Engenharia de Computação

LABORATÓRIO

Coordenação: Prof. Dr. Angelo Sebastião Zanini

INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

Time de Computação

Coordenação:

Prof. Dr. Ângelo Sebastião Zanini

Professores:

Prof. Nuncio Perrella

Prof. Everson Denis

Corpo Técnico:

Eng. Gabriel Godoy

Enga. Ana Helena

Eng. Matheus Coelho

Meio Ambiente – Poluição Sonora

Poluição sonora: excesso de sons que afetam os seres humanos e o ambiente.

Fontes Sonoras: Trânsito, indústrias, construção civil, eventos, aeroportos, entre outros.

Impactos

Saúde

- Fisiológicos: perda auditiva, aumento da pressão arterial, estresse.
- Psicológicos: irritabilidade, ansiedade, dificuldades de concentração e distúrbios do sono.
- Desempenho Cognitivo: redução da capacidade de aprendizado e produtividade.

Meio Ambiente

- Fauna: alteração nos padrões de comunicação dos animais, afetando suas interações sociais e reprodutivas.
- Flora: embora menos evidente, há estudos que indicam impactos no crescimento de plantas expostas a altos níveis de ruído.

Níveis Sonoros

A medição de níveis sonoros é feita utilizando escala em **decibéis (dB)**.

A **escala em decibéis (dB)** é usada para quantificar níveis sonoros e comparar a intensidade do som. Como o ouvido humano percebe sons de forma não linear, a escala dB é baseada em **razões logarítmicas**, tornando-a mais adequada para representar nossa percepção auditiva.

O equipamento mais comum para essa medição é o **medidor de nível de pressão sonora (Decibelímetro ou Sonômetro)**, que capta as variações de pressão do ar causadas pelas ondas sonoras.

Decibel

O **decibel (dB)** é uma unidade de medida relativa que expressa a razão entre dois valores de pressão sonora.

Ele é definido pela seguinte fórmula:

$$L = 10 \cdot \log\left(\frac{P^2}{P_0^2}\right) = 20 \cdot \log\left(\frac{P}{P_0}\right)$$

Onde:
L = Nível de pressão sonora em **dB**
P = Pressão sonora medida (**Pa**)
P0 = Pressão de referência (geralmente 20 µPa, o limiar da audição humana)

Características da Escala Logarítmica

A escala de decibéis é **logarítmica**, o que significa que:

- Um **aumento de 10 dB** representa um **som 10 vezes mais intenso**.
- Um **aumento de 20 dB** representa um **som 100 vezes mais intenso**.
- Um **aumento de 30 dB** representa um **som 1.000 vezes mais intenso**.

Exemplo:

•**40 dB** (conversa em tom baixo)
•**50 dB** (ruído de escritório) → 10 vezes mais intenso que 40 dB
•**60 dB** (conversa normal) → 100 vezes mais intenso que 40 dB

Por isso, um aumento de apenas alguns decibéis pode significar uma grande diferença na percepção do som.

Níveis Sonoros

Tipo de área	Diurno (7h-22h)	Noturno (22h-7h)
Área estritamente residencial	50 dB	45 dB
Área predominantemente residencial	55 dB	50 dB
Área mista (residencial/comercial)	60 dB	55 dB
Área comercial/administrativa	65 dB	60 dB
Área industrial	70 dB	65 dB

ABNT NBR 10.151/2019

Níveis Sonoros

Nível (dB)	Fonte do Som	Efeito no Ouvido Humano
10 dB	Respiração, estúdio de gravação silencioso	Quase imperceptível
20 dB	Sussurro, biblioteca silenciosa	Muito baixo
30 dB	Quarto silencioso à noite	Tranquilo
40 dB	Sala de estar, conversação baixa	Som ambiente confortável
50 dB	Escritório silencioso, chuva fraca	Pouco incômodo
60 dB	Conversa normal, ar-condicionado	Nível moderado
70 dB	Trânsito leve, aspirador de pó	Limite confortável por longos períodos
80 dB	Trânsito intenso, liquidificador	Pode ser incômodo
90 dB	Moto, secador de cabelo	Prolongado pode causar danos
100 dB	Show de rock, buzina	Exposição prolongada pode causar perda auditiva
110 dB	Serra elétrica, boate	Muito alto e prejudicial
120 dB	Motor de avião a 30m, trovão	Dores no ouvido
130 dB	Tiro de arma de fogo, fogos de artifício	Risco imediato de dano auditivo
140+ dB	Explosão, motor de jato próximo	Limiar da dor

Dano Auditivo e Segurança

A exposição prolongada a níveis sonoros elevados pode causar **danos auditivos permanentes**.

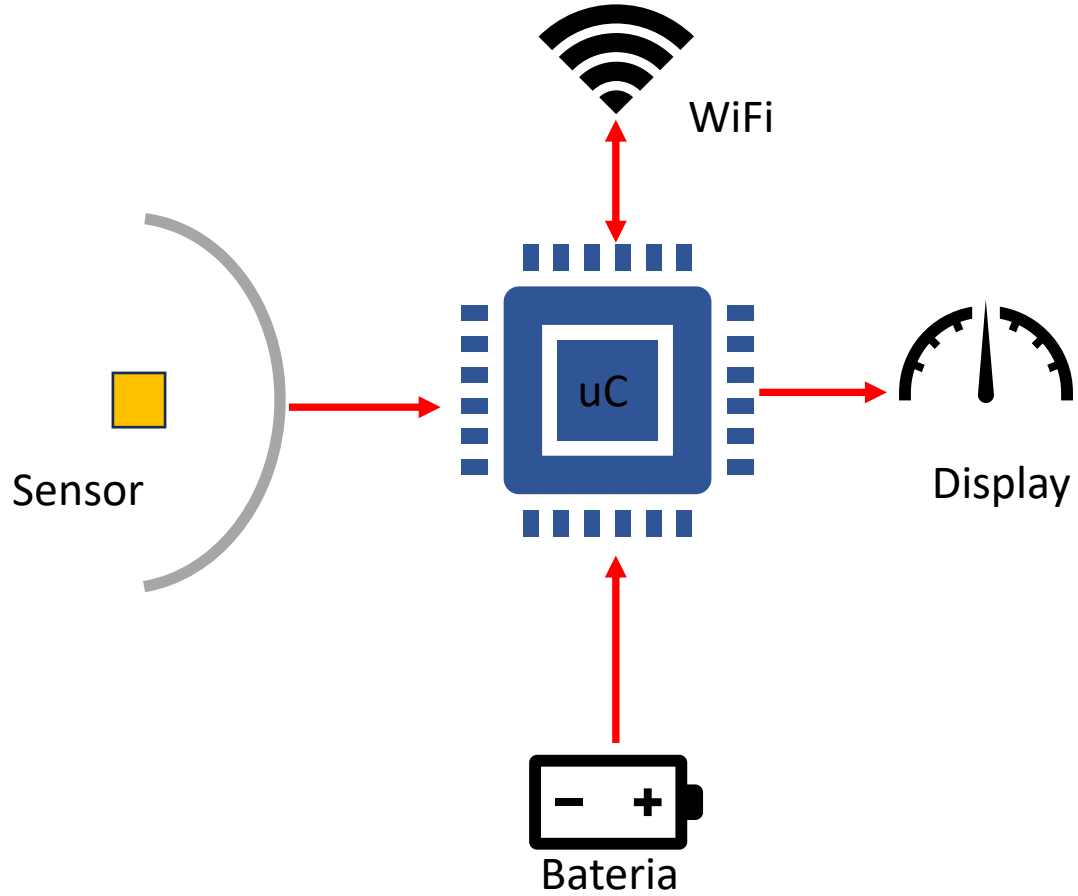
Segundo normas de segurança (como a **NR-15** no Brasil):

- **Acima de 85 dB** → Pode causar danos auditivos após exposição prolongada.
- **Acima de 120 dB** → Pode causar dor e danos imediatos.

Projeto

- Criar sistema de tempo real para monitoramento sonoro gerando alarmes e alertas.

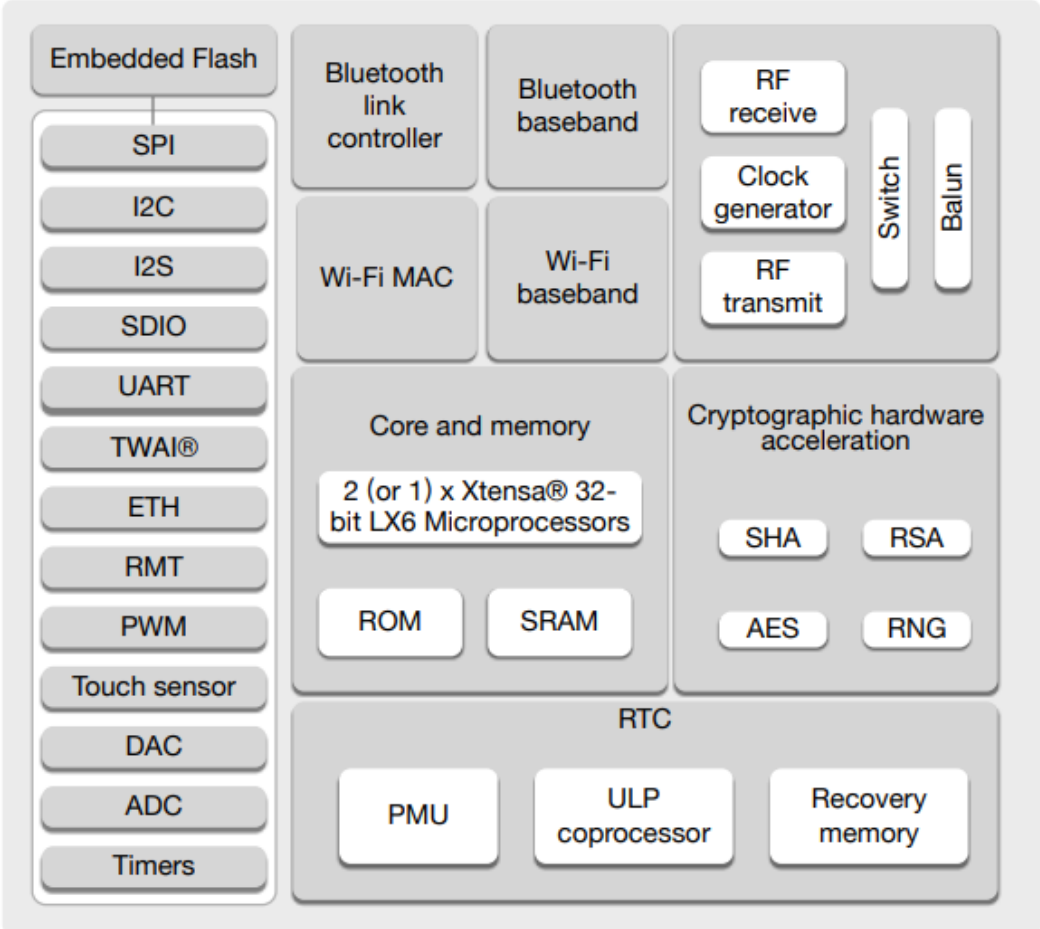
Diagrama de Blocos do Projeto



Dashboard

Roteador

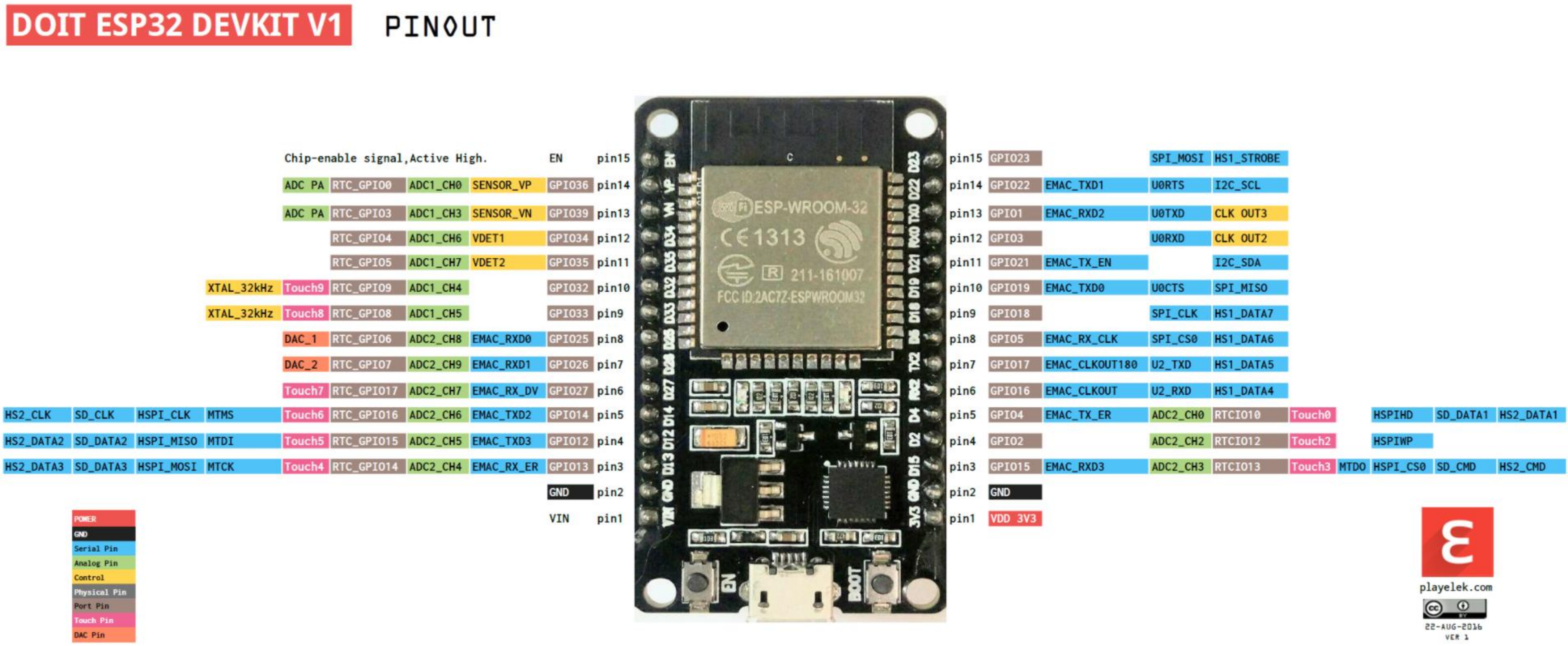
uC - ESP32



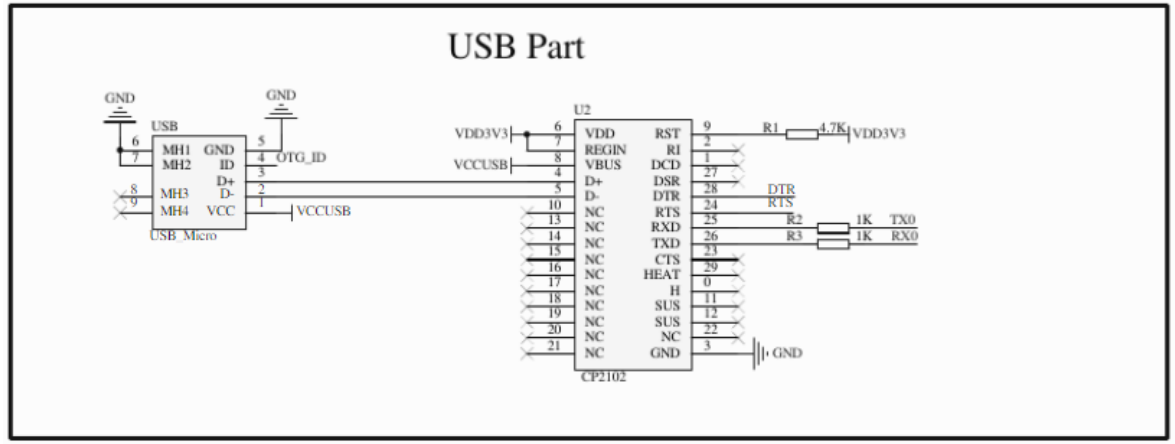
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

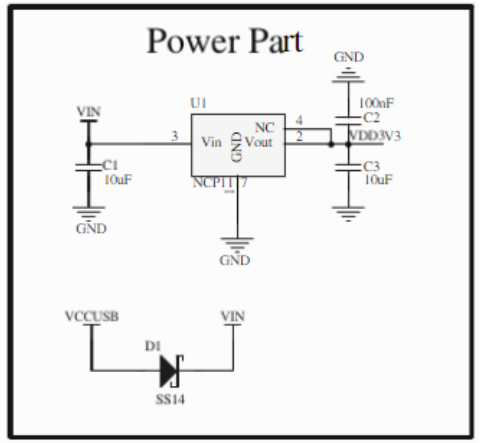
ESP32 DOIT DEV KIT V1 - HW Básico

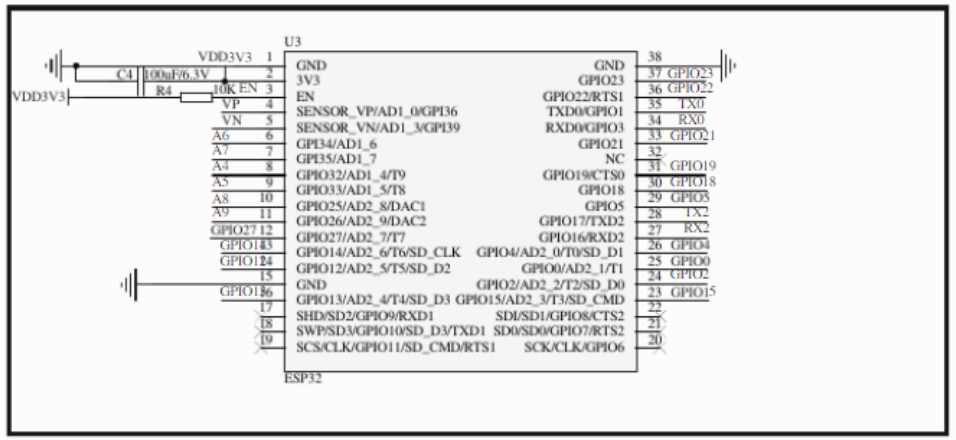
INSTITUTO MAUÁ DE TECNOLOGIA
MAUÁ

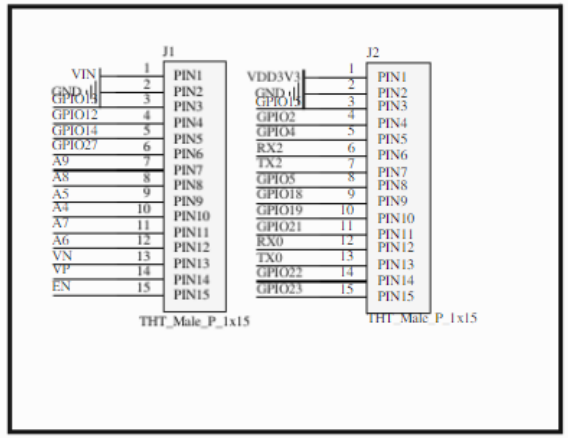


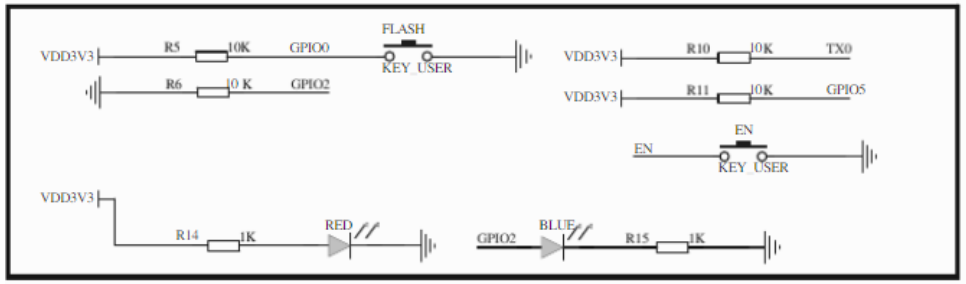
ESP32 DOIT DEV KIT V1 - HW Básico

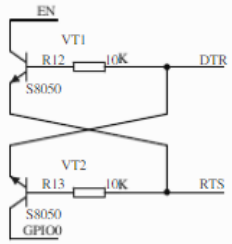




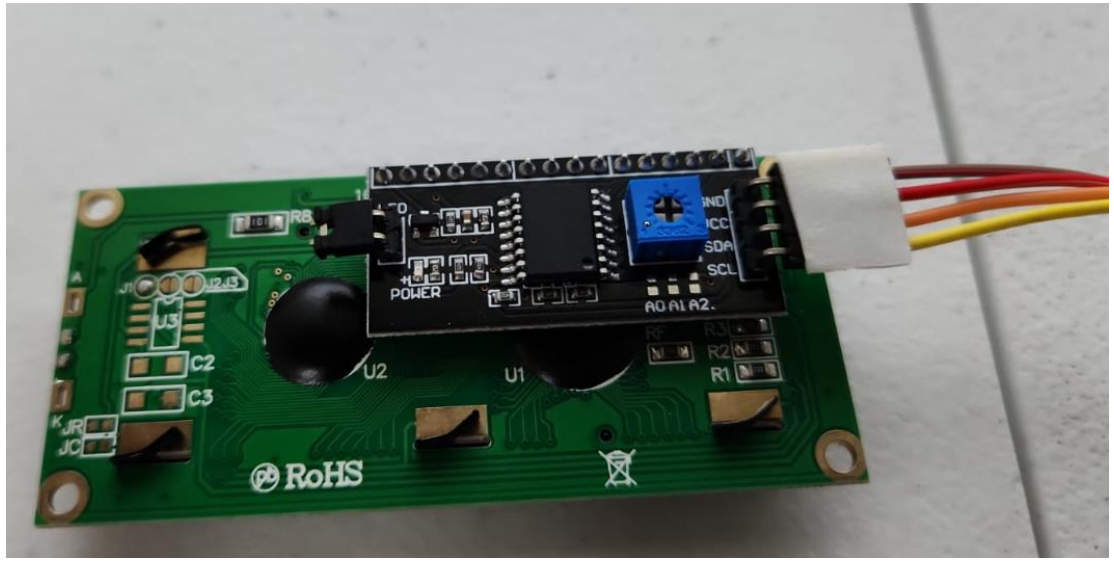


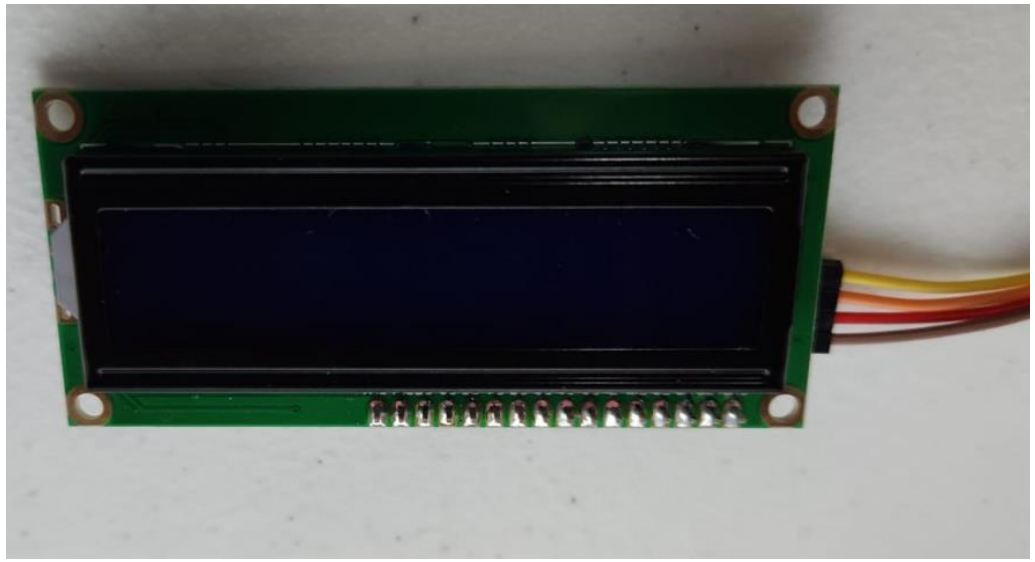






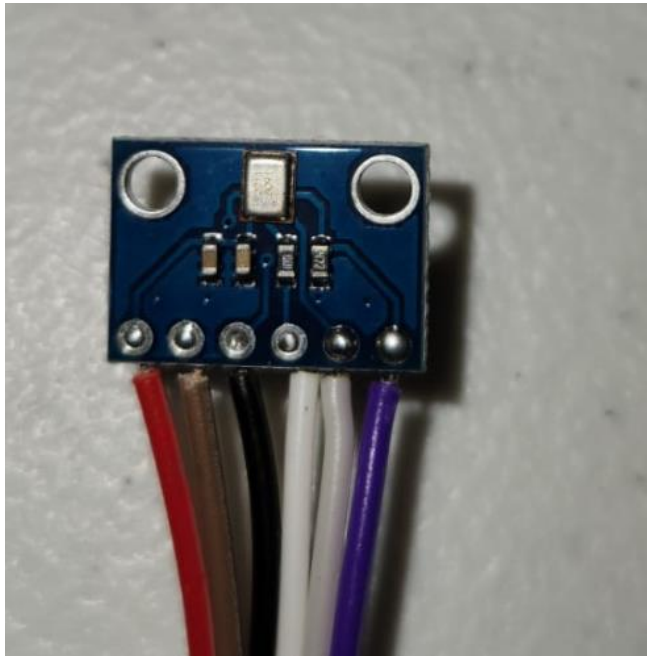
Display

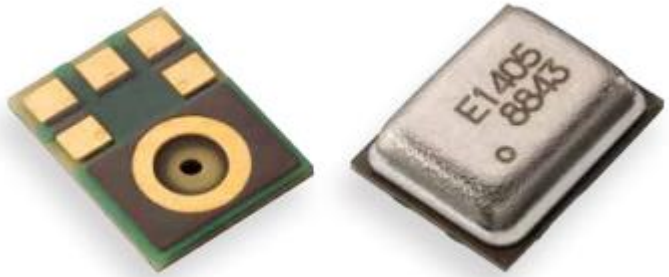


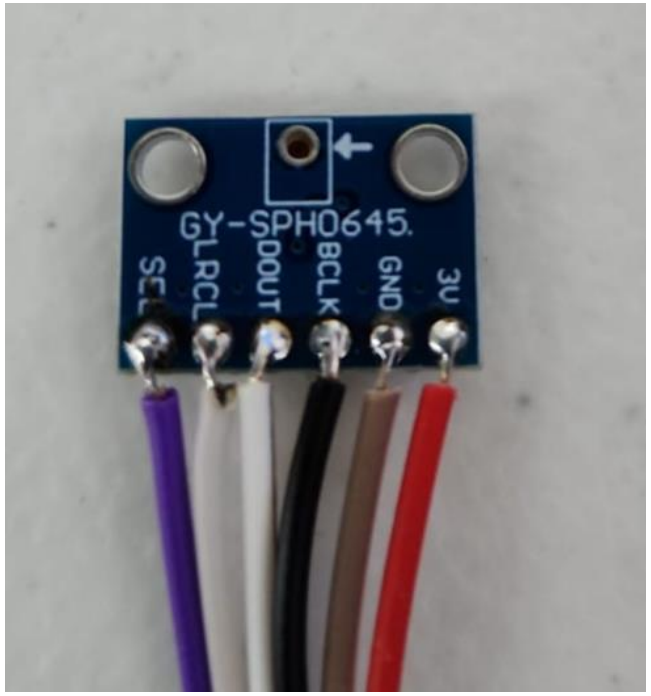


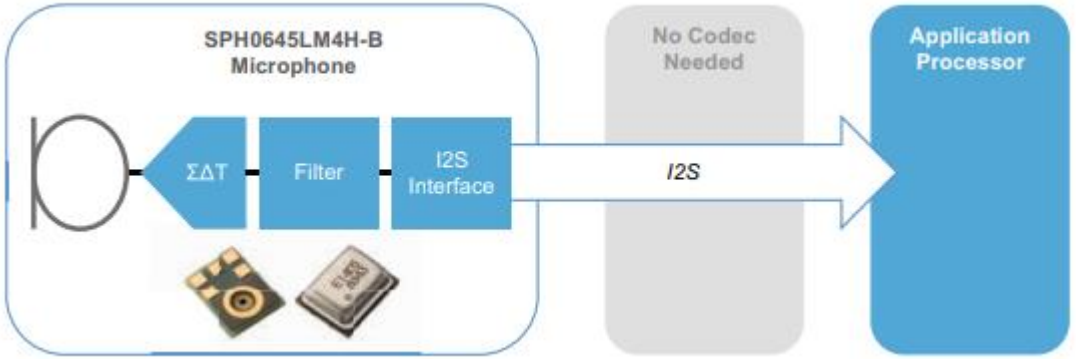
Vermelho → Vin (5V)
Marrom → GND
Laranja → D21 (ESP32)
Amarelo → D22 (ESP32)

Sensor SPH0645

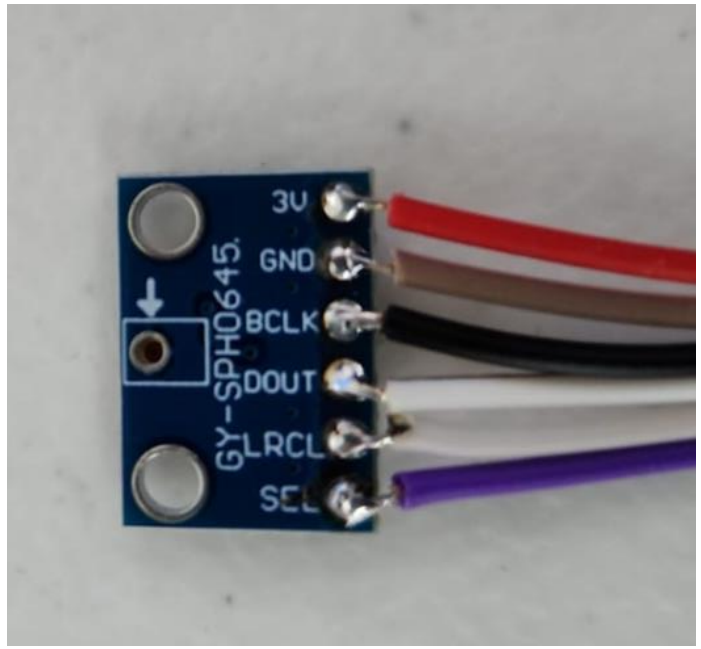






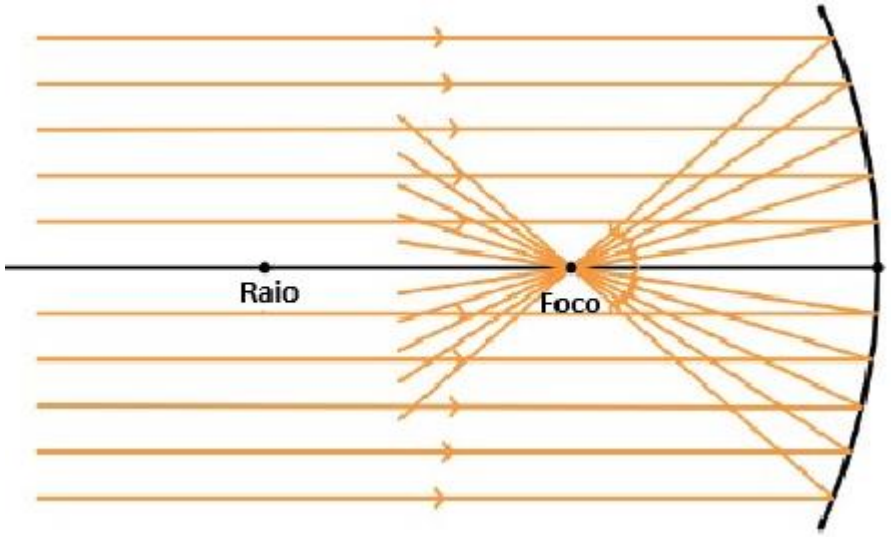


Sensor SPH0645



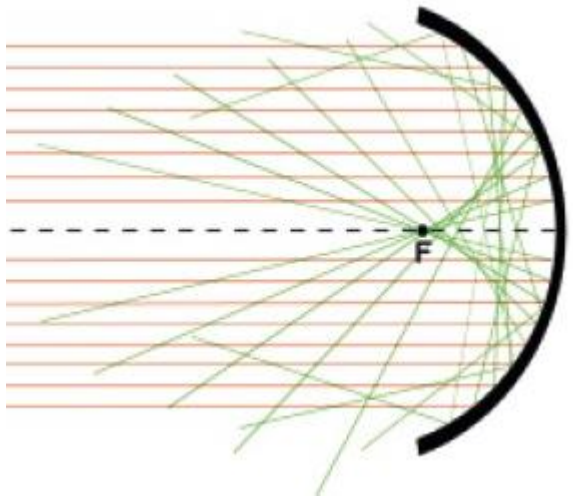
Vermelho → 3V3
Marrom → GND
Preto → D23 (ESP32)
Branco → D32 (ESP32)
Cinza → D33 (ESP32)
Roxo → GND

Refletor Esférico Gaussiano

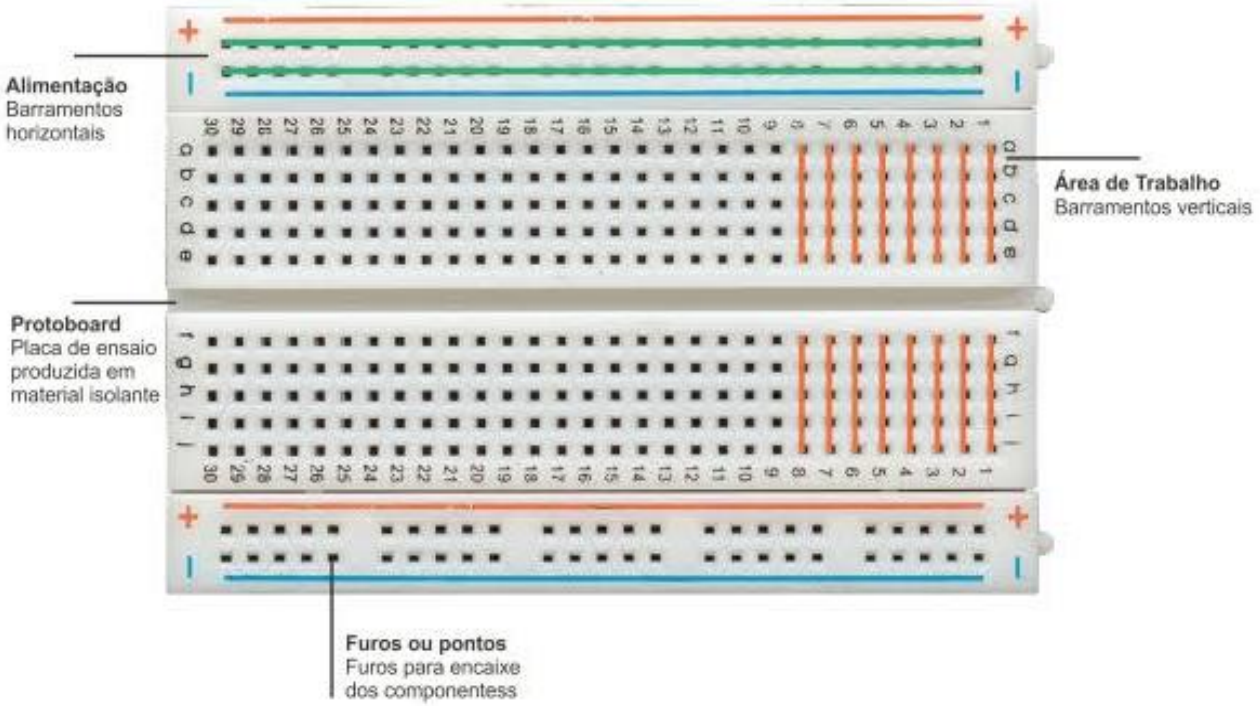


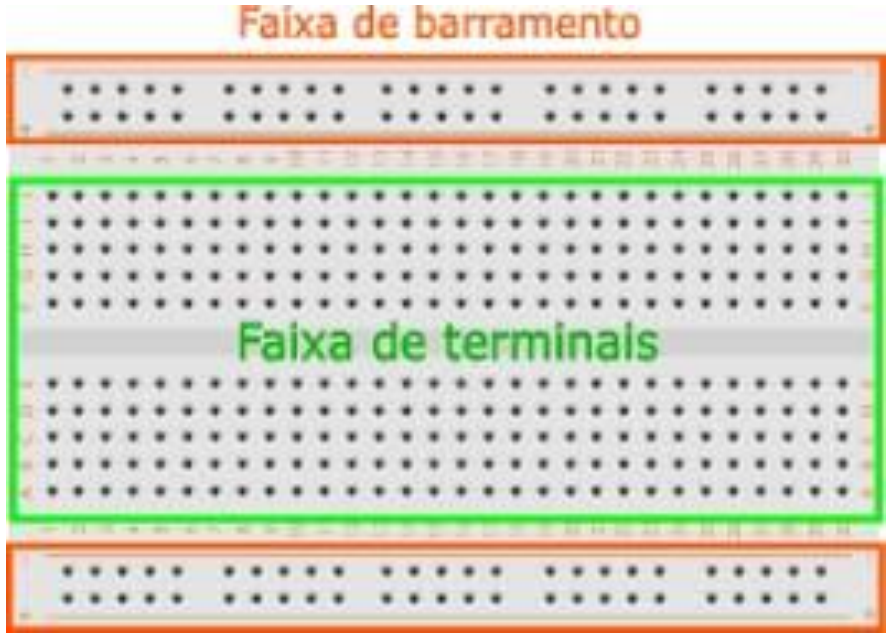
Foco = Raio / 2

Refletor Esférico Gaussiano



Protoboard


Alimentação
Barramentos horizontais
Área de Trabalho
Barramentos verticais
Protoboard
Placa de ensaio produzida em material isolante
Furos ou pontos
Furos para encaixe dos componentess


Faixa de barramento
Faixa de terminais

Ambiente de Desenvolvimento (IDE)

Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embarcados, uma linguagem de programação padrão,[a qual tem origem em Wiring, e é essencialmente C/C++.

O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e ferramentas mais complicadas.

Pode ser usado para o desenvolvimento de objetos interativos independentes, ou ainda para ser conectado a um computador.

Uma típica placa Arduino é composta por um controlador, alguns pinos de E/S digital e analógica, além de uma interface serial ou USB, para interligar-se ao computador, que é usado para programá-la e interagir em tempo real. A placa em si não possui qualquer recurso de rede, porém é comum combinar um ou mais Arduinos deste modo, usando extensões apropriadas chamadas de shields.
A interface do computador é simples, podendo ser escrita em várias linguagens. A mais popular é a Processing, mas outras que podem comunicar-se com a conexão serial são: Max/MSP, Pure Data, SuperCollider, ActionScript e Java.

O nome Arduino vem de um bar em Ivrea, Itália, onde alguns dos fundadores do projeto costumavam se reunir.

Fonte: https://pt.wikipedia.org/wiki/Arduino (17/11/2021)

Arduino

ARDUINO

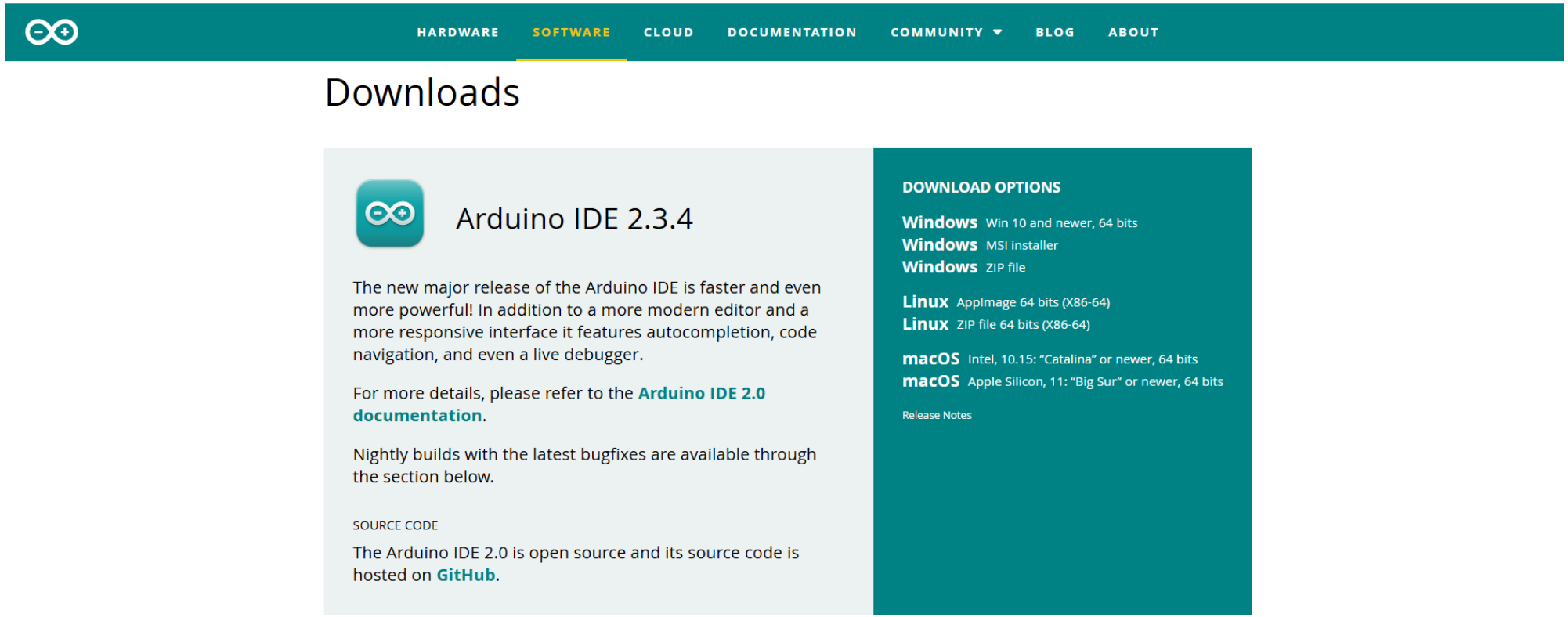
Arduino Uno

Desenvolvedor	• Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. • Baseado no *Processing*, de Casey Reas e Ben Fry. • Comunidade Código aberto.
Plataforma	C/C++
Lançamento	2005
Versão estável	1.8.11 (27 de janeiro de 2020; há 21 meses [1])
Versão em teste	1.9.0
Idioma(s)	67 idiomas[2]
Escrito em	Java
Sistema operacional	Microsoft Windows, Linux, Mac OS X[3][4]
Gênero(s)	Ambiente de desenvolvimento integrado
Licença	• Software em LGPL ou GPL • Hardware em Creative Commons
Estado do desenvolvimento	Ativo
Página oficial	http://www.arduino.cc/en/ (em inglês)

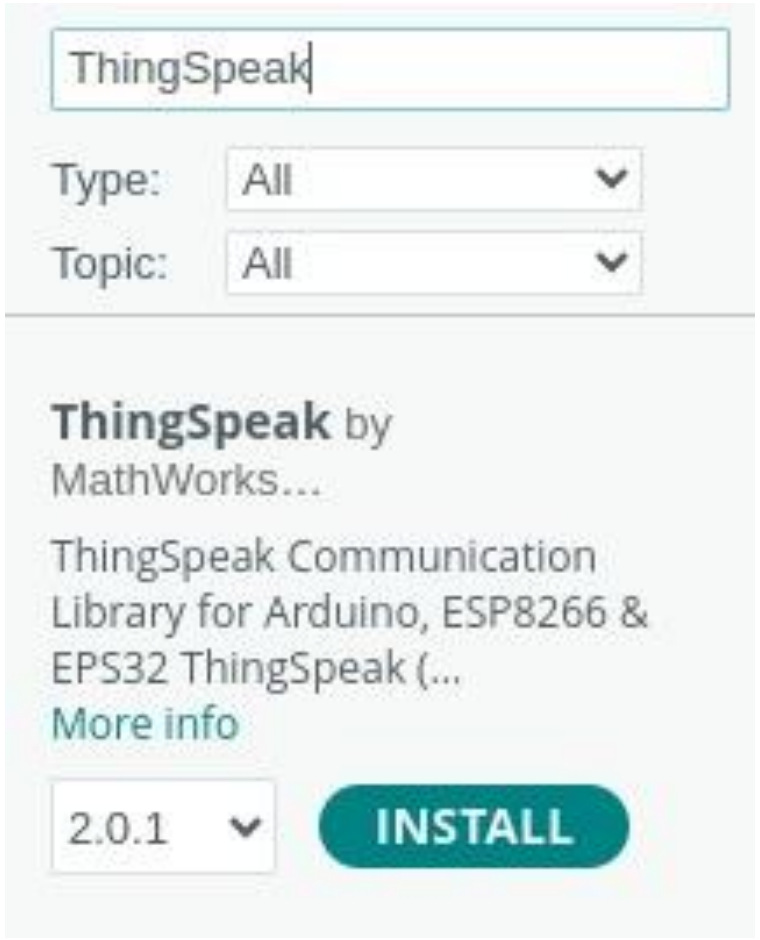
Ambiente de Desenvolvimento (IDE)

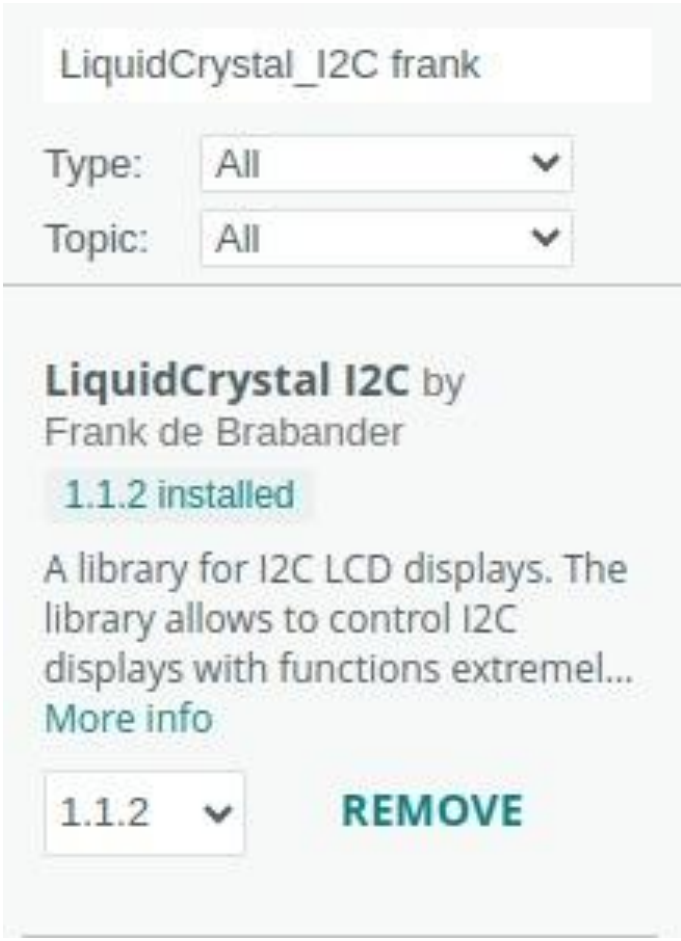
Fazer Download e instalar Arduino IDE (versão mais recente)

https://www.arduino.cc/en/Main/Software

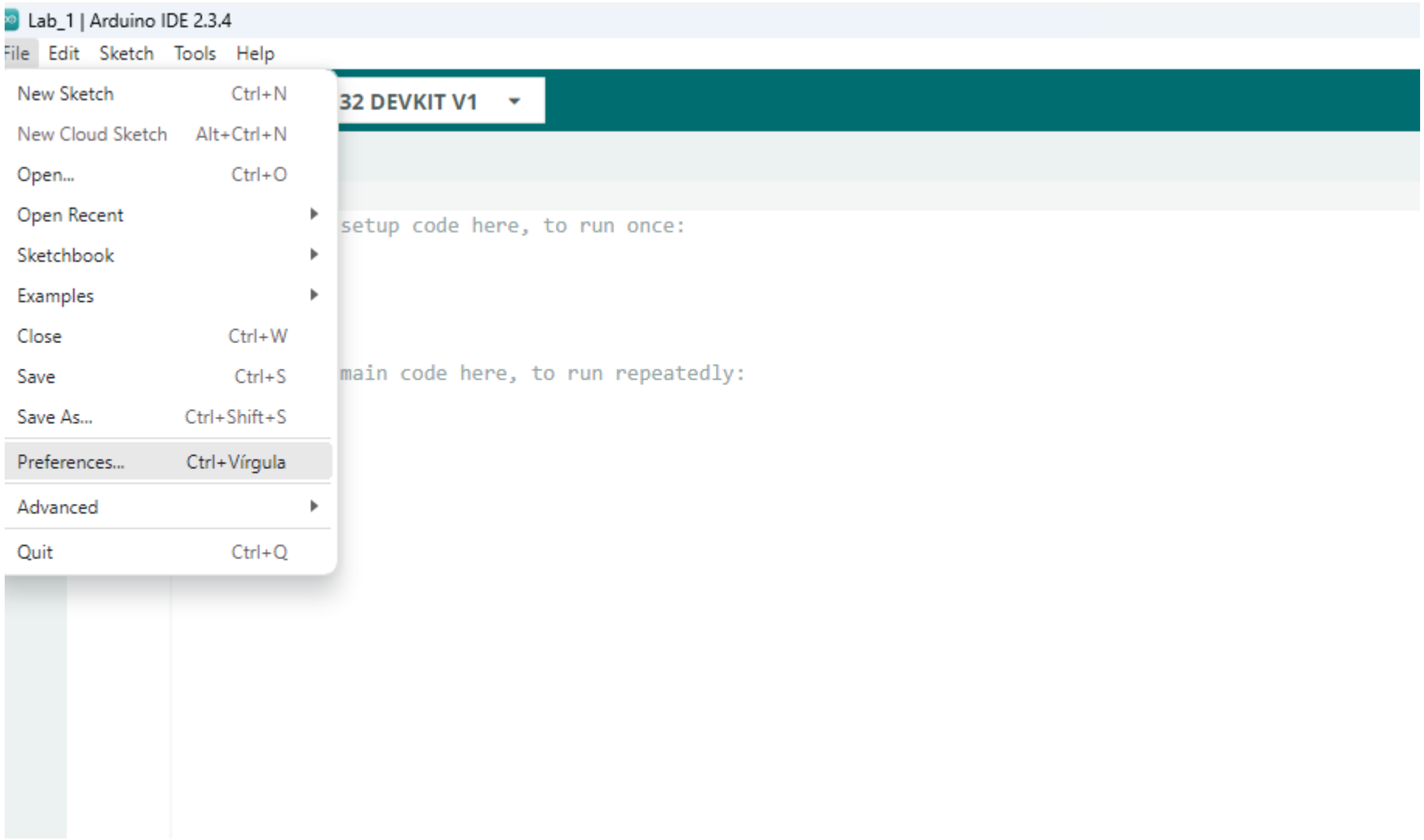


Bibliotecas Instaladas

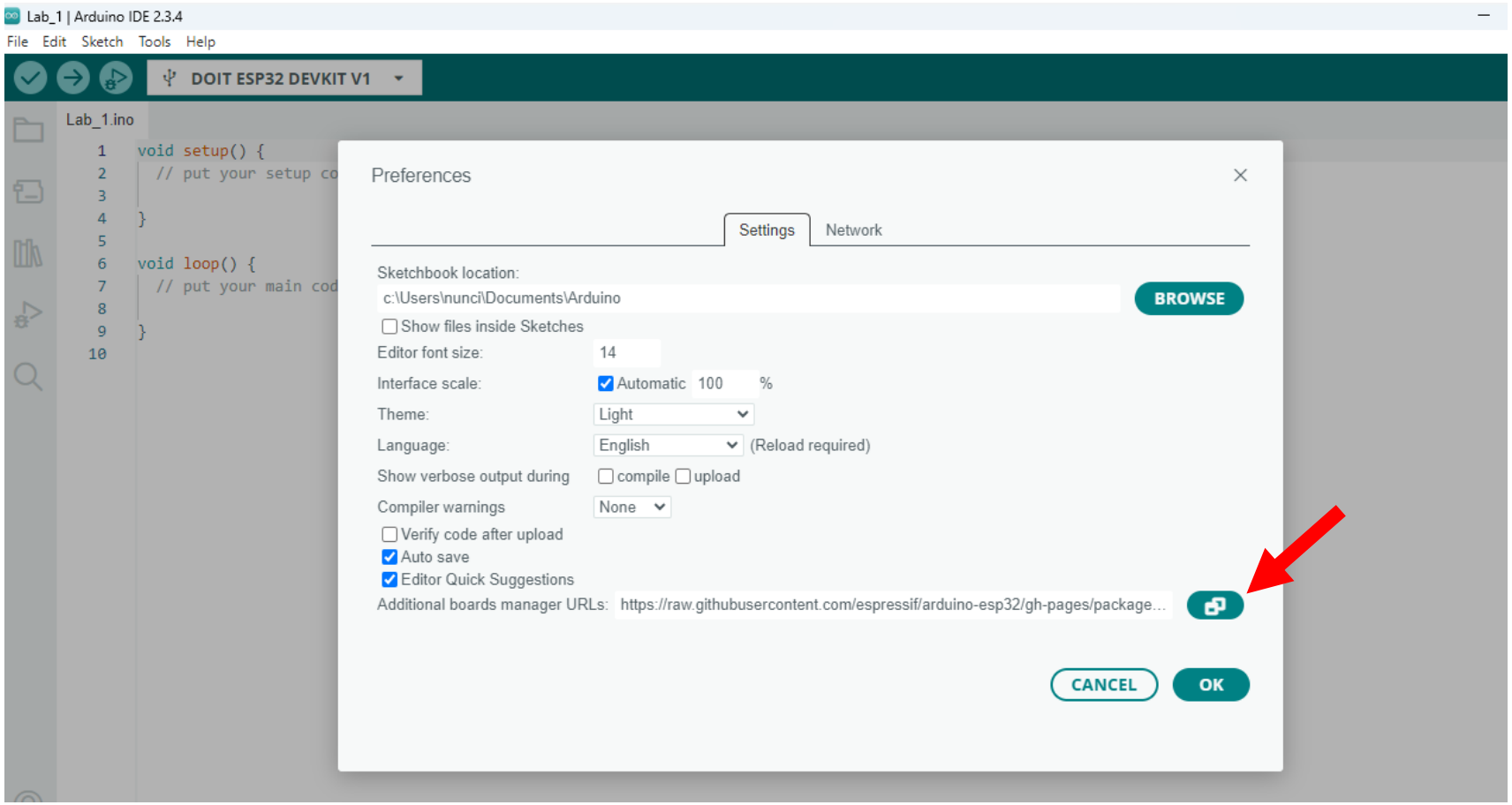




Configurando IDE

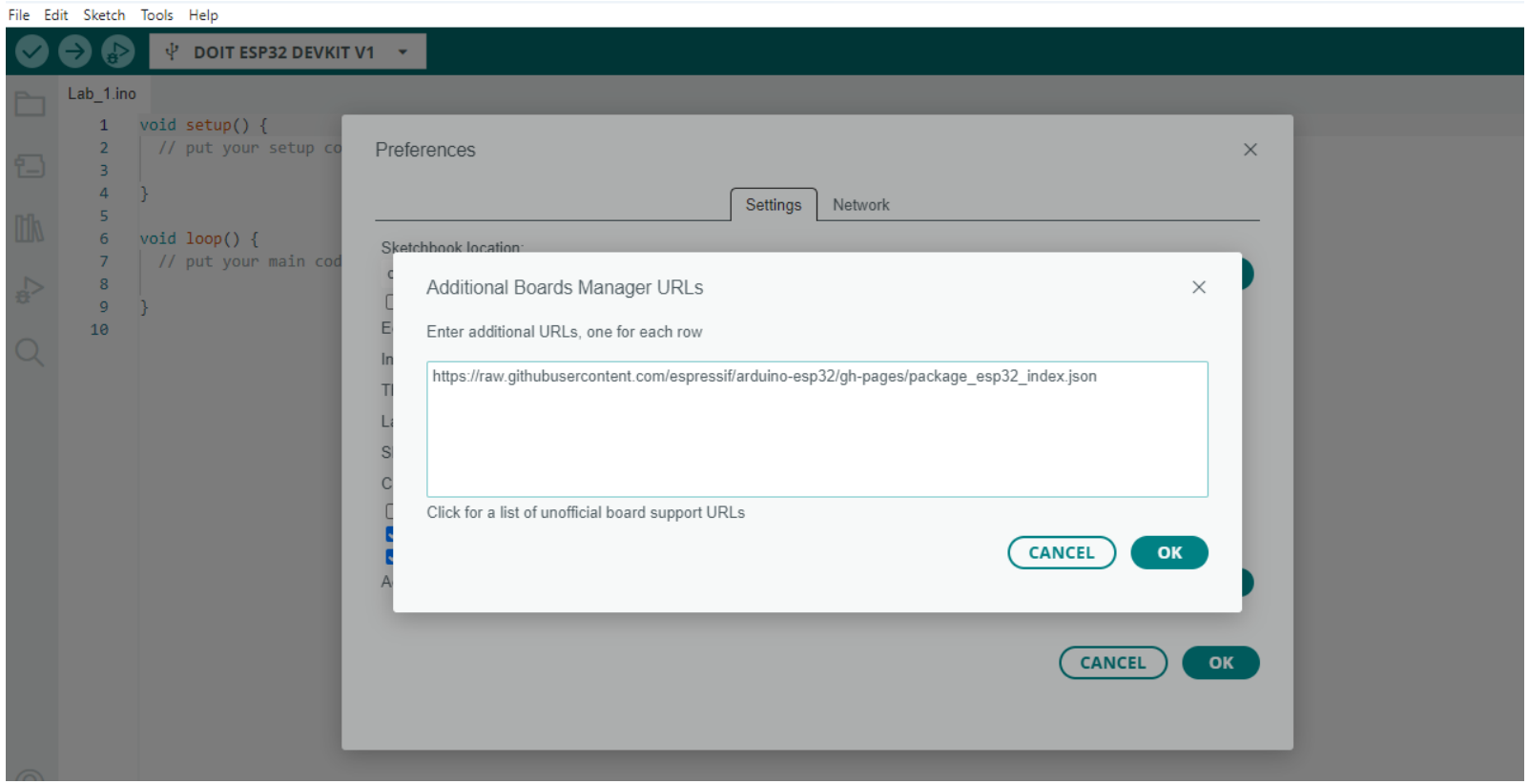


Configurando IDE

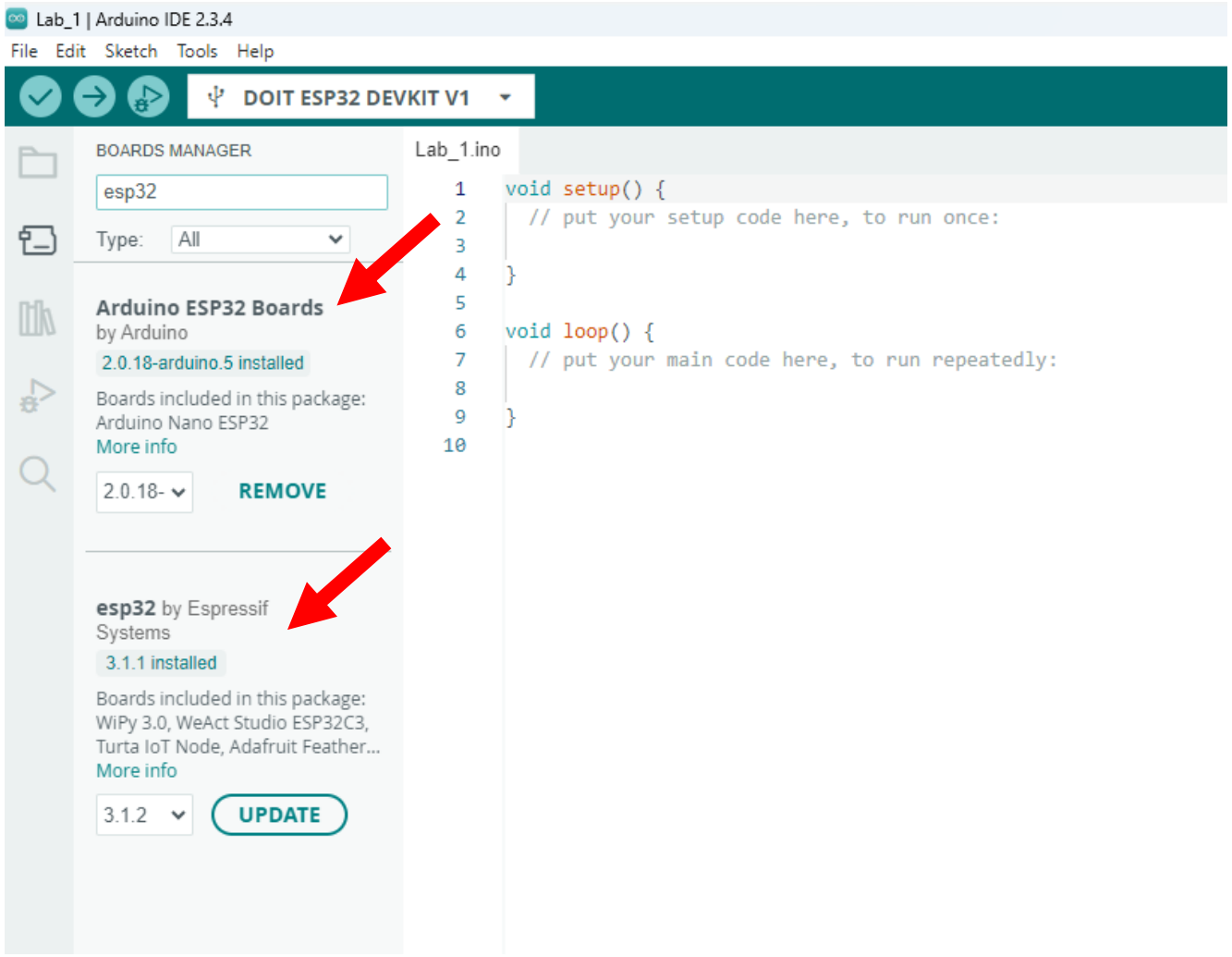


Configurando IDE

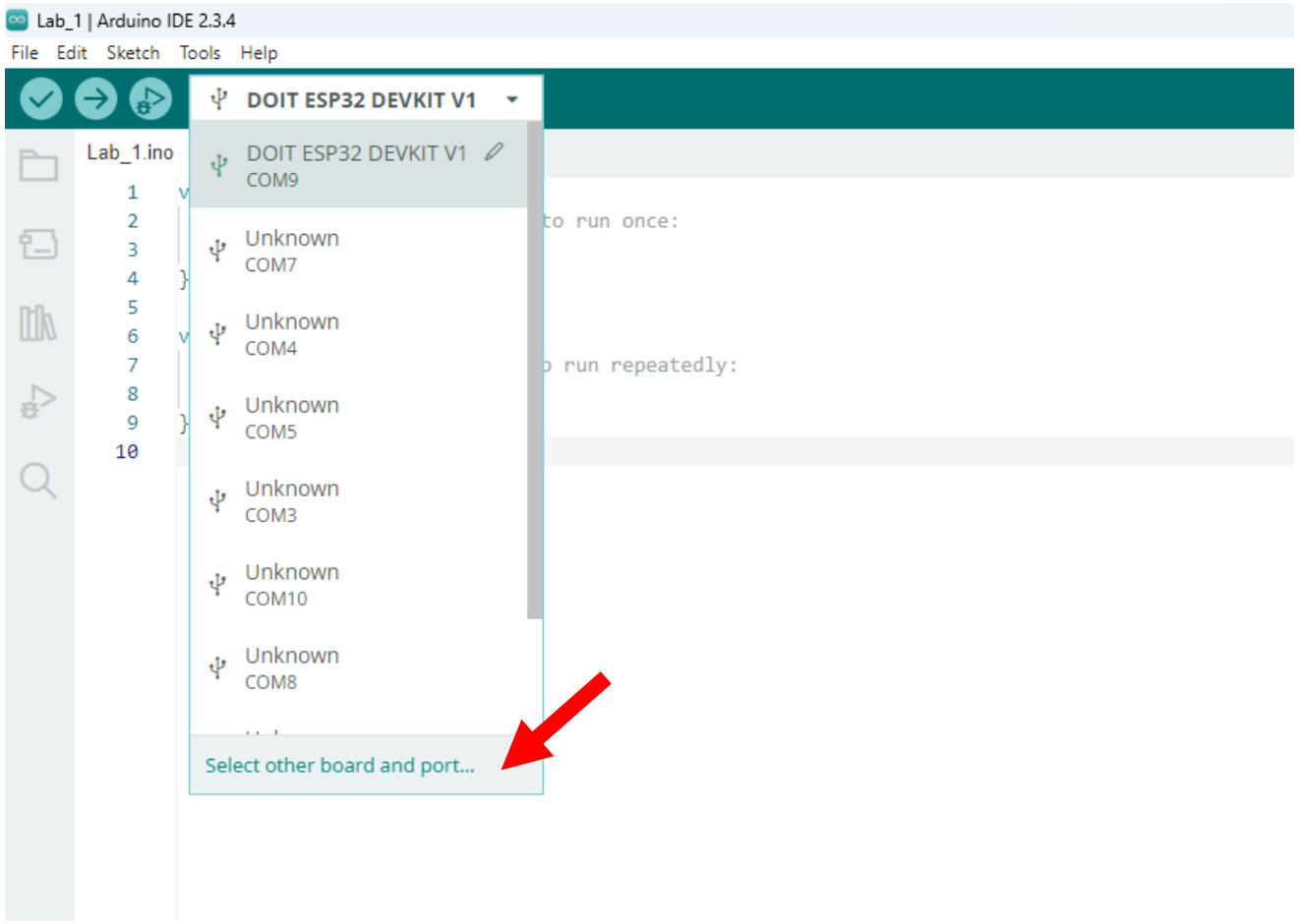
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



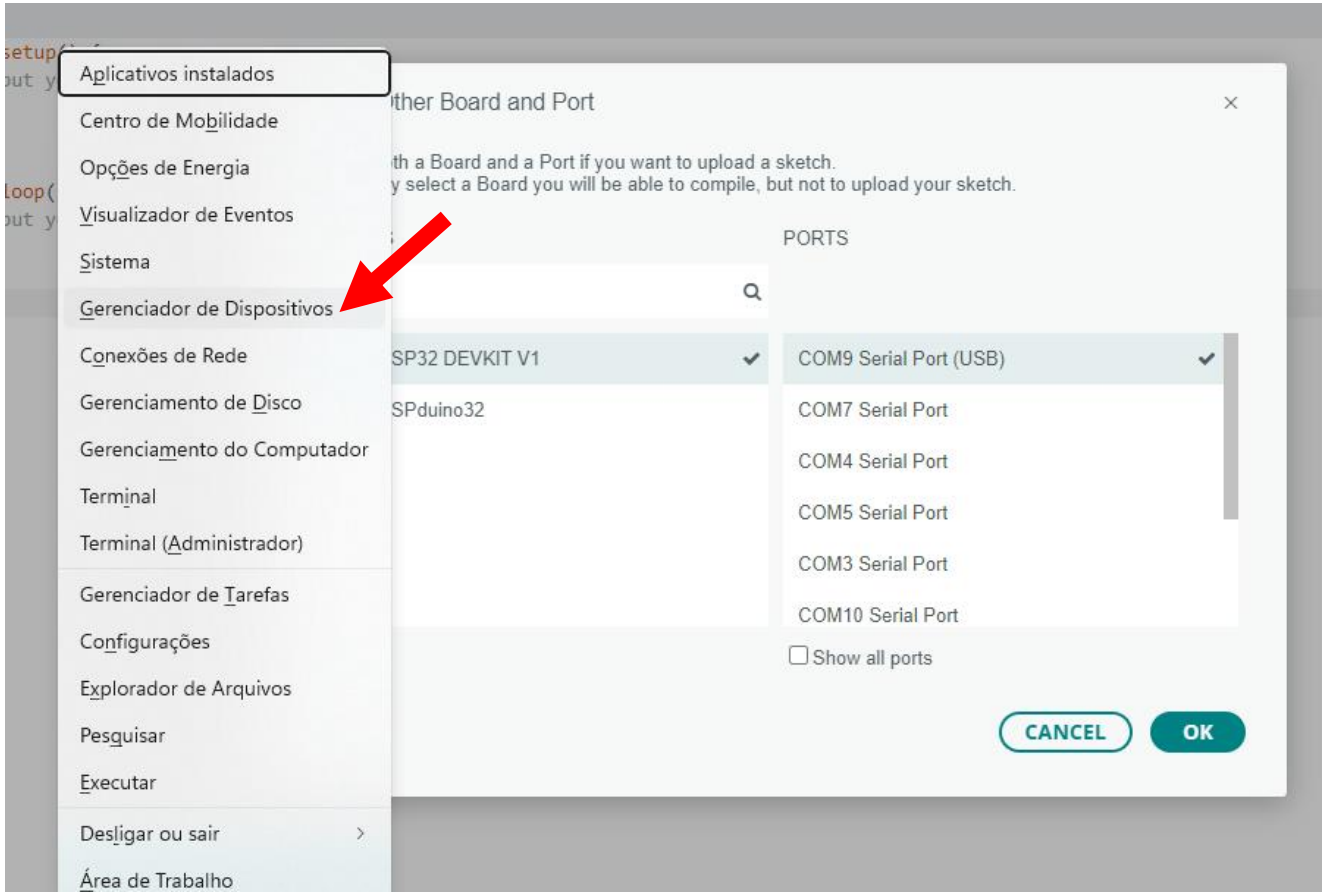
Configurando IDE – Procurar ESP32



Configurando IDE – ESP32 DOIT DEVKIT V1

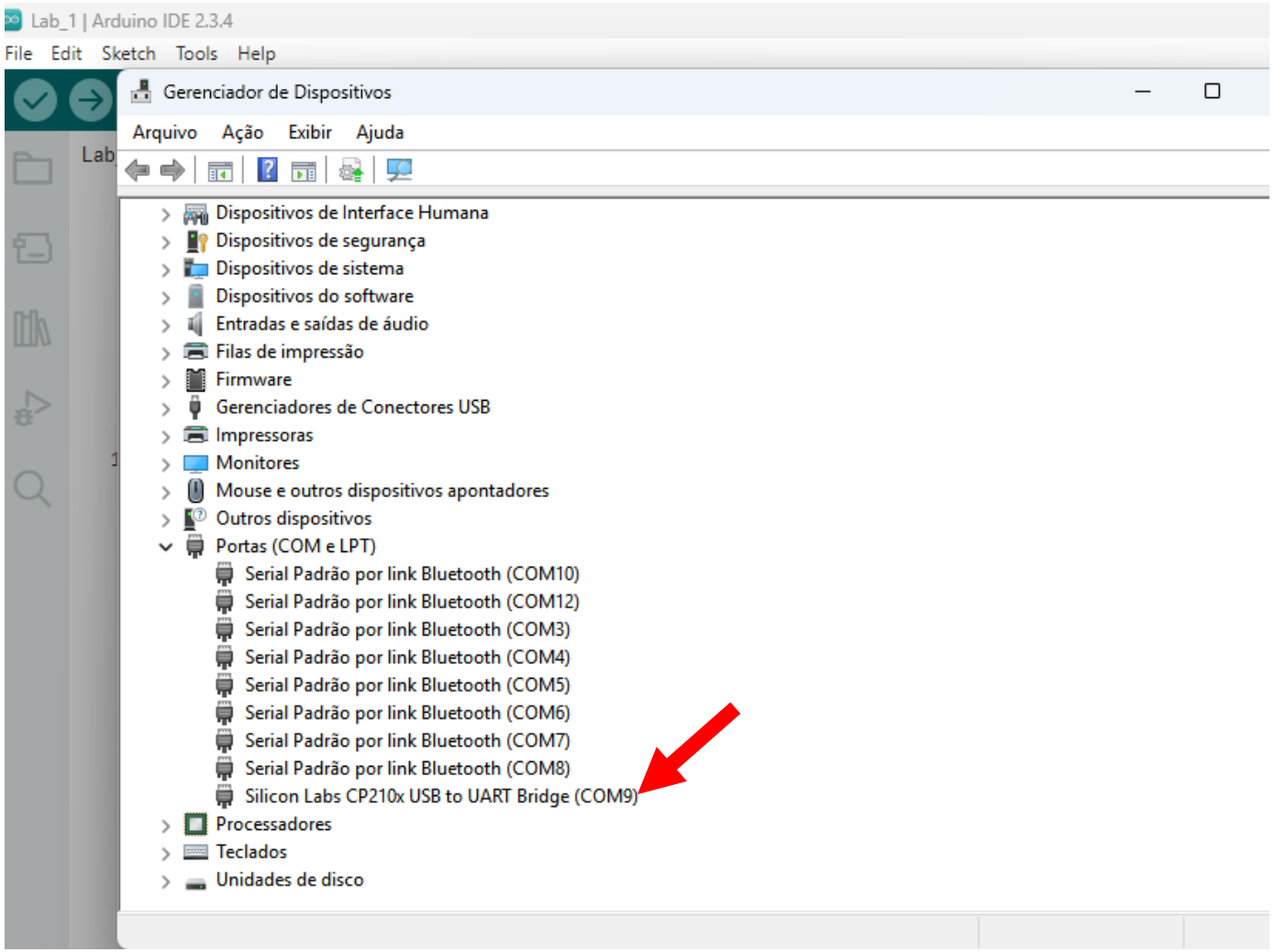


Configurando IDE – ESP32

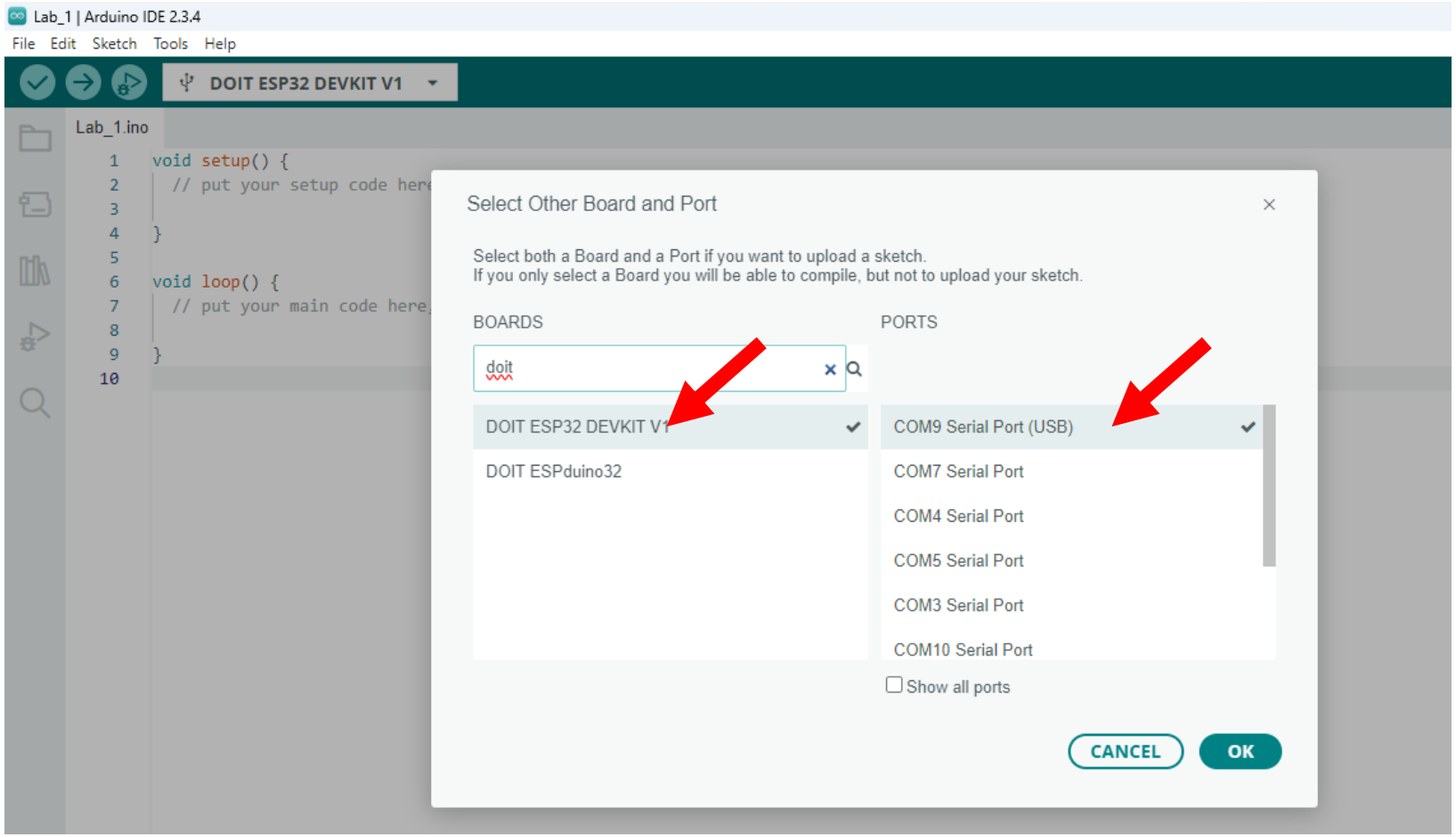


Teclas: Windows + x

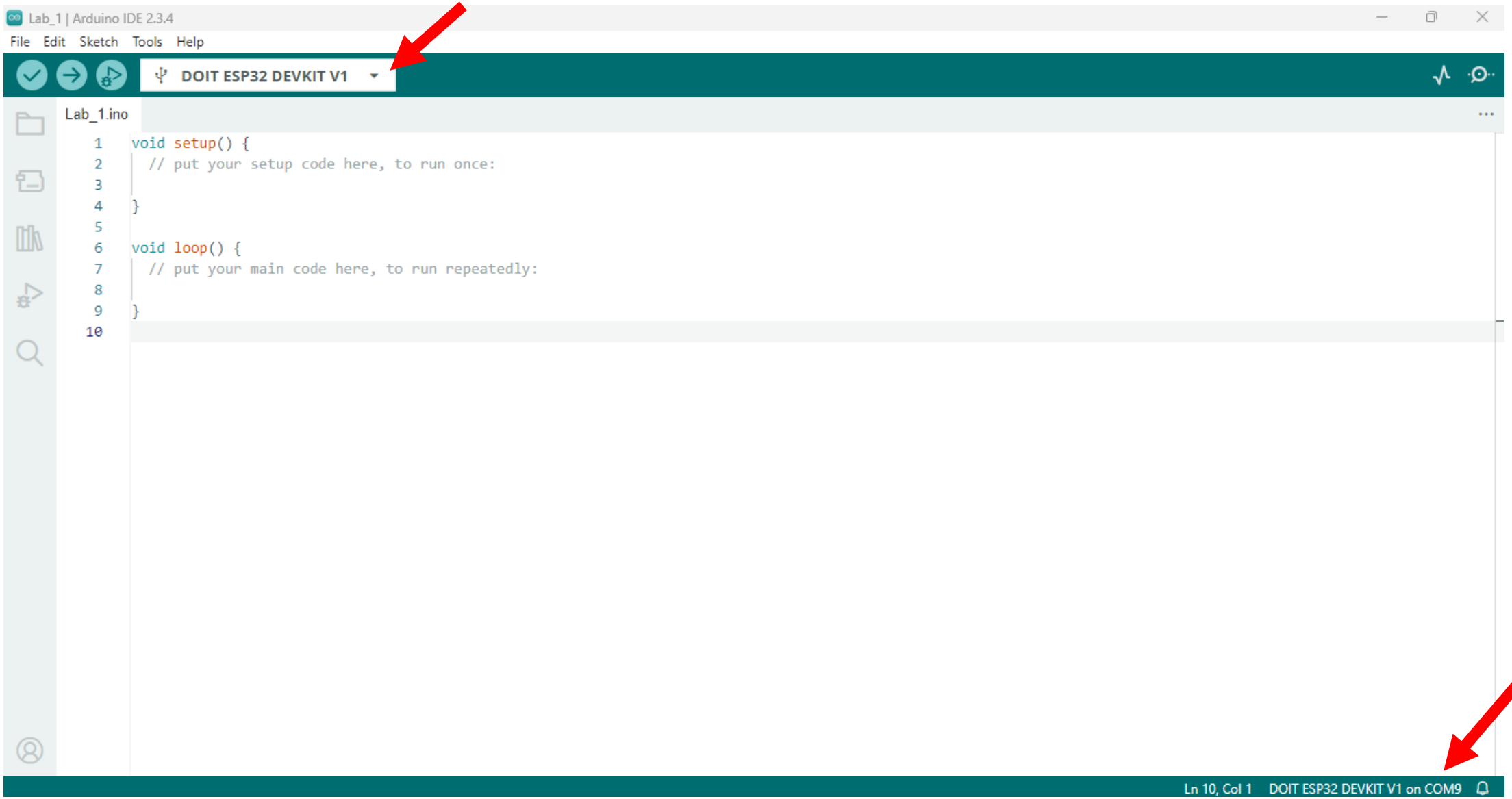
Configurando IDE – ESP32



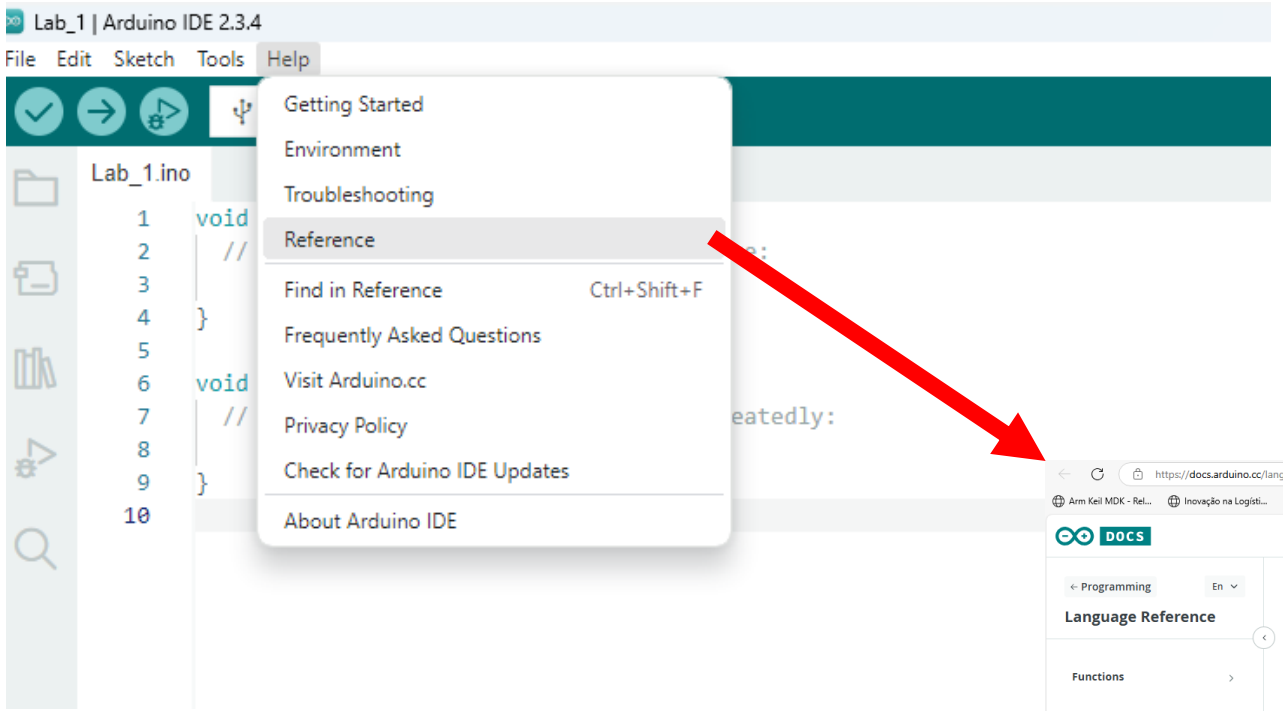
Configurando IDE – ESP32 DOIT DEVKIT V1

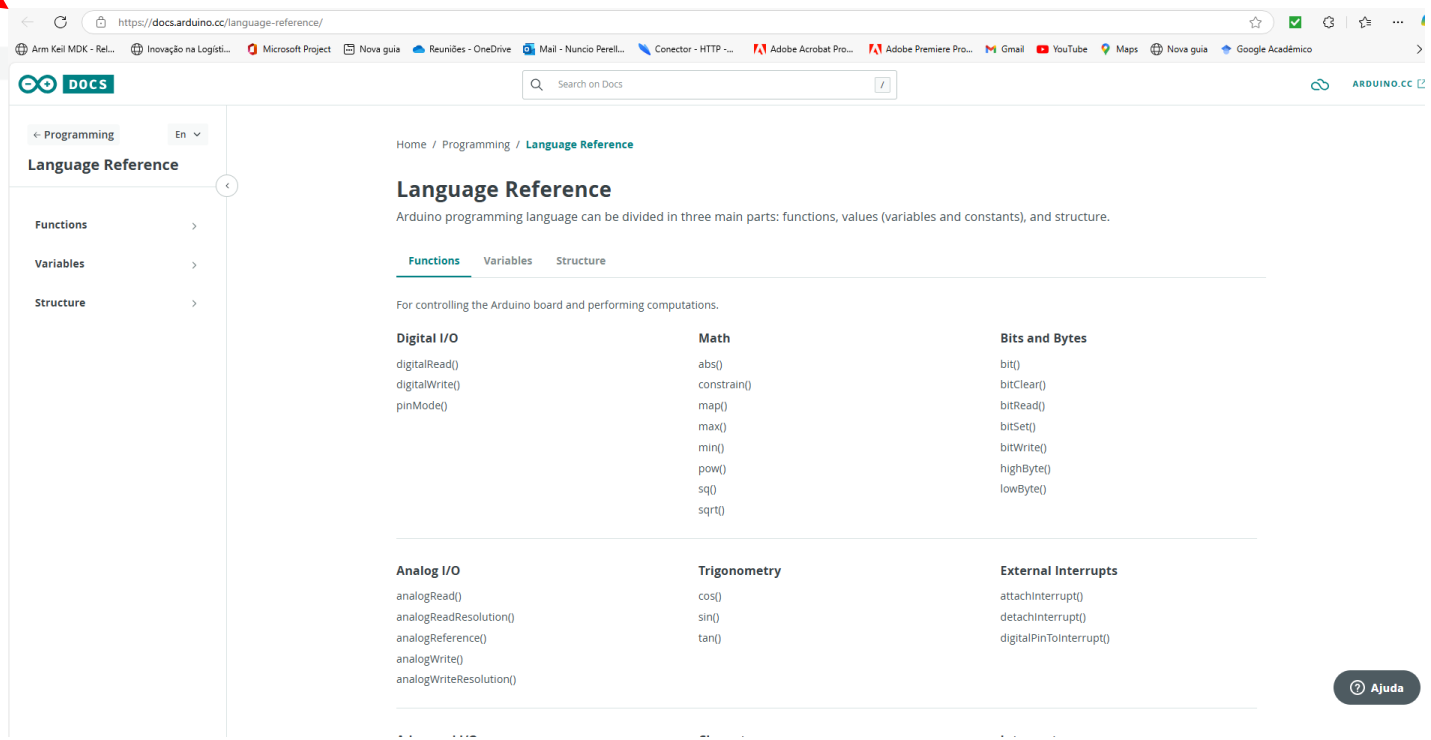


Configurando IDE – ESP32 DOIT DEVKIT V1

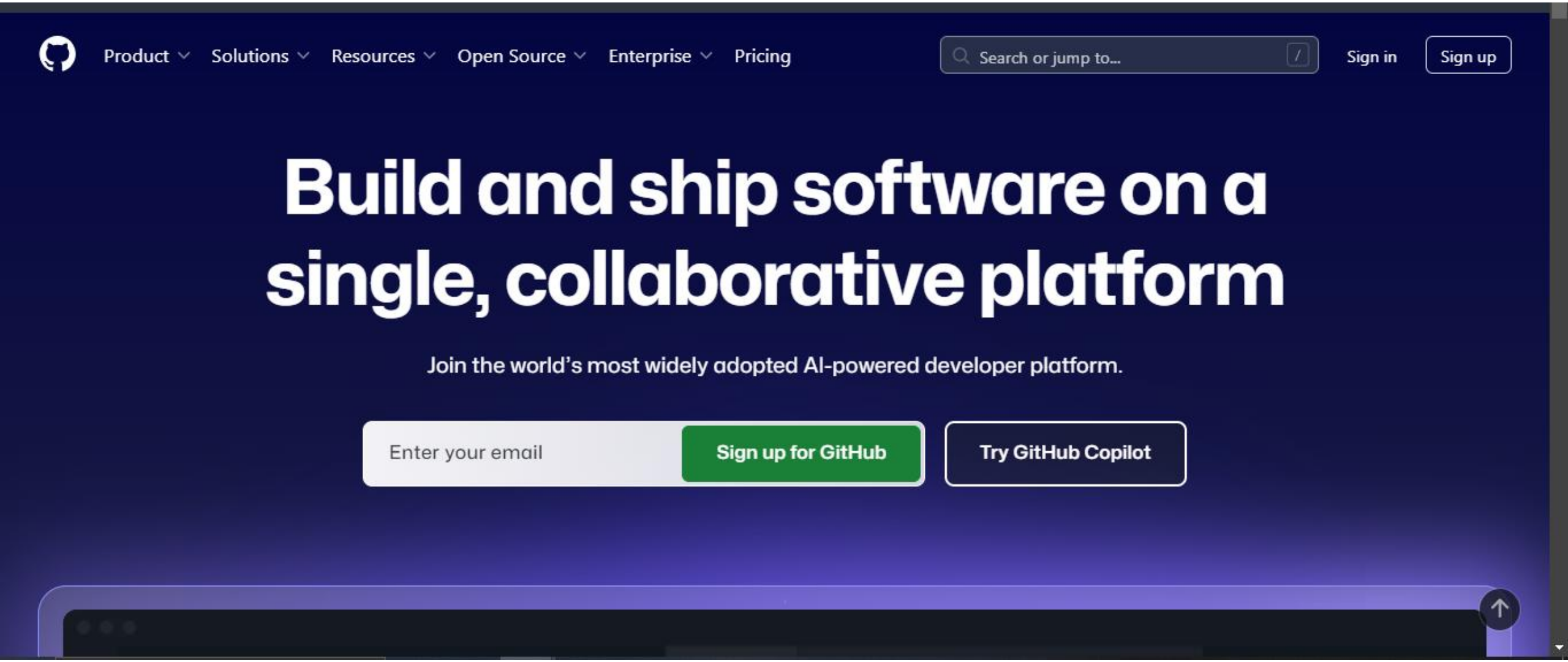


Ajuda Arduino



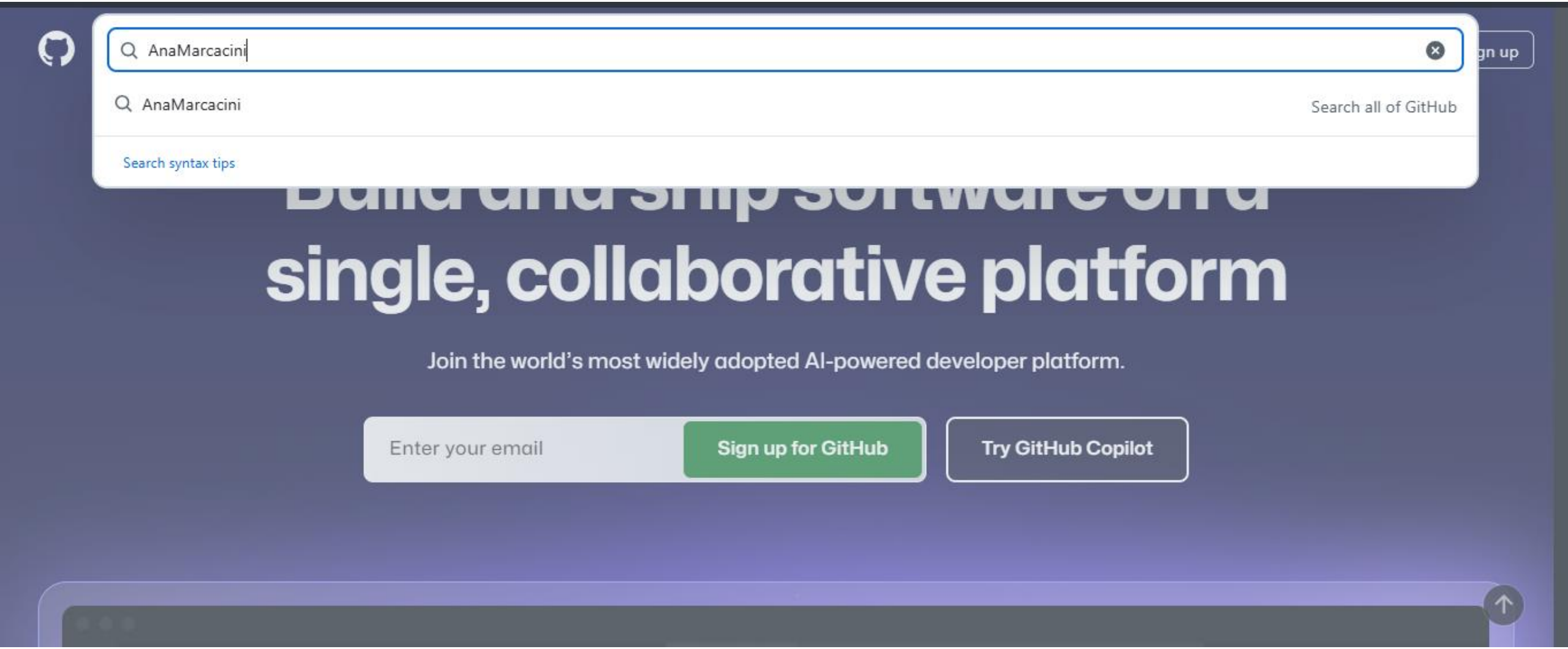


Acesso aos Códigos (Software)

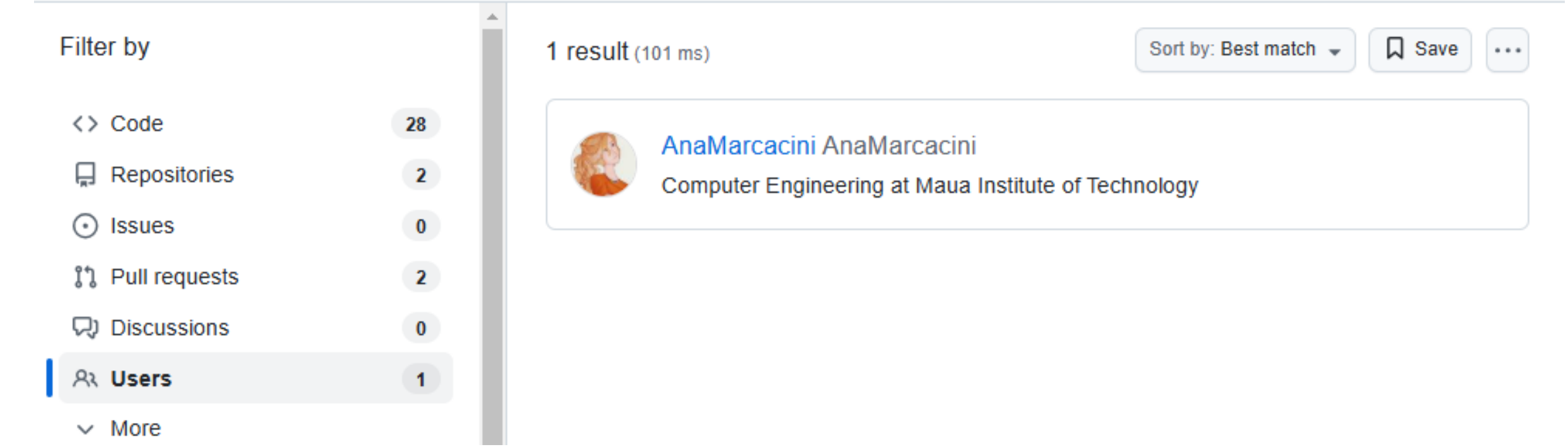


Acesso aos Códigos (Software)

Acesso aos Códigos (Software)



Acesso aos Códigos (Software)

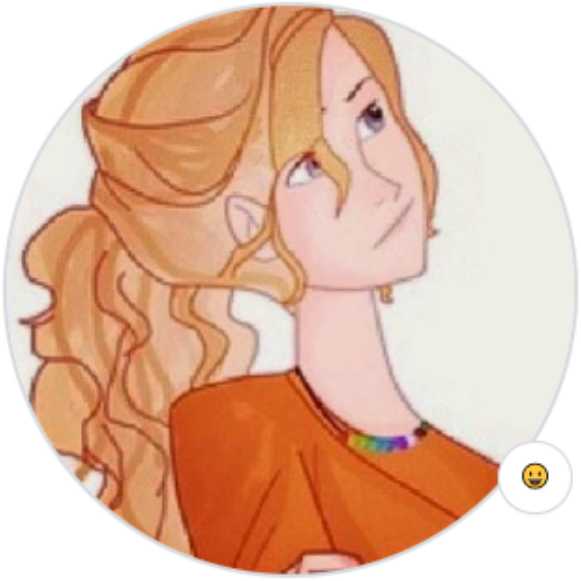


Acesso aos Códigos (Software)

AnaMarcacini

Type / to search

Overview | Repositories 90 | Projects | Packages | Stars 93



AnaMarcacini

AnaMarcacini · she/her

Computer Engineering at Maua Institute of Technology

AnaMarcacini / README.md

Hi, I'm Ana! Welcome

Computer Enginner at Instituto Mauá de Tecnologia

Former Senior Consultant Mauá Jr

Former Calculus Tutor Instituto Mauá de Tecnologia

Former Intern at Research Center at Mauá Institute of Technology

AnaHelena Follow 14

▸ *knowledge and Languages*

Acesso aos Códigos (Software)

AnaMarcacini

AnaMarcacini · she/her

Computer Engineering at Maua Institute of Technology

AnaMarcacini

Type / to search

Overview | Repositories 90 | Projects | Packages | Stars 93

Find a repository… | Type | Language | Sort | New

SensorSonoro Public — Star

C++ Updated 3 hours ago

AnaMarcacini Public — Star

Config files for my GitHub profile.

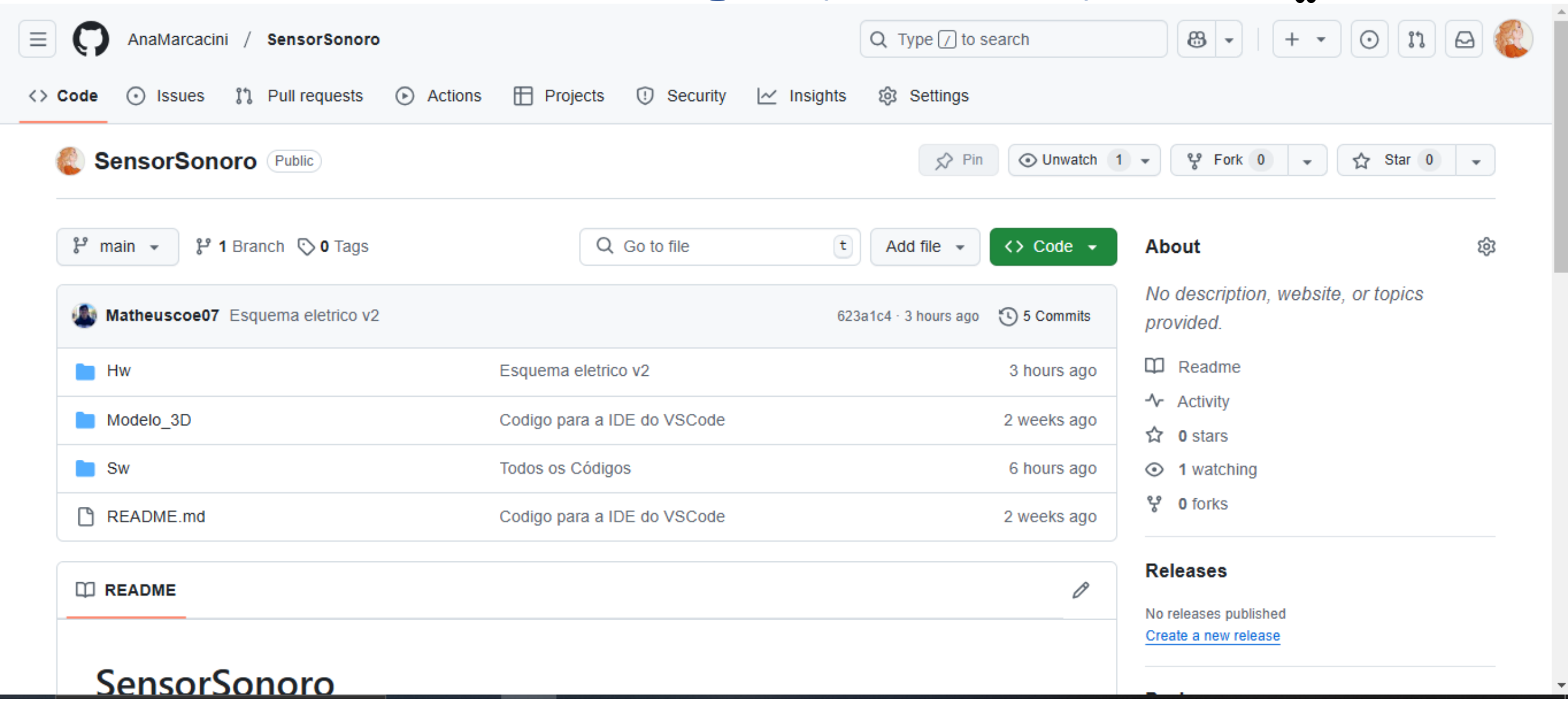
config github-config

Python Updated 16 hours ago

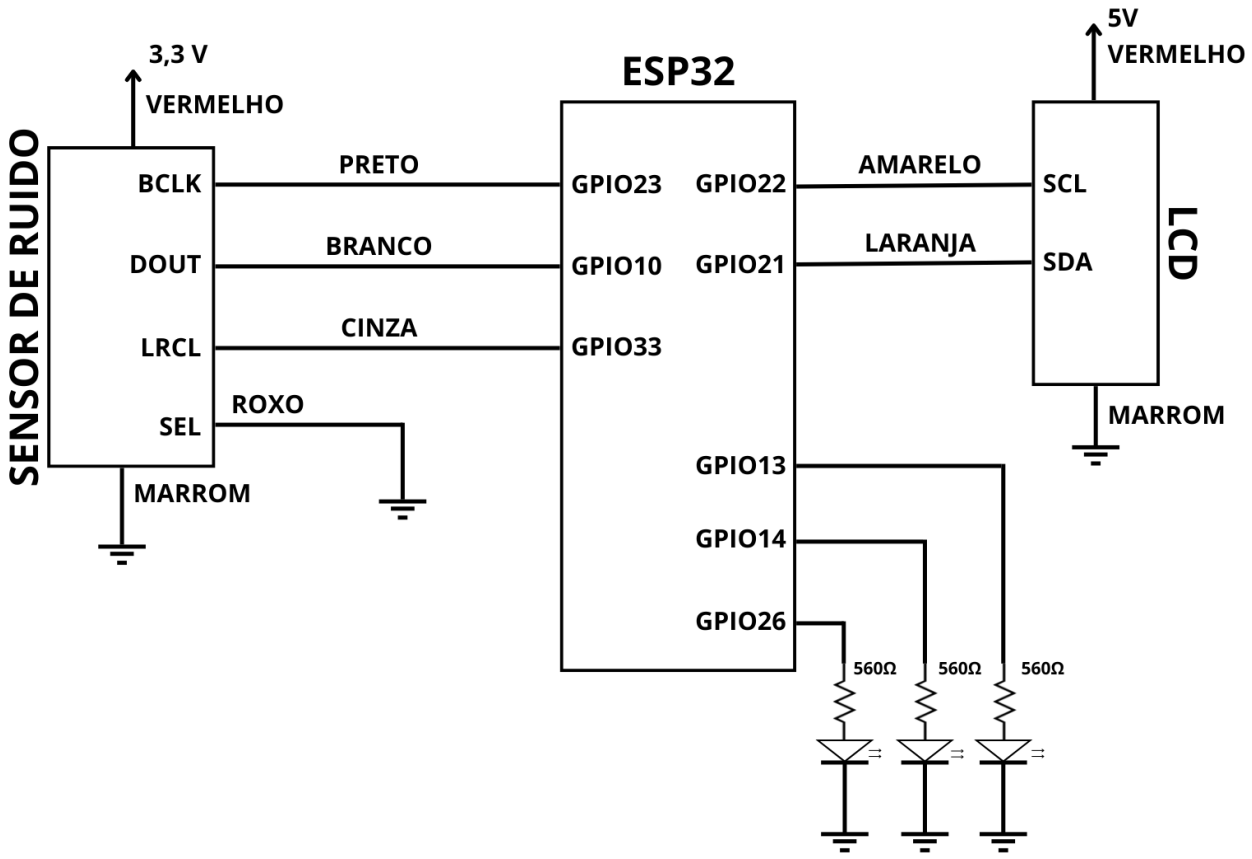
a Private — Star

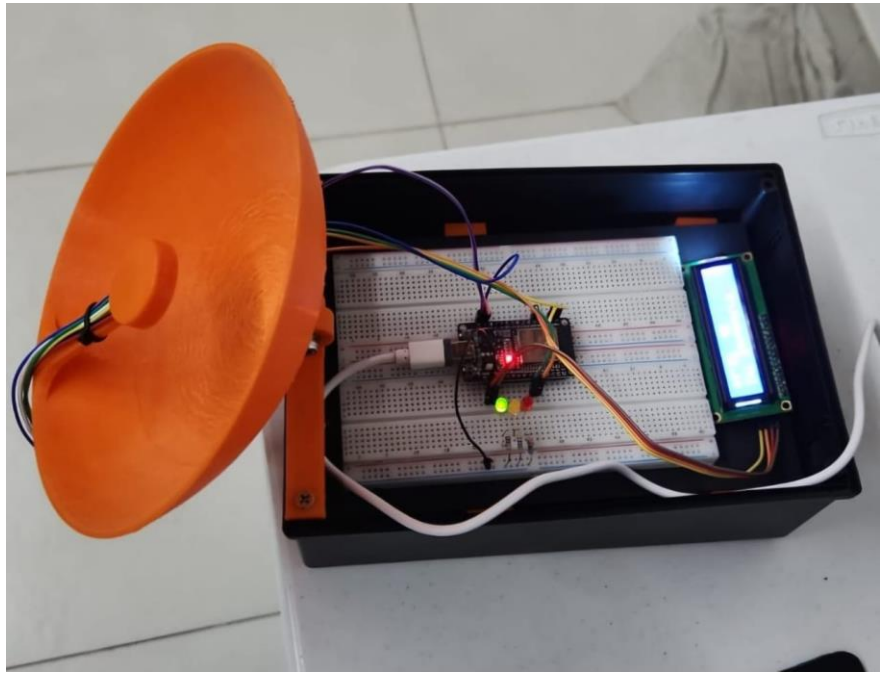
Updated last week

Acesso aos Códigos (Software)



Projeto Desenvolvido



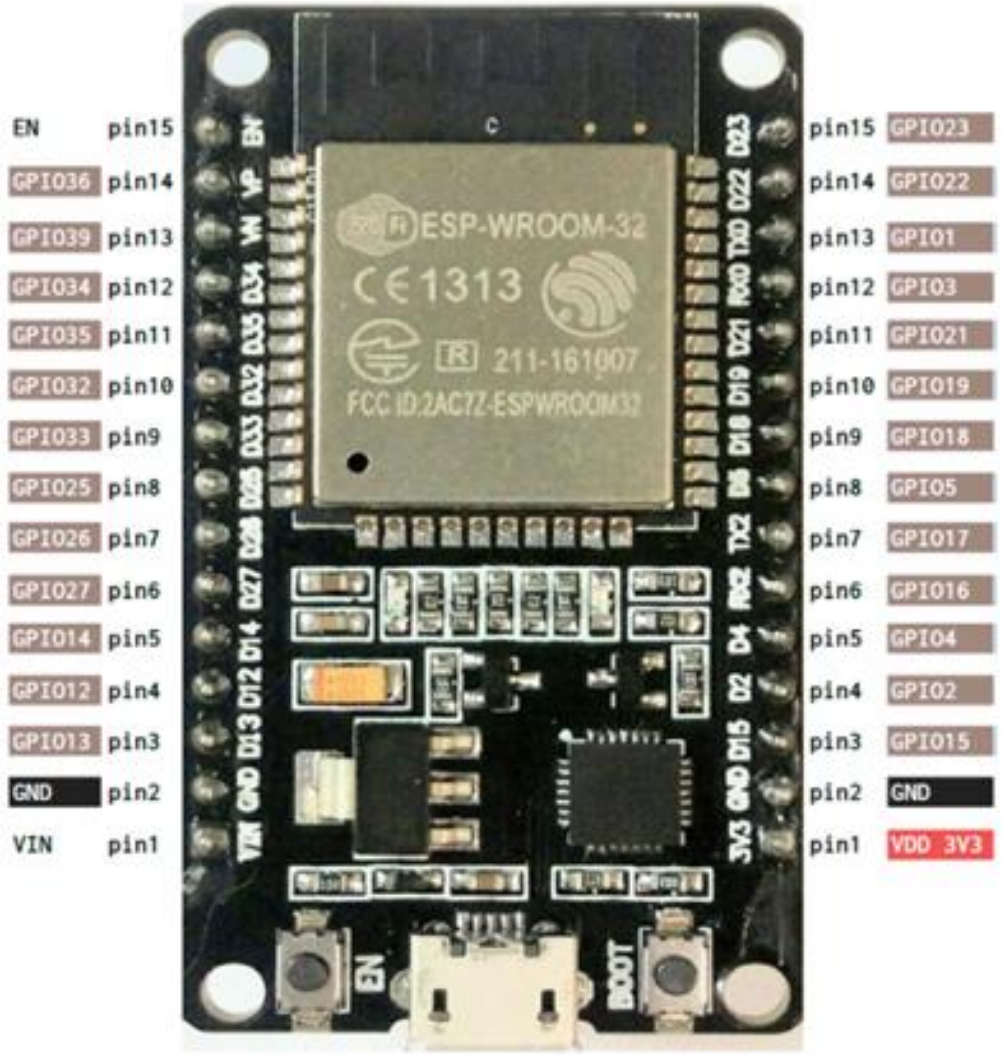


INSTITUTO MAUÁ DE TECNOLOGIA

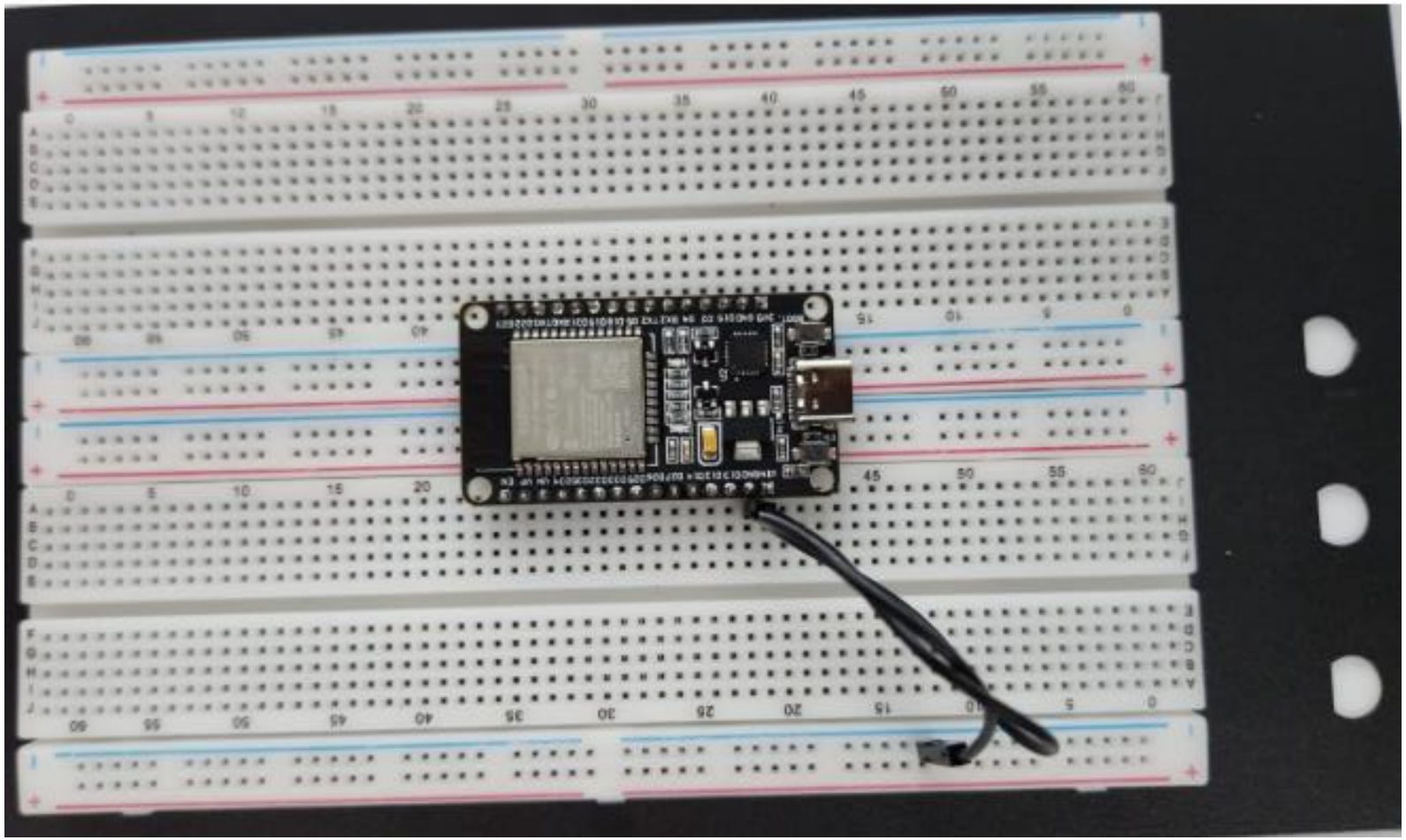
MAUÁ

LAB 1

Lab 1 – Montagem (HW)

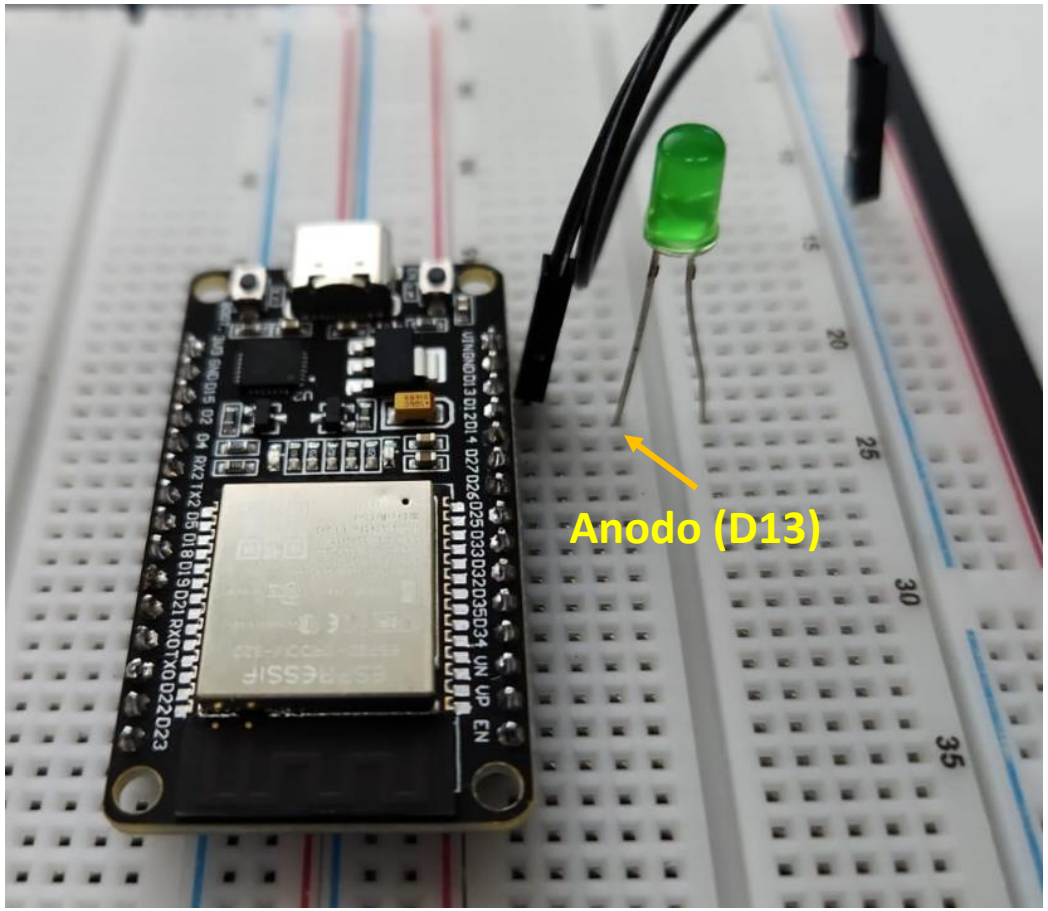


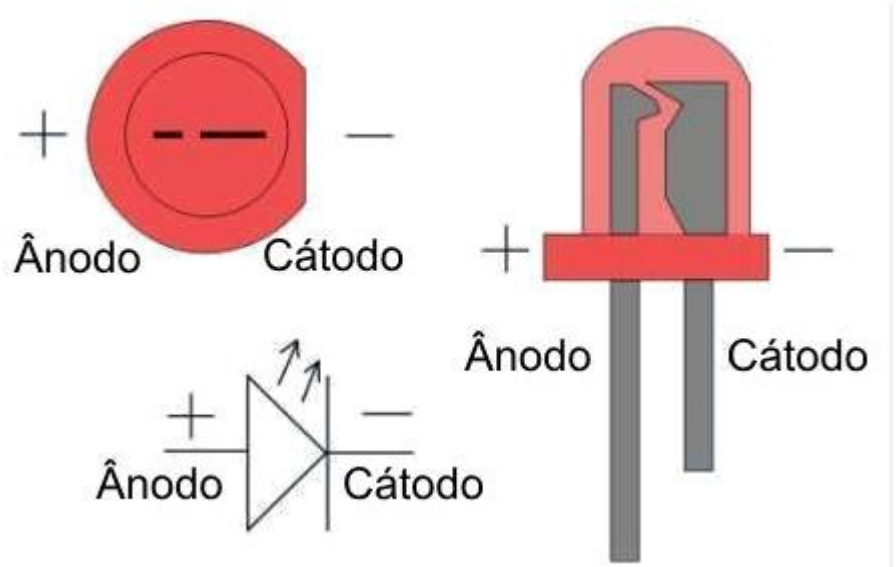
Lab 1 – Montagem (HW)



1 – Ligar Fio Preto Linha Azul – GND ESP32

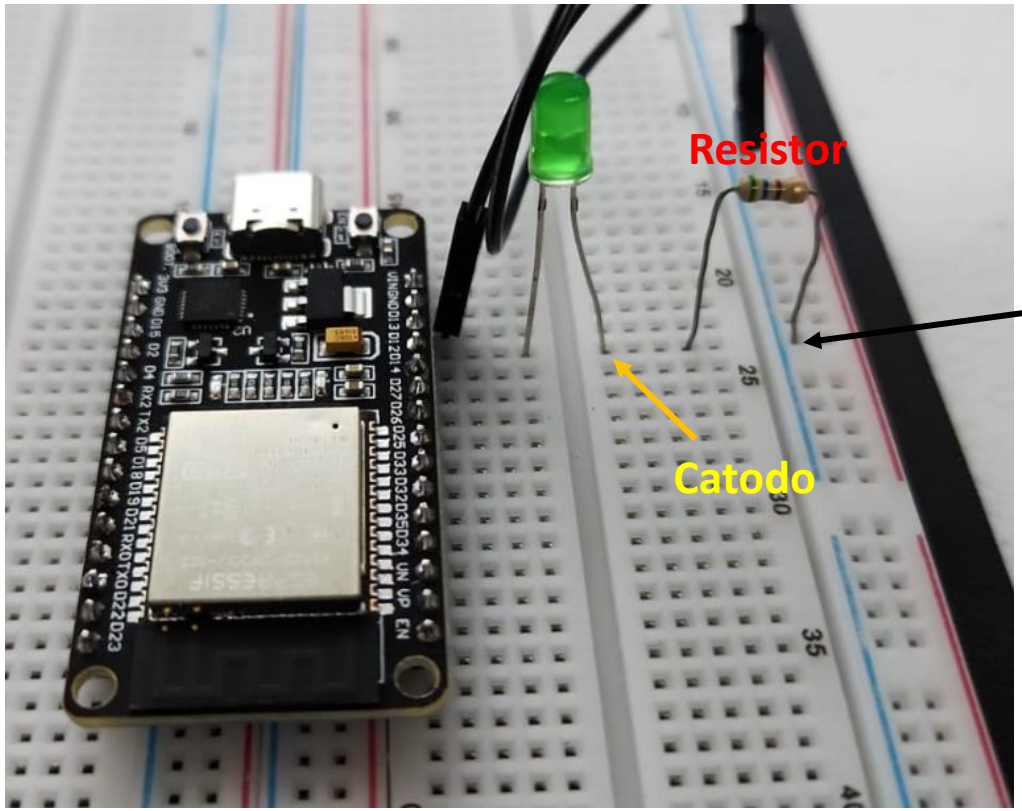
Lab 1 – Montagem (HW)





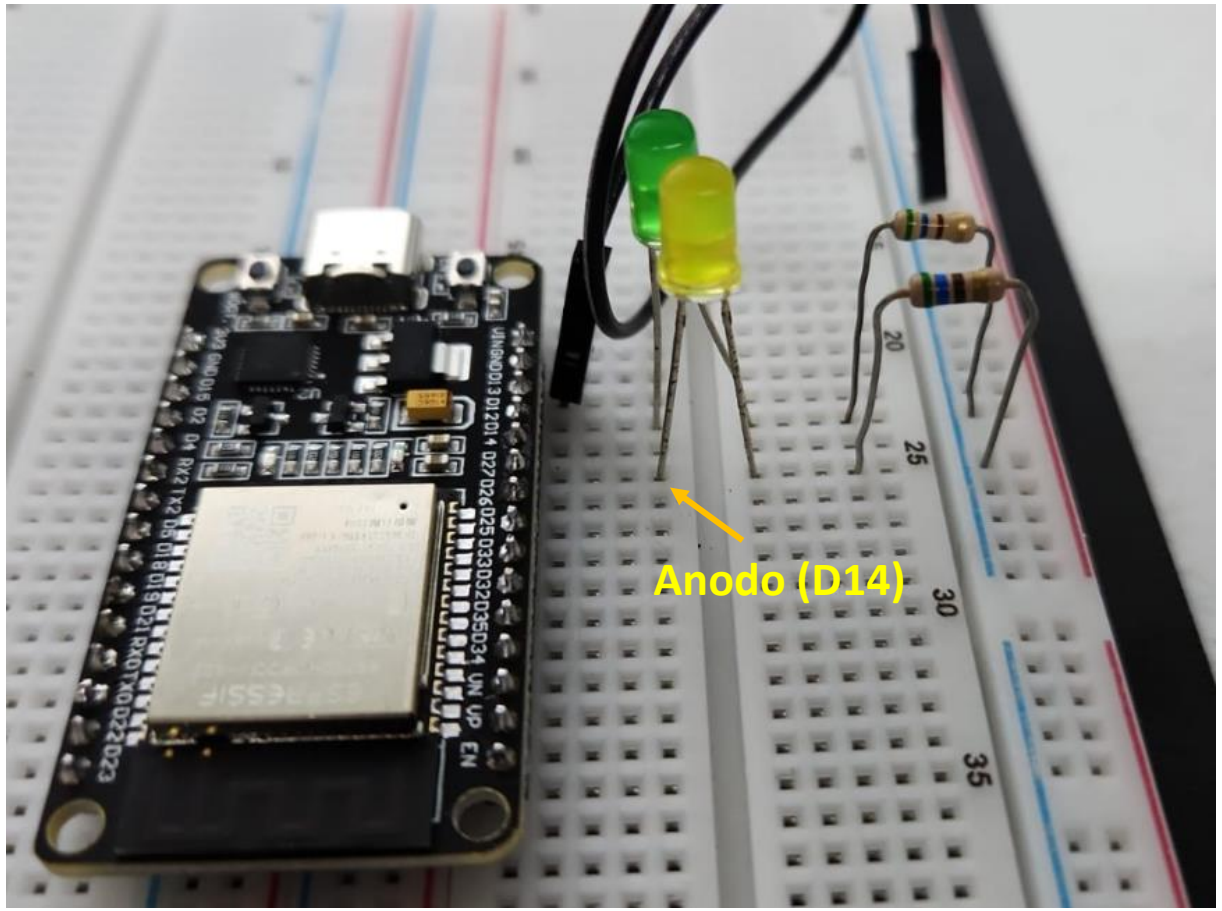
2 – Ligar Anodo do Led Verde no pino D13 ESP32

Lab 1 – Montagem (HW)



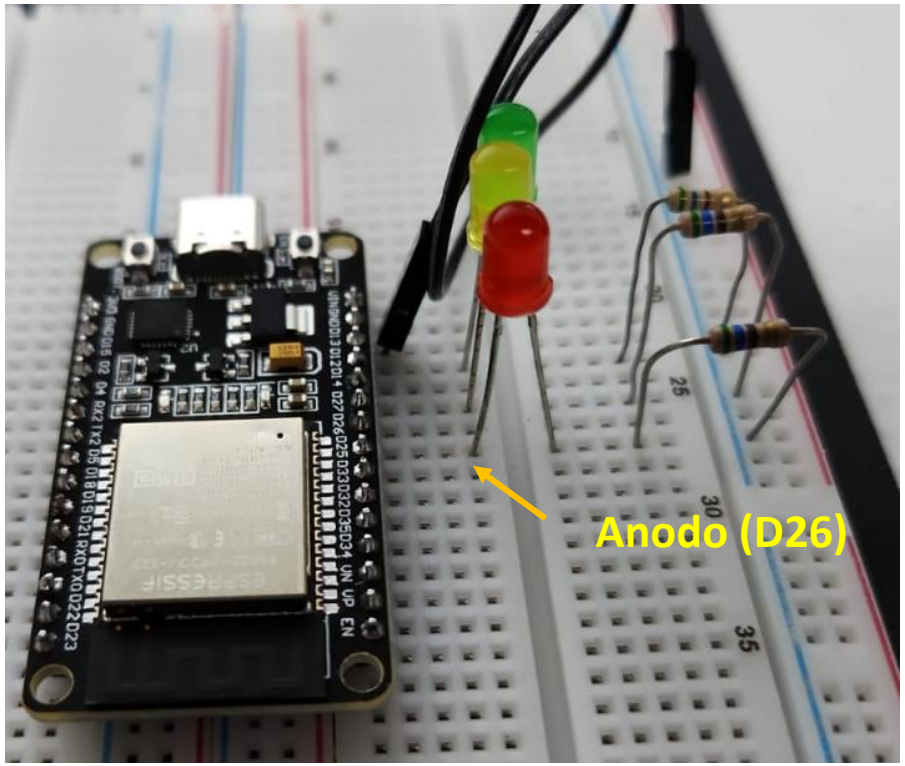
3 – Ligar terminal de resistor Catodo do do Led Verde outro terminal ligar na linha Azul

Lab 1 – Montagem (HW)



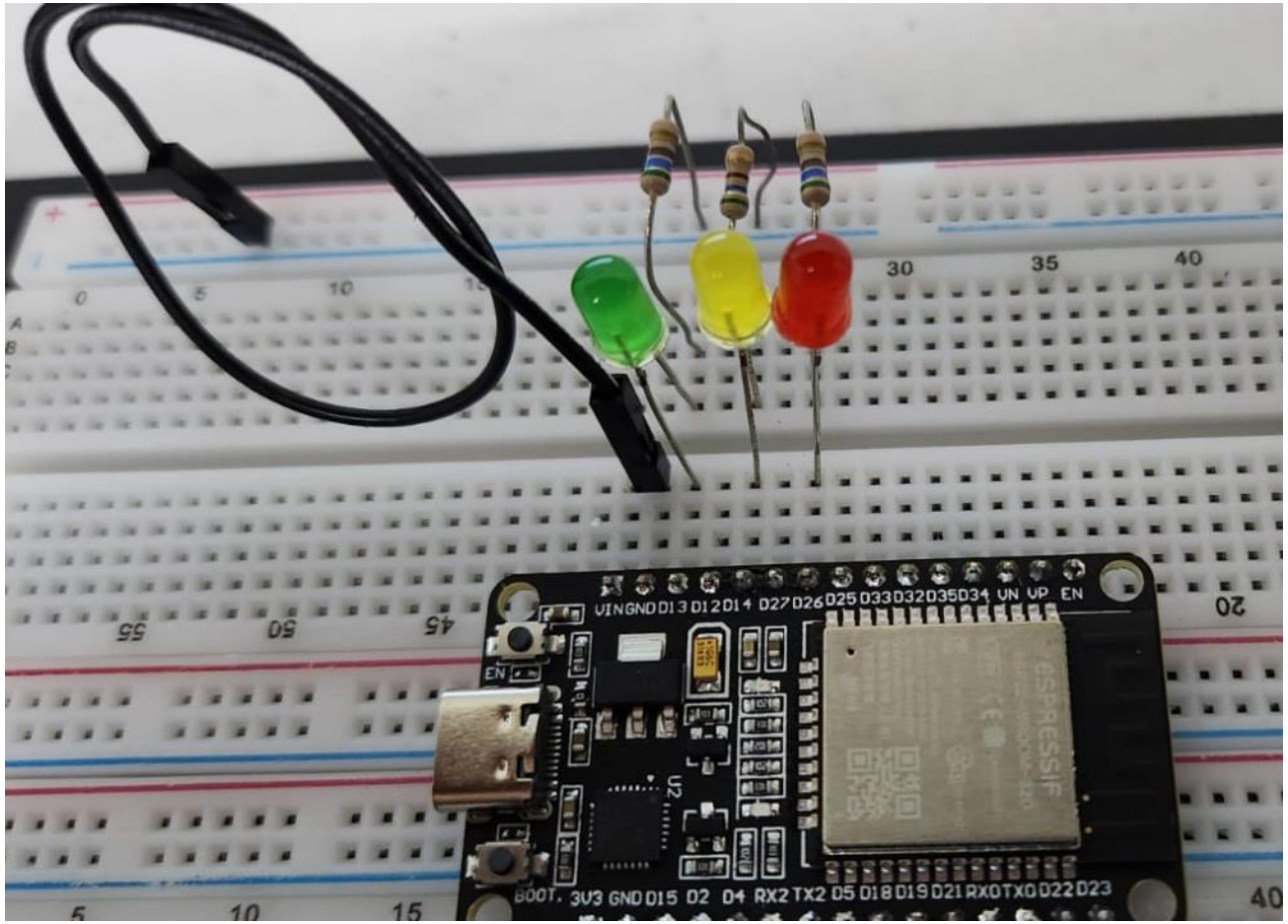
4 – Ligar Anodo do do Led Amarelo no pino D14 ESP32, ligar o resistor conforme ilustra a figura

Lab 1 – Montagem (HW)

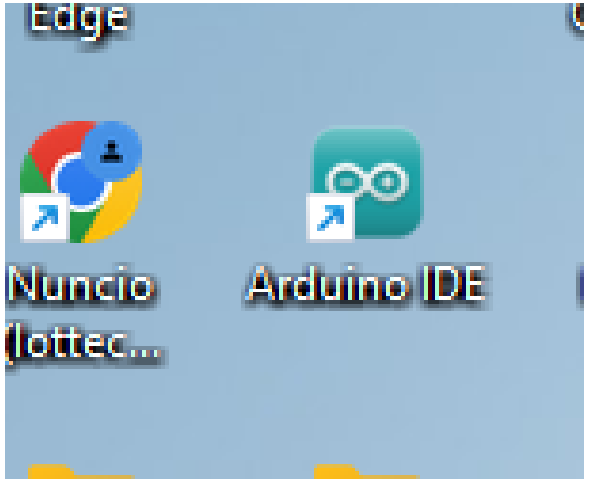


5 – Ligar Anodo do do Led Vermelho no pino D26 ESP32, ligar o resistor conforme ilustra a figura

Lab 1 – Montagem (HW)

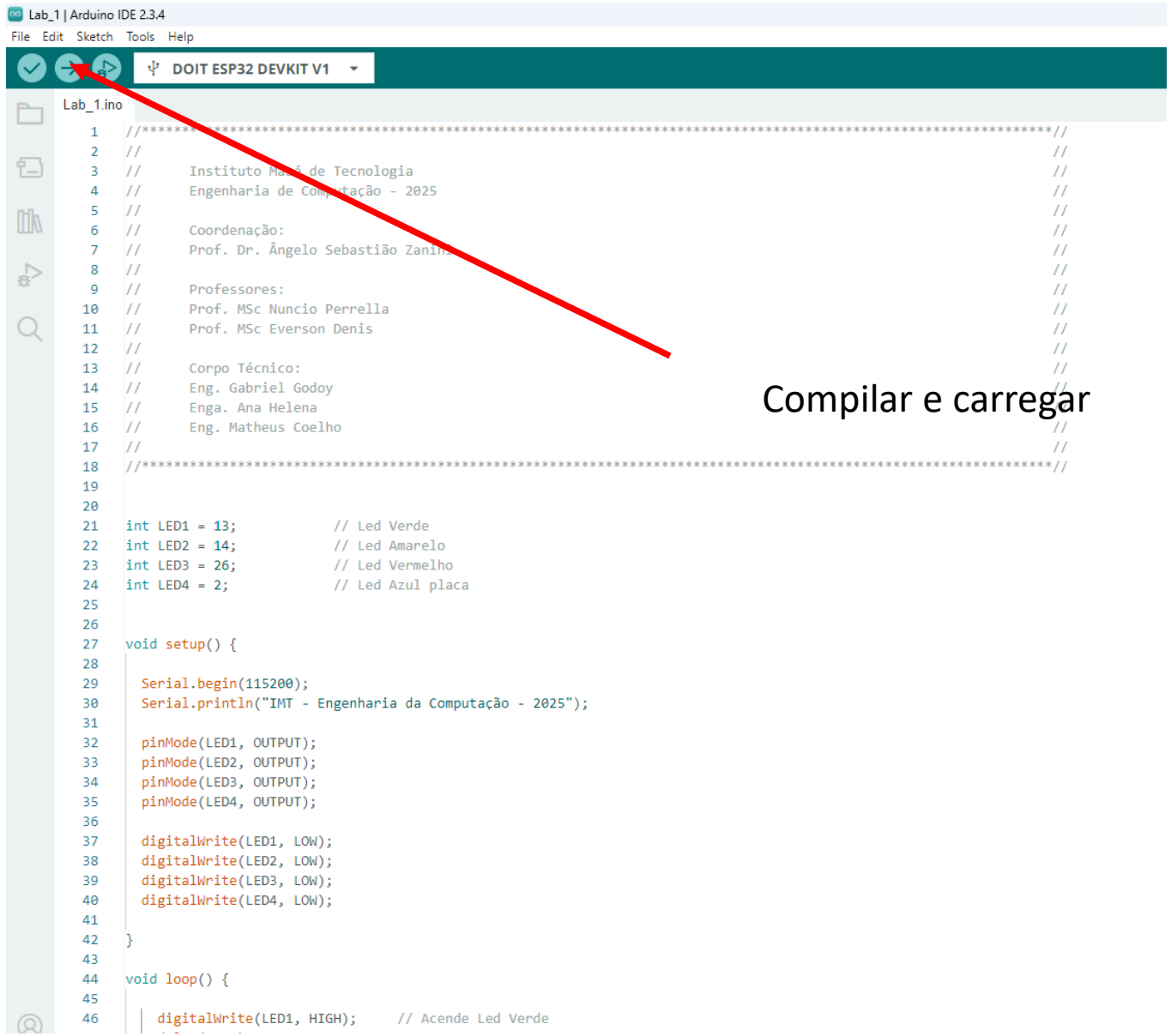


Lab 1 – SW



1- Abrir o IDE Arduino
2- Carregar Sketch: Lab_1.ino

Lab 1 – SW




```
//*************************************************************************************************************//
//                                                                                                             //
//      Instituto Mauá de Tecnologia                                                                           //
//      Engenharia de Computação - 2025                                                                        //
//                                                                                                             //
//      Coordenação:                                                                                           //
//      Prof. Dr. Ângelo Sebastião Zanini                                                                      //
//                                                                                                             //
//      Professores:                                                                                           //
//      Prof. MSc Nuncio Perrella                                                                              //
//      Prof. MSc Everson Denis                                                                                //
//                                                                                                             //
//      Corpo Técnico:                                                                                         //
//      Eng. Gabriel Godoy                                                                                     //
//      Enga. Ana Helena                                                                                       //
//      Eng. Matheus Coelho                                                                                    //
//                                                                                                             //
//*************************************************************************************************************//


int LED1 = 13;            // Led Verde
int LED2 = 14;            // Led Amarelo
int LED3 = 26;            // Led Vermelho
int LED4 = 2;             // Led Azul placa


void setup() {

  Serial.begin(115200);
  Serial.println("IMT - Engenharia da Computação - 2025");

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);

  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);

}

void loop() {

    digitalWrite(LED1, HIGH);     // Acende Led Verde
```

Lab 1 – SW

Lab_1 | Arduino IDE 2.3.4

File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

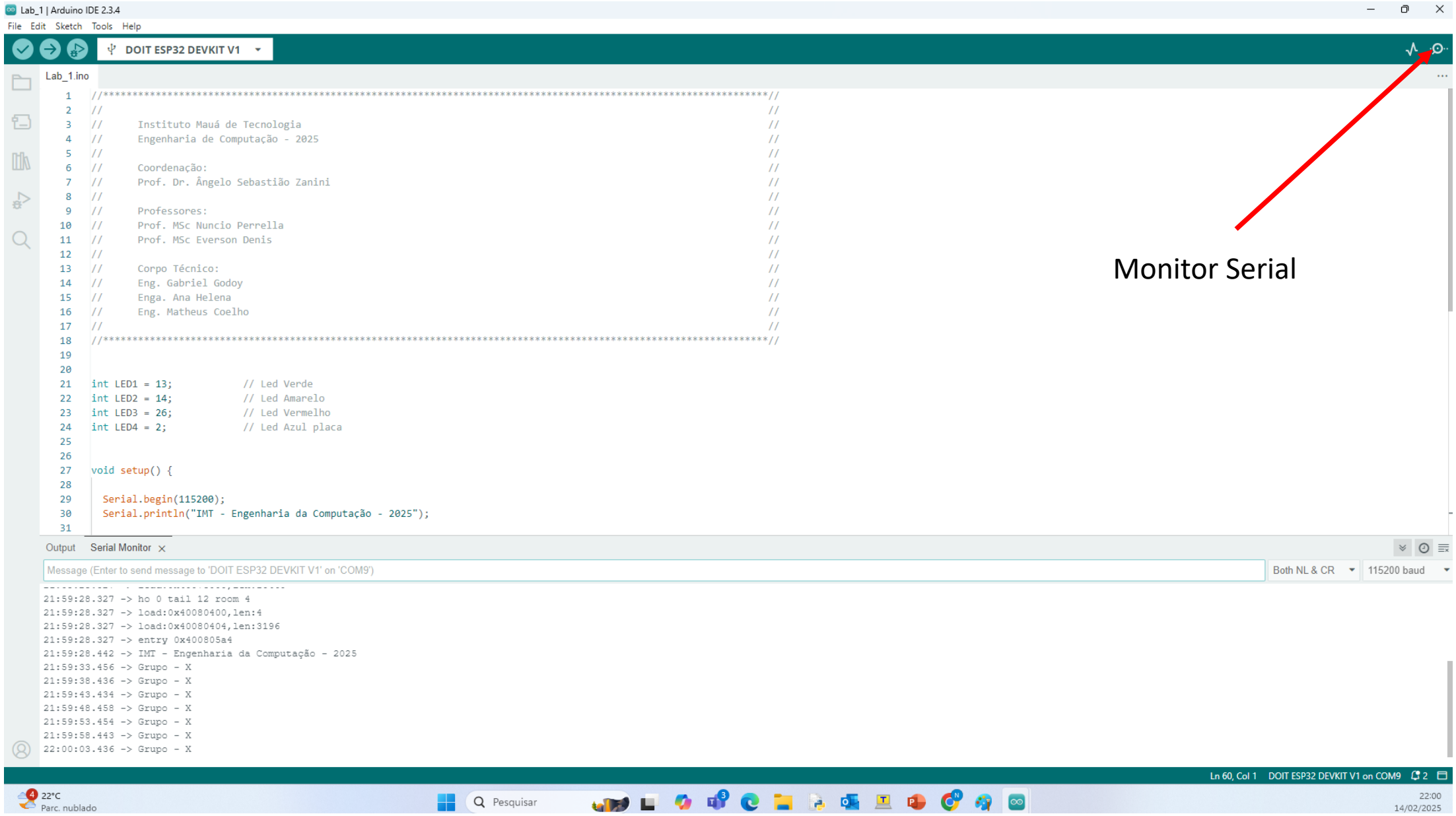
Lab_1.ino

```
//**************************************************************************************************************************//
//                                                                                                                           //
//      Instituto Mauá de Tecnologia                                                                                         //
//      Engenharia de Computação - 2025                                                                                      //
//                                                                                                                           //
//      Coordenação:                                                                                                         //
//      Prof. Dr. Ângelo Sebastião Zanini                                                                                    //
//                                                                                                                           //
//      Professores:                                                                                                         //
//      Prof. MSc Nuncio Perrella                                                                                            //
//      Prof. MSc Everson Denis                                                                                              //
//                                                                                                                           //
//      Corpo Técnico:                                                                                                       //
//      Eng. Gabriel Godoy                                                                                                   //
//      Enga. Ana Helena                                                                                                     //
//      Eng. Matheus Coelho                                                                                                  //
//                                                                                                                           //
//**************************************************************************************************************************//


int LED1 = 13;            // Led Verde
int LED2 = 14;            // Led Amarelo
int LED3 = 26;            // Led Vermelho
int LED4 = 2;             // Led Azul placa


void setup() {

  Serial.begin(115200);
  Serial.println("IMT - Engenharia da Computação - 2025");

```

Output

```
Writing at 0x00028bd8... (30 %)
Writing at 0x0002e214... (40 %)
Writing at 0x00033f5f... (50 %)
Writing at 0x0003942c... (60 %)
Writing at 0x0003e98f... (70 %)
Writing at 0x00044066... (80 %)
Writing at 0x0004b06c... (90 %)
Writing at 0x00054cd6... (100 %)
Wrote 300896 bytes (160347 compressed) at 0x00010000 in 3.0 seconds (effective 790.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Lab 1 – SW



Monitor Serial

Lab 1 – Conclusões

Listar as observações e conclusões do laboratório

Lab 1 – Conclusões

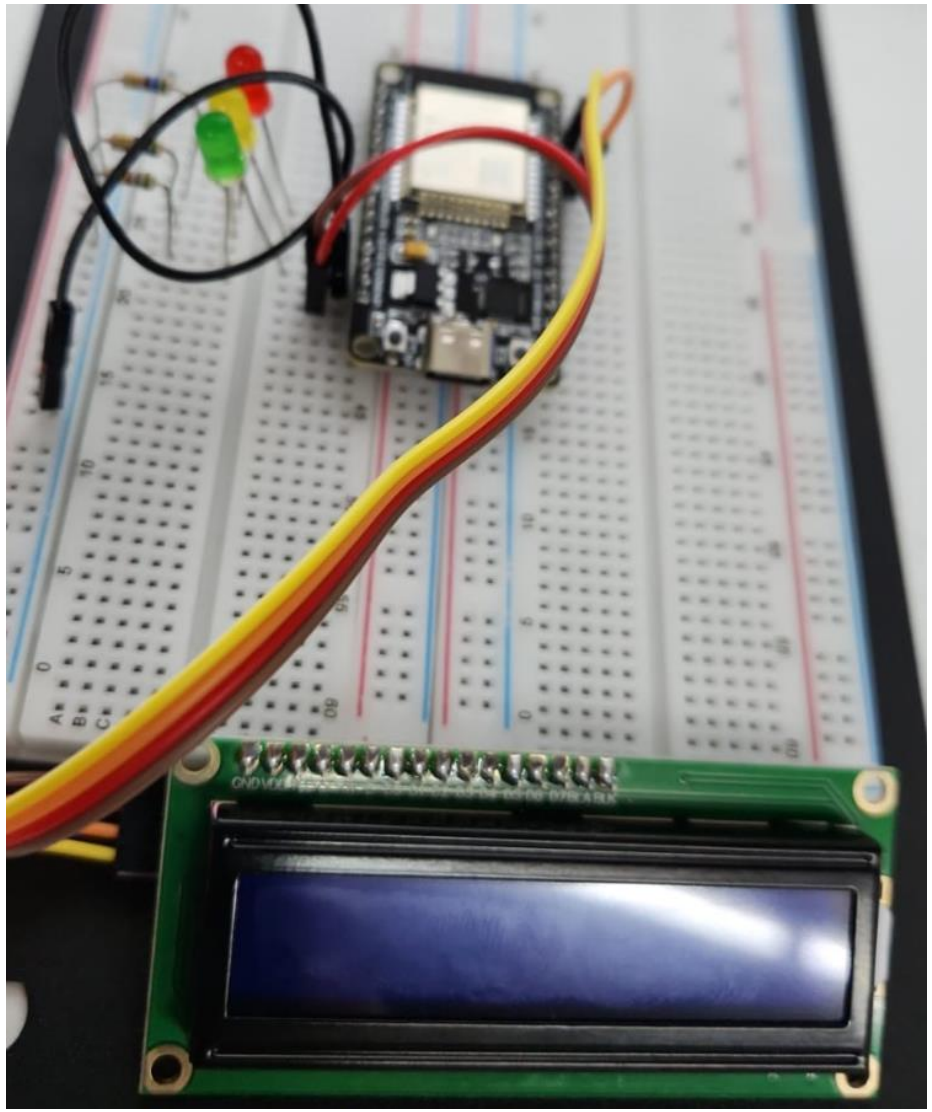
Parabéns !!! Se você consegue acender um LED você Pode fazer qualquer coisa em sistemas embarcados!!!

INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

LAB 2

Lab 2 – Montagem (HW)



Ligar os fios dos display

Vermelho → Vin (5V)
Marrom → GND
Laranja → D21 (ESP32)
Amarelo → D22 (ESP32)

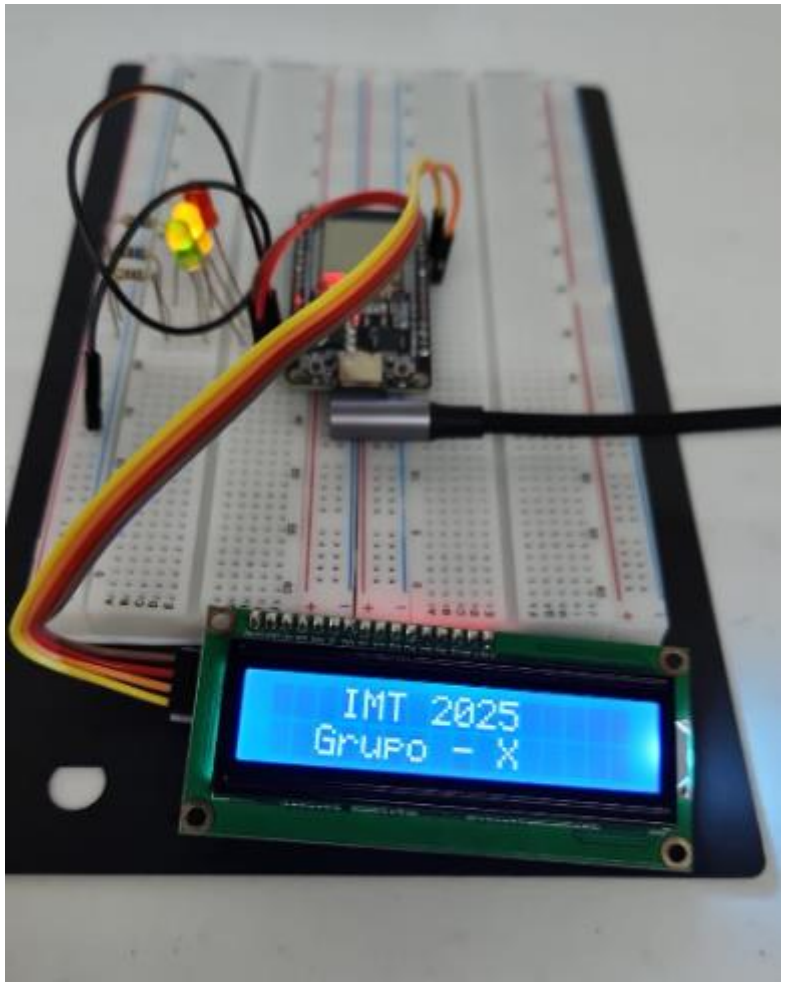
Lab 2 – SW

1- Abrir o IDE Arduino
2- Carregar Sketch: Lab_2.ino
3- Compilar e carregar

Lab 2 – Conclusões

Listar as observações e conclusões do laboratório

Lab 2 – Conclusões

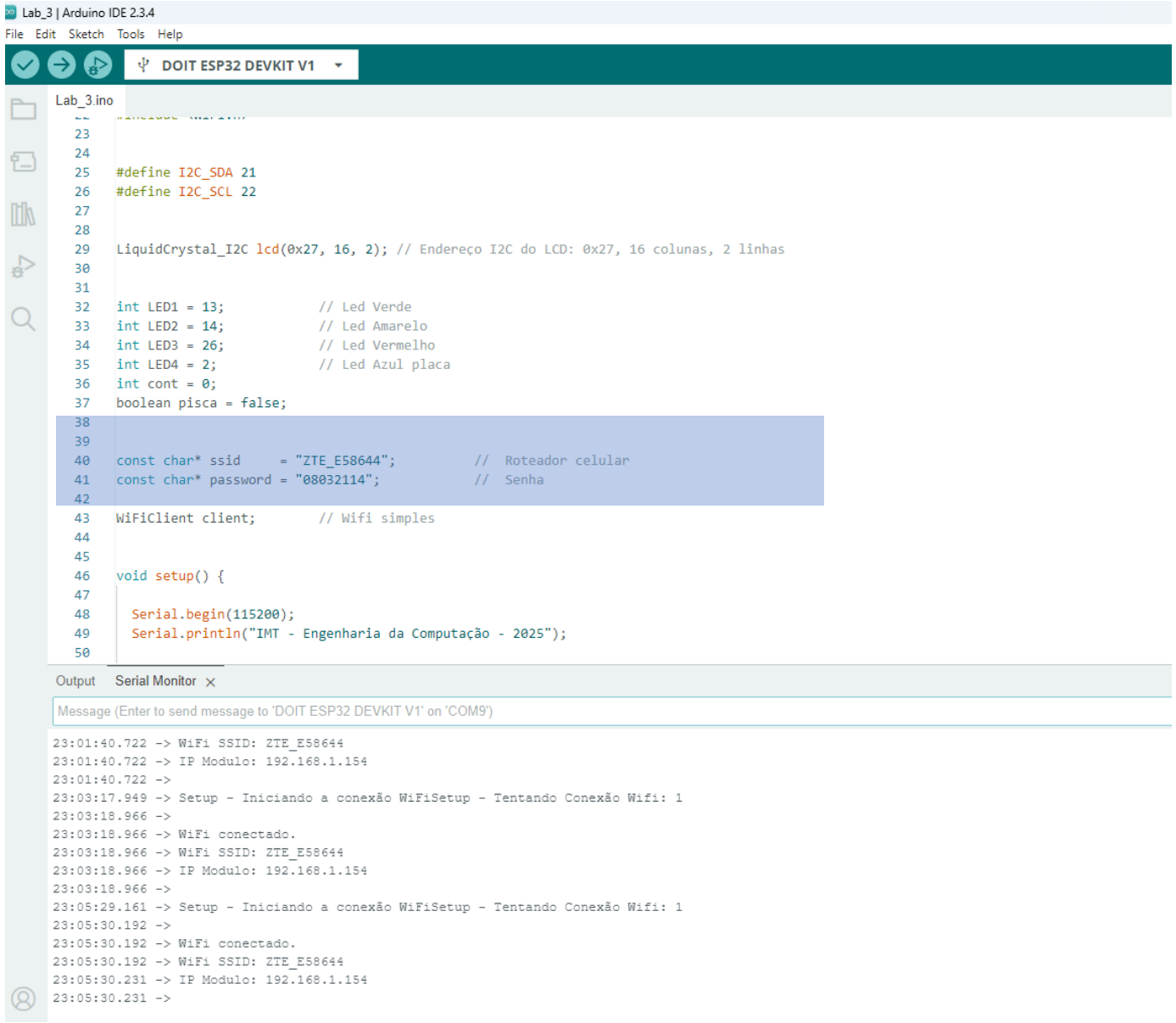


INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

LAB 3

Lab 3 – SW

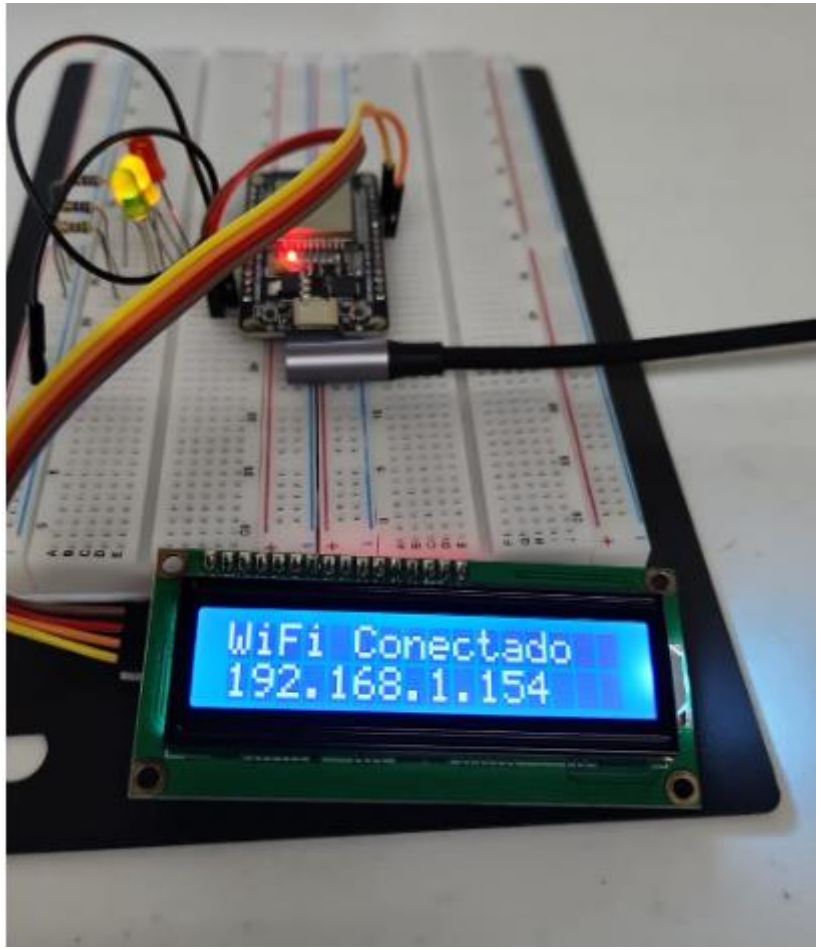


1 - Abrir o IDE Arduino
2 - Carregar Sketch: Lab_3.ino
3 - Colocar celular modo Roteador
4 - Linha 40 carregar SSID do roteador do seu celular
5 – Linha 41 carregar a senha do seu roteador
6- Compilar e carregar

Lab 3 – Conclusões

Listar as observações e conclusões do laboratório

Lab 3 – Conclusões

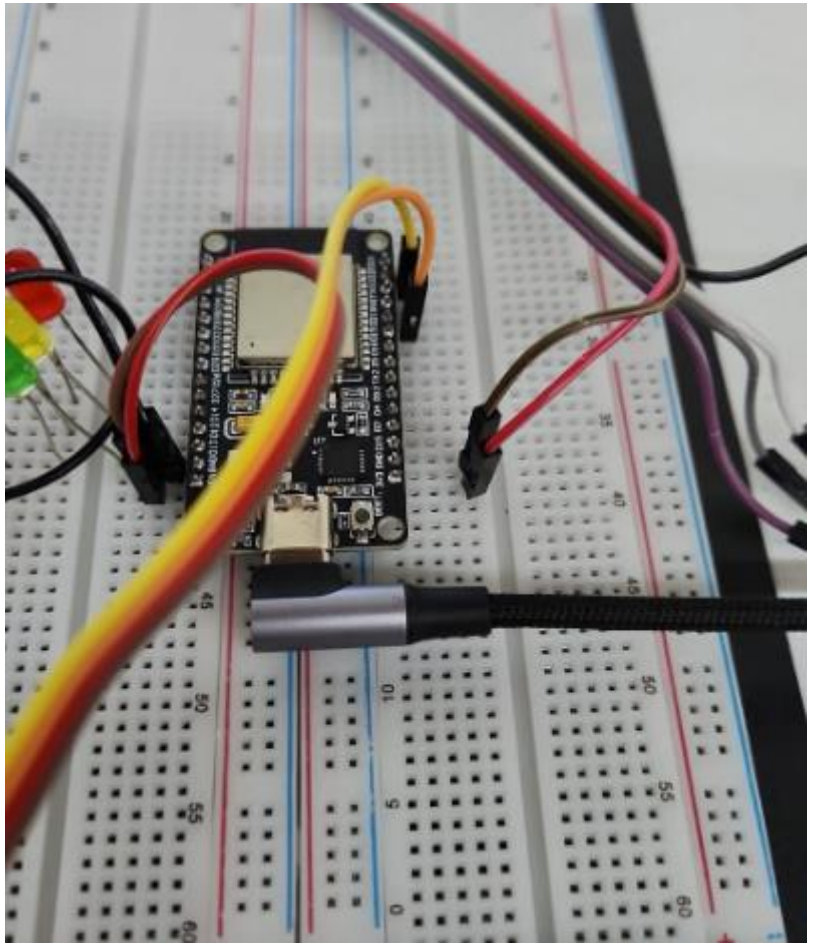


INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

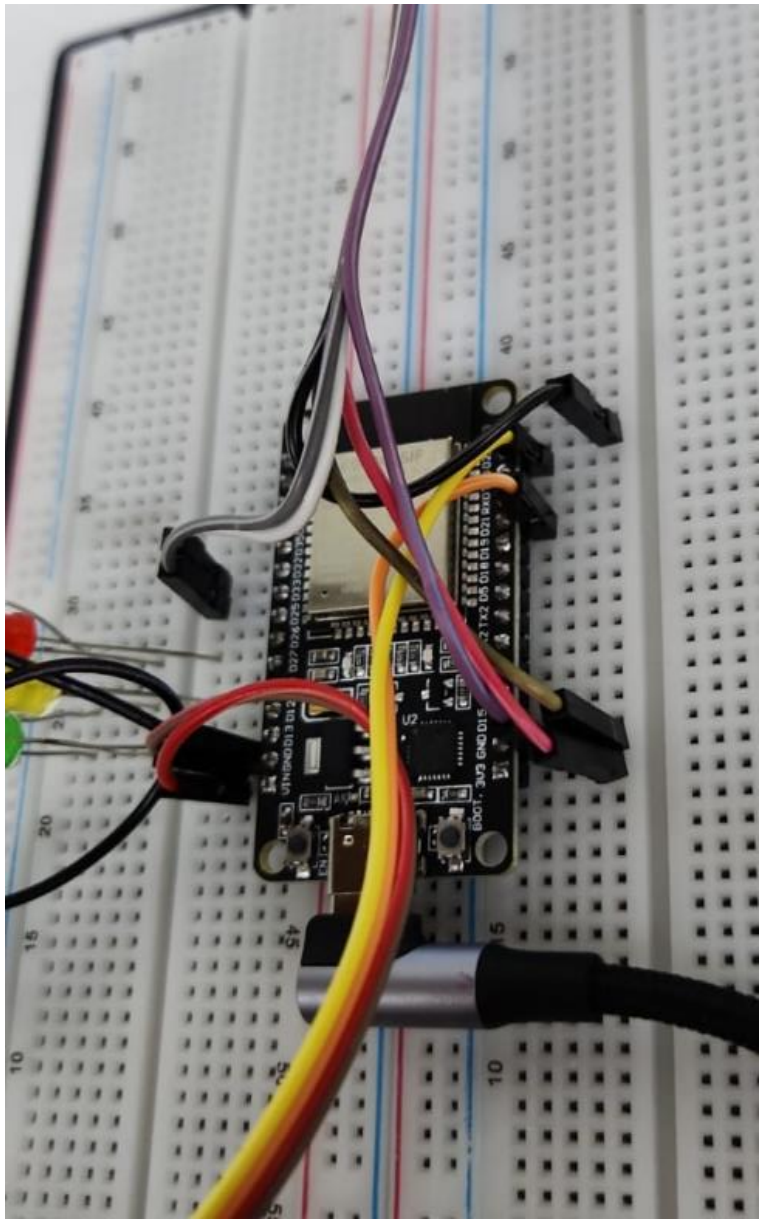
LAB 4

Lab 4 – Montagem (HW)



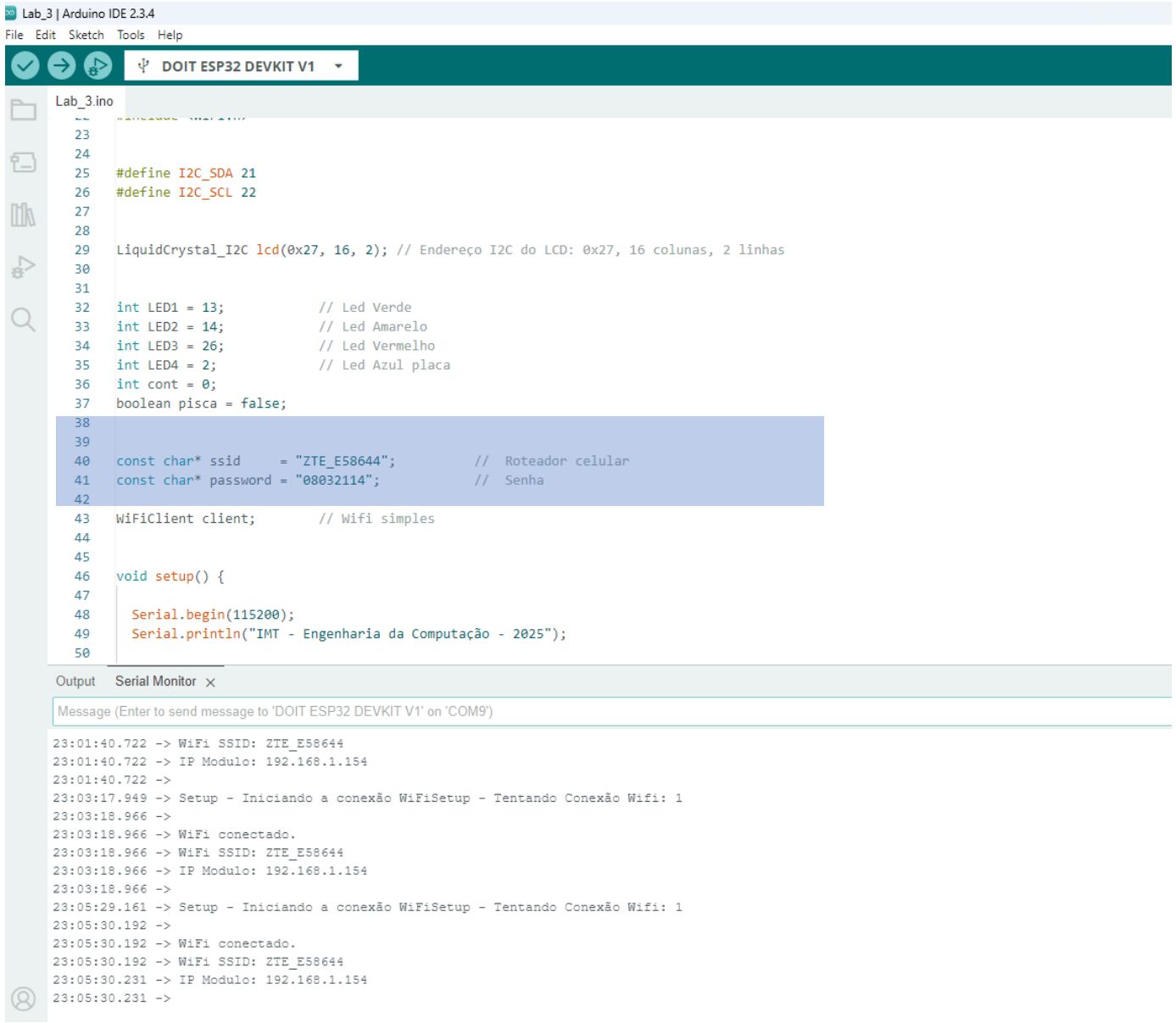
1 – Ligar Fio Vermelho do Sensor – 3V3 ESP32
2 – Ligar Fio Marrom do Sensor – GND ESP32

Lab 4 – Montagem (HW)



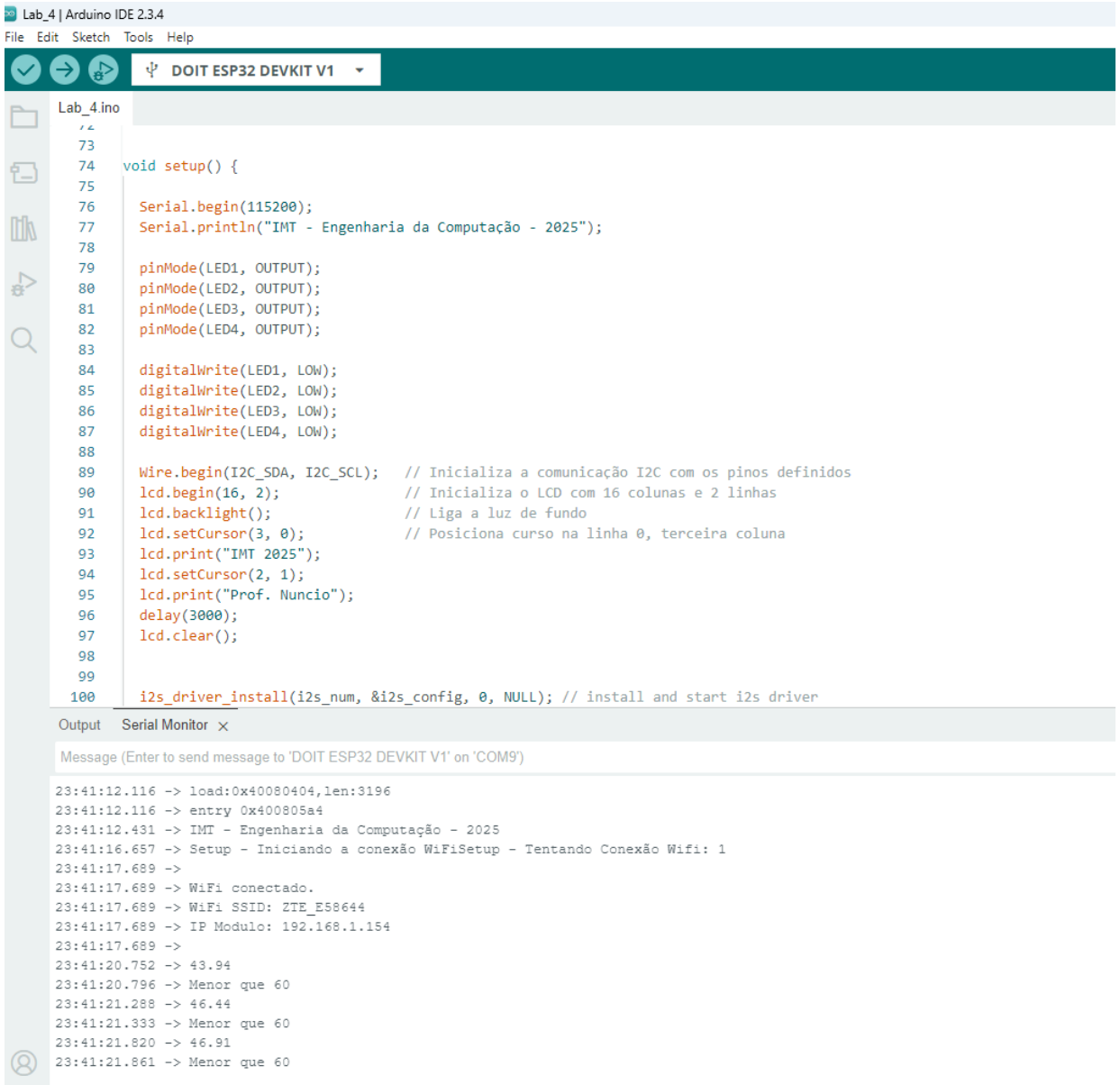
1 – Ligar Fio Preto do Sensor – D23 ESP32
2 – Ligar Fio Branco do Sensor – D32 ESP32
3 – Ligar Fio Cinza do Sensor – D33 ESP32
4 – Ligar Fio Roxo do Sensor – GND ESP32

Lab 4 – SW



1 - Abrir o IDE Arduino
2 - Carregar Sketch: Lab_4.ino
3 - Colocar celular modo Roteador
4 - Linha 40 carregar SSID do roteador do seu celular
5 – Linha 41 carregar a senha do seu roteador
6- Compilar e carregar

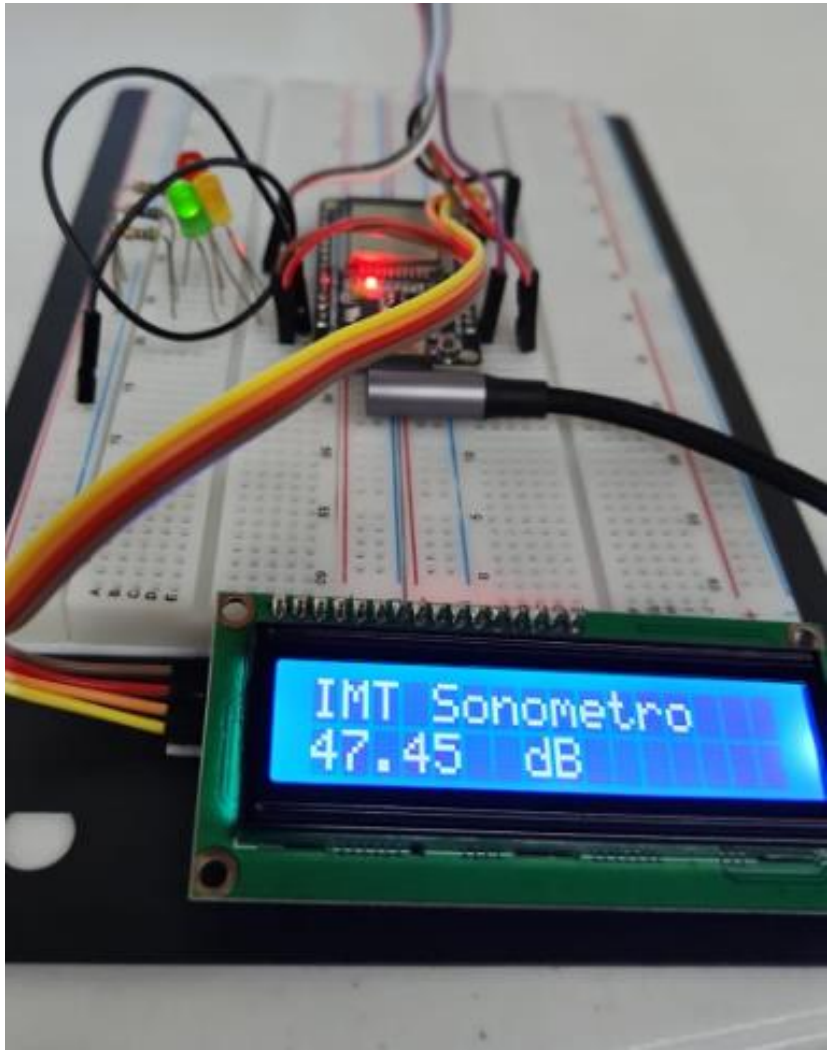
Lab 4 – SW


Lab_4 | Arduino IDE 2.3.4
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
Lab_4.ino
73
74 void setup() {
75
76 Serial.begin(115200);
77 Serial.println("IMT - Engenharia da Computação - 2025");
78
79 pinMode(LED1, OUTPUT);
80 pinMode(LED2, OUTPUT);
81 pinMode(LED3, OUTPUT);
82 pinMode(LED4, OUTPUT);
83
84 digitalWrite(LED1, LOW);
85 digitalWrite(LED2, LOW);
86 digitalWrite(LED3, LOW);
87 digitalWrite(LED4, LOW);
88
89 Wire.begin(I2C_SDA, I2C_SCL); // Inicializa a comunicação I2C com os pinos definidos
90 lcd.begin(16, 2); // Inicializa o LCD com 16 colunas e 2 linhas
91 lcd.backlight(); // Liga a luz de fundo
92 lcd.setCursor(3, 0); // Posiciona curso na linha 0, terceira coluna
93 lcd.print("IMT 2025");
94 lcd.setCursor(2, 1);
95 lcd.print("Prof. Nuncio");
96 delay(3000);
97 lcd.clear();
98
99
100 i2s_driver_install(i2s_num, &i2s_config, 0, NULL); // install and start i2s driver
Output Serial Monitor
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM9')
23:41:12.116 -> load:0x40080404,len:3196
23:41:12.116 -> entry 0x400805a4
23:41:12.431 -> IMT - Engenharia da Computação - 2025
23:41:16.657 -> Setup - Iniciando a conexão WiFiSetup - Tentando Conexão Wifi: 1
23:41:17.689 ->
23:41:17.689 -> WiFi conectado.
23:41:17.689 -> WiFi SSID: ZTE_E58644
23:41:17.689 -> IP Modulo: 192.168.1.154
23:41:17.689 ->
23:41:20.752 -> 43.94
23:41:20.796 -> Menor que 60
23:41:21.288 -> 46.44
23:41:21.333 -> Menor que 60
23:41:21.820 -> 46.91
23:41:21.861 -> Menor que 60

Lab 4 – Conclusões

Listar as observações e conclusões do laboratório

Lab 4 – Conclusões



INSTITUTO MAUÁ DE TECNOLOGIA

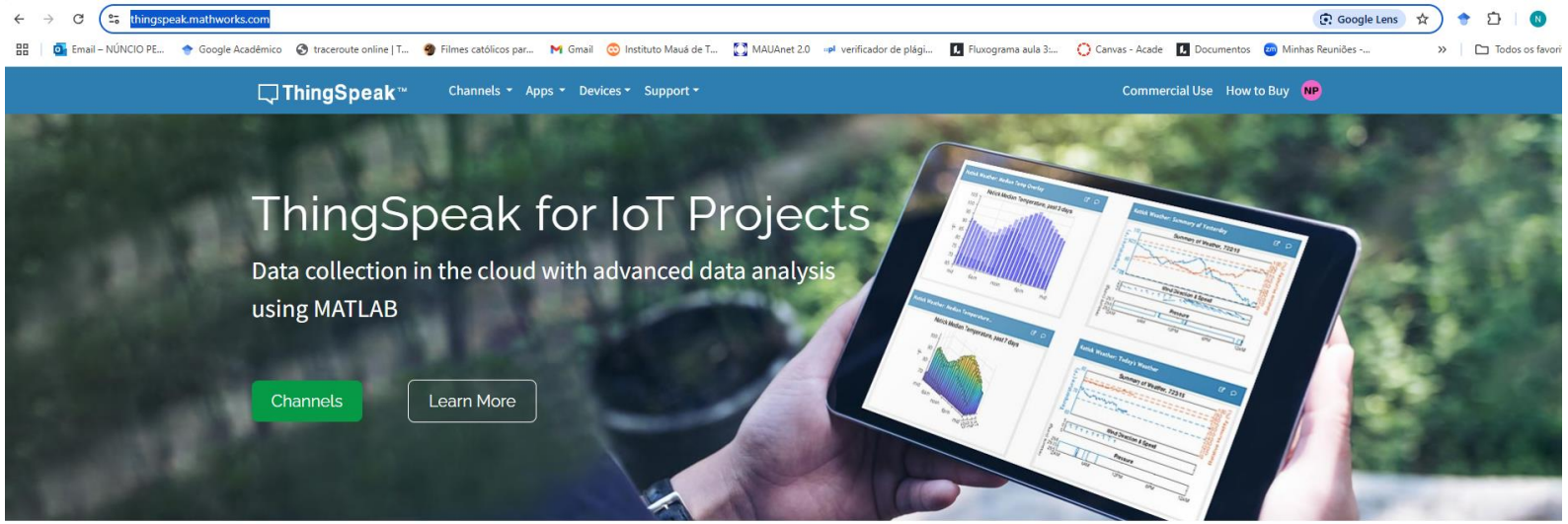
MAUÁ

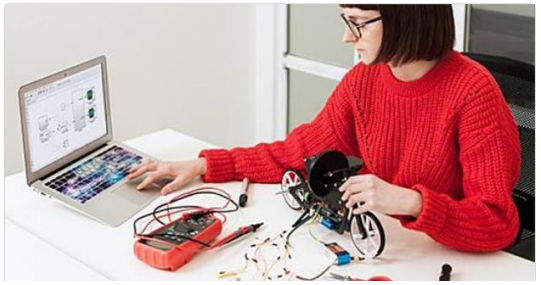
LAB 5

ThingSpeak

Acessar: https://thingspeak.mathworks.com/

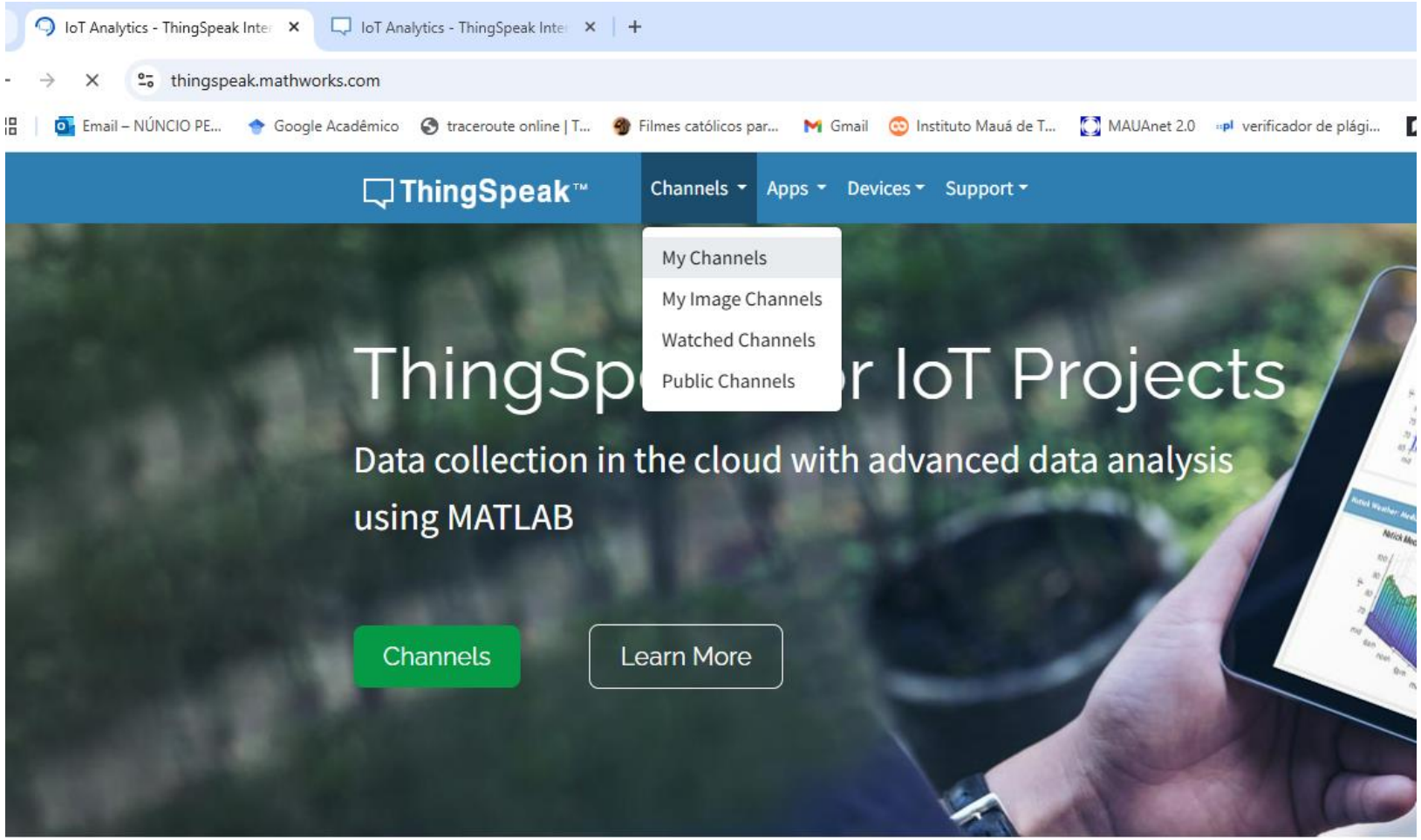
Criar conta



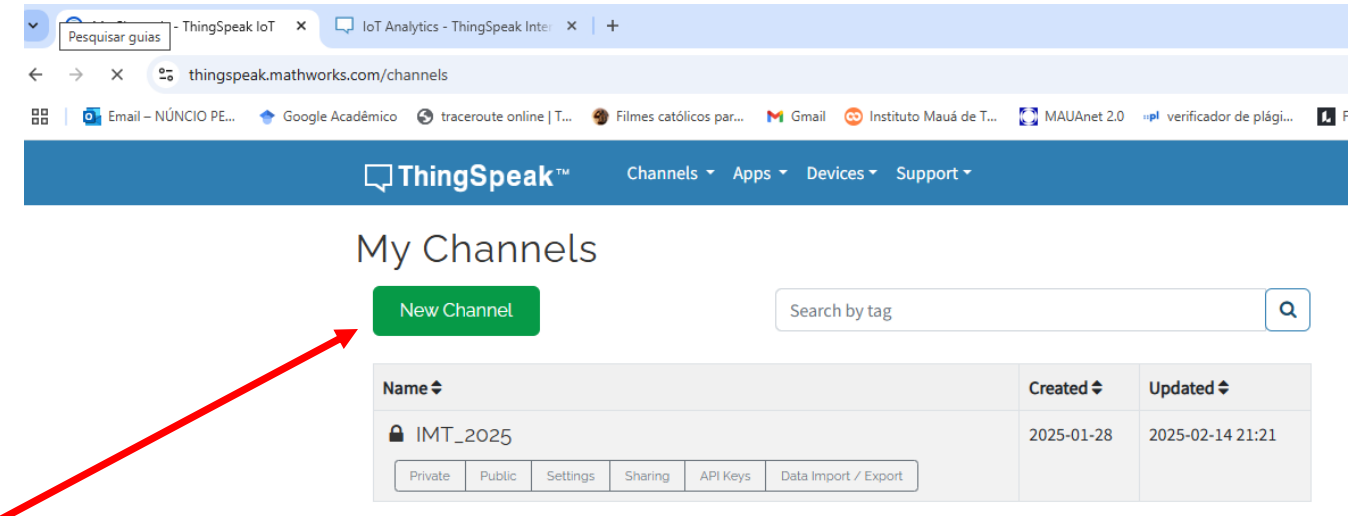


ThingSpeak

→ My Channels

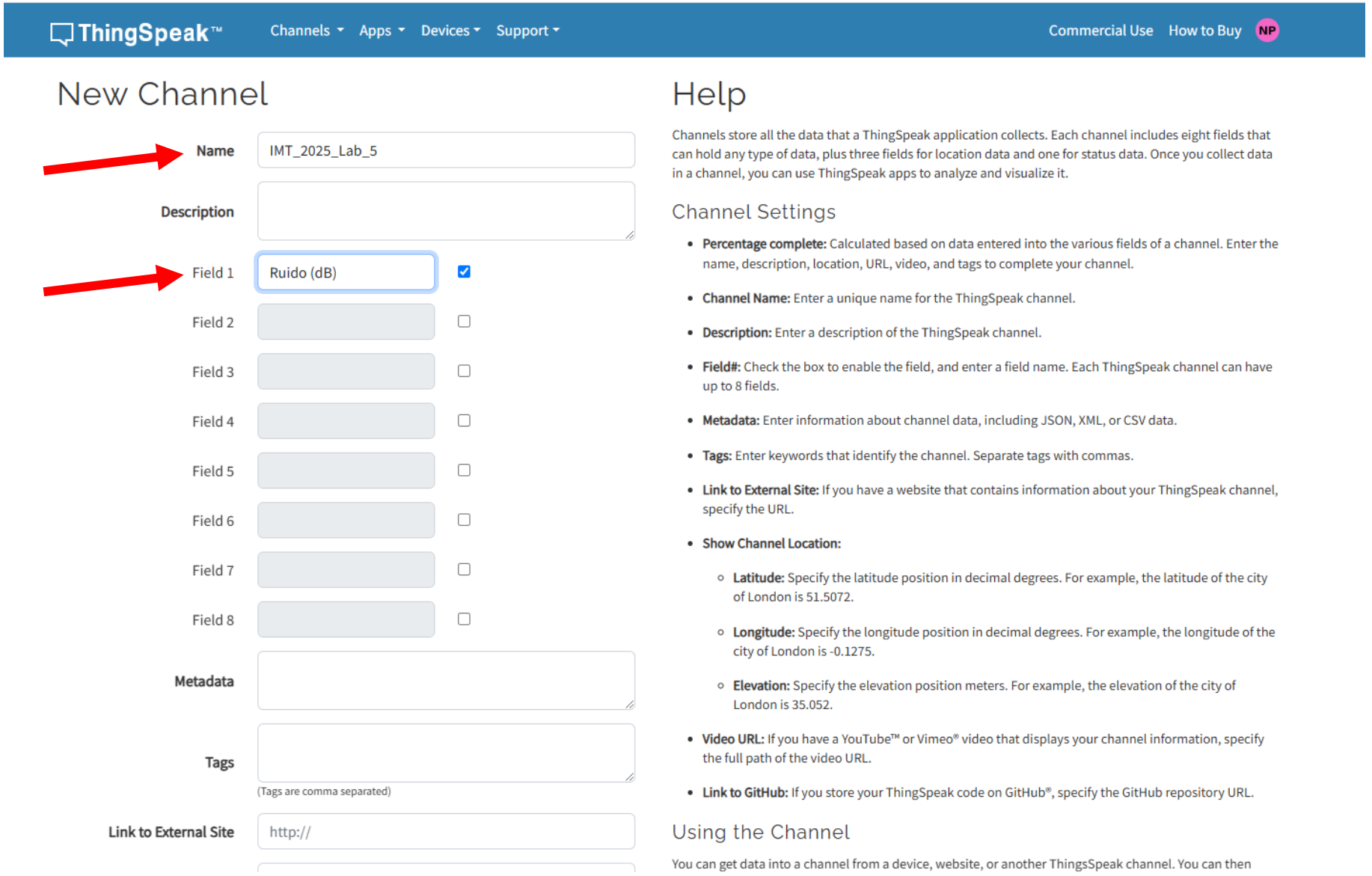


ThingSpeak

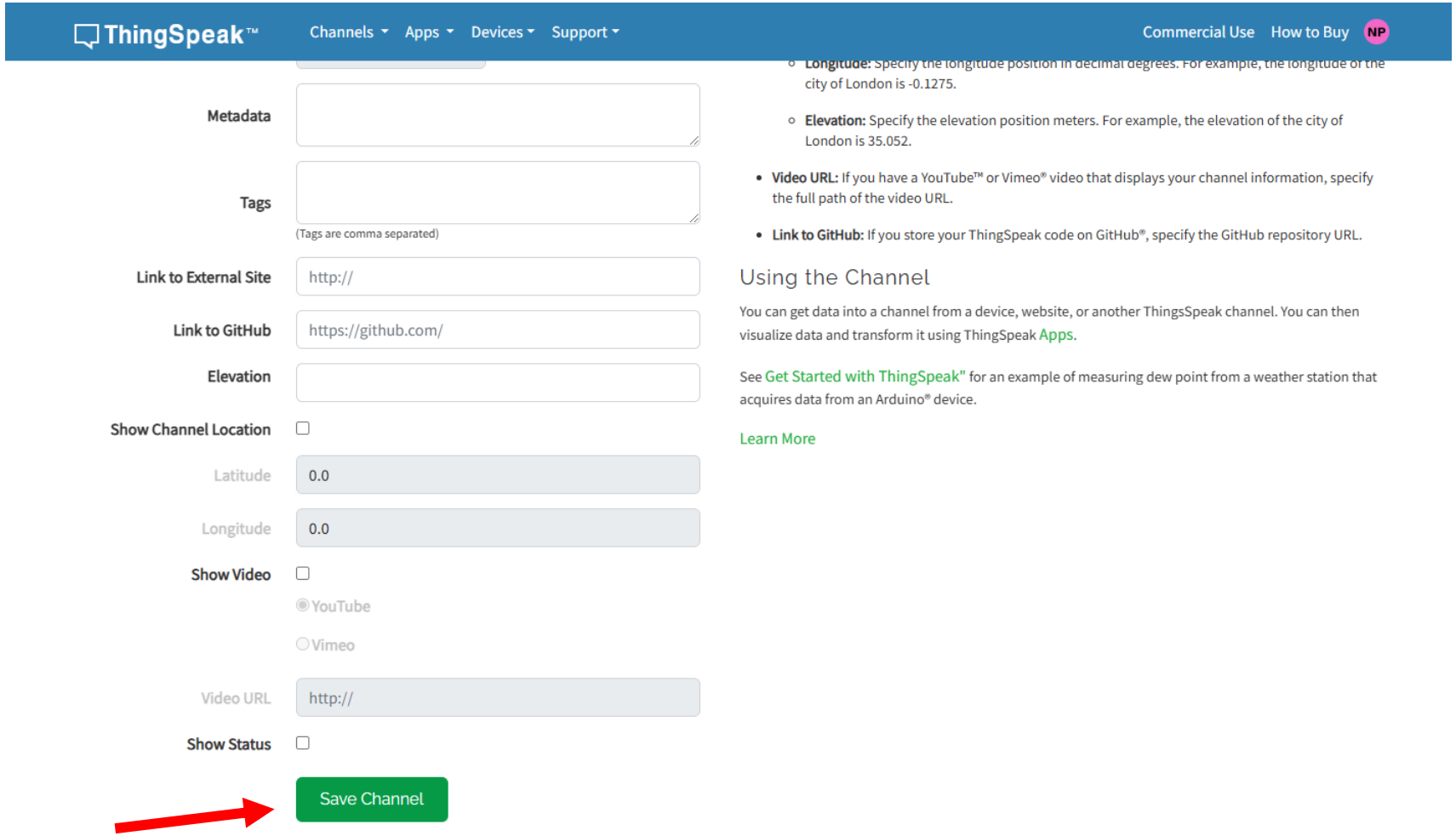


New Channel

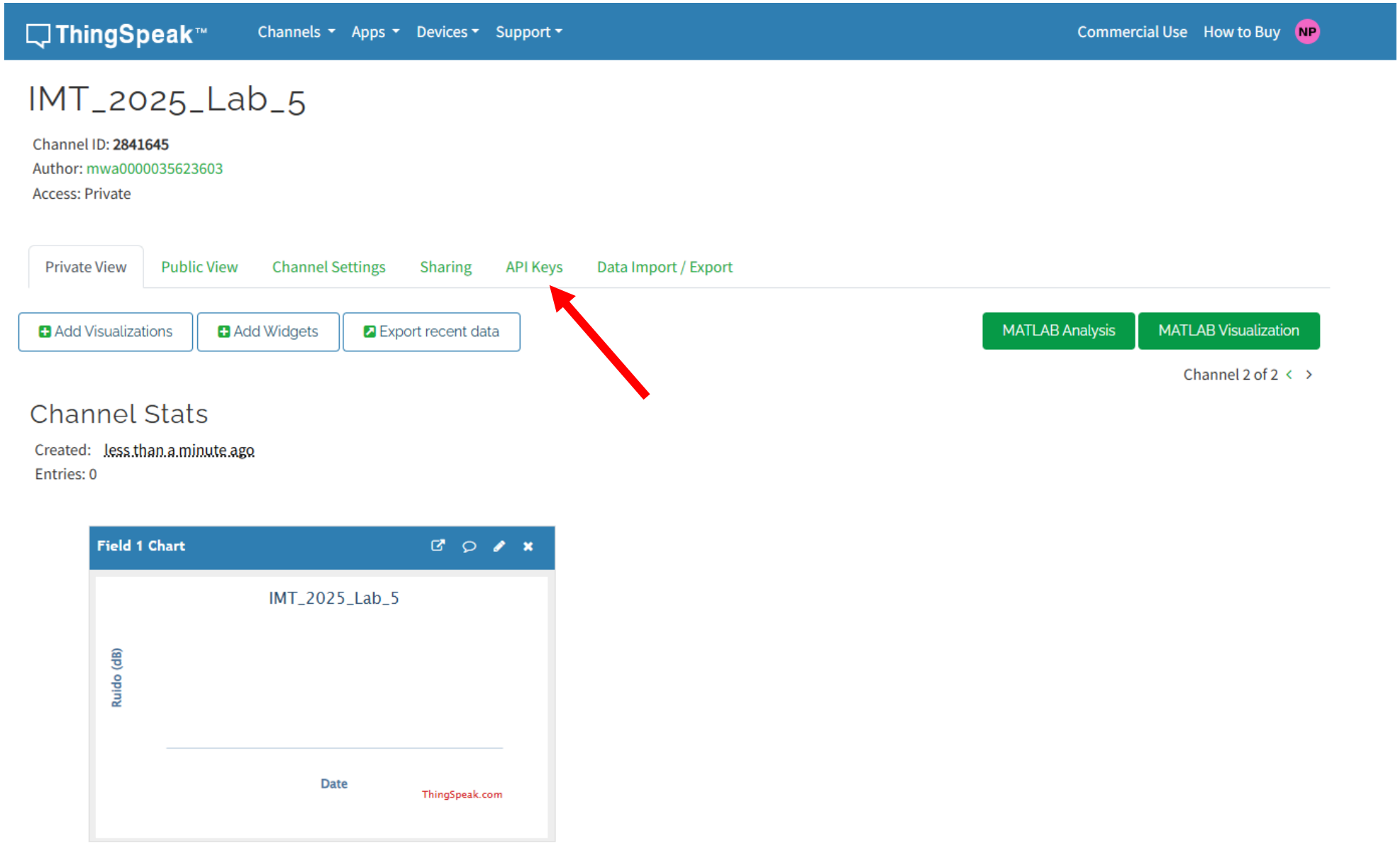
ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
NP
New Channel
Name
IMT_2025_Lab_5
Description
Field 1
Ruido (dB)
Field 2
Field 3
Field 4
Field 5
Field 6
Field 7
Field 8
Metadata
Tags
(Tags are comma separated)
Link to External Site
http://
Help
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.
Channel Settings
Percentage complete: Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
Channel Name: Enter a unique name for the ThingSpeak channel.
Description: Enter a description of the ThingSpeak channel.
Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
Metadata: Enter information about channel data, including JSON, XML, or CSV data.
Tags: Enter keywords that identify the channel. Separate tags with commas.
Link to External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL.
Show Channel Location:
Latitude: Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
Longitude: Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
Elevation: Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
Video URL: If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
Link to GitHub: If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.
Using the Channel
You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then

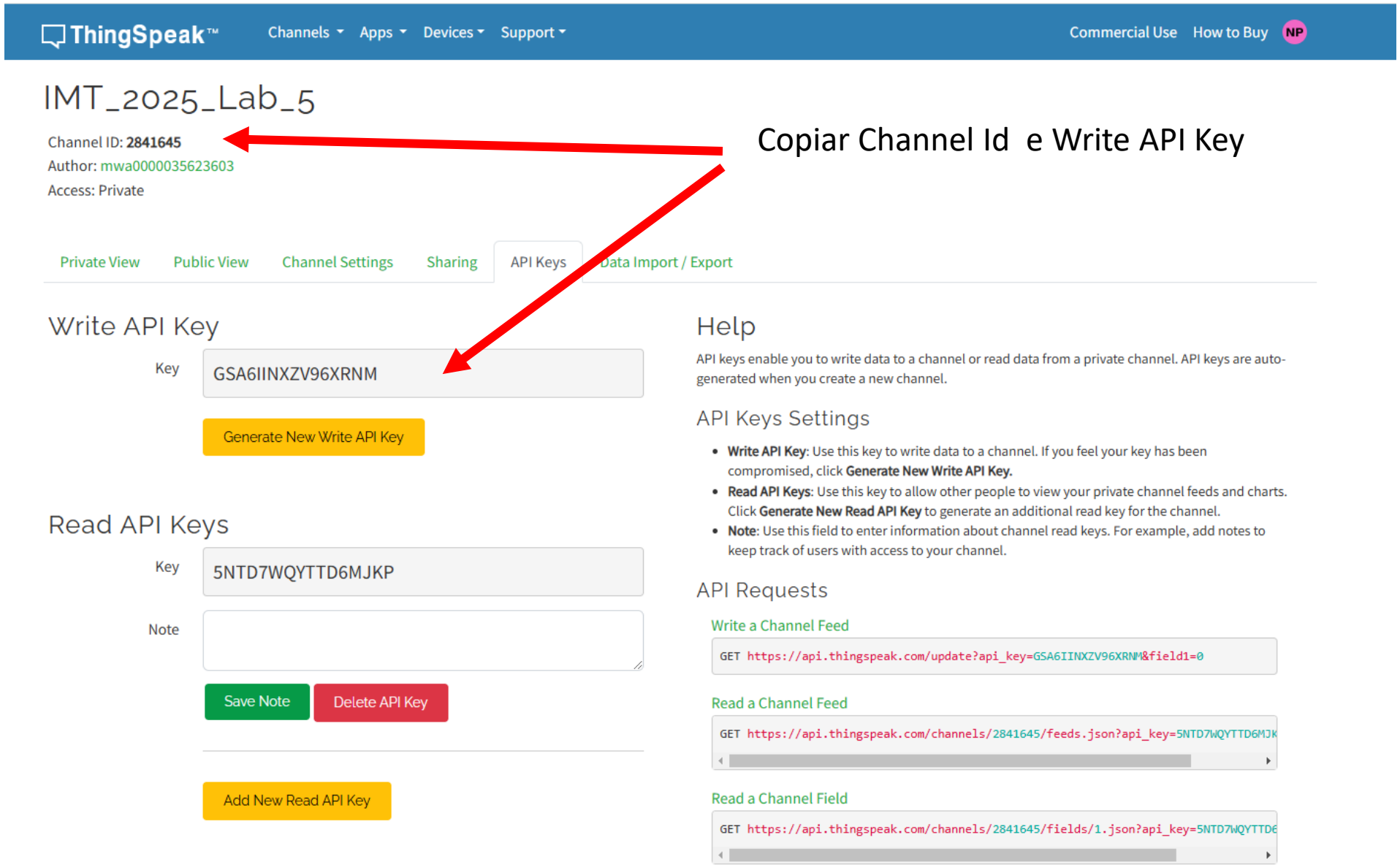
ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
NP
Metadata
Tags
(Tags are comma separated)
Link to External Site
http://
Link to GitHub
https://github.com/
Elevation
Show Channel Location
Latitude
0.0
Longitude
0.0
Show Video
YouTube
Vimeo
Video URL
http://
Show Status
Save Channel
Longitude: Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
Elevation: Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
Video URL: If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
Link to GitHub: If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.
Using the Channel
You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.
See Get Started with ThingSpeak" for an example of measuring dew point from a weather station that acquires data from an Arduino® device.
Learn More

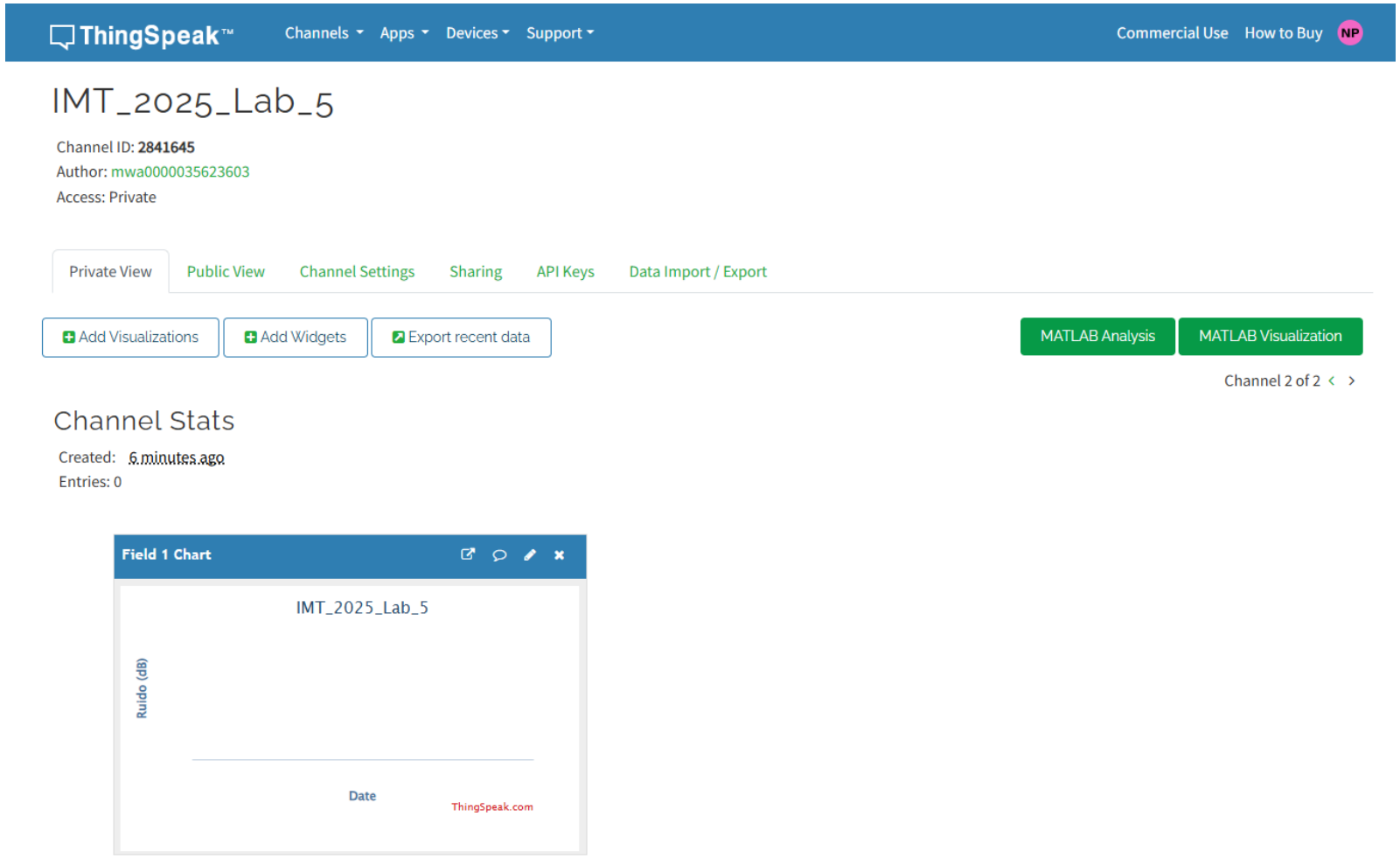
ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
NP
IMT_2025_Lab_5
Channel ID: 2841645
Author: mwa0000035623603
Access: Private
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
MATLAB Analysis
MATLAB Visualization
Channel 2 of 2
Channel Stats
Created: less than a minute ago
Entries: 0
Field 1 Chart
IMT_2025_Lab_5
Ruido (dB)
Date
ThingSpeak.com

ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
NP
IMT_2025_Lab_5
Channel ID: 2841645
Author: mwa0000035623603
Access: Private
Copiar Channel Id e Write API Key
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Write API Key
Key
GSA6IINXZV96XRNM
Generate New Write API Key
Read API Keys
Key
5NTD7WQYTTD6MJKP
Note
Save Note
Delete API Key
Add New Read API Key
Help
API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.
API Keys Settings
Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click Generate New Write API Key.
Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click Generate New Read API Key to generate an additional read key for the channel.
Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.
API Requests
Write a Channel Feed
GET https://api.thingspeak.com/update?api_key=GSA6IINXZV96XRNM&field1=0
Read a Channel Feed
GET https://api.thingspeak.com/channels/2841645/feeds.json?api_key=5NTD7WQYTTD6MJK
Read a Channel Field
GET https://api.thingspeak.com/channels/2841645/fields/1.json?api_key=5NTD7WQYTTD6

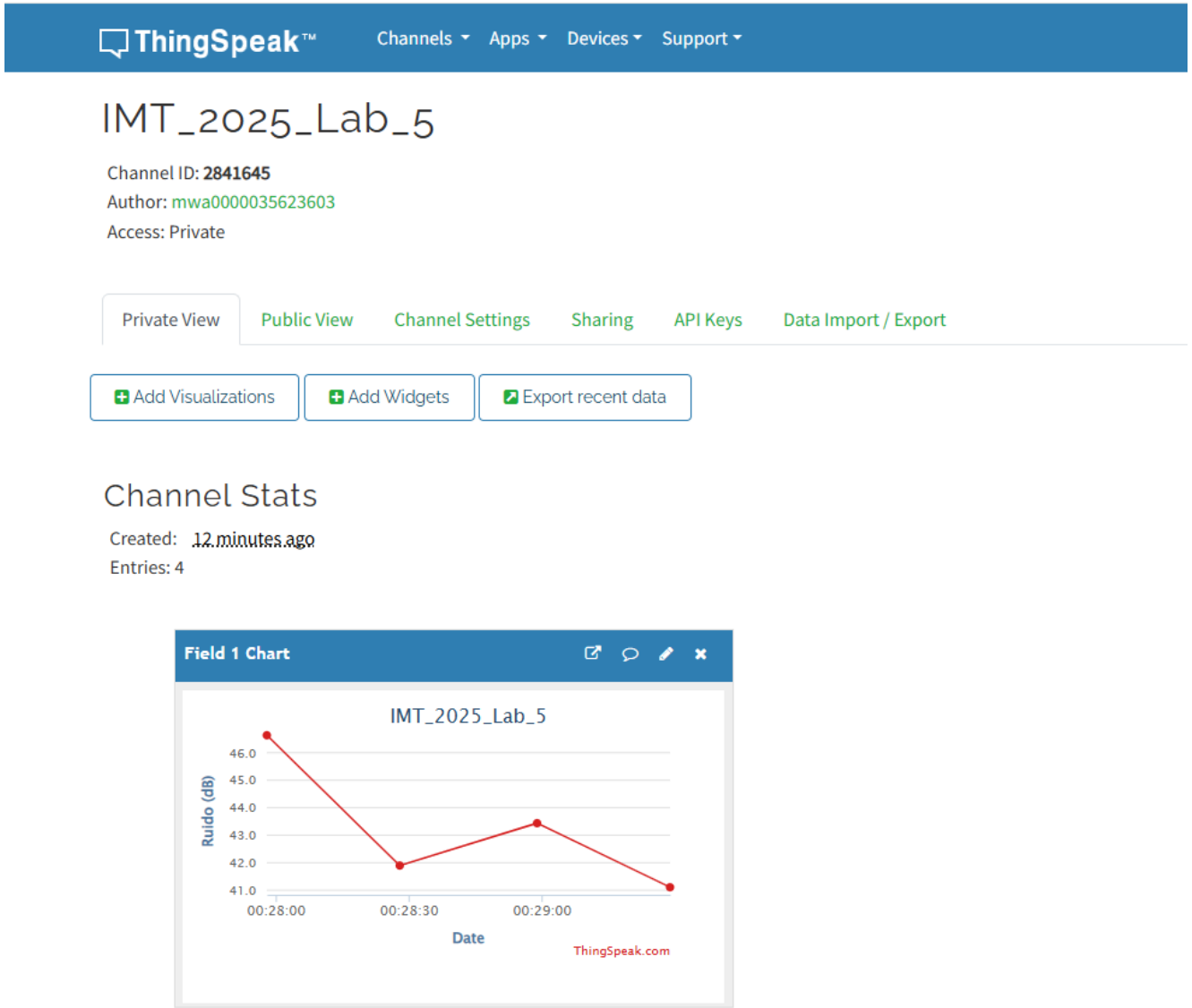
ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
NP
IMT_2025_Lab_5
Channel ID: 2841645
Author: mwa0000035623603
Access: Private
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
MATLAB Analysis
MATLAB Visualization
Channel 2 of 2
Channel Stats
Created: 6 minutes ago
Entries: 0
Field 1 Chart
IMT_2025_Lab_5
Ruído (dB)
Date
ThingSpeak.com

Lab 5 – SW

1 - Abrir o IDE Arduino
2 - Carregar Sketch: Lab_5.ino
3 - Colocar celular modo Roteador
4 - Linha 44 carregar SSID do roteador do seu celular
5 - Linha 45 carregar a senha do seu roteador
6 – Linha 48 colar API Key ThingSpeak
7 – Linha 49 colar Channel ID ThingSpeak
9- Compilar e carregar

ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
IMT_2025_Lab_5
Channel ID: 2841645
Author: mwa0000035623603
Access: Private
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
Channel Stats
Created: 12 minutes ago
Entries: 4
Field 1 Chart
IMT_2025_Lab_5
Ruido (dB)
46.0
45.0
44.0
43.0
42.0
41.0
00:28:00
00:28:30
00:29:00
Date
ThingSpeak.com

Lab 5 – Conclusões

Listar as observações e conclusões do laboratório

INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

LAB 6

Lab 6 – Trabalho de Campo

1. Desligar o Cabo USB do Computador.
2. Montar sistema com antena esférica
3. Ligar sistema Power Bank e verificar se os dados estão subindo para nuvem
4. Desligar o sistema.
5. Cada grupo irá para uma localidade da Mauá, definida
6. Nesta localidade ligar o sensor e aguardar 10 minutos de aquisição de dados
7. Retornar para laboratório para análise de dados.

Lab 6 – Trabalho de Campo

Localidades de Inspeção

1. Biblioteca
2. Laboratório de Motores
3. Cantina da Biblioteca
4. Centro Acadêmico
5. Recepção Reitoria
6. Laboratório de Eletrônica W300
7. Sala U22
8. Fablab
9. Sala Professores
10. Secretaria

Lab 6 – Análise de Dados / Dashboard.

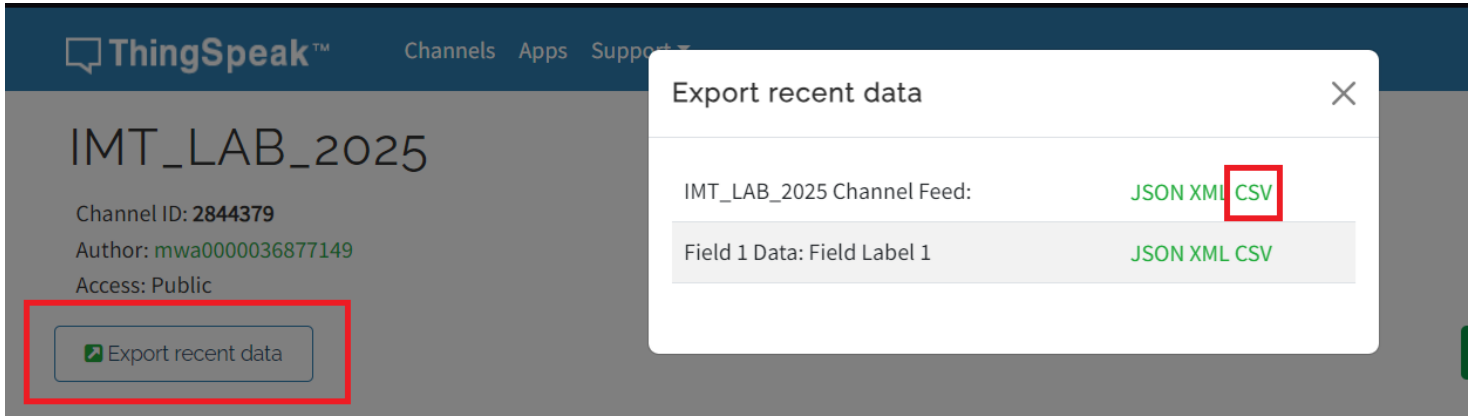
Preparando o ambiente para análise de dados:

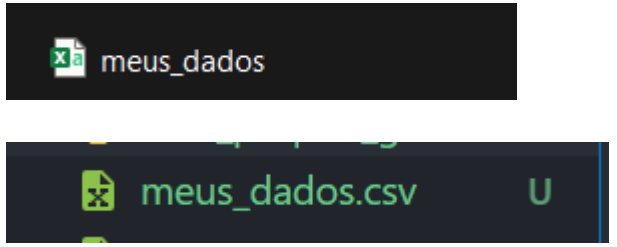
1. Acessar Github: github.com/GabrielGodoy01/mapa_maua
2. Clonar (copiar) o repositório em sua máquina:
   - Clique em "Code" > "Download Zip" > Extraia o conteúdo
3. Abra o aplicativo "VS Code" ou "Visual Studio Code"
4. Clique em "File" > "Open Folder" > Selecione a pasta extraída
5. Abra um terminal no VS Code: "Terminal" > "New Terminal"
   - Use o seguinte comando no terminal para instalar as bibliotecas do projeto:
     - **pip install pandas numpy matplotlib folium**

Lab 6 – Análise de Dados / Dashboard.

Faça download dos dados coletados no ThingSpeak:

- Clique em "Export Recent Data"
- Depois em "CSV" (nome da extensão em formato "excel")
- Salve o arquivo como "meus_dados" dentro da pasta extraída do Github





Lab 6 – Análise de Dados / Dashboard.

Analisando seus próprios dados:

1. No arquivo "meu_proprio_grafico.py" podemos usar o Python para fazer qualquer tipo de gráfico sem depender do ThingSpeak, por exemplo.
2. Abra novamente o terminal e utilize o comando para abrir o gráfico:
   - **python meu_proprio_grafico.py**
3. Aguarde, um gráfico personalizado será aberto

Lab 6 – Análise de Dados / Dashboard.

Demonstração de análise dos dados de todos os grupos:

Lab 6 – Conclusões

Listar as observações e conclusões do laboratório

Coordenador Prof. Dr. Angelo Sebastião Zanini
angelo.zanini@maua.br

INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

Campus São Caetano do Sul
Praça Mauá, 01 - São Caetano do Sul - SP

Prof. Nuncio Perrella
nuncio.perrella@maua.br