trilha na protoboard. Isso, por sua vez, irá reproduzir o sinal para o eletrodo do LED.
- 2** O pino **GND** conecta-se ao resistor a partir de uma trilha da protoboard. O outro lado do resistor está conectando as trilhas que replicam o sinal até o cátodo do LED. Assim, o LED encontra-se conectado ao GND e a porta 7 do Arduino através da protoboard.

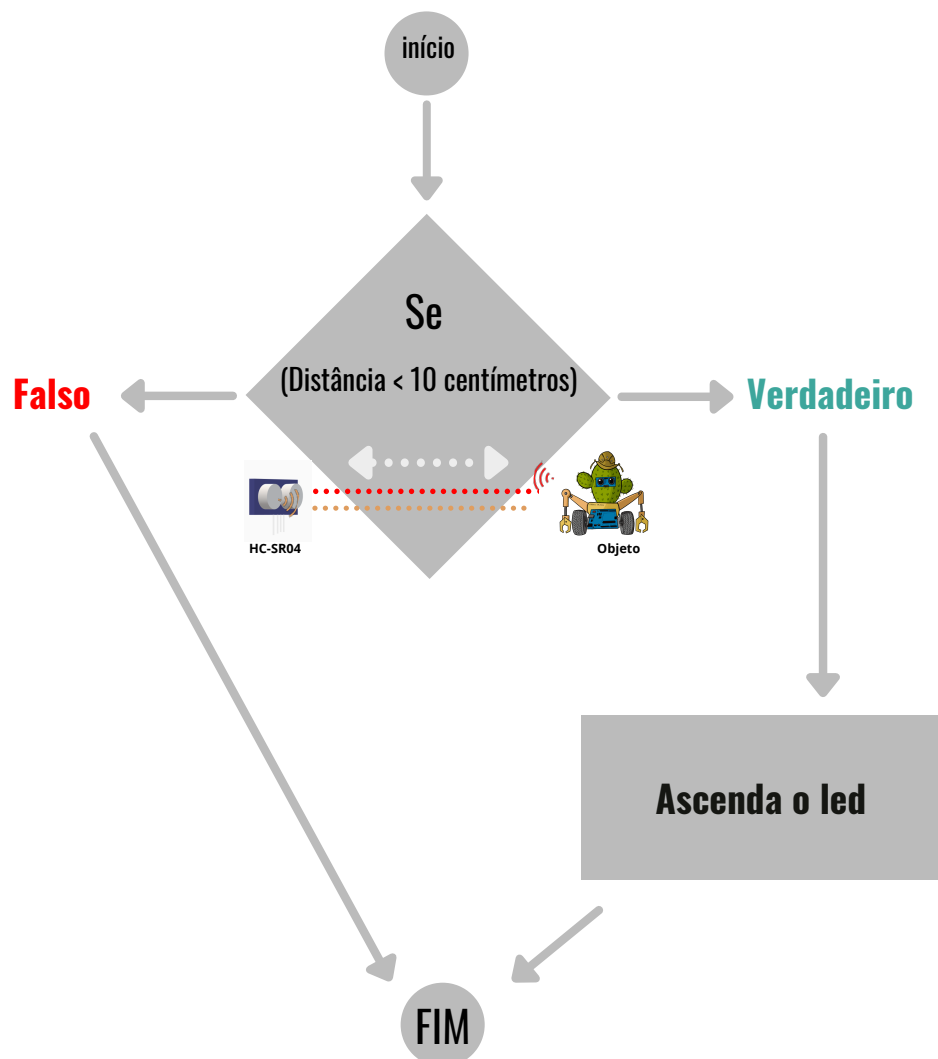
# Projetando o futuro

## 5.4.2 Programando o sistema de alerta

Vamos fazer o nosso projeto, mas você deve ter percebido que neste caso existe uma condição que faz com que o LED só acenda "quando a distância entre o objeto e o sensor for menor que 10 centímetros", então, como faremos isso?

Para resolver este problema, usaremos a **estrutura de repetição "se" ou "if"**. Esta estrutura indica quais instruções nosso código executará a partir da condição. Se esta condição for **verdadeira**, os comandos da estrutura serão executados, se a condição for **falsa** nosso robô não executará este bloco de código. Uma vez implementada, esta estrutura apresenta duas opções que nosso programa pode seguir. O que vai determinar o caminho será a condição, que nesse caso é "**distância < 10 centímetros**".

Fluxograma da condicional if



# Projetando o futuro

*Vamos ao código!*

```
#include <Ultrasonic.h>
```

```
Ultrasonic sensor(3, 4);
```

```
int led_verde = 8;
```

Variável led\_verde recebe o valor 8, esse é o número da porta que utilizaremos para o LED.

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  Serial.println("Lendo dados do sensor...");
```

```
  pinMode(led_verde, OUTPUT);
```

```
}
```

Configurando a porta "led\_verde" como saída

```
void loop() {
```

```
  float centimetros;
```

```
  long microsec = sensor.timing();
```

```
  centimetros = sensor.convert(microsec, Ultrasonic::CM);
```

```
  Serial.print("Distancia em cm: ");
```

```
  Serial.print(centimetros);
```

```
  Serial.println("");
```

```
  if(centimetros<10) {
```

```
    digitalWrite(led_verde, HIGH);
```

```
    delay(2000);
```

```
    digitalWrite(led_verde, LOW);
```

```
  }
```

```
}
```

Condicional

Comando para ligar a porta "led\_verde"

Comando para desligar a porta "led\_verde"

# Projetando o futuro

---

Observe no código que declaramos a variável **"led\_verde"** para receber o valor da porta que conectamos o LED, nesse caso a porta 8. Portanto, se você deseja adicionar mais diodos ao seu projeto declare outras variáveis com portas diferentes. Pouco depois, definimos o pino **"led\_verde"** como saída, o que permite que pulsos elétricos sejam enviados para ligar o elemento conectado.

Na condicional **"if (centimetros<10)"** estamos criando dois caminhos que dependem dessa condição.

- Se o valor for menor que 10, ascenderemos o LED com o comando **"digitalWrite(led\_verde, HIGH);"**, esperamos 2 segundos com a função "delay" e depois desligamos o LED com o comando **"digitalWrite(led\_verde, LOW);"**.
  - Se o valor for maior ou igual a 10 centímetros, o LED permanece apagado.
-

# Projetando o futuro

---

## 6. Atividade M.IV

1) Assista o seguinte vídeo na plataforma do YouTube para completar nosso conteúdo.



Título: **Robôcactus - projeto guiado II**

Plataforma: YouTube

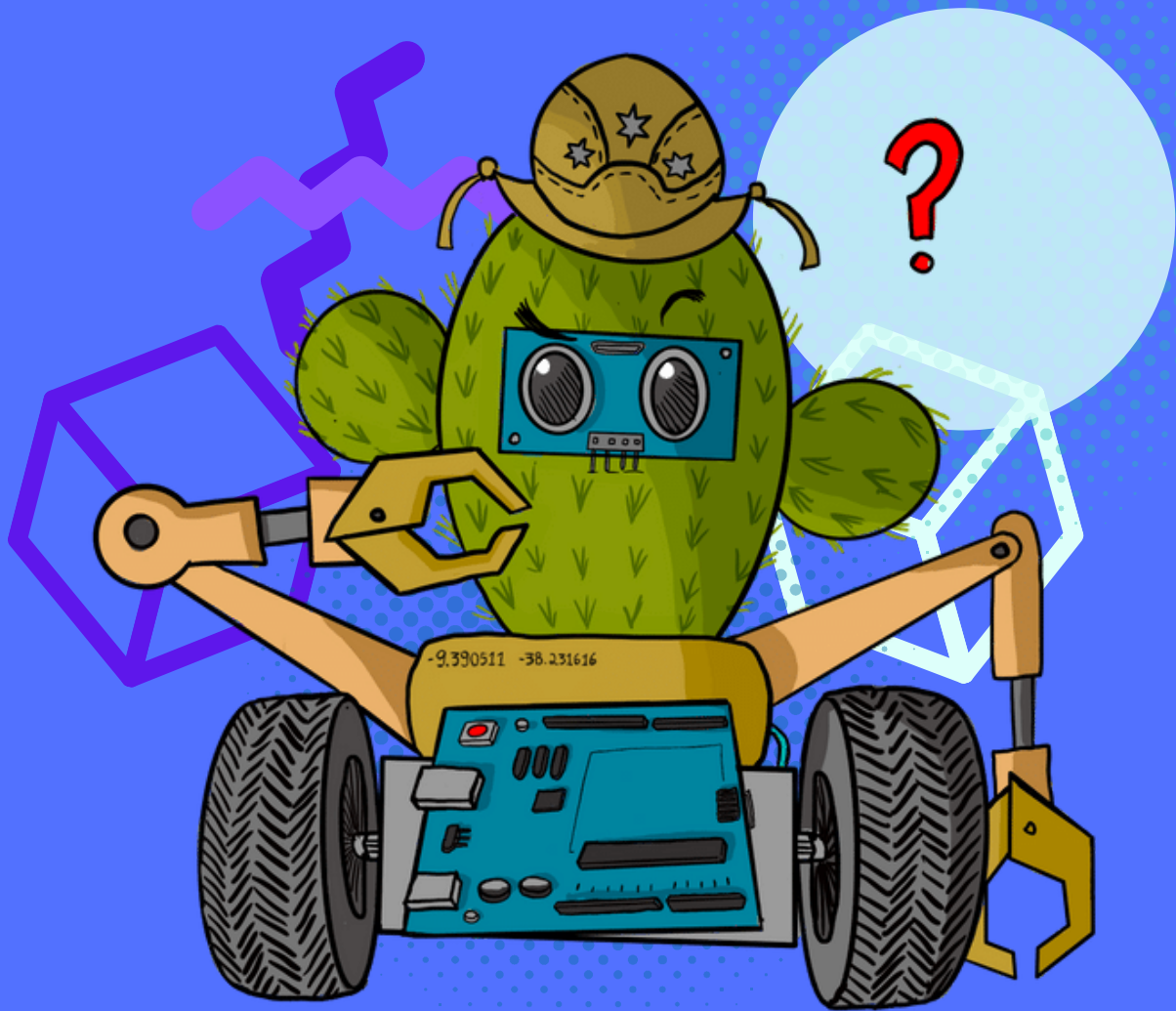
Canal: Bruno Amorim

link: <https://youtu.be/p6gdbNmWBKA>

2) Monte o projeto "sistema de alerta" e envie um vídeo de no máximo 1 minuto apresentando os componentes e seu funcionamento .

---

# DESAFIO!



AGORA QUE VOCÊ JÁ ESTA COM A MÃO NA MASSA,  
IMPLEMENTE O PROJETO ABAIXO E ENVIE UM VÍDEO  
DE NO MÁXIMO 1 MINUTO EXPLICANDO OS  
COMPONENTES UTILIZADOS E SEU FUNCIONAMENTO

**BOA SORTE, VOCÊ CONSEGUE!!**

SUPONHA QUE VOCÊ RECEBEU O SEGUINTE EMAIL:



Olá, nós da empresa *Se avexe não soluções tecnológicas* trabalhamos com diversos sistemas robóticos na região e encontramos seu contato através de uma busca em nosso banco de dados.

Temos um problema em um dos nossos sistemas, criamos um semáforo especial que fecha o sinal quando alguém se aproxima da faixa de pedestres. Funciona assim, quando um pedestre se aproxima de um semáforo a menos de 30 centímetros de distância da faixa de pedestres o sinal fecha para que ele passe, mas para isso é necessário acender o semáforo amarelo por 8 segundos para que os carros percebam que o sinal está fechando e possam diminuir a velocidade. Após esse tempo, o semáforo amarelo é apagado para acender o semáforo vermelho, obrigando os carros a parar por 30 segundos, tempo da travessia do pedestre. Após este processo, o sinal reabre com uma luz verde.

Enviamos a você um kit com elementos para representar o semáforo. Utilizamos um sensor ultrassônico (HC-SR04) para fazer a leitura da distância e as luzes serão representadas por LEDs. Nossos técnicos de hardware criaram o projeto elétrico do sistema, mas nossa única programadora está de férias e precisamos gerar o código para controlar o sistema. Siga o desenho técnico em anexo. Queremos que você construa um código capaz de controlar o sistema de hardware mostrado aqui. Se você puder nos contatar e enviar o código e o vídeo, no máximo 1 minuto explicando como você fez e contando sobre as peças utilizadas no projeto, por favor use o link:

[https://docs.google.com/forms/d/e/1FAIpQLSfMEzkwVqBSIIyu-ebojnG-WIFpf70PRGZ\\_o3Cu93F-pP40DQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfMEzkwVqBSIIyu-ebojnG-WIFpf70PRGZ_o3Cu93F-pP40DQ/viewform?usp=sf_link)

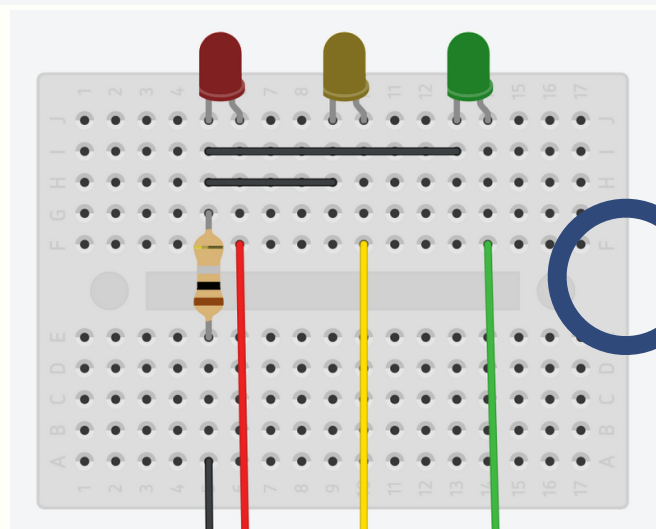
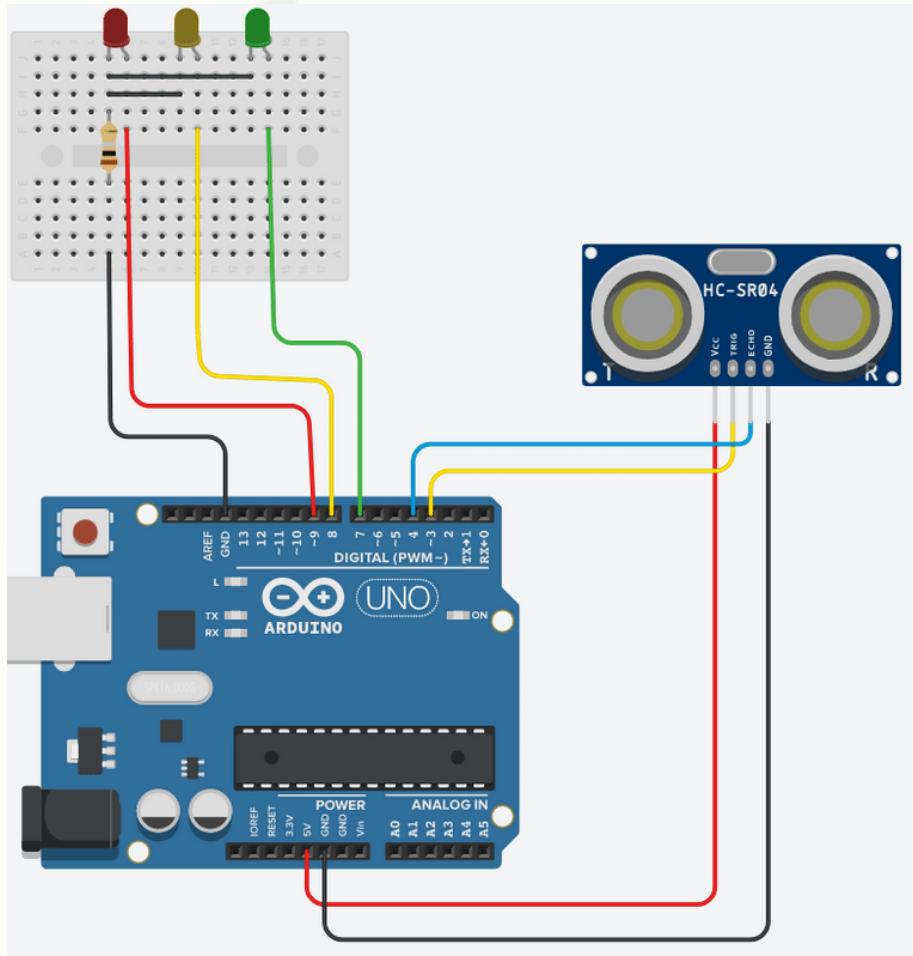
Contamos com você.

Atenciosamente, Se avexe não soluções tecnológicas.

Nota dos técnicos em hardware:

*Usamos alguns jumpers para conectar o resistor aos cátodos dos LEDs, as portas 7,8 e 9 estão sendo utilizadas para os LEDs verde, amarelo e vermelho. Para o sensor estamos trabalhando com a porta 3 e 4. Vamos anexar algumas fotos do projeto montado, mas com jumpers de outra cor.*

### Esquema elétrico do projeto



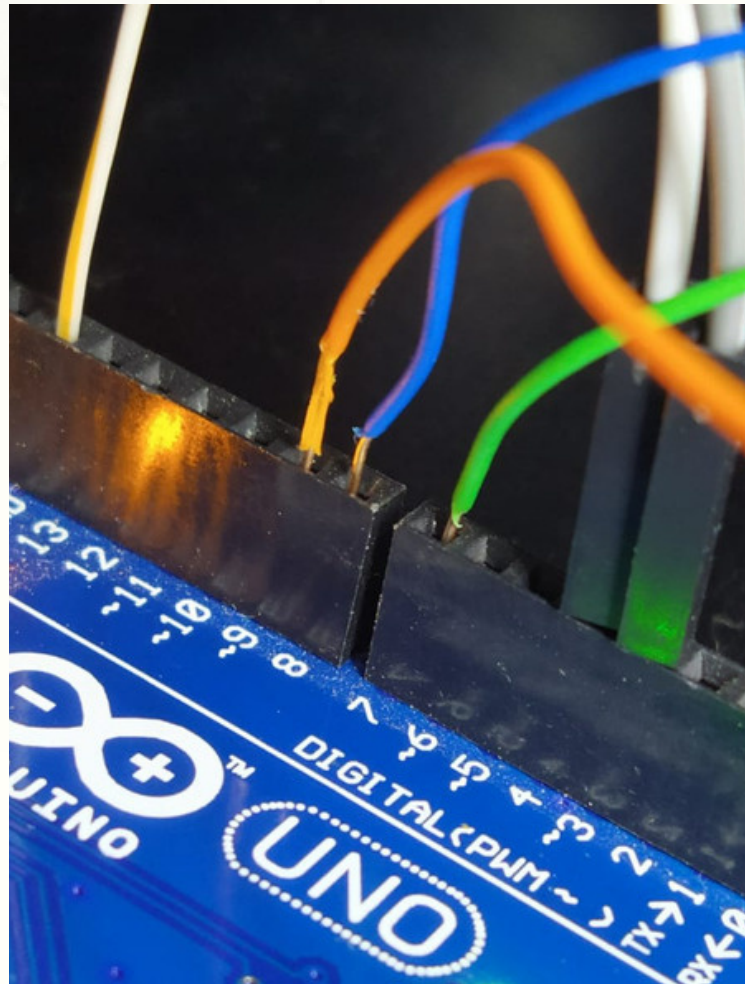
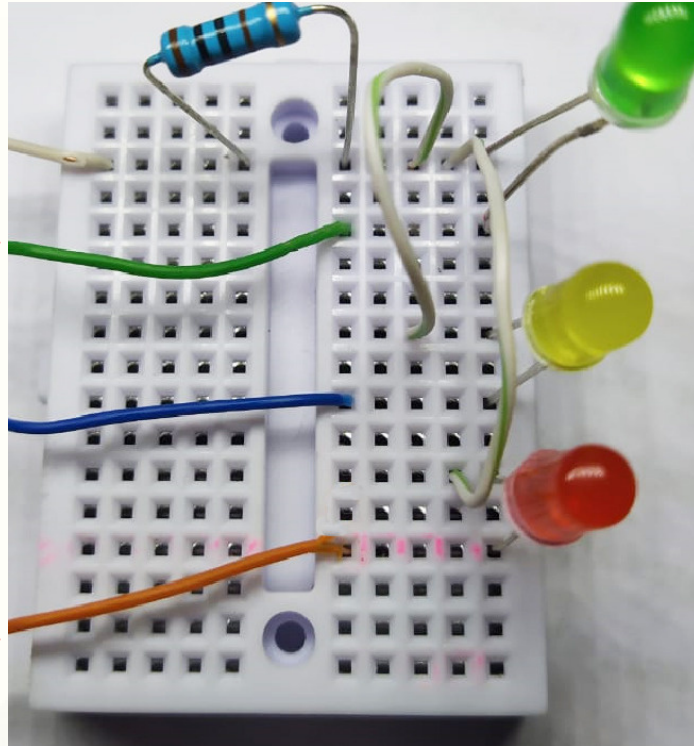
**Olha o Zoom!**

## Fotos do esquema elétrico do projeto

LED verde  
→  
porta 7

LED Amarelo  
→  
porta 8

LED Vermelho  
→  
porta 9



## OBRIGADO!

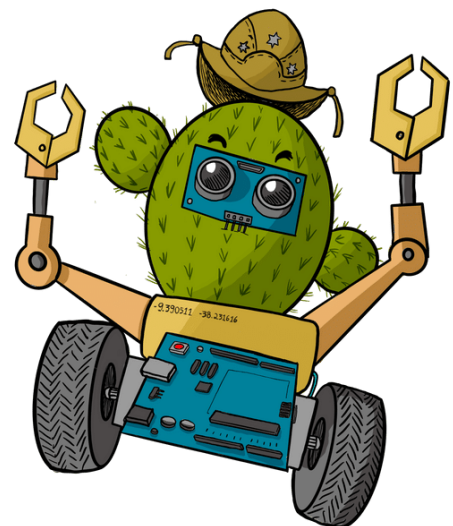
E assim encerramos nossa jornada, espero que você tenha absolvido o máximo de conhecimento desta ciência, e também tenha se divertido na caminhada. Agora que você já conhece os fundamentos da construção de projetos robóticos, não pare por aí, coloque-os em prática e continue seus estudos neste campo. Espero encontrá-lo novamente na próxima versão desta apostila.

Obrigado e se avexe não, é cada um no seu tempo!

Atenciosamente,

**ROBÔCACTUS**

**PROF. BRUNO AMORIM**  
**PROF. DR. EDUARDO CARDOSO MORAES**



*Para mais informações:*

[brunno00amorim@gmail.com](mailto:brunno00amorim@gmail.com)

## **REFERENCIAL TEÓRICO**

**MONK, S. Programação com Arduino: começando com Sketches. Porto Alegre: Bookman Editora, 2013. 200 p.**

**MCROBERTS, M. Arduino básico: tudo sobre o popular microcontrolador Arduino. 2. ed. São Paulo: Novatec Editora, 2015.**

**McRoberts, M., 2018. Arduino básico. Novatec Editora.**

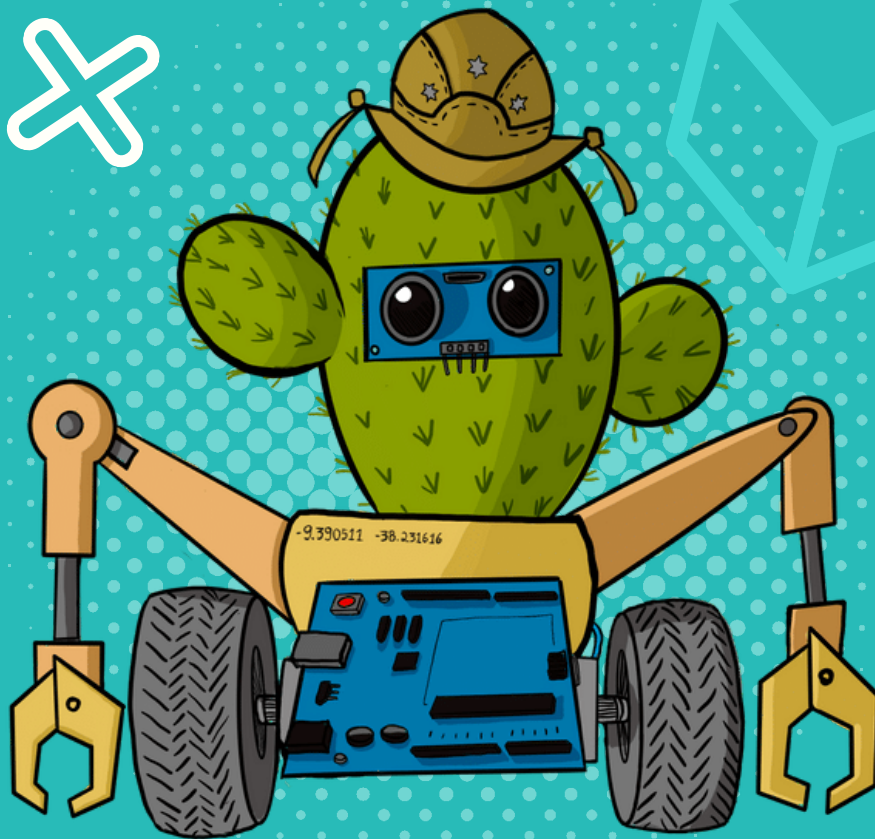
**Braga NC. Eletrônica básica para mecatrônica. São Paulo: Saber; 2005.**

**KOFAC. Korea Foundation for the Advancement of Science and Creativity. How IoT(Internet of Things) Has Changed Our Lives, 2017. Disponível em: <<https://community.computingschool.org.uk/resources/5376/single/>>.**

**ARDUINO.CC. Arduino, c2019. Página inicial. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 16 de jun. de 2019.**

# ROBÔCACTUS

Sua apostila para o estudo de robótica, lógica de programação e Arduino



## Créditos:

Elaboração do material didático – Bruno Amorim

Diagramação – Bruno Amorim

Idealização do mascote Robôcactus – Bruno Amorim

Arte do mascote Robôcactus – Lucas Vieira (@vieira.art)