

PYTHON PARA ARDUINO

Mateus Antonio da Silva

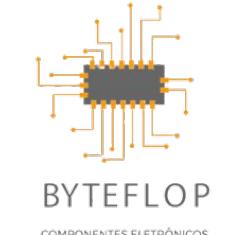
Vice-Presidente RAS UFPB



Realização:

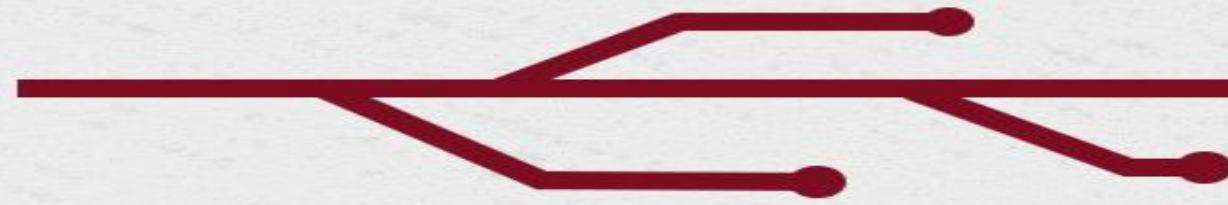


Patrocínios:

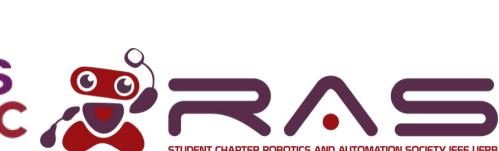


Sumário

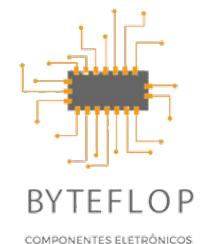
- Introdução
- Objetivo
- Arduino
- Python
- Aplicações



Realização:



Patrocínios:



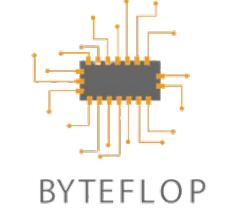
Introdução



Realização:



Patrocínios:



Expectativa

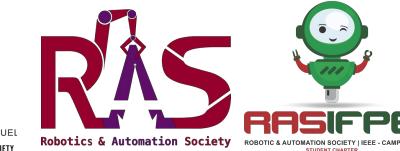


+

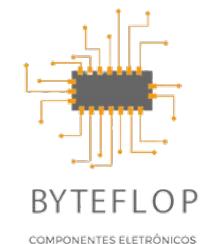
```
31 self._file = None
32 self._fingerprints = set()
33 self._logdups = True
34 self._debug = debug
35 self._logger = logging.getLogger(__name__)
36 if path:
37     self._file = open(os.path.join(path, 'fingerprint.log'), 'a')
38     self._file.seek(0)
39     self._fingerprints.update([line.strip() for line in self._file])
40
41 @classmethod
42 def from_settings(cls, settings):
43     debug = settings.getbool('logger.debug')
44     return cls(job_dir(settings), debug)
45
46 def request_seen(self, request):
47     fp = self.request_fingerprint(request)
48     if fp in self._fingerprints:
49         return True
50     self._fingerprints.add(fp)
51     if self._file:
52         self._file.write(fp + os.linesep)
53
54 def request_fingerprint(self, request):
55     return request_fingerprint(request)
```



Realização:



Patrocínios:



Realidade



+

```
31     self.file = None
32     self.fingerprints = set()
33     self.logdups = True
34     self.debug = debug
35     self.logger = logging.getLogger(__name__)
36     if path:
37         self.file = open(os.path.join(path, 'fingerprint.dat'))
38         self.file.seek(0)
39         self.fingerprints.update(pickle.load(self.file))
40
41     @classmethod
42     def from_settings(cls, settings):
43         debug = settings.getbool('supervisor.debug')
44         return cls(job_dir(settings), debug)
45
46     def request_seen(self, request):
47         fp = self.request_fingerprint(request)
48         if fp in self.fingerprints:
49             return True
50         self.fingerprints.add(fp)
51         if self.file:
52             self.file.write(fp + os.linesep)
53
54     def request_fingerprint(self, request):
55         return request_fingerprint(request)
```



=



Facilidade do Python



Curso Python #01 - Seja um Programador - YouTube

YouTube · Curso em Vídeo



Python para Iniciantes

Para que não sabe Python, é aqui que deve começar!

Tiago Miguel

4,1 ★★★★★ (11.365)

1 total hora • 12 aulas • Iniciante

Gratuito



Introdução à linguagem Python

Aprenda a desenvolver programas de computador usando a linguagem mais popular do mercado de trabalho

Diego Mariano, Ph.D.

4,4 ★★★★★ (18.109)

2 total horas • 38 aulas • Iniciante

Gratuito



Python 3 na Prática

Aprenda a programar em Python

João Batista

4,2 ★★★★★ (508)

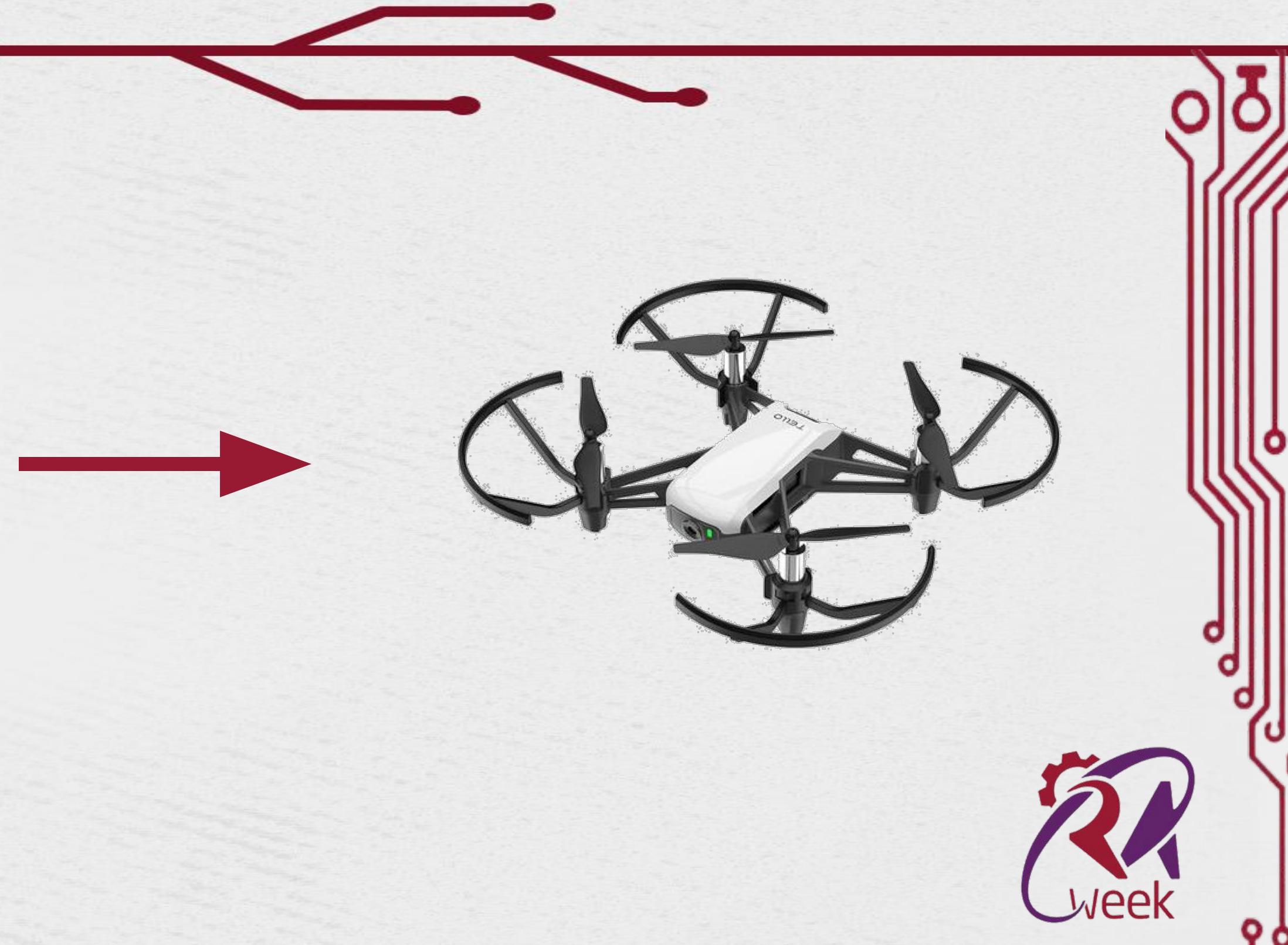
2,5 total horas • 26 aulas • Iniciante

Gratuito



Facilidade do Python

```
1  from djitellopy import Tello  
2  import time  
3  
4  tello = Tello()  
5  
6  tello.connect()  
7  tello.takeoff()  
8  
9  tello.move_left(100)  
10 tello.rotate_counter_clockwise(45)  
11  
12 tello.land()  
13 tello.end()
```



Objetivo



Realização:



Patrocínios:



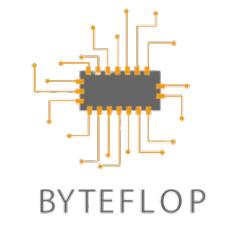
Virar o Batman?



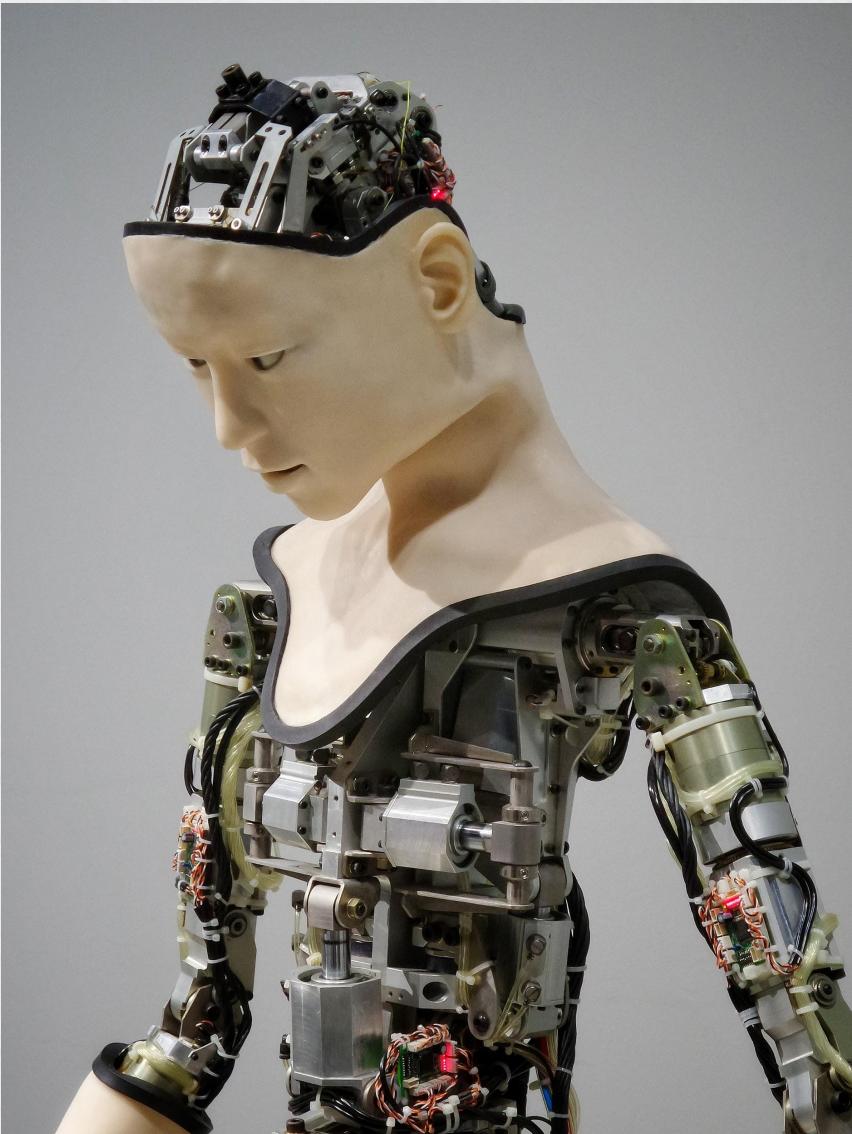
Realização:



Patrocínios:



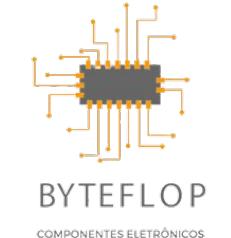
Leque de opções



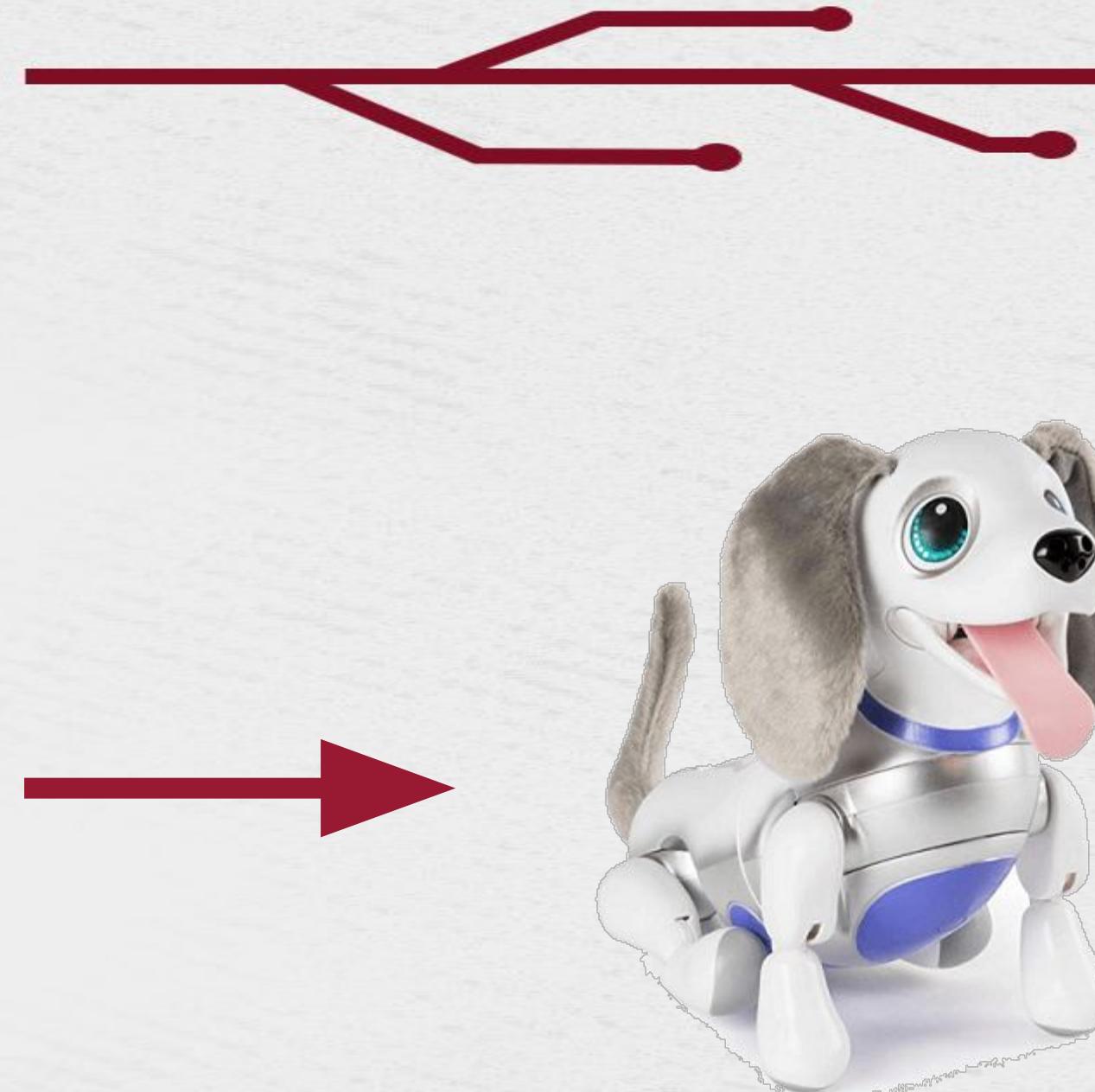
Realização:



Patrocínios:



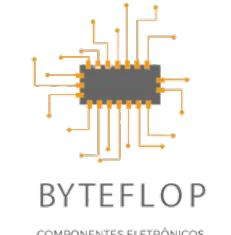
Leque de opções



Realização:

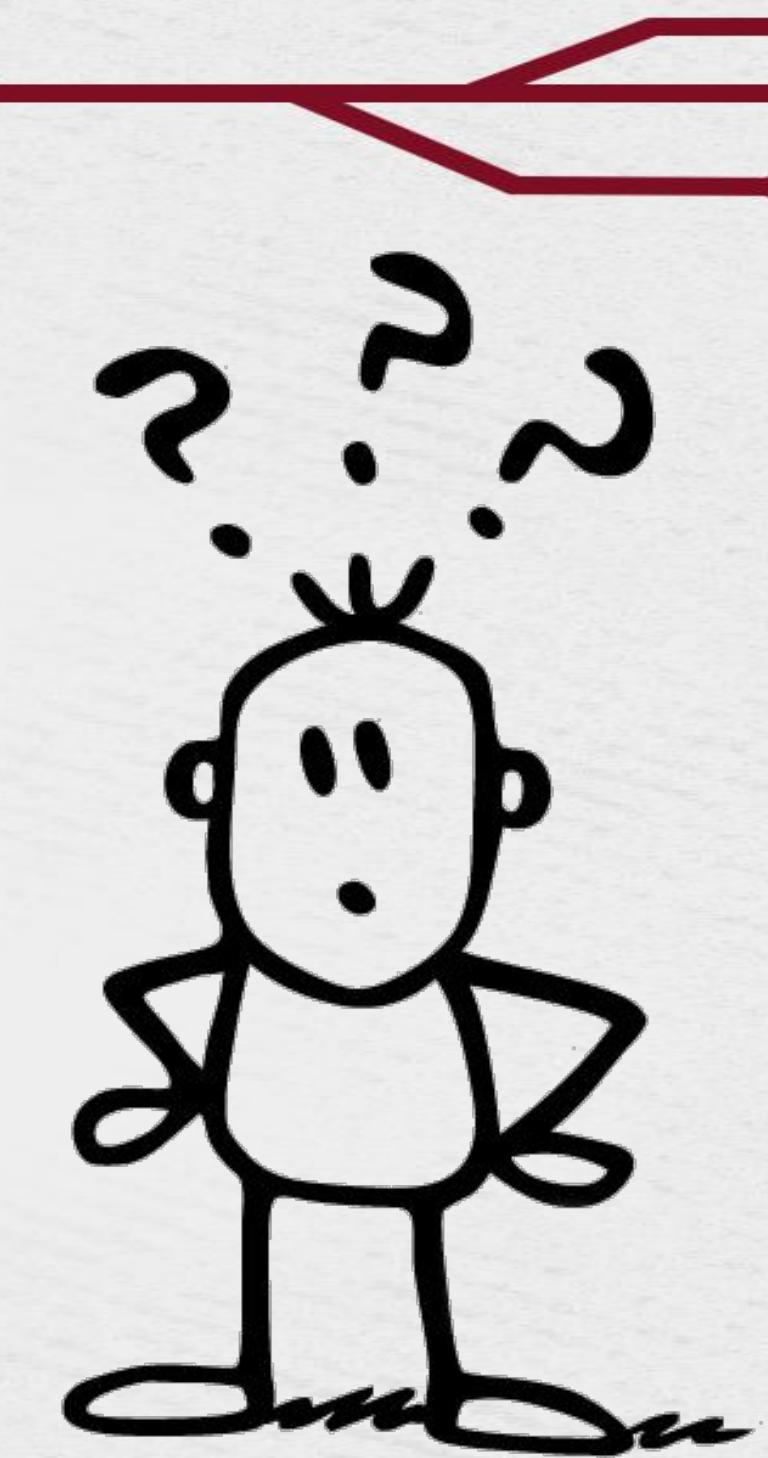


Patrocínios:



Utilidades

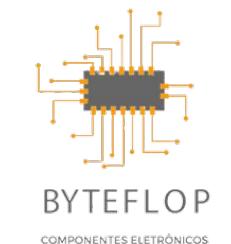
- Hackathon
- Contests
- Projeto de Disciplina
- Projeto de Pesquisa
- Projetos Pessoais
- Ensino



Realização:



Patrocínios:



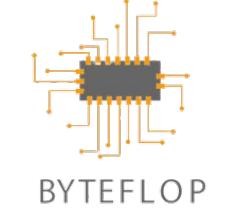
Materiais e Métodos



Realização:

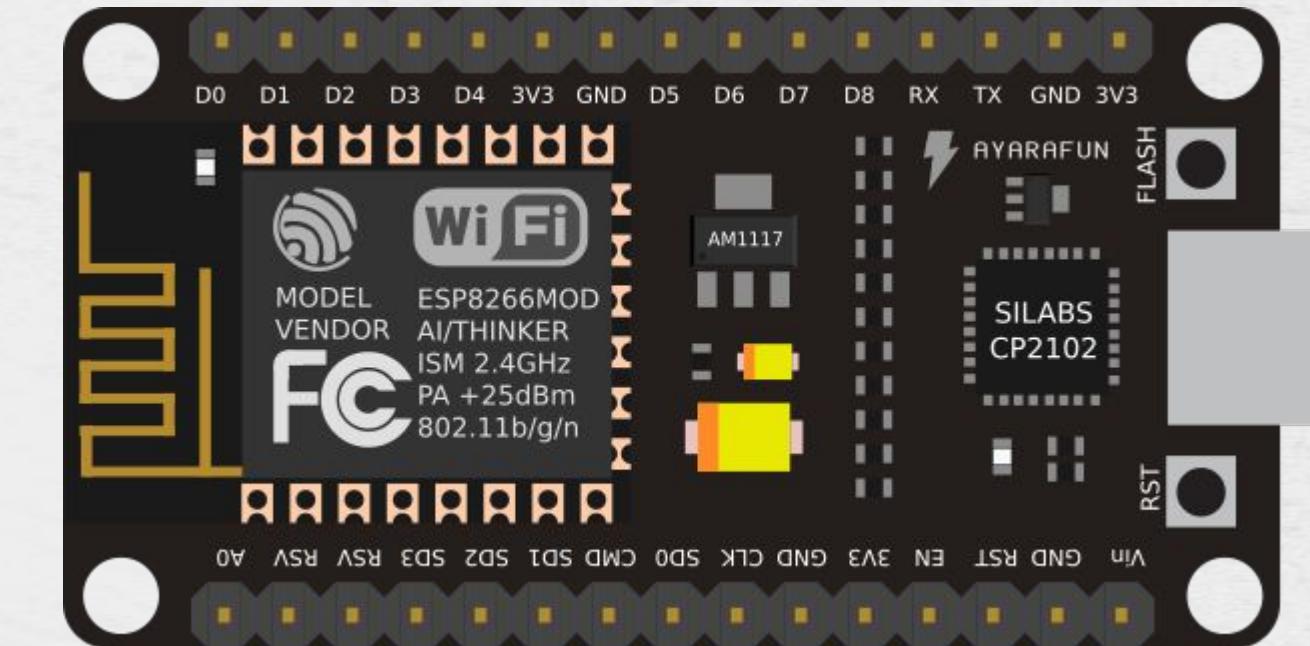


Patrocínios:



[Arduino]

Arduino + Extra



[Arduino]

Arduino + Extra



Arduino Uno



Arduino Leonardo



Arduino Due



Arduino Yun



Arduino Tré



Arduino Micro



Arduino Robot



Arduino Esplora



Arduino Mega ADK



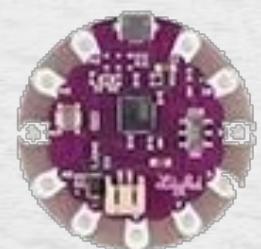
Arduino Ethernet



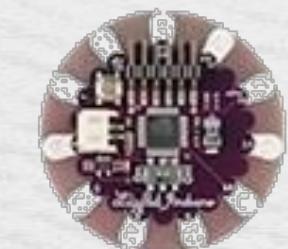
Arduino Mega 2560



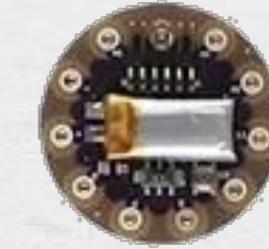
Arduino Mini



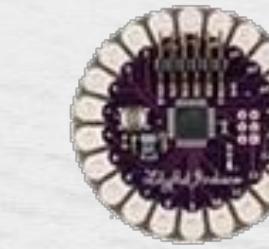
LilyPad Arduino USB



LilyPad Arduino Simple



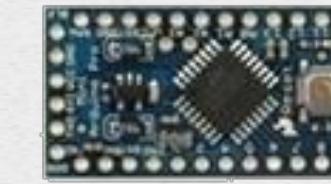
LilyPad Arduino SimpleSnap



LilyPad Arduino



Arduino Nano



Arduino Pro Mini



[Arduino]

O que dá pra fazer?

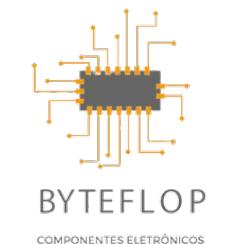
- Sensores
- Atuadores
- Integração com Web
- Integração com Banco de Dados
- Etc



Realização:

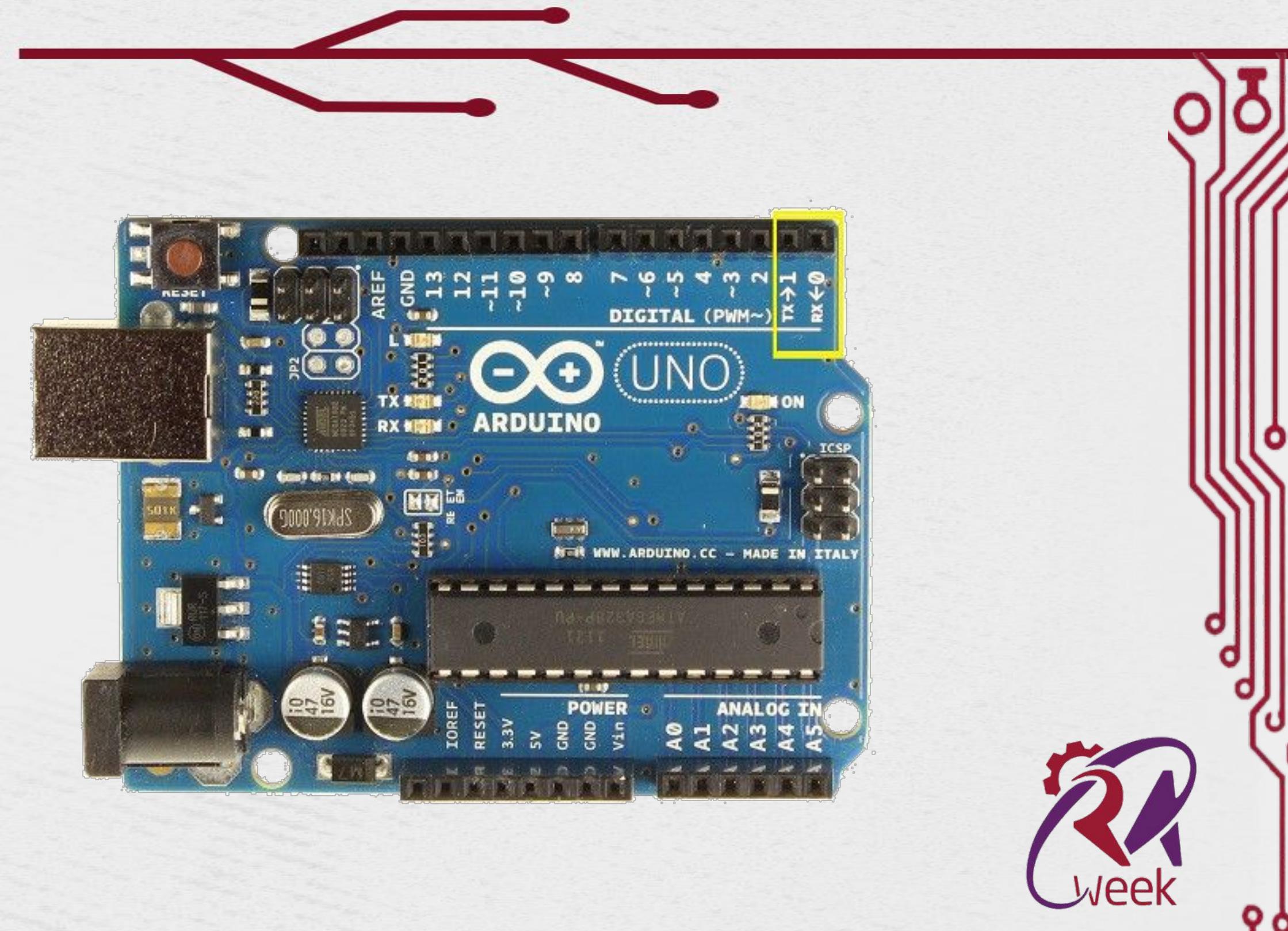


Patrocínios:



Comunicação Serial

- Digital 0 - RX
- Digital 1 - TX
- Vendor ID
- Product ID
- Porta
 - COM1, COM2, COM3...
 - /dev/ttyACM, /dev/ttyUSB...
- Baudrate
 - 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200



Comunicação Serial

Documentação

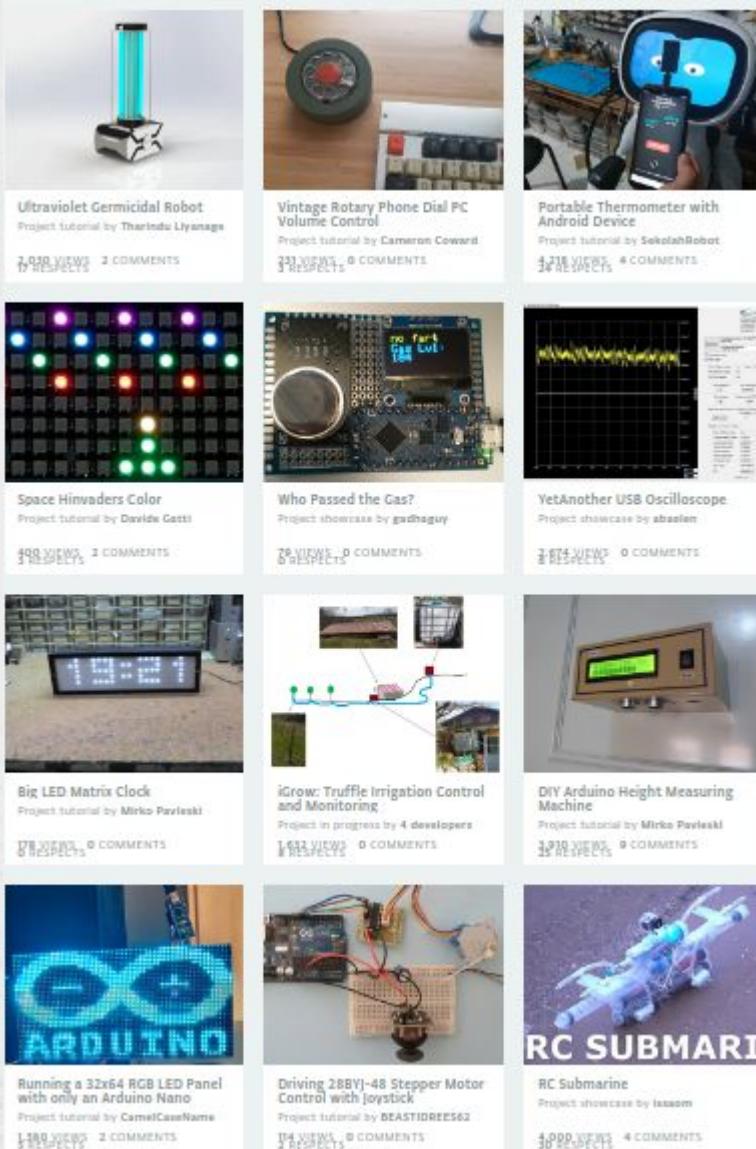
```

1 #define led 13 // Porta onde o led está conectado
2
3 void setup(){
4     Serial.begin(9600);    // Velocidade padrão para comunicação
5     pinMode(led, OUTPUT); // Porta onde o led será acionado, configurado como saída
6 }
7
8 void loop(){
9     if (Serial.available() > 0){
10         char leitura = Serial.read(); // Variavel que receberá os valores enviados pelo programa em python
11
12         if(leitura == 'a'){
13             digitalWrite(led, HIGH); // Liga a porta 13 se o valor recebido for 1
14             Serial.println("Acionou led!");
15         }
16
17         else if(leitura == 'b'){
18             digitalWrite(led, LOW); // Desliga a porta 13 se o valor recebido for 2
19             Serial.println("Desligou led!");
20         }
21     }
22 }
```



[Arduino]

Imagine Possibilidades



Arduino Project HUB

Three project cards from Hackster.io:

- Arduino Bluetooth Basic Tutorial** by Mayoogh Girish: A basic tutorial for controlling electronic devices via Bluetooth.
- Arduino Based Mini CNC 2D Plotter** by MrInnovative: A mini CNC 2D plotter made from old DV equipment.
- Arduino Sonar** by Tony ZHANG: Using HC-SR04 on Arduino with Processing.

Hackster.io

Three project cards from Instructables:

- Getting Started With Arduino** by bekathwia: An introductory guide to Arduino.
- Arduino Ble Rover** by Arbot in Arduino: A BLE-powered remote-controlled rover.
- Custom Arduino guitar** by MrAtkinson: A guitar built around an Arduino.
- Arduino Egg Incubator** by craineum in Gardening: An incubator for eggs using an Arduino.
- Arduino Light Theremin** by zacharyianhoward in Arduino: A light-activated theremin.
- Arduino Game "NIM"** by Dzefri in Arduino: A game of NIM implemented on an Arduino.

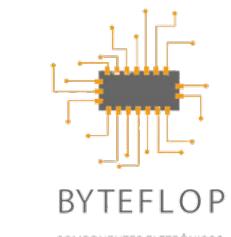
Instructables



Realização:



Patrocínios:



[Python]

Comunicação Serial

Documentação

Escrita

```
1 # -*- coding: iso-8859-1 -*-
2 import serial
3
4 arduino = serial.Serial(port='/dev/ttyACM0', baudrate=9600, timeout=1) # open serial port
5 arduino.write(b'a')          # escreve uma string
6 #arduino.write(bytes('a', encoding='utf-8'))           # escreve uma string
7 arduino.close()             # Fecha porta de comunicação
```

Leitura

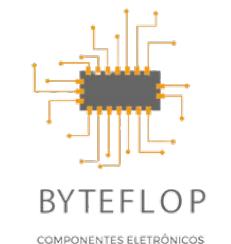
```
1 # -*- coding: iso-8859-1 -*-
2 import serial
3
4 ser = serial.Serial(port='/dev/ttyACM0', baudrate=9600, timeout=1) # open serial port
5
6 while True:
7     msg = ser.readline().decode('ascii')
8     print(msg)
```



Realização:



Patrocínios:



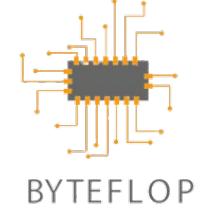
Aplicações



Realização:



Patrocínios:



[Python]

Comunicação Serial

Documentação

Escrita e
Leitura

```
1 import serial
2
3 arduino = serial.Serial(port='/dev/ttyACM0', baudrate=9600, timeout=1)
4
5 while True:
6     msg = input('Envie um comando: ')
7
8     arduino.write(bytes(msg, encoding='utf-8'))
9
10    retorno = arduino.readline().decode('ascii')
11    print('Retorno: ' + retorno)
```



O que vem depois?

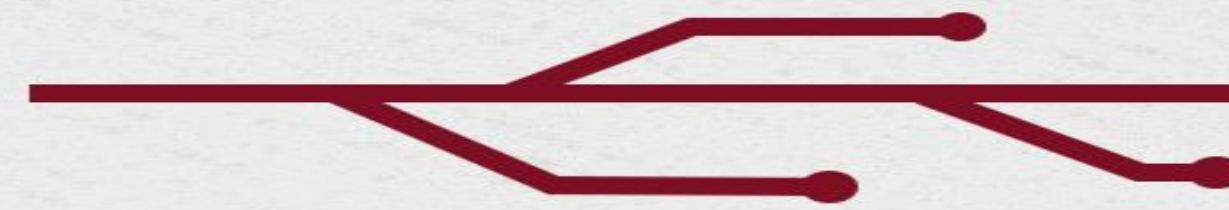
“Are you a one or a zero? That's the question you have to ask yourself. Are you a yes or a no? Are you going to act or not?”

“Você é um ou zero? Esse é o questionamento que você tem que fazer a si mesmo. Você é um sim ou um não? Você vai agir ou não?”

Mr. Robot



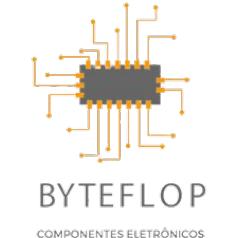
Speech to text



Realização:



Patrocínios:



Speech to text

Documentação

Solta a voz!

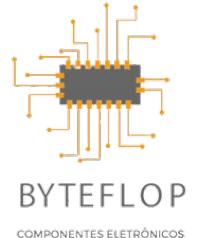
- [CMU Sphinx](#) (works offline)
- Google Speech Recognition
- [Google Cloud Speech API](#)
- [Wit.ai](#)
- [Microsoft Bing Voice Recognition](#)
- [Houndify API](#)
- [IBM Speech to Text](#)
- [Snowboy Hotword Detection](#) (works offline)



Realização:



Patrocínios:



Speech to text

```
1 import speech_recognition as sr
2
3 #Funcao responsavel por ouvir e reconhecer a fala
4 def ouvir_microfone():
5     #Habilita o microfone para ouvir o usuario
6     microfone = sr.Recognizer()
7     with sr.Microphone() as source:
8         #Chama a funcao de reducao de ruido disponivel na speech_recognition
9         microfone.adjust_for_ambient_noise(source, duration=1)
10        #Avisa ao usuario que esta pronto para ouvir
11        print("Diga alguma coisa: ")
12        #Armazena a informacao de audio na variavel
13        audio = microfone.listen(source, timeout=2, phrase_time_limit=4)
14    try:
15        #Passa o audio para o reconhecedor de padroes do speech_recognition
16        frase = microfone.recognize_google(audio, language='pt-BR')
17
18        #Caso nao tenha reconhecido o padrao de fala, exibe esta mensagem
19    except sr.UnknownValueError:
20        print("Não entendi")
21
22    return frase
23
24 frase = ouvir_microfone()
25 print('Resposta: ' + frase)
```



Algumas outras

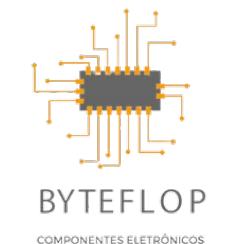
- [PyGame](#)
- Banco de Dados
- [Telegram](#)
- [Chatbot](#)
- [Firebase](#)
- IA
- [Flask](#)
- Entre muitos outros



Realização:



Patrocínios:



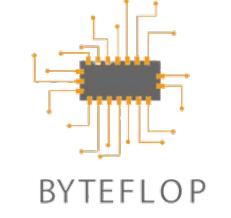
Lembretes



Realização:



Patrocínios:



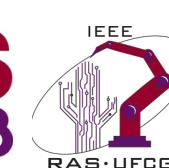
Workshop de ROS



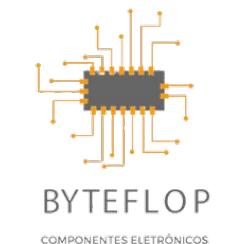
<https://bit.ly/40IEEEna-ROS>

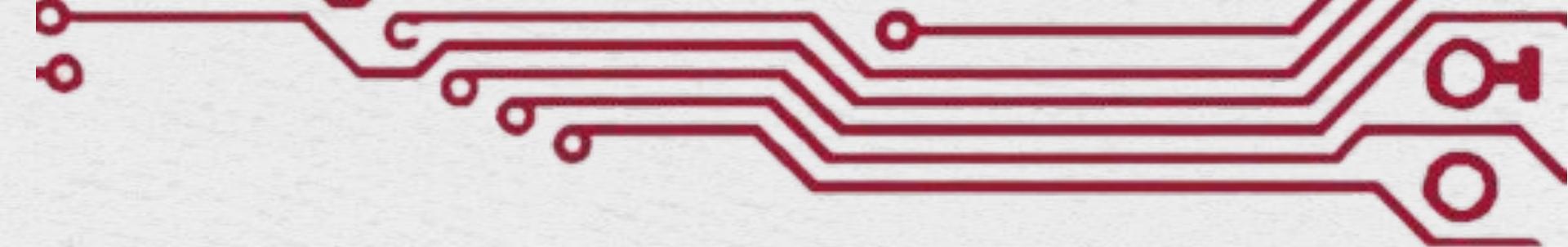


Realização:



Patrocínios:





Obrigado pela participação!

Email para contato:

mateusantonio@eng.ci.ufpb.br



Realização:

RAS UFPB
 RAS UFCG
 IEEE RAS JEL
Robotics & Automation Society

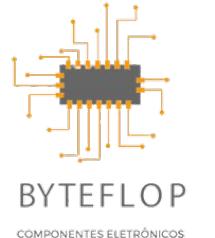
RAS IFPB
Robotics & Automation Society

RAS UNIFACS
Robotics & Automation Society

IEEE RAS CIMATEC
Robotics and Automation Society

RAS
STUDENT CHAPTER ROBOTICS AND AUTOMATION SOCIETY IEEE UFRB

Patrocínios:



BYTEFLOP



teorienta!



CNN
Centro Avançado de
Neuro e Coluna