

Sistemas Embarcados

Introdução aos Sistemas Embarcados

- Sistemas embarcados são computadores dedicados a realizar funções específicas.
- São integrados a dispositivos e otimizados para desempenho e eficiência. Ex: automação, IoT, automóveis, eletrodomésticos.

Características dos Sistemas Embarcados

- Tempo real
- Baixo consumo de energia
- Confiabilidade
- Recursos limitados (memória, processamento)
- Alta integração com sensores e atuadores

Plataformas Populares de Sistemas Embarcados

- Exemplos: Arduino, ESP32, NodeMCU e Raspberry Pi
- Cada uma possui características específicas, aplicações distintas e diferentes capacidades de hardware.

Arduino

- Microcontrolador de 8/16 bits
- Fácil de programar (linguagem baseada em C++)
- Ideal para projetos educacionais e protótipos
- Baixo custo e vasta comunidade

ESP32

- Microcontrolador da Espressif
- Dual core, Wi-Fi e Bluetooth integrados
- Mais potente que o Arduino
- Ideal para aplicações IoT e conectividade sem fio

NodeMCU

- Placa baseada no ESP8266
- Wi-Fi integrado
- Programável com Lua ou Arduino IDE
- Popular em projetos de IoT de baixo custo

Diferença entre NodeMCU e ESP

- ESP8266 / ESP32 (ESP)
- O que são: São microcontroladores fabricados pela Espressif. O ESP8266 e o ESP32 são chips integrados com Wi-Fi (e Bluetooth no caso do ESP32).
- Características:
- ESP8266: processador de 80/160 MHz, Wi-Fi, memória limitada.
- ESP32: mais poderoso, dual-core, Wi-Fi + Bluetooth, mais GPIOs, mais memória.
- Uso típico: São os "cérebros" dos dispositivos. Sozinhos, precisam de placa de desenvolvimento para serem facilmente usados por makers.

Diferença entre NodeMCU e ESP

- NodeMCU
- O que é: É uma placa de desenvolvimento que utiliza o chip ESP8266.
- Inclui:
- Chip ESP8266.
- Interface USB-serial (para programação via cabo USB).
- Regulador de tensão, pinos organizados, botão de reset, etc.
- Vantagens:
- Fácil de usar com Arduino IDE.
- Compatível com protoboards.
- Projeto open-source baseado em firmware Lua, mas hoje amplamente programado com Arduino IDE.

Diferença entre NodeMCU e ESP

Comparação Rápida

Característica	ESP8266 / ESP32 (Chip)	NodeMCU (Placa)
Tipo	Microcontrolador	Placa de desenvolvimento com ESP8266
Produzido por	Espressif	Comunidade/open-source
Interface USB	Não possui	Sim
Regulador de tensão	Não	Sim
Uso direto	Exige mais conhecimento técnico	Pronto para uso em prototipagem

Raspberry Pi

- Computador de placa única (SBC)
- Roda sistemas operacionais Linux
- Suporta multitarefa e interface gráfica
- Ideal para aplicações mais complexas e educativas

Comparação Geral

- Arduino: simples, sem conectividade nativa
- ESP32: potente, conectividade avançada
- NodeMCU: ideal para automação Wi-Fi simples
- Raspberry Pi: completo, ideal para sistemas mais robustos

Aplicações Típicas

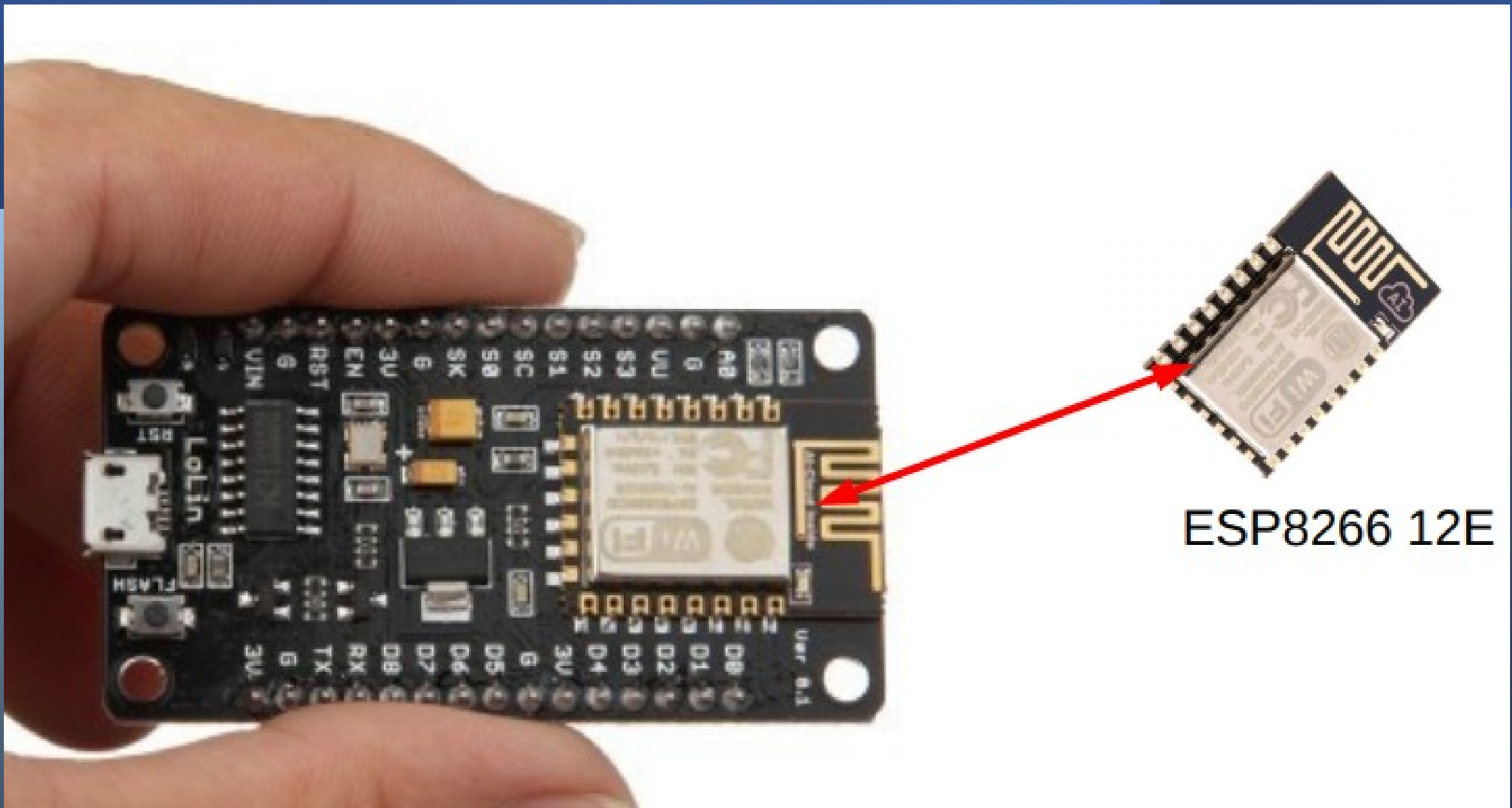
- Arduino: sensores e atuadores, automação básica
- ESP32/NodeMCU: IoT, monitoramento remoto
- Raspberry Pi: servidores, hubs IoT, processamento de imagem

Esp 8266, Esp 32 e
NodeMCU

VARIANT

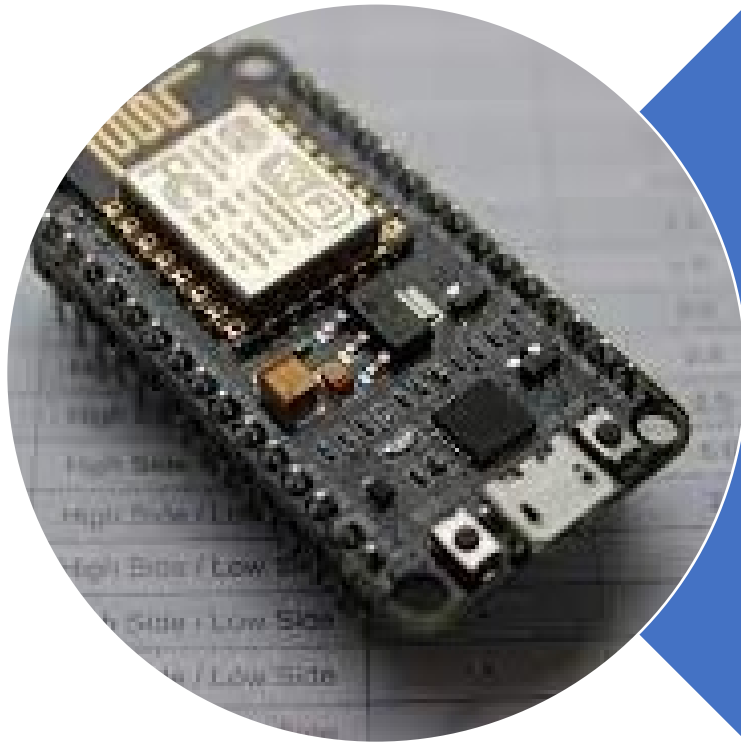
ES



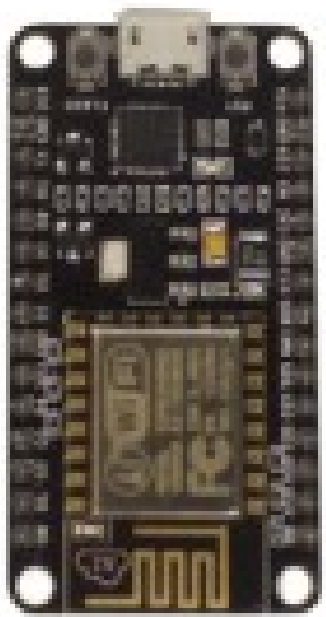


ESP8266 12E

NodeMCU



Módulo **NodeMCU** é uma plataforma de internet das coisas, de baixo custo e open-source, e que foi criada para facilitar o desenvolvimento de aplicações para o módulo **ESP8266 ESP-12**.



Ver 0.1

MODEL VENDOR
ESP8266MOD
ISM 2.4GHz
PA +25dBm
DBZ 11b/g/n

WiFi

FC



AD G UW S3 S2 S1 SC SO SK G 3U EN RST G VIN

00 01 02 03 04 3U G 05 06 07 08 RX TX G 3U

FLASH

RST





KODEL
VENDOR

ESP8266F00



ISM 2.4GHz

PA +25dBm

802.11b/g/n

1. install CH340G driver
2. use 9600bps baud rate
3. Connect to wifi

new Nodemcu V3



AN OPEN PROJECT WRITTEN, DEBUGGED,
AND SUPPORTED BY ARDUINO.CC AND
THE ARDUINO COMMUNITY WORLDWIDE

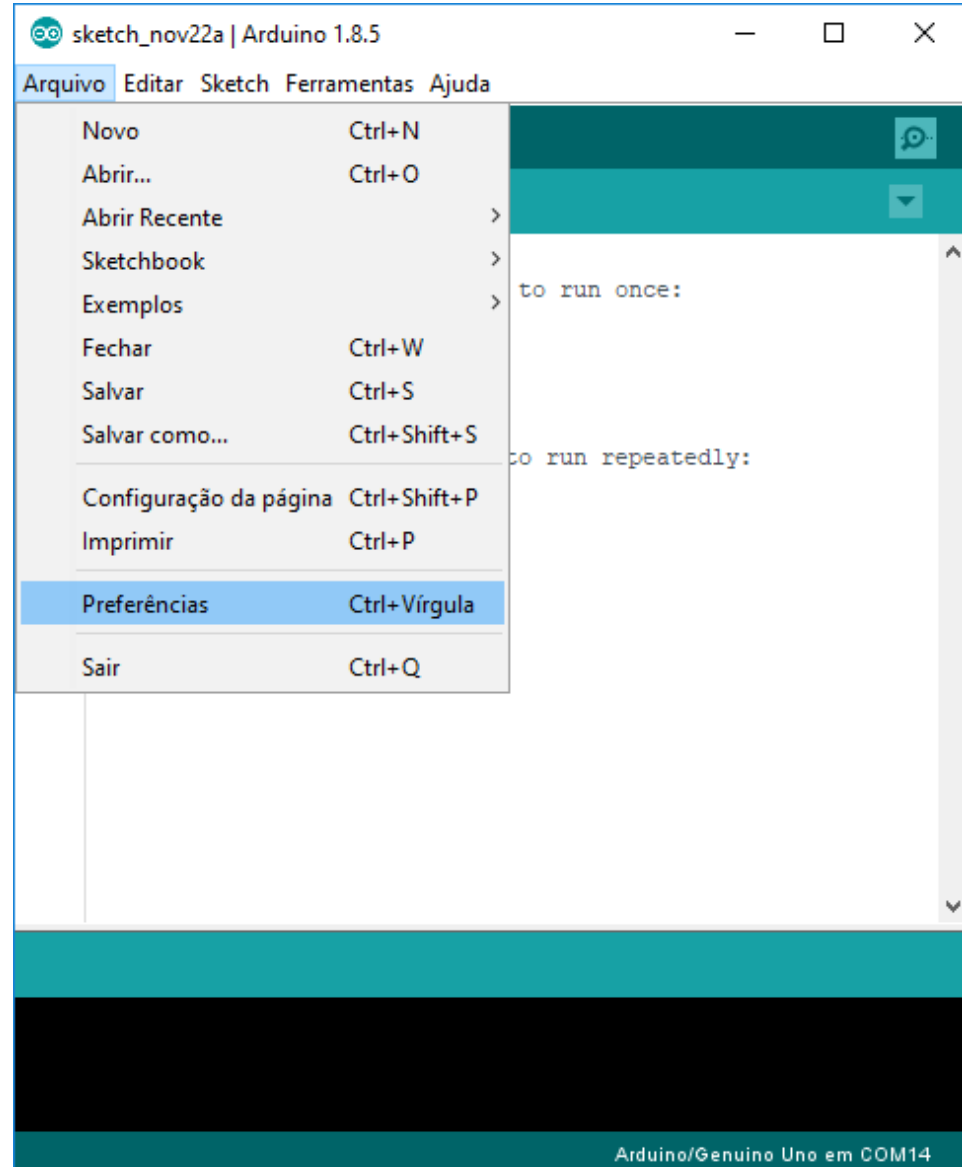
LEARN MORE ABOUT THE CONTRIBUTORS
OF [ARDUINO.CC](https://arduino.cc) AT arduino.cc/credits



Configurando a Arduino IDE

Para NodeMCU

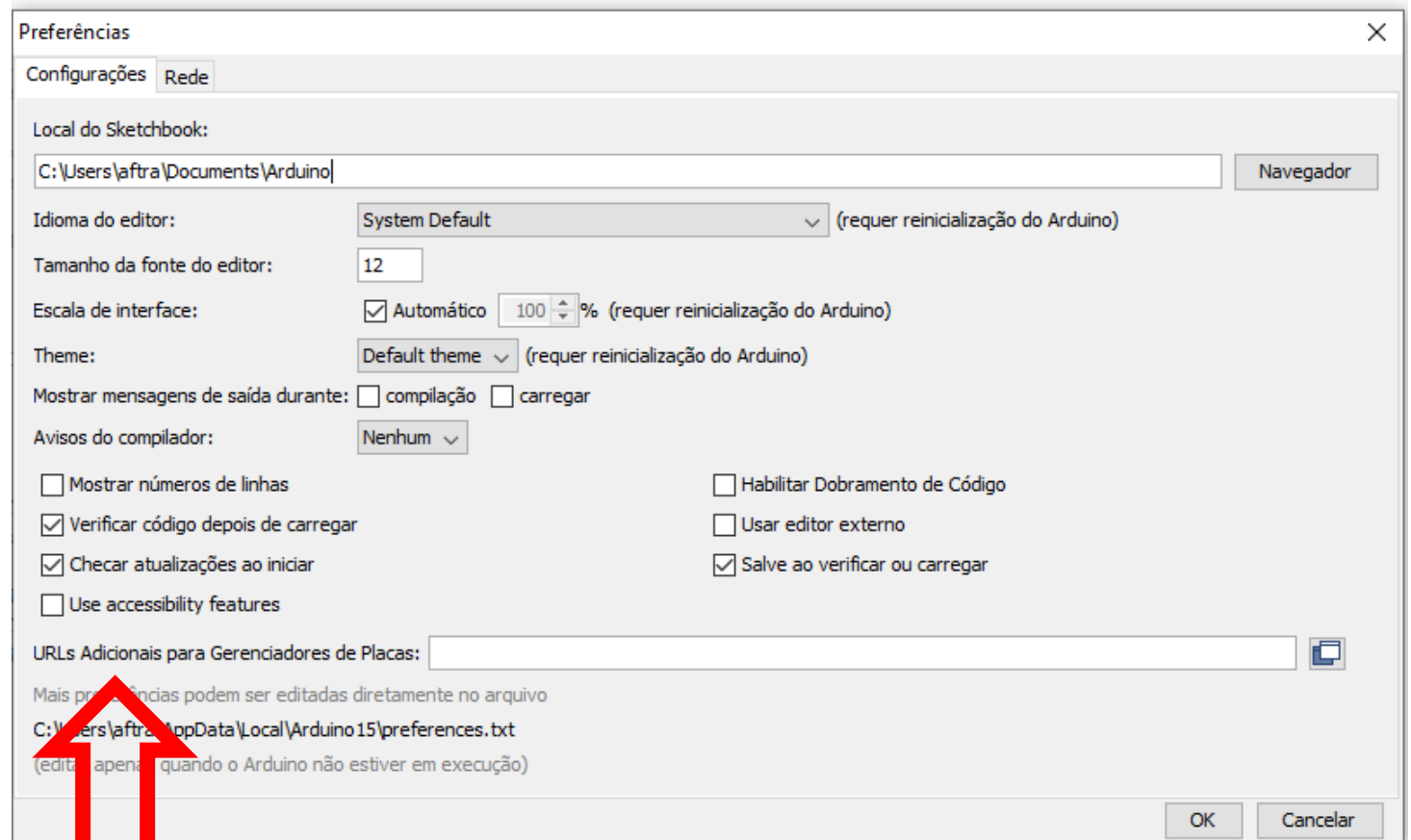
A primeira coisa a se fazer é abrir a sua Arduino IDE. Depois vá em **Arquivo->Preferências:**



Assim...

- Procure pelo campo **URLs adicionais de Gerenciadores de Placas:** e insira o link abaixo:

http://arduino.esp8266.com/stable/package_esp8266com_index.json



Aqui



Configurações Rede

Local do Sketchbook:

C:\Users\fernando\Documents\Arduino

Navegador

Idioma do editor:

System Default

(requer reinicialização do Arduino)

Tamanho da fonte do editor:

12

Escala de interface:

 Automático

100%

(requer reinicialização do Arduino)

Theme:

Default theme

(requer reinicialização do Arduino)

Mostrar mensagens de saída durante:

 compilação carregar

Avisos do compilador:

Nenhum

 Mostrar números de linhas Habilitar Dobramento de Código Verificar código depois de carregar Usar editor externo Checar atualizações ao iniciar Salve ao verificar ou carregar Use accessibility features

URLs Adicionais para Gerenciadores de Placas:

Mais preferências podem ser editadas diretamente no arquivo

C:\Users\fernando\AppData\Local\Arduino15\preferences.txt

(editar apenas quando o Arduino não estiver em execução)

URLs Adicionais para Gerenciadores de Placas



Entre com URLs adicionais, uma por linha

**[http://arduino.esp8266.com/stable/
package_esp8266com_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)**

Clique para ver uma lista de URLs de suporte de placas não oficiais

OK

Cancelar

Local do Sketchbook:

C:\Users\marce\Documents\Arduino

Navegador

Idioma do editor:

Português (Brasil) (Português (Brasil))

(requer reinicialização do Arduino)

Tamanho da fonte do editor:

12

Escala de interface:

 Automático 100% (requer reinicialização do Arduino)Mostrar mensagens de saída durante: compilação carregar

Avisos do compilador:

Todos

- Mostrar números de linhas
- Habilitar Dobramento de Código
- Verificar código depois de carregar
- Usar editor externo
- Aggressively cache compiled core
- Checar atualizações ao iniciar
- Atualizar arquivos de sketch para nova extensão ao salvar (.pde -> .ino)
- Salve ao verificar ou carregar

URLs Adicionais para Gerenciadores de Placas: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Mais preferências podem ser editadas diretamente no arquivo

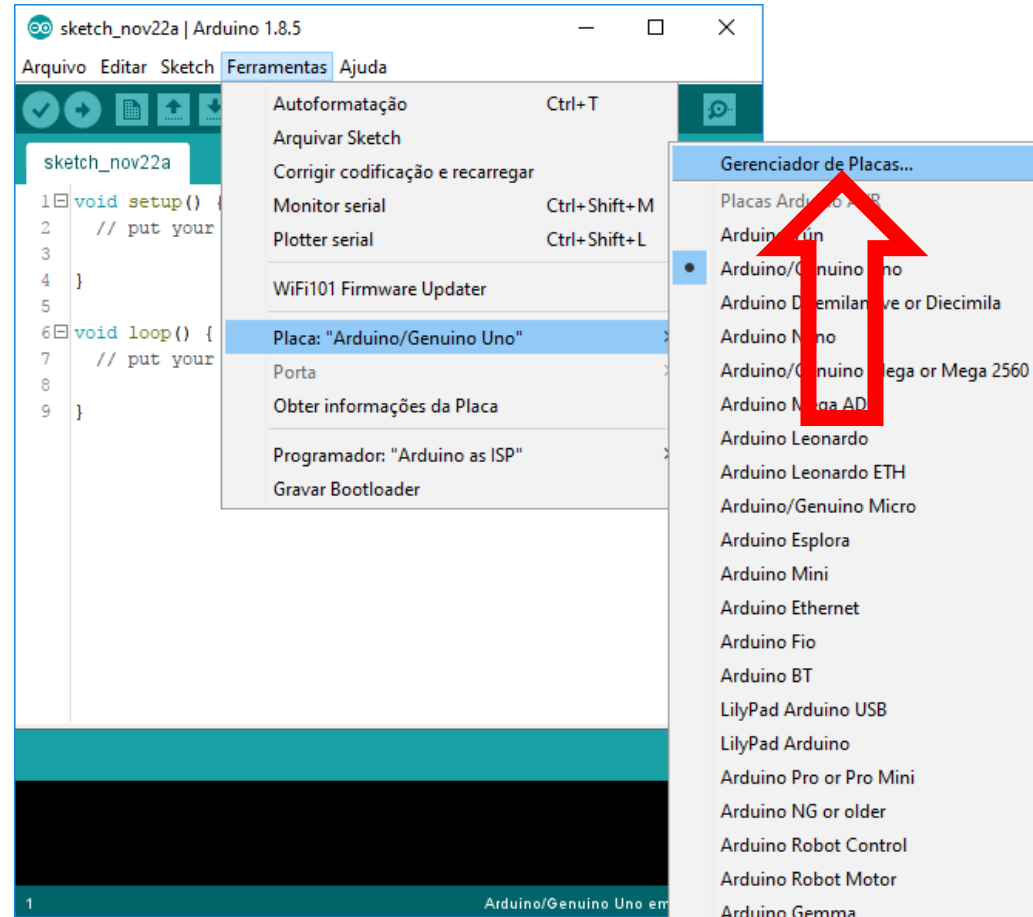
C:\Users\marce\AppData\Local\Arduino15\preferences.txt

(editar apenas quando o Arduino não estiver em execução)

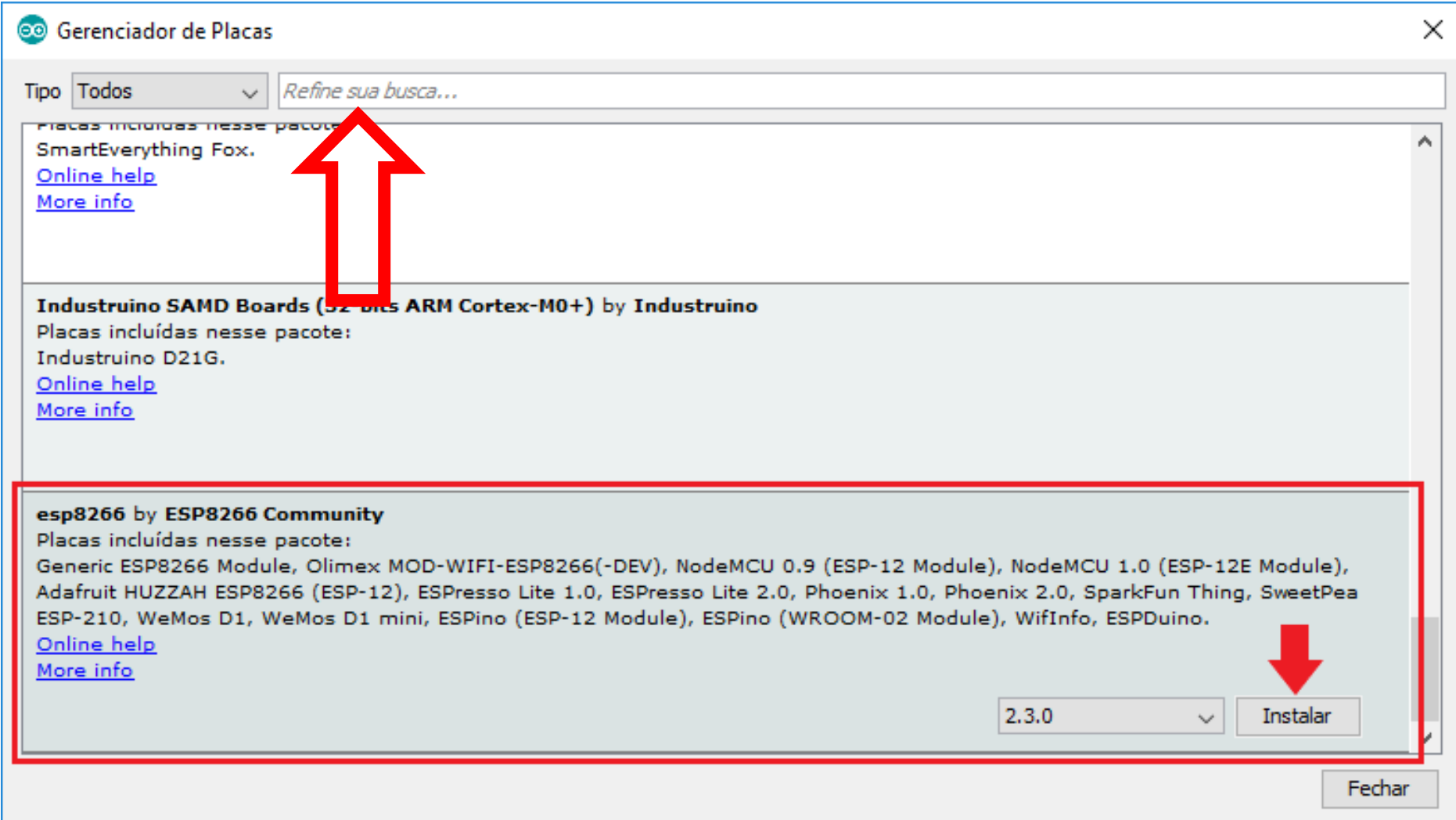
OK

Cancelar

Agora clique em **FERRAMENTAS -> PLACA -> GERENCIADOR DE PLACAS:**



Busque pela opção esp8266 by ESP8266 Community e clique no botão Instalar.



The screenshot shows the 'Gerenciador de Placas' (Board Manager) window in the Arduino IDE. The window title is 'Gerenciador de Placas'. At the top, there is a search bar with the text 'Refine sua busca...'. Below the search bar, there are three package entries:

- SmartEverything Fox.** (with links for [Online help](#) and [More info](#))
- Industruino SAMD Boards (32-bits ARM Cortex-M0+) by Industruino** (with links for [Online help](#) and [More info](#))
- esp8266 by ESP8266 Community** (with links for [Online help](#) and [More info](#))

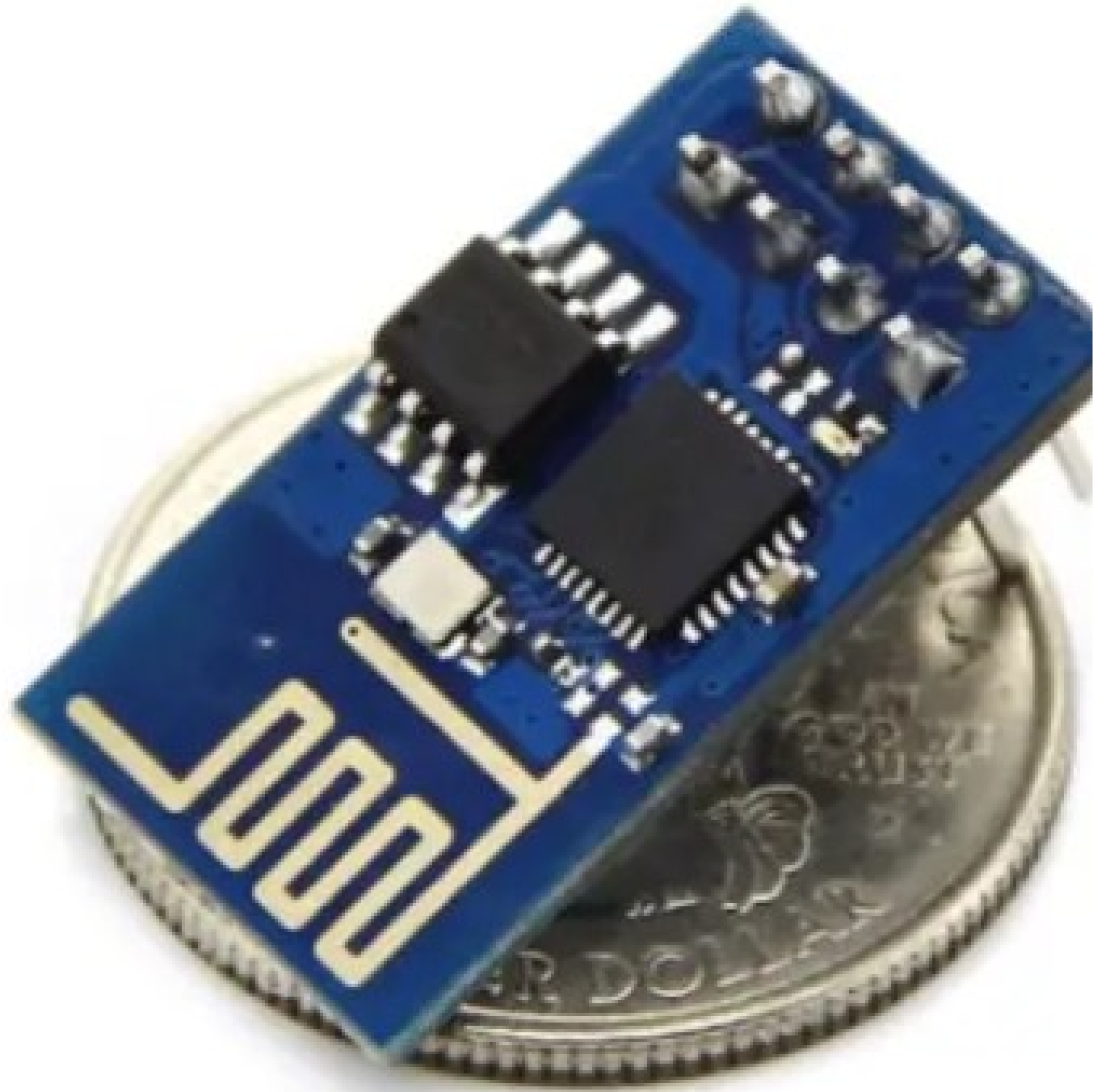
The 'esp8266 by ESP8266 Community' package is highlighted with a red box. Below the package name, it lists the boards included in the package: Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, and ESPDuino. At the bottom right of the package entry, there is a version dropdown menu showing '2.3.0' and an 'Instalar' button. A red arrow points to the 'Instalar' button. At the bottom right of the window, there is a 'Fechar' button.

```
/*
 ESP8266 Blink by Simon Peter
 Blink the blue LED on the ESP-01 module
 This example code is in the public domain
 The blue LED on the ESP-01 module is connected to GPIO1
 (which is also the TXD pin; so we cannot use Serial.print() at the same time)
 Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
 */
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage
level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000); // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
  delay(2000); // Wait for two seconds (to demonstrate the active
low LED)
}
```

MÓDULO Wireless ESP8266



Módulo Wireless ESP8266



Linguagens:

C

- Espressif SDK
- Arduino

Python

- MicroPython

Lua

- NodeMCU
- LuaNode

Basic

- Esp8266 BASIC
- Zbasic for ESP8266

JavaScript

- Espruino

Módulo Wireless ESP8266

Com o Módulo Wireless ESP8266 você pode conectar o Arduino nas redes wireless 802.11 b/g/n, enviando e recebendo dados nos modos AP (Access Point/Ponto de acesso) e STA (Station).

Módulo Wireless ESP8266

Esse módulo se comunica com o microcontrolador utilizando interface serial e seu firmware pode ser atualizado, se necessário.

Possui ainda 2 pinos GPIO (General Purpose Input Output, ou Entrada e Saída de uso geral), permitindo que o módulo seja programado diretamente e a GPIO acionada sem a necessidade de uso de um microcontrolador.



Características do **Modulo Wireless ESP8266:**

Conexão à redes padrão 802.11 B/G/N

Alcance aproximado: 91 metros

Tensão de operação : 3.3 VDC

Comunicação serial: pinos TX e RX

Modos de operação : Cliente, Access Point, Cliente+Access Point

Modos de segurança wireless : OPEN/WEP/WPA_PSK/WPA2_PSK/WPA_WPA2_PSK.

Suporta comunicação TCP e UDP, com até 5 conexões simultâneas

Características do **Modulo Wireless ESP8266:**

É um System-On-Chip (SOC) com Wi-Fi embutido;

Tem conectores GPIO, barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura;

CPU que opera em 80MHz, com possibilidade de operar em 160MHz;

Arquitetura RISC de 32 bits;

32KBytes de RAM para instruções;

96KBytes de RAM para dados;

64KBytes de ROM para boot;

Possui uma memória Flash SPI Winbond W25Q40BVNIG de 512KBytes;

O núcleo é baseado no IP Diamand Standard LX3 da Tensilica;

Fabricado pela Espressif;

Características do **Modulo Wireless ESP8266:**

É um System-On-Chip (SOC) com Wi-Fi embutido;

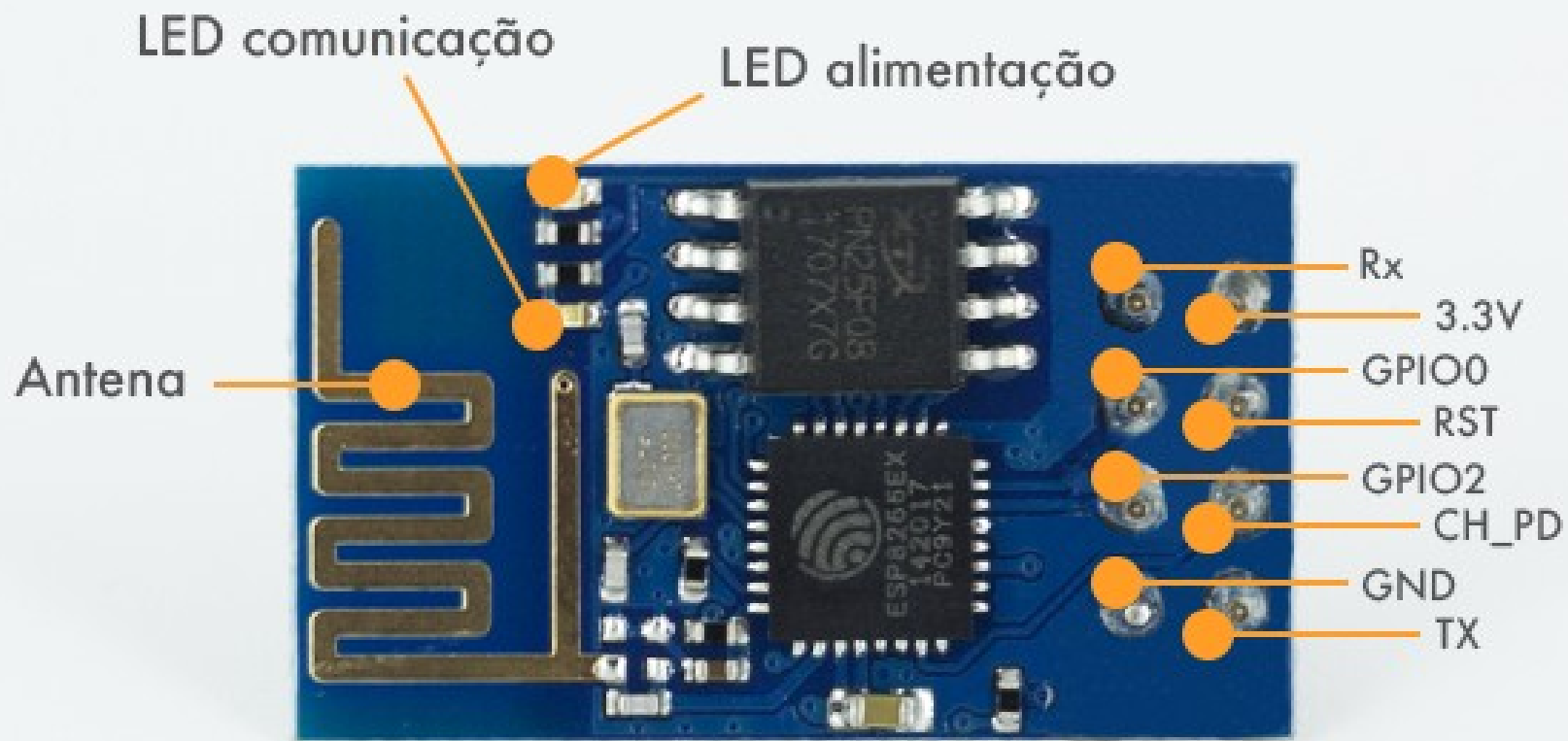
Tem conectores GPIO, barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura;

CPU que opera em 80MHz, com possibilidade de operar em 160MHz;

Arquitetura RISC de 32 bits;

32KBytes de RAM para instruções;

96KBytes de RAM para dados;



Ligação do módulo ESP8266 ao Arduino

1

Ligação do ESP8266 com o Arduino usa-se um resistor de 1K entre o Vcc (3.3V) e o pino **CH_PD** (Chip Enable).

2

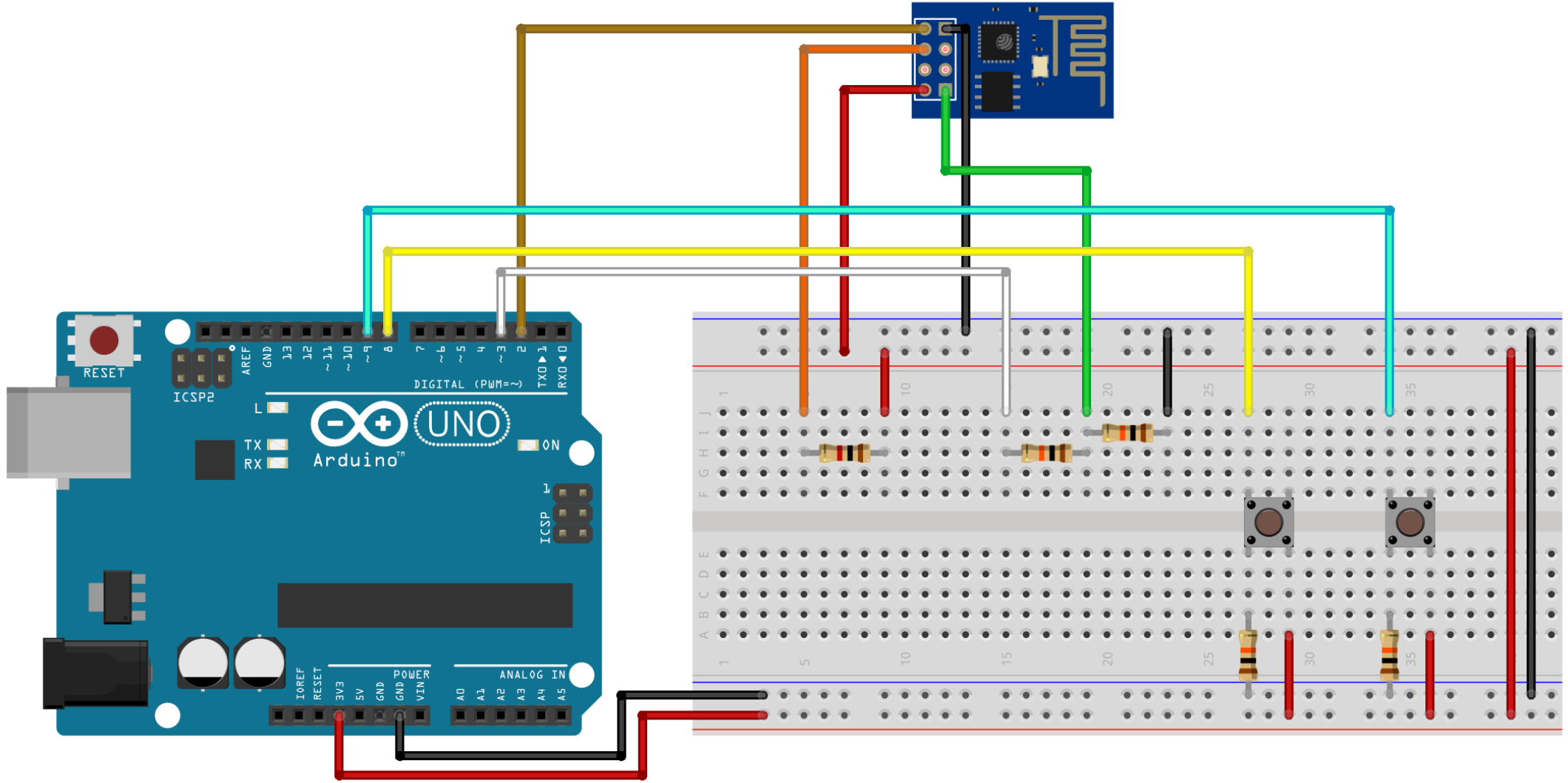
Para o divisor de tensão, utiliza-se 2 resistores de 10K, o que diminuiu a tensão do nível de sinal para um valor suficiente para os testes.

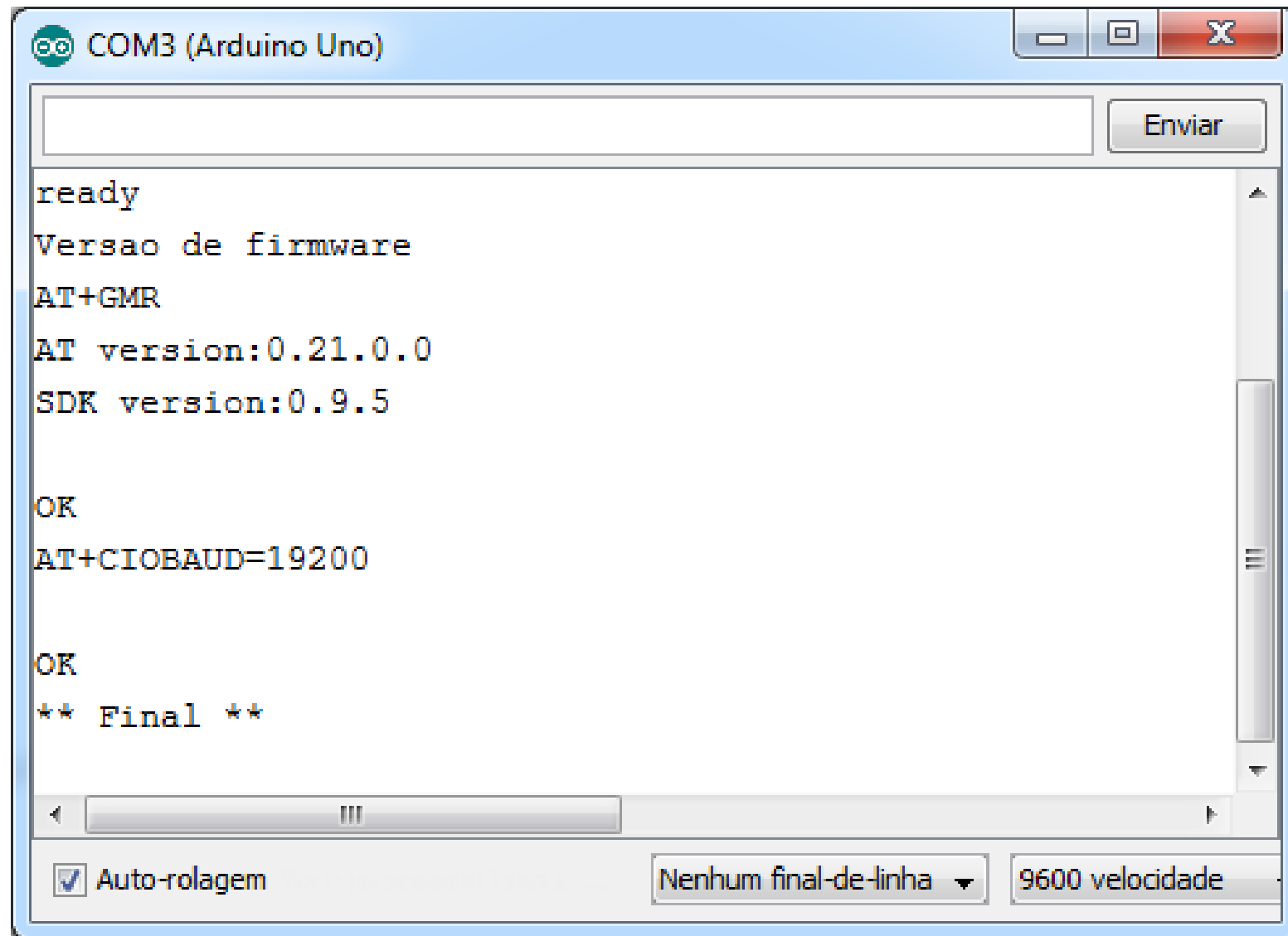
3

Os dois push-buttons do circuito serão utilizados para enviar informações à uma página web, utilizando um web server.

4

Os resistores utilizados nos botões também são de 10K.





```
// Programa: Versao firmware modulo ESP8266 e
//      mudanca de baud rate
// Autor : FILIPEFLOP


#include <SoftwareSerial.h>
//RX pino 2, TX pino 3
SoftwareSerial esp8266(2, 3);

#define DEBUG true
void setup()
{
  Serial.begin(9600);
  // Configure na linha abaixo a velocidade inicial do
  // modulo ESP8266
  esp8266.begin(115200);
  sendData("AT+RST\r\n", 2000, DEBUG);
  delay(1000);
  Serial.println("Versao de firmware");
  delay(3000);
  sendData("AT+GMR\r\n", 2000, DEBUG); // rst
  // Configure na linha abaixo a velocidade desejada para a
  // comunicacao do modulo ESP8266 (9600, 19200, 38400, etc)
  sendData("AT+CIOBAUD=19200\r\n", 2000, DEBUG);
  Serial.println("** Final **");
}
```


```
void loop() {}

String sendData(String command, const int timeout, boolean debug)
{
  // Envio dos comandos AT para o modulo
  String response = "";
  esp8266.print(command);
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (esp8266.available())
    {
      // The esp has data so display its output to the serial window
      char c = esp8266.read(); // read the next character.
      response += c;
    }
  }
  if (debug)
  {
    Serial.print(response);
  }
  return response;
}
```

No programa será utilizada a biblioteca **SoftwareSerial** para efetuar a comunicação com o módulo usando os pinos 2 (RX) e 3 (TX).



Assim, pode-se utilizar o serial monitor para acompanhar o envio dos comandos ao módulo.



Na linha 18 do programa, substitua as informações de SSID (nome da rede wireless) e SENHA pelas informações da rede à qual o módulo irá se conectar.

COM3 (Arduino Uno)

Enviar

```
AT+CWMODE=1  
no change  
AT+CIFSR  
192.168.0.27 → Endereço IP  
  
OK  
AT+CIPMUX=1  
  
OK  
AT+CIPSERVER=1,80  
  
OK
```

Auto-rolagem Nenhum final-de-linha ▼ 9600 velocidade

Para testar o funcionamento do circuito, abra um browser (Chrome, Firefox, etc) e digite na barra de endereços o endereço IP que foi mostrado no passo anterior, no caso deste exemplo 192.168.0.27:



<https://www.makerhero.com/blog/esp8266-arduino-tutorial/>

<https://espressif.com/>

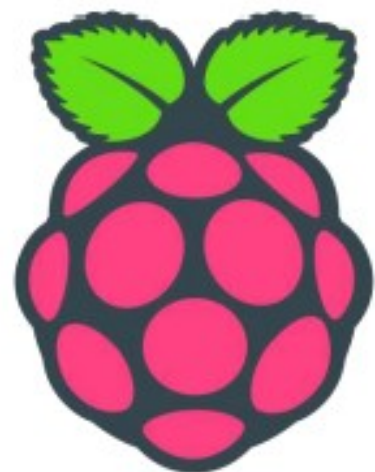
<https://www.embarcados.com.br/modulo-esp8266/>

Raspberry

Definição – Microcontroladores

- Computadores embutidos, às vezes chamados de microcontroladores [4] ou computadores em um único chip, podem ser considerados – essencialmente – dispositivos compostos por um processador dedicado com finalidade de executar uma aplicação específica, diferentemente de computadores de propósito geral. Para tal, esses dispositivos apresentam, em menor quantidade, os mesmos elementos de um computador reunidos em um único chip.

Sistemas Embarcados - Introdução



A forma de classificar os sistemas computacionais variaram com o tempo.



Até a **década 80** a classificação dos computadores **são relacionadas ao seu tamanho físico**. Exemplo:

- Computador - grandes instalações em CPD's (Ex: IBM370 e os DEC10);
- Mini-computadores – Utilizado em laboratórios acadêmicos e pequenas instituições (Ex: PDP11);
- Micro-computadores, que surgiam com os computadores pessoais, exemplo maior desta classe foram os IBM-PC.

Sistemas Embarcados - Introdução



Com a evolução do tempo, passou a prevalecer **outras formas de classificação**. Hoje em dia, segundo Patterson & Hennessy (2018), os computadores podem ser classificados como:

-
- **Computadores pessoais**, usados por um indivíduo;
 - **Servidores**, usados principalmente através de acesso remoto via rede, rodando múltiplas aplicações simultâneas;
 - **Supercomputadores**, são sistema com maiores desempenho, principalmente para aplicações científicas complexas.
 - **Computadores embarcados**, formam a maior classe de computadores e abrangem o mais largo campo de aplicações e desempenho. Em geral, trata-se de um computador dentro de outro dispositivo usado para rodar uma predeterminada aplicação ou software.



Computadores embarcados / Sistemas embarcados

A classe (**Computadores embarcados ou Sistemas embarcados**) que este curso irá tratar.

São projetados para rodar uma aplicação ou um conjunto de aplicações interrelacionadas, **normalmente integradas com o hardware**, e trabalham como um sistema único; assim, independente do grande número de computadores envolvidos, os usuários nunca perceberão que eles estão usando um computador. (Patterson & Hennessy, 2018)

Outra definição: Um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. Diferentemente de computadores de propósito geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos.

Sistemas Embarcados

Um sistema embarcado pode conter toda a funcionalidade sob a forma de software executado em processadores ou sob a forma de hardware especializado.

É utilizado como complemento ao software e hardware clássicos.

O projeto de sistemas embarcados normalmente exige o desenvolvimento de **software dependente de hardware** e **de hardware dependente de software**.





Sistemas Embarcados

As interfaces hardware-software necessitam de ser projetadas conjuntamente, requerendo **novas competências** que combinam o conhecimento de **hardware com o de software**. (INForum, 2019)

O computador embarcado está embutido em aplicações com pouca ou nenhuma interação com o ser humano.

Se por um lado isto simplifica a programação da interface humano-máquina, por outro lado, impõe diversos requisitos, como por exemplo, de temporização, de confiabilidade e de segurança.

Conhecendo a plataforma Raspberry PI



O Raspberry foi inicialmente planejado para ser um computador exatamente do tamanho de um Cartão de Crédito:

Raspberry

Desde a primeira versão, ele vem evoluindo e ganhando muito poder de processamento, e em sua versão 4, já está bem próximo de um computador "normal".

Porém ainda necessita de um pouco de entusiasmo para conseguir formatar e operar, apesar de ser simples.



YouTube

[https://
www.youtube.com/
watch?v=C70EfRTgr-
g](https://www.youtube.com/watch?v=C70EfRTgr-g)

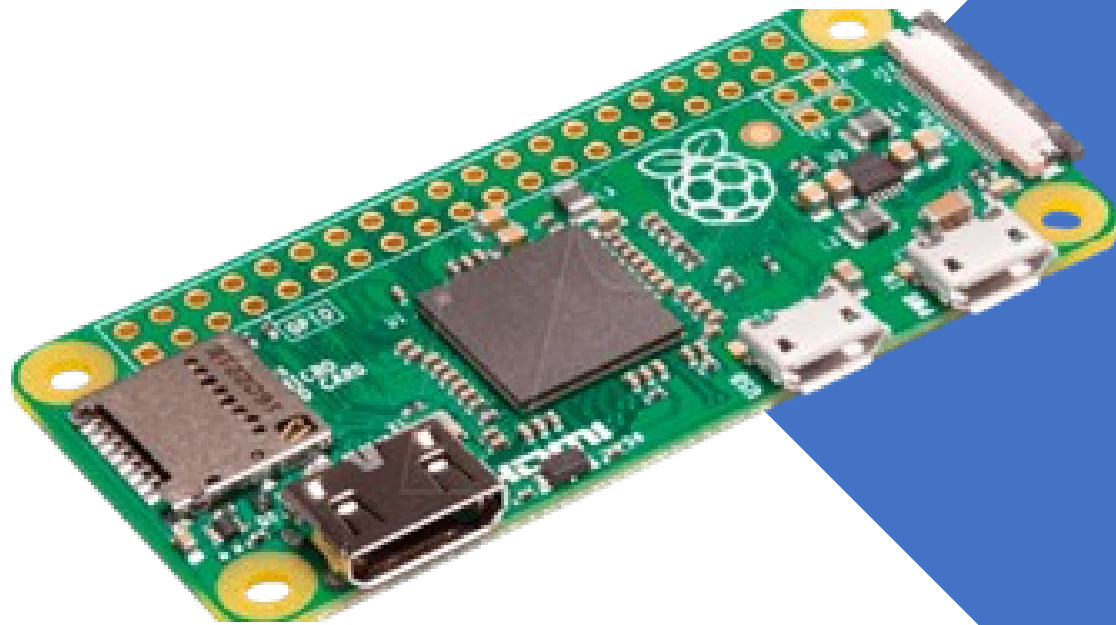
Conhecendo outros tipos de Raspberry



PI 400

- Um teclado com o Raspberry pi 4 embutido. Com os pinos de IO (Entrada e saída)

Conhecendo outros tipos de Raspberry



RASPBERRY PI ZERO

- O menor Raspberry, caçula da família.

Alguns modelos de Raspberry



Raspberry
Pi Zero W



Raspberry
Pi 3 A+



Raspberry
Pi 3 B+



Raspberry
Pi 4 B



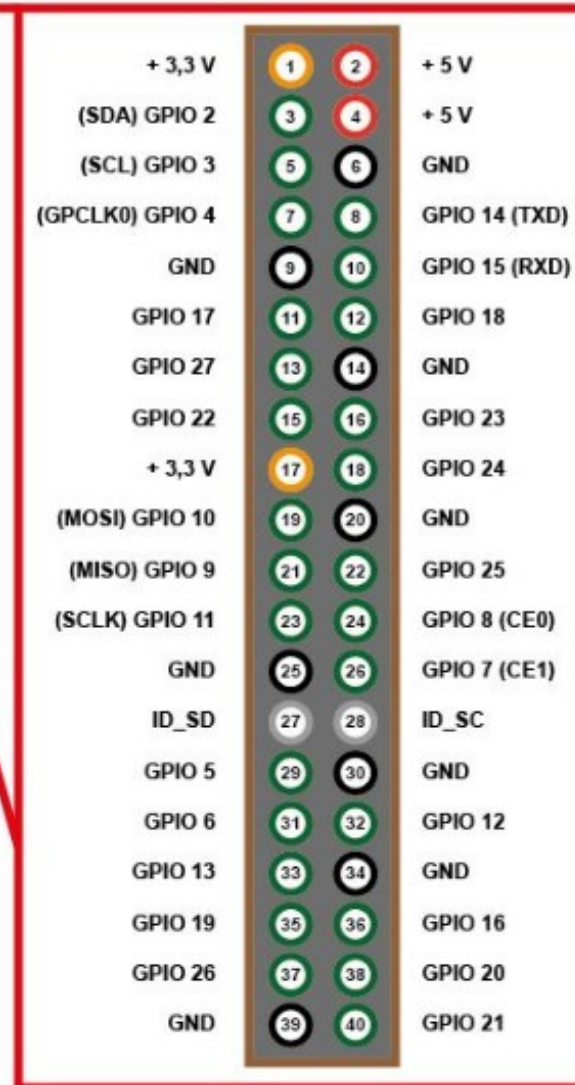
Com case de
proteção e
refrigeração

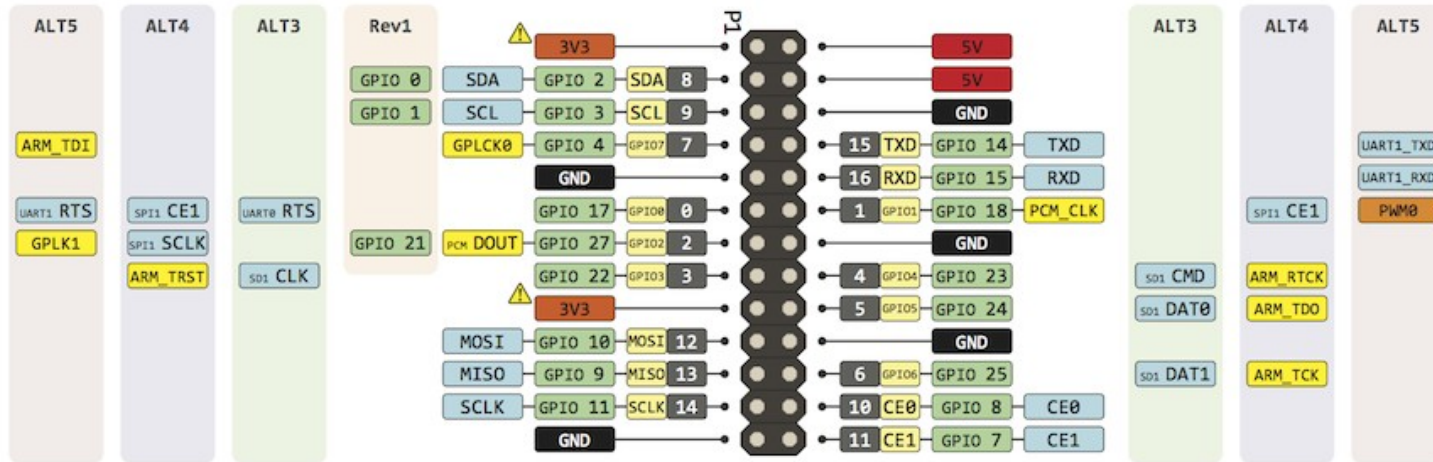


Você pode transformar seu Raspberry em um Notebook com um kit, de nome PI-TOP

O case que funciona como corpo do laptop agora traz uma tela melhor, de 14 polegadas e resolução Full HD, bateria com carga suficiente para oito horas de uso, além de teclado em padrão norte-americano e touchpad.







FIM

