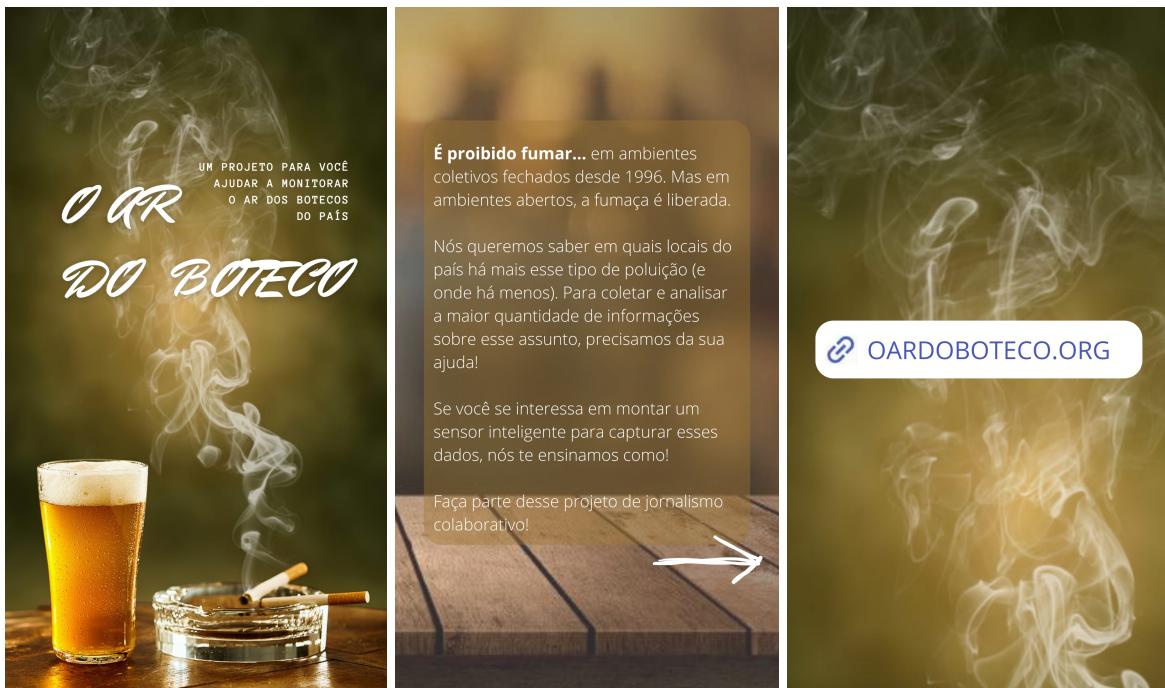


O AR DO BOTECO



A ideia do projeto é coletar dados sobre bares e restaurantes em todo o país, de ambiente aberto (onde é permitido fumar), para saber em quais locais do Brasil a contaminação pela fumaça de cigarro nesses locais ocorre mais.

A ideia surgiu depois que a autora do projeto se mudou de Brasília -DF para Salvador - BA e notou que havia diferenças comportamentais nas saídas para botecos. Na capital baiana, por exemplo, é menos comum o uso do cigarro em saídas para beber do que em Brasília e outros estados.

Uma curiosidade que pode apontar para dados de saúde de uma população, baseada em comportamentos sociais de determinado lugar. A ideia é que a coleta de dados possa ser vista como uma brincadeira divertida por quem topar o desafio... afinal, levar um notebook para o bar não é coisa mais fácil a se fazer, né?

Acima, a ideia para 3 stories no Instagram de divulgação e convite para participar do projeto (que é fictício). Ao clicar no link do projeto, a ideia é cair num site que seja o manual das próximas páginas.

Acesso à planilha do sensor:

<https://docs.google.com/spreadsheets/d/1nizUj1AbfYW-S1B94739828BcPRY5LpKmghnQ78uVm/edit#gid=0>

MANUAL - O AR DO BOTECO

Aqui, vamos te dar o passo a passo para você montar o seu próprio sensor, coletar os dados do ar e nos enviar para que possamos monitorar se a lei que proíbe fumar em ambientes coletivos - como bares - vale ou não vale no Brasil.

(afinal de contas... quem nunca voltou pra casa cheirando cigarro depois de uma noitada, mesmo não tendo fumado nada?)

Siga o passo a passo abaixo e, se der certo, envie um e-mail para: info@oardoboteco.org

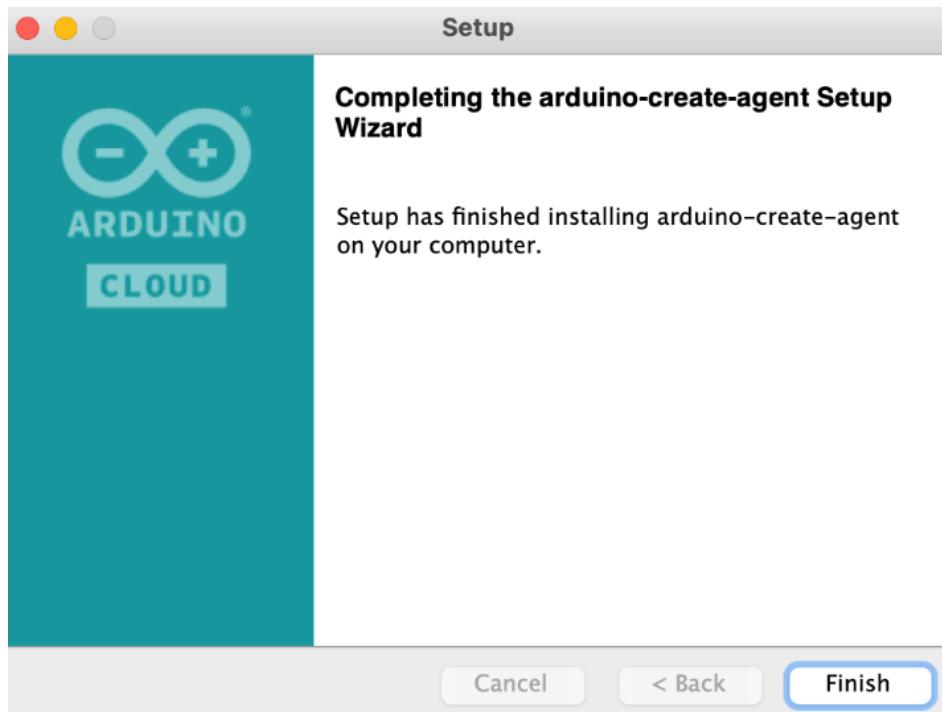
Caso você tenha qualquer tipo de problema, não deixe de nos avisar, queremos ajudar:
problema@oardoboteco.org

MATERIAIS

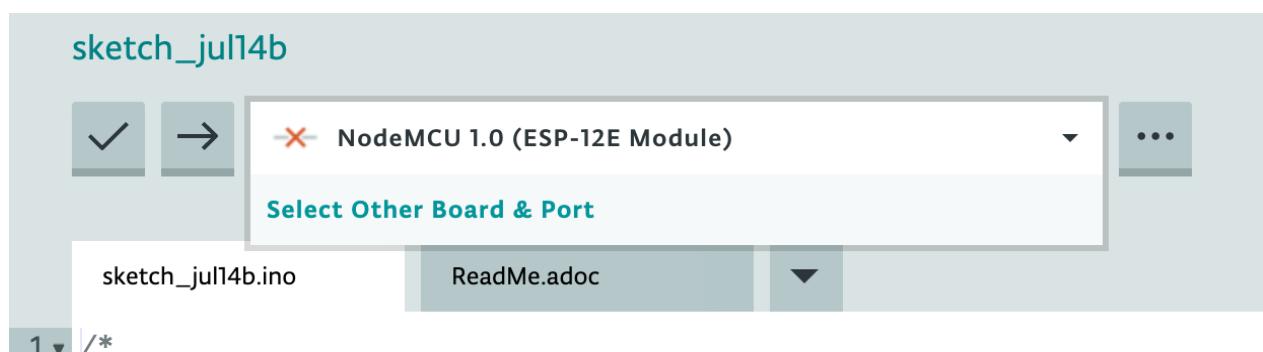
1 <u>Microcontrolador NodeMCU com pinos pré-soldados</u>	 A NodeMCU microcontroller board, which is a CH32V003 chip on a PCB with pre-soldered pins.
1 <u>Mini protoboard</u>	 A white rectangular breadboard with two vertical columns of holes and a central horizontal row of holes.
1 <u>Sensor MQ135 (qualidade do ar)</u>	 A blue printed circuit board with a metal mesh cylindrical sensor at the top and four pins extending from the bottom.
<u>Jumpers Macho-Fêmea</u>	 A bundle of multi-colored male-female jumper wires.
1 <u>Cabo micro-USB de dados</u>	 A black micro-USB data cable with a standard USB-A connector on one end and a micro-USB-B connector on the other.

PREPARAÇÃO

- Acesse o site do [Arduino Web Editor](#) e crie uma conta
- Faça o download do [Arduino Create Agent](#) e instale o programa

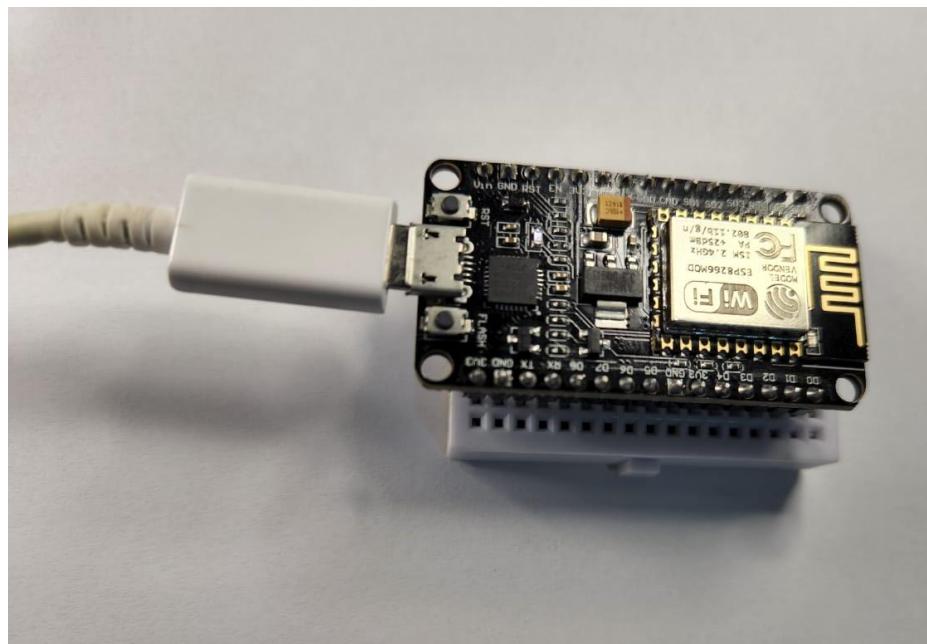


- Abra o [Arduino Web Editor](#)
- Clique em "New Sketch" e na caixa de seleção, opte por "Select Other Board or Port"
- Uma nova caixa vai se abrir, digite NodeMCU e selecione "NodeMCU 1.0 (ESP-12E Module)"
- Não é necessário selecionar nada em "FLAVOURS". Clique em "OK".
- É normal que haja um X na frente do nome, porque a placa ainda não foi conectada

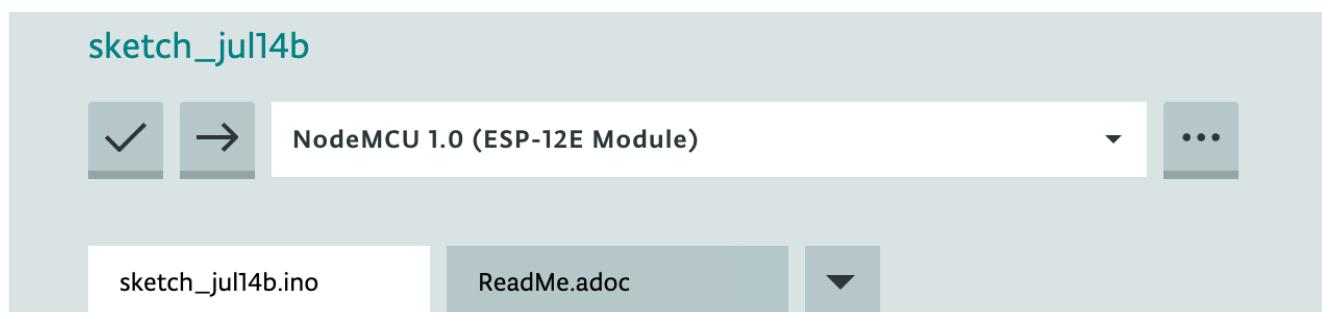


MONTAGEM DO SENSOR

- Encaixe a placa no protoboard com a parte do plugue na primeira fileira de entradas. Conecte a placa de modo que os pinos da esquerda fiquem do lado de uma metade (há a divisão no protoboard) e os pinos da direita na outra metade
- Ligue o cabo USB



- Ao conectar o cabo USB ao computador, o **X** no Arduino Editor em frente ao nome da placa deve desaparecer. É sinal que o Arduino reconheceu a placa.



- Depois de checar se está conectando, retire o cabo USB da placa
- Agora, vamos conectar os jumpers. Atenção nessa montagem.
- Pegue os jumpers fêmea-macho (solte os cabos do bloco e vamos definir as cores).

- Neste exemplo:

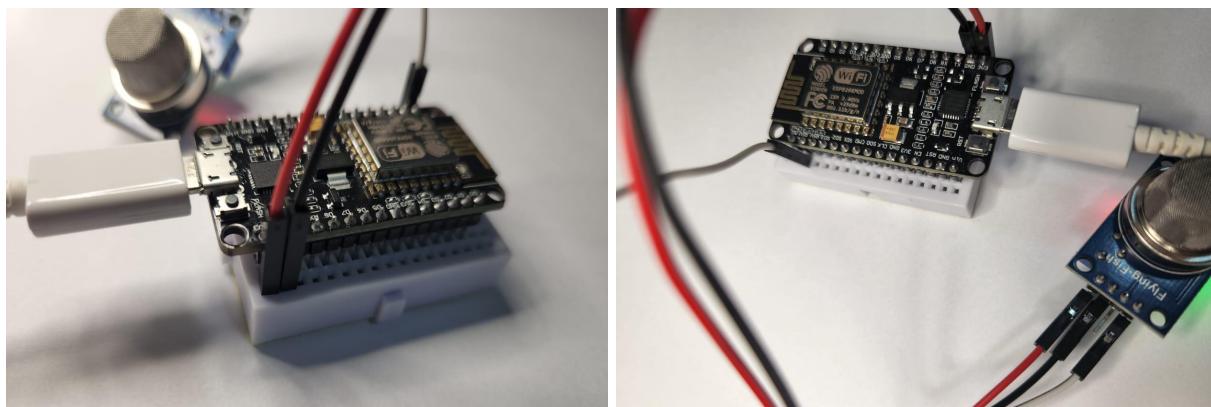
- . **vermelho** - positivo
- . **preto** - negativo
- . **branco** - dados

No **protoboard** (com a placa já encaixada):

- Plugue o cabo **vermelho** na primeira entrada do lado esquerdo (ao lado de onde conecta o USB), onde está escrito **3V3**
- Plugue o cabo **preto** do lado do vermelho, onde está escrito **GND**
- Conecte o cabo **branco** no último buraco do lado oposto dos outros 2, onde está escrito **A0**

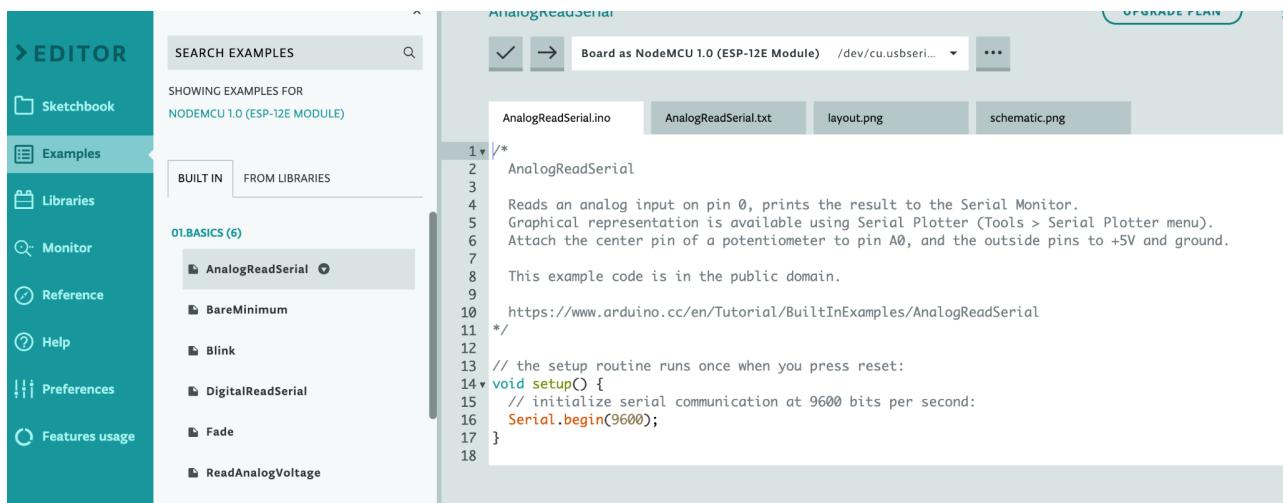
No **sensor**:

- Plugue o cabo **vermelho** no pino **VCC** (o primeiro, da direita para esquerda, com o sensor virado para baixo)
- Plugue o cabo **preto** ao lado **vermelho**, no pino **GND**
- Plugue o cabo **branco** no pino **A0** (o primeiro pino do lado esquerdo)
- A placa deve ficar conectada assim:

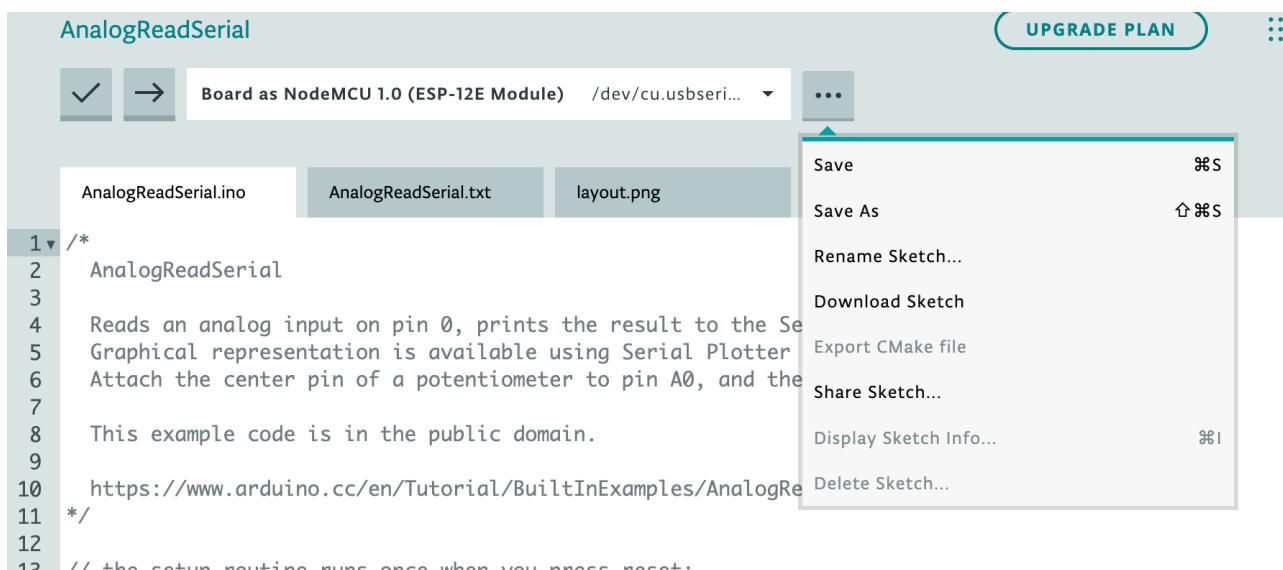


COLETANDO OS DADOS

- Para coletar os dados, é preciso levar o seu notebook para o boteco
- Com o sensor montado o Arduino Web Editor preparado, conecte o sensor no notebook
- Em "Examples", selecione "Basics" e depois selecione "AnalogReadSerial"



- Clique nos 3 pontinhos (...) do lado direito da caixa onde está escrito "Board as NodeMCU 1.0 (ESP-12E Module)" e clique em "Save"



- Uma cópia irá aparecer nos seus Sketchbooks, clicando no menu esquerdo
- Clique na seta → (upload) do lado direito da caixa onde está escrito "Board as NodeMCU 1.0 (ESP-12E Module)"

- O código será instalado no sensor
- No menu esquerdo, clique em "Monitor"
- Os números que devem começar a aparecer já são as informações que precisamos.
- Colete a primeira amostra de números em um ambiente sem fumaça de cigarro ou odores fortes
- Depois, pegue seu notebook e corra para um bar para, é claro, só fazer a medição.

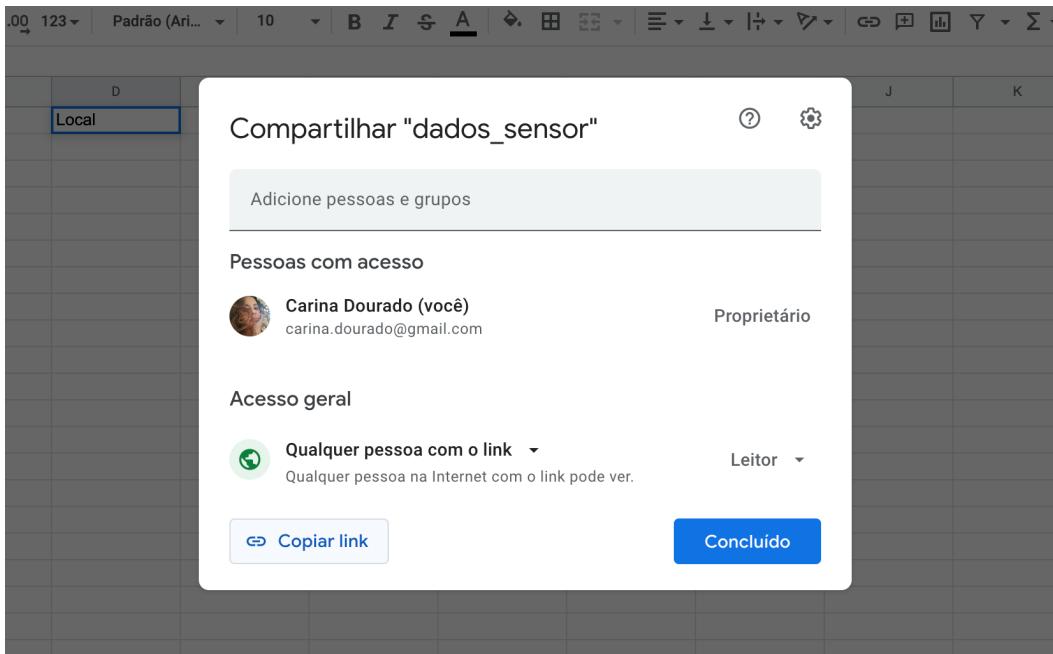
Você pode deixar rodando por 30 minutos, depois copiá-las e nos enviar para o e-mail:
info@oardoboteco.org

É importante nos falar o local (boteco, cidade e estado), a data e hora do início e fim da coleta. E, claro, os seus dados para que a gente possa te creditar como colaborador.

Se você achou fácil até aqui, podemos acrescentar um pouco mais de dificuldade, mas que facilitaria a nossa análise de dados. Basta seguir os próximos passos.

CONFIGURAÇÃO DO GOOGLE SHEETS

- Vá no seu navegador e [crie uma nova planilha no Google Sheets](#)
- Salve com um nome
- Nomeie as colunas: Dia, Hora, Valor e Local
- Clique em "Compartilhar" no canto superior direito da página
- Mude o acesso de "Restrito" para "Qualquer pessoa com o link"
- Clique em "Concluído"



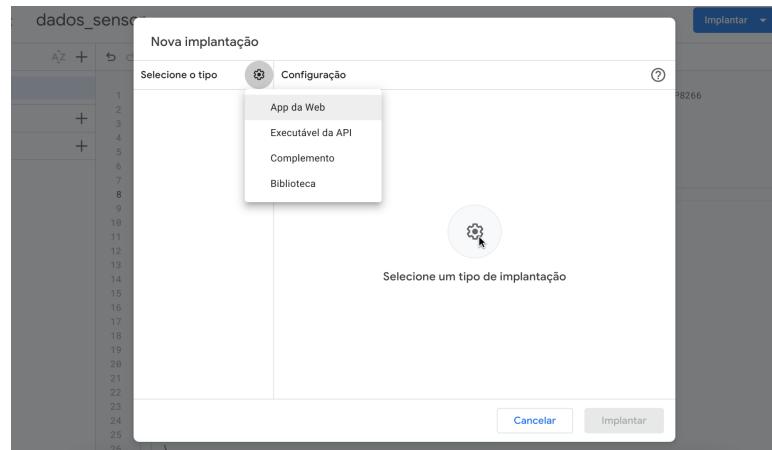
- Copie o ID da planilha. Você o encontra no endereço que está no navegador, entre "d/" e "/edit" (destacado na figura abaixo)

<https://docs.google.com/spreadsheets/d/1nizUj1AbfYW-S1B94739828BcPRY5LpKmghnQ78uVtM/edit#gid=0>

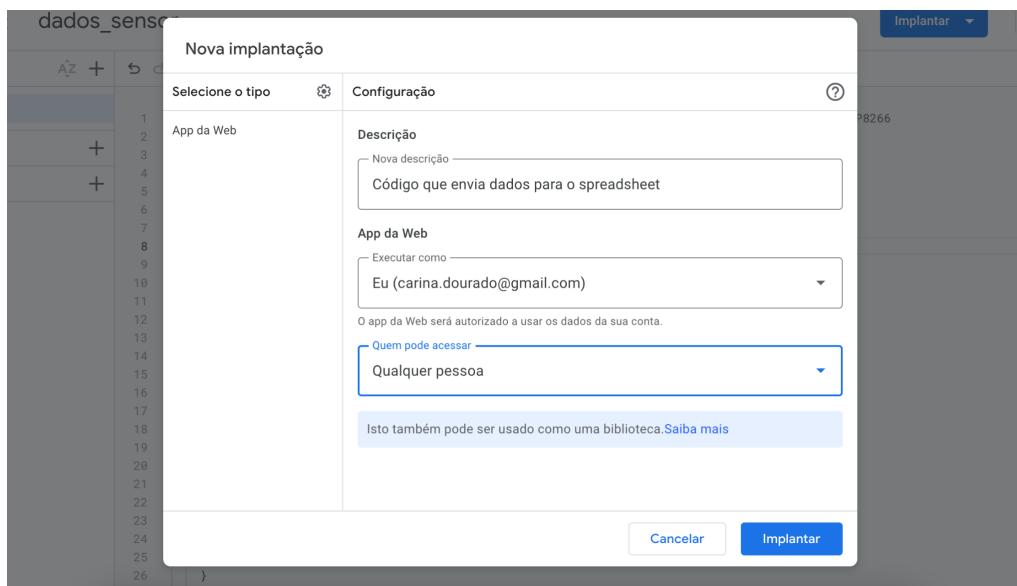
- No menu, clique em "Extensões" e depois em "Apps Script"

- Nomeie o seu projeto
 - Em outra aba, abra o [repositório](#) e o código contido nele
 - Volte na página do Apps Script, apague o código que há lá e cole o que você acabou de copiar.
 - Vá até a linha 8 do código, onde está escrito **INSIRA SEU ID**:
- ```
var SS = SpreadsheetApp.openById('INSIRA SEU ID');
```

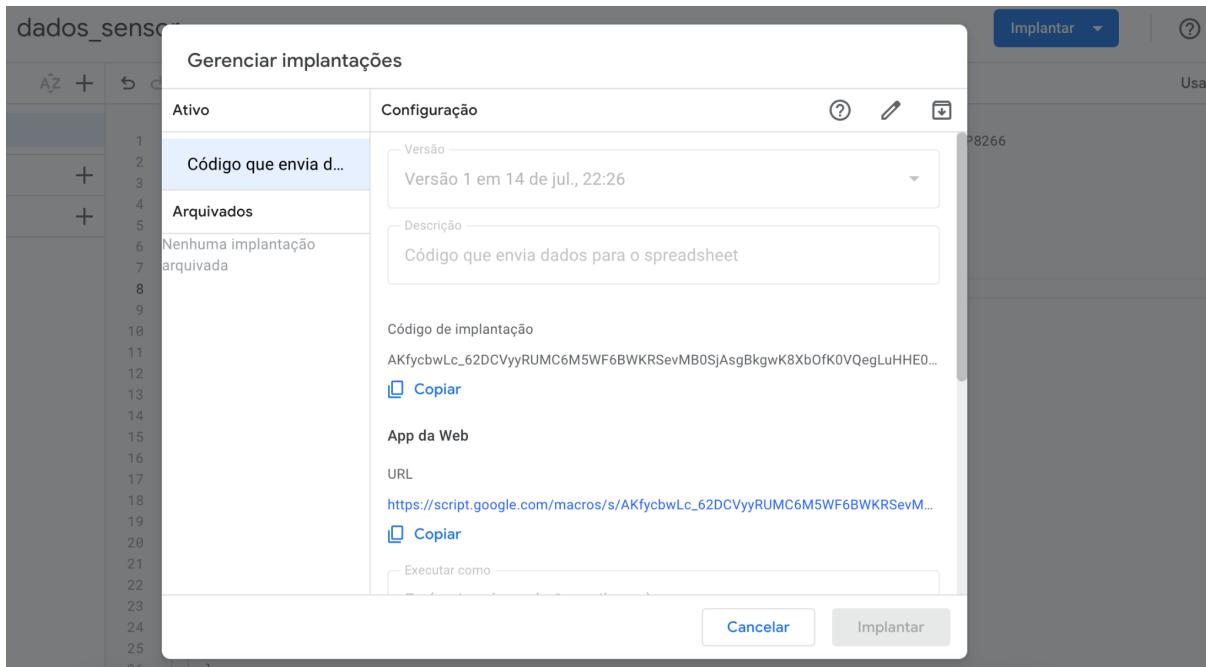
- Substitua o que está entre parênteses pelo seu ID (que você copiou da página do Sheets)
- Salve o projeto no ícone do "disquete"
- Clique em "Implantar" e selecione "Nova implantação"
- Na caixa que abre, clique no ícone de engrenagem e selecione "Web app".



- Adicione uma descrição. Em "Quem pode acessar", mude para "Qualquer pessoa". Clique em "Implantar"



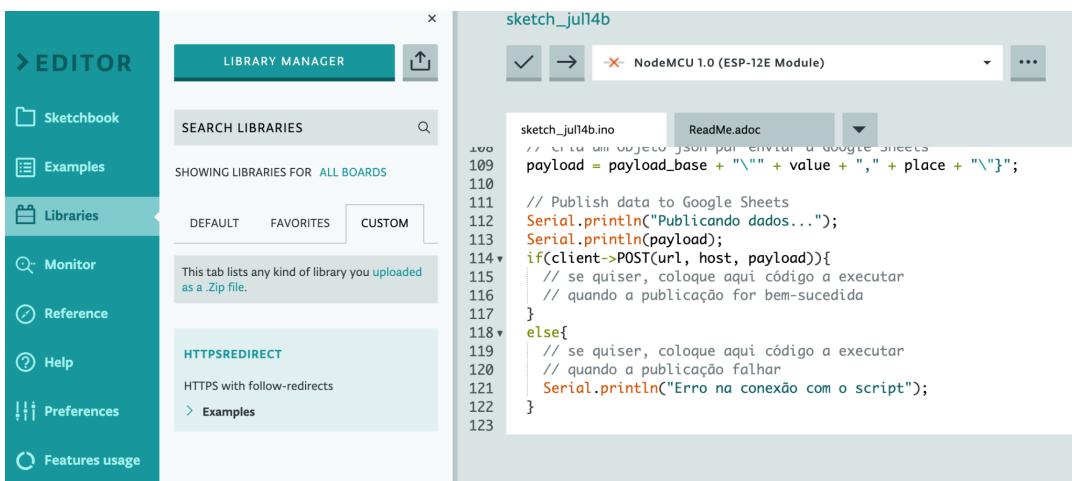
- Clique em "Autorizar acesso". Uma página de alerta será aberta, clique em "Advanced" e autorize o acesso à sua conta, selecionando a conta que quer vincular.
- Copie o código de implantação que aparece (caso você precise localizar esse código depois, basta clicar em "Implantar" e depois em "Gerenciar implantações")



- Na barra acima do código, clique em "Executar" para rodar o código.

## INTEGRAÇÃO - GOOGLE SHEETS E ARDUÍNO

- Abra o [Arduino Web Editor](#) e crie um novo Sketch
- Copie o código do [repositório](#) e cole no código do novo Sketch
- Baixe a biblioteca do [repositório](#)
- Volte no Arduino Web Editor e clique em "Libraries", no menu esquerdo
- Clique no botão de seta para cima, ao lado de "LIBRARY MANAGER" para importar o arquivo que você baixou do repositório



- No código, nas linhas 12 e 13, mude o "NOME-WIFI" para o nome da sua rede de wifi e "senha-wifi" para a senha que você usa para logar neste wifi.

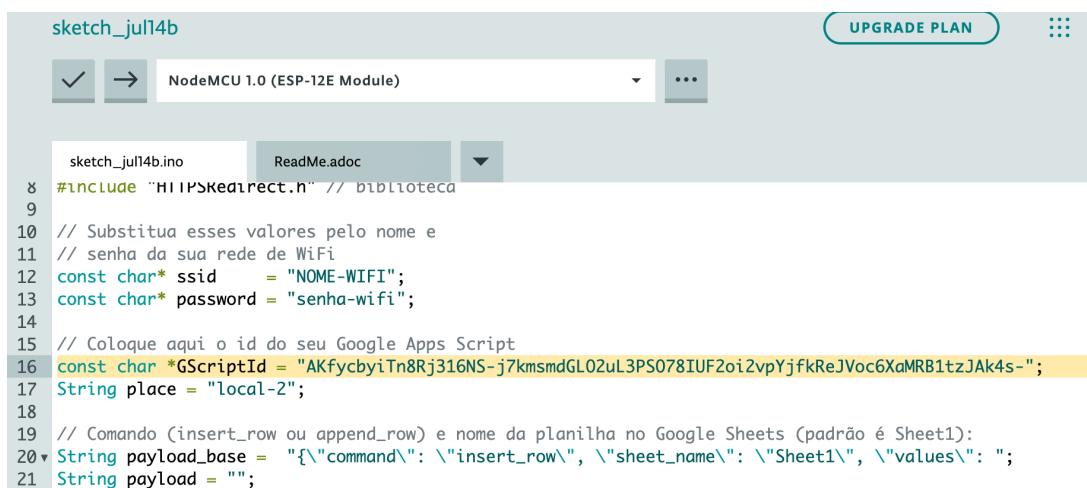


```

sketch_jul14b.ino ReadMe.adoc
8 #include "HTTPSSRedirect.h" // biblioteca
9
10 // Substitua esses valores pelo nome e
11 // senha da sua rede de WiFi
12 const char* ssid = "NOME-WIFI";
13 const char* password = "senha-wifi";
14
15 // Coloque aqui o id do seu Google Apps Script
16 const char *GscriptId = "AKfycbyiTn8Rj316NS-j7kmsmdGL02uL3PS078IUF2oi2vpYjfkReJVoc6XaMRB1tzJAk4s-";
17 String place = "local-2";
18
19 // Comando (insert_row ou append_row) e nome da planilha no Google Sheets (padrão é Sheet1):
20 String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\", \"values\": \"";
21 String payload = "";
22
23 // Google Sheets setup (do not edit)

```

- Na linha 16, mude o GscriptId para o do seu código de implantação, do Apps Script

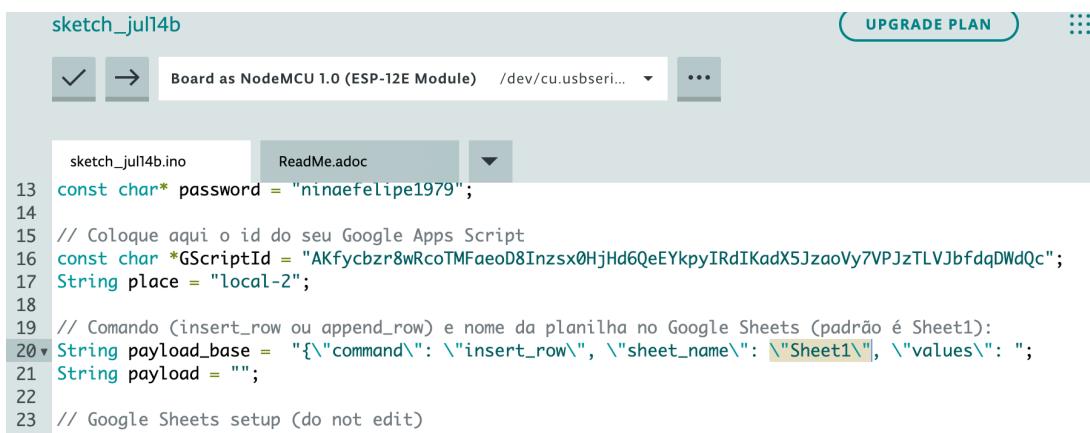


```

sketch_jul14b.ino ReadMe.adoc
8 #include "HTTPSSRedirect.h" // biblioteca
9
10 // Substitua esses valores pelo nome e
11 // senha da sua rede de WiFi
12 const char* ssid = "NOME-WIFI";
13 const char* password = "senha-wifi";
14
15 // Coloque aqui o id do seu Google Apps Script
16 const char *GscriptId = "AKfycbyiTn8Rj316NS-j7kmsmdGL02uL3PS078IUF2oi2vpYjfkReJVoc6XaMRB1tzJAk4s-";
17 String place = "local-2";
18
19 // Comando (insert_row ou append_row) e nome da planilha no Google Sheets (padrão é Sheet1):
20 String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\", \"values\": \"";
21 String payload = "";
22
23 // Google Sheets setup (do not edit)

```

- Cheque na linha 20 se a página da planilha é Sheet1 (inglês) ou Página1 (português)



```

sketch_jul14b.ino ReadMe.adoc
13 const char* password = "ninaefelipe1979";
14
15 // Coloque aqui o id do seu Google Apps Script
16 const char *GscriptId = "AKfycbzcwRcoTMFaeoD8Inzsx0HjHd6QeYkpyIRdIKadX5JzaoV7VPJzTLVJbfdqDWdQc";
17 String place = "local-2";
18
19 // Comando (insert_row ou append_row) e nome da planilha no Google Sheets (padrão é Sheet1):
20 String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\", \"values\": \"";
21 String payload = "";
22
23 // Google Sheets setup (do not edit)

```

- Clique na seta → ao lado, em "Upload and Save"

```

sketch_jul14b
UPGRADE PLAN
✓ → Board as NodeMCU 1.0 (ESP-12E Module) /dev/cu.usbseri... ...
sketch_jul14b.ino ReadMe.adoc

1 // Exemplo de envio de dados do NodeMCU para Google Sheets
2
3 // Este exemplo foi criado pa partir do exemplo no link a seguir
4 // TUTORIAL AQUI: https://github.com/carolinex/Google-Sheets-Logging-MQ135
5
6 #include <Arduino.h> // biblioteca
7 #include <ESP8266WiFi.h> // biblioteca
8 #include "HTTPSRedirect.h" // biblioteca
9
10 // Substitua esses valores pelo nome e
11 // senha da sua rede de WiFi
12 const char* ssid = "JANESCLEIDEBERENICE";
13 const char* password = "ninaefelipe1979";
14
15 // Coloque aqui``. o id do seu Google Apps Script
16 const char *GscriptId = "AKfycbyV7B-YkMq0VOZUyy_vHYVf10eCT-NSHBTxjBPGWowqjxvGp3LQPTjXg7peAwdIzKT5SQ";
17 String place = "local-2";
18
Success: Saved on your online Sketchbook and done uploading sketch_jul14b
espcomm_send_command: sending command payload
espcomm_send_command: receiving 2 bytes of data
closing bootloader

```

- Se estiver tudo correto, os dados começarão a ser atualizados a cada 2 minutos na planilha do Google Sheets que você criou.

| A1 | A          | B        | C     | D       |
|----|------------|----------|-------|---------|
| 1  | Dia        | Hora     | Valor | Local   |
| 2  | 15/07/2022 | 22:44:37 | 483   | local-2 |
| 3  | 15/07/2022 | 22:44:25 | 484   | local-2 |
| 4  |            |          |       |         |

- Colete a primeira amostra de números em um ambiente sem fumaça de cigarro ou odores fortes

Pegue seu notebook e o leve para beber... digo, para analisar o ar que você anda respirando nos botecos da sua cidade.

E, claro, não se esqueça de depois nos enviar a sua planilha para análise, com seus dados para que você possa ser creditado como colaborador, para o e-mail: [info@oardoboteco.org](mailto:info@oardoboteco.org)

# PROBLEMAS ENFRENTADOS

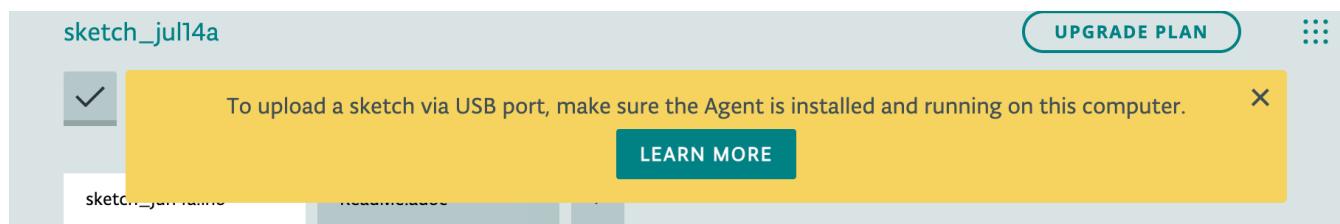
1. No fim do processo, dando "Success" ao rodar o código, no Monitor havia o erro:

```
Conectando à rede Wifi JANESCLEIDEBERENICE ...
...
Conexão estabelecida!
IP: 192.168.0.227
Conectando a script.google.com
NodeMCU conectado ao host
Publicando dados...
{"command": "insert_row", "sheet_name": "Sheet1", "values": "559,local-2"}
<!DOCTYPE html><html><head><link rel="shortcut icon" href="//ssl.gstatic.com/docs/script/images/favicon.ico"><title>Erro</title><style type="text/css" nonce="4OILnbGKjAnjpBGp_c_TuA">body {background-color: #fff; margin: 0; padding: 0;}.errorMessage {font-family: Arial,sans-serif; font-size: 12pt; font-weight: bold; line-height: 150%; padding-top: 25px;}</style></head><body style="margin:20px"><div></div><div style="text-align:center;font-family:monospace;margin:50px auto 0;max-width:600px">TypeError: Cannot read property 'insertRows' of null
(linha 43, arquivo "Código")</div></body></html>
```

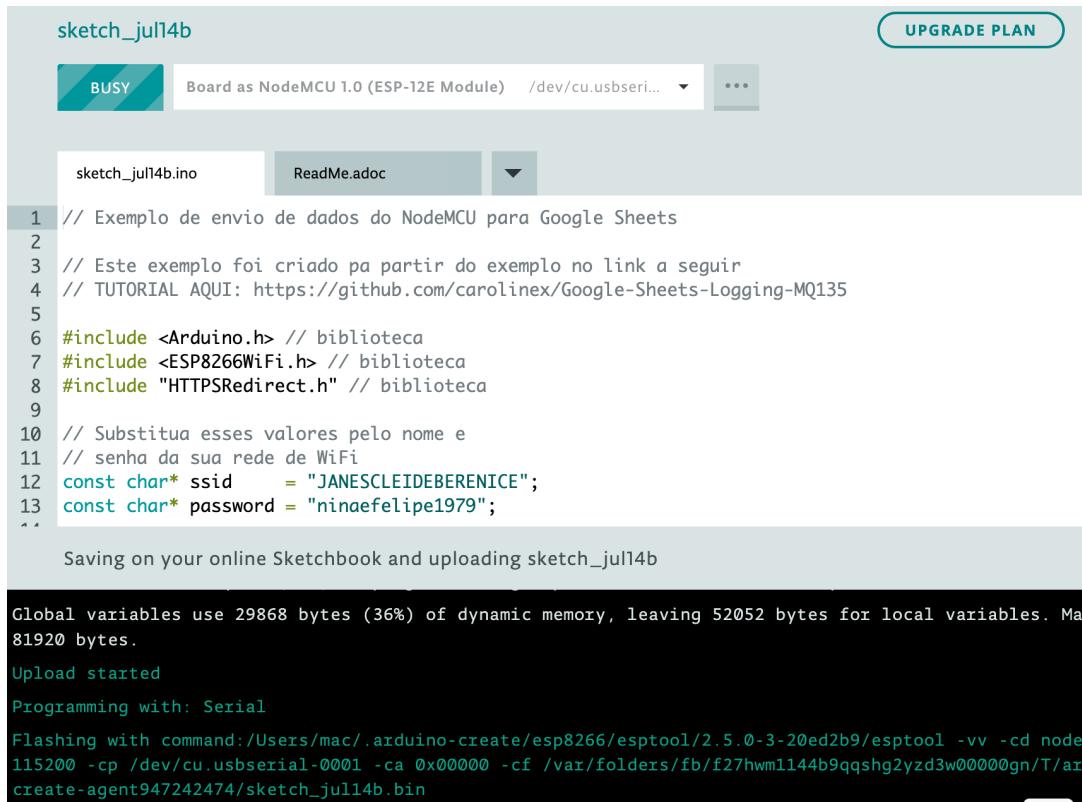
Problema resolvido alterando no código do Arduino o nome da tabela no Google Sheets, porque o Sheets em Português salva como **Página1**:

```
// Comando (insert_row ou append_row) e nome da planilha no Google Sheets (padrão é Sheet1):
String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\",\n\"values\": \"
```

2. Também tive que reinstalar, a pedido do próprio arduino, o **Agent** por 4 vezes.



3. A placa várias vezes apagava e quando pedia para rodar o código, ela ficava em "busy" eternamente, sem nenhuma alteração. Deixei ela desligada uns minutos, resetei, troquei o cabo USB, troquei o jumper de dados (branco), ela voltou a funcionar, mas depois de algum tempo de uso, voltou a dar defeito e não consegui concluir a atualização planilha por esse motivo (o busy que não terminava nunca). A luz azul mais forte já não acendia.



```

sketch_jul14b
UPGRADE PLAN

BUSY Board as NodeMCU 1.0 (ESP-12E Module) /dev/cu.usbseri... ⚙️

sketch_jul14b.ino ReadMe.adoc ▾

1 // Exemplo de envio de dados do NodeMCU para Google Sheets
2
3 // Este exemplo foi criado pa partir do exemplo no link a seguir
4 // TUTORIAL AQUI: https://github.com/carolinex/Google-Sheets-Logging-MQ135
5
6 #include <Arduino.h> // biblioteca
7 #include <ESP8266WiFi.h> // biblioteca
8 #include "HTTPSRedirect.h" // biblioteca
9
10 // Substitua esses valores pelo nome e
11 // senha da sua rede de WiFi
12 const char* ssid = "JANESCLEIDEBERENICE";
13 const char* password = "ninaefelipe1979";
```
Saving on your online Sketchbook and uploading sketch_jul14b

Global variables use 29868 bytes (36%) of dynamic memory, leaving 52052 bytes for local variables. Max 81920 bytes.
Upload started
Programming with: Serial
Flashing with command:/Users/mac/.arduino-create/esp8266/esptool/2.5.0-3-20ed2b9/esptool -vv -cd node115200 -cp /dev/cu.usbserial-0001 -ca 0x00000 -cf /var/folders/fb/f27hwm1144b9qqshg2yzd3w00000gn/T/arduino-agent947242474/sketch_jul14b.bin

```

Pesquisando na internet, acabei achando um [comentário num fórum sobre Arduino](#) e que resolveu meu problema, consegui instalar o código no sensor e finalmente a planilha ser atualizada.



who_took_my_nick Karma: 1+

jan '18 **post #7**

Following that tutorial and Tool V3.4.4 i get invalid head of packet. But using V2.3 i get this from the first massage. Firmware is 2.0 in both cases. Except he first press reset and then flash, but in my case and on bunch of cases on the net first is need to press flash and then reset. Release first flash and then reset. If i do as guy on the link i can not enter boot mode.

After success flash i should press RST, power down ESP and power it up.

Not sure do i have to press something at the very last boot sequence. Juraj mention to put GPIO0 on HIGH but i never see someone do that.

4. De tanto tentar corrigir o problema, meu limite da conta grátis de compilações no Arduino chegou ao limite. Tive que criar uma outra conta, com outro e-mail, para poder continuar tentando corrigir os problemas

