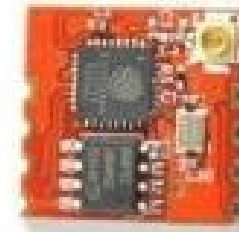


Introdução ao ESP8266 NodeMCU



ESP-01



ESP-02



ESP-03



ESP-04



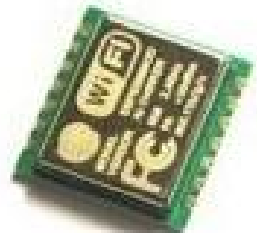
ESP-05



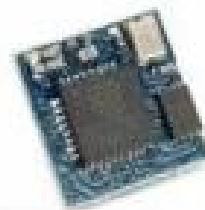
ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12E



ESP-13

Introdução ao ESP8266 NodeMCU

- ESP8266 é um microcontrolador (MCU) de baixo custo com conexão Wi-Fi desenvolvido pela Espressif Systems.
- Pode ser utilizado em uma aplicação isolada (standalone) ou como um componente para outro microcontrolador conectar ao Wi-Fi.
- Pode tornar seu projetos online, ideal para automações e lots.
- Baixo custo: a partir de 1 dollar
- Baixo consumo: baixíssimo consumo comparado com outros MCUs e possui deep sleep mode.
- Linguagem nativa: eLua - Embedded LUA.
- Compatível com Arduino IDE, MicroPython e outras linguagens (SDK).

Introdução ao ESP8266 NodeMCU

- Wi-Fi:
 - pode ser configurado como roteador (access point) ou se conectar a uma rede (estação).
 - pode ser usado para conectar serviço online (HTTP)
 - subir dados para a nuvem (MQTT)
 - pode ser um servidor de web e controlar componentes pelo navegador web.

Comparação

	ESP8266	ESP32	Arduino UNO
Corrente	197mA	220mA	40mA
Núcleo	1	2	1
Arquitetura	32 bits	32 bits	8 bits
Clock	80 – 160 MHz	160-240 MHz	16MHz
Bluetooth	Não	Clássico e BLE (Bluetooth Low Energy)	Não
WiFi	Sim	Sim	Não
RAM	160KB	520KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	13	34	14
DAC	0	2	0
ADC	1	18	6
Interfaces	SPI, I2C, UART e I2S	SPI, I2C, UART, I2S e CAN	SPI, I2C e UART

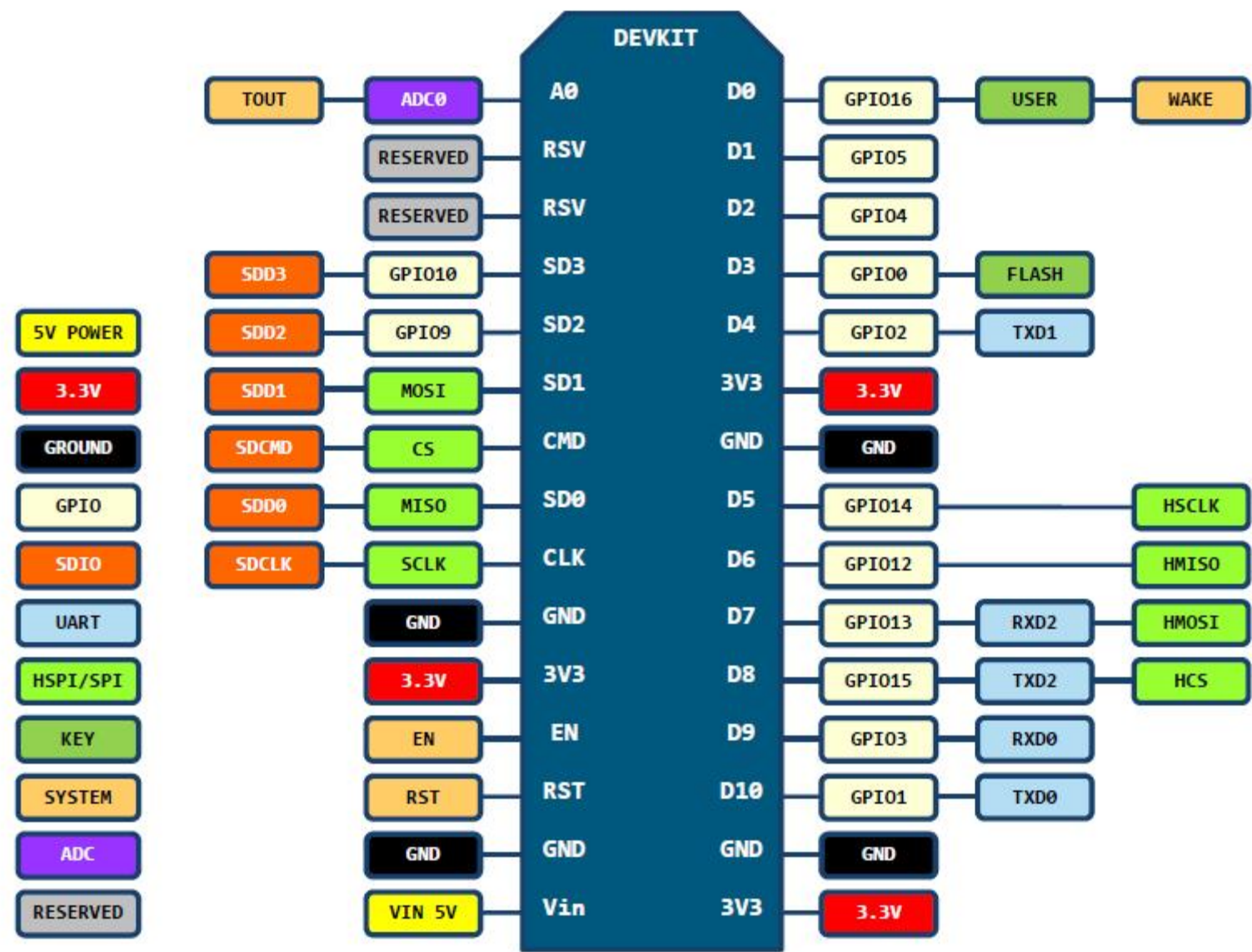
O que fazer com o ESP8266?

- Um web server para controlar equipamentos.
- Um servidor web para mostrar leituras de sensores.
- Enviar solicitações HTTP.
- controlar saídas e ler entradas como um microcontrolador comum.
- Projetos com leitura e controle de dados.
- Comunicação com outros componente e serviços.
- Criar aplicações web.
- enviar emails, notificações para web e aplicativos.

Links

- <https://www.espressif.com/>
- <https://github.com/esp8266>
- <https://arduino-esp8266.readthedocs.io/en/3.0.2/>
- <http://www.nodemcu.com/>
- <https://github.com/nodemcu>
- <https://en.wikipedia.org/wiki/ESP8266>

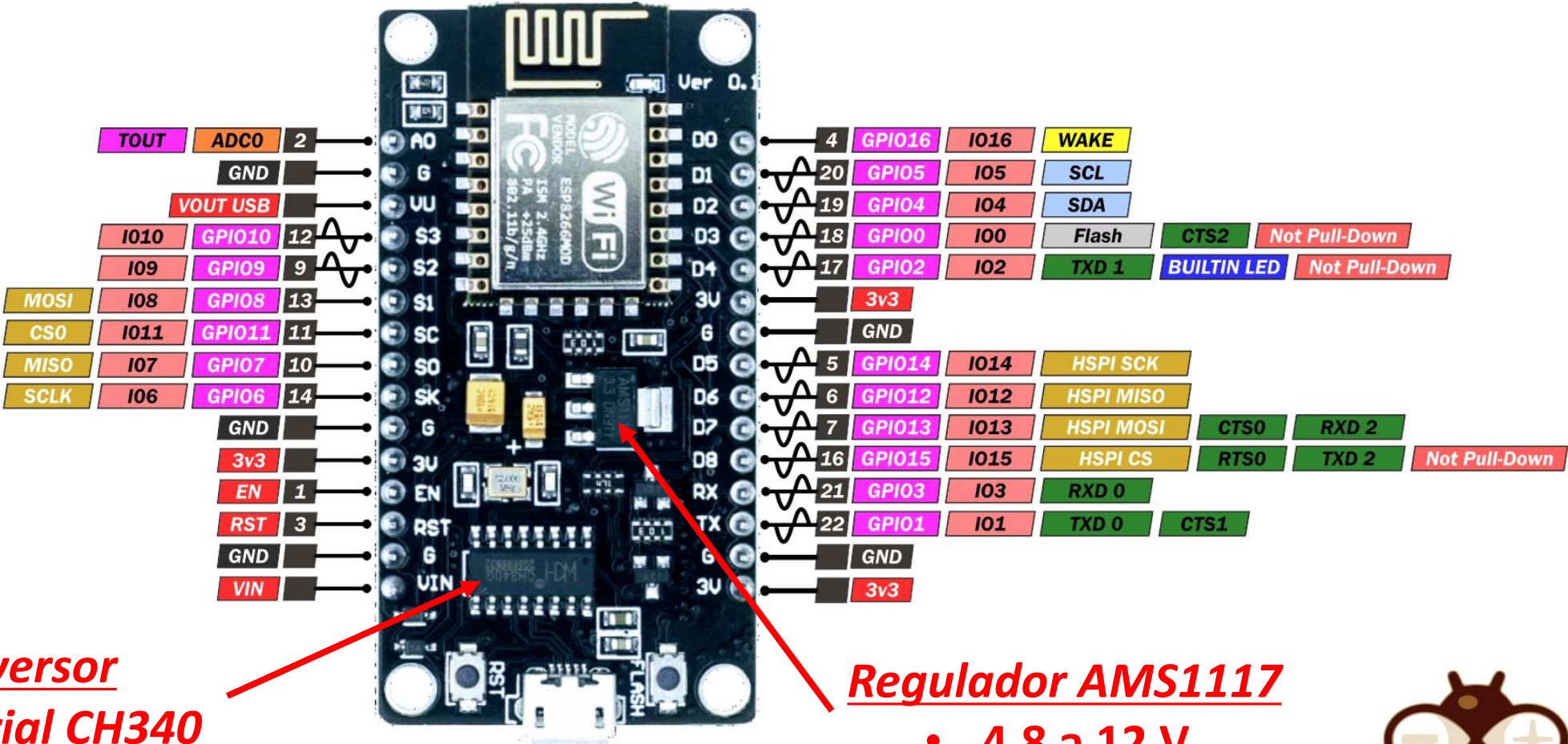
Pinagem (Pinout)



D0(GPI016) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

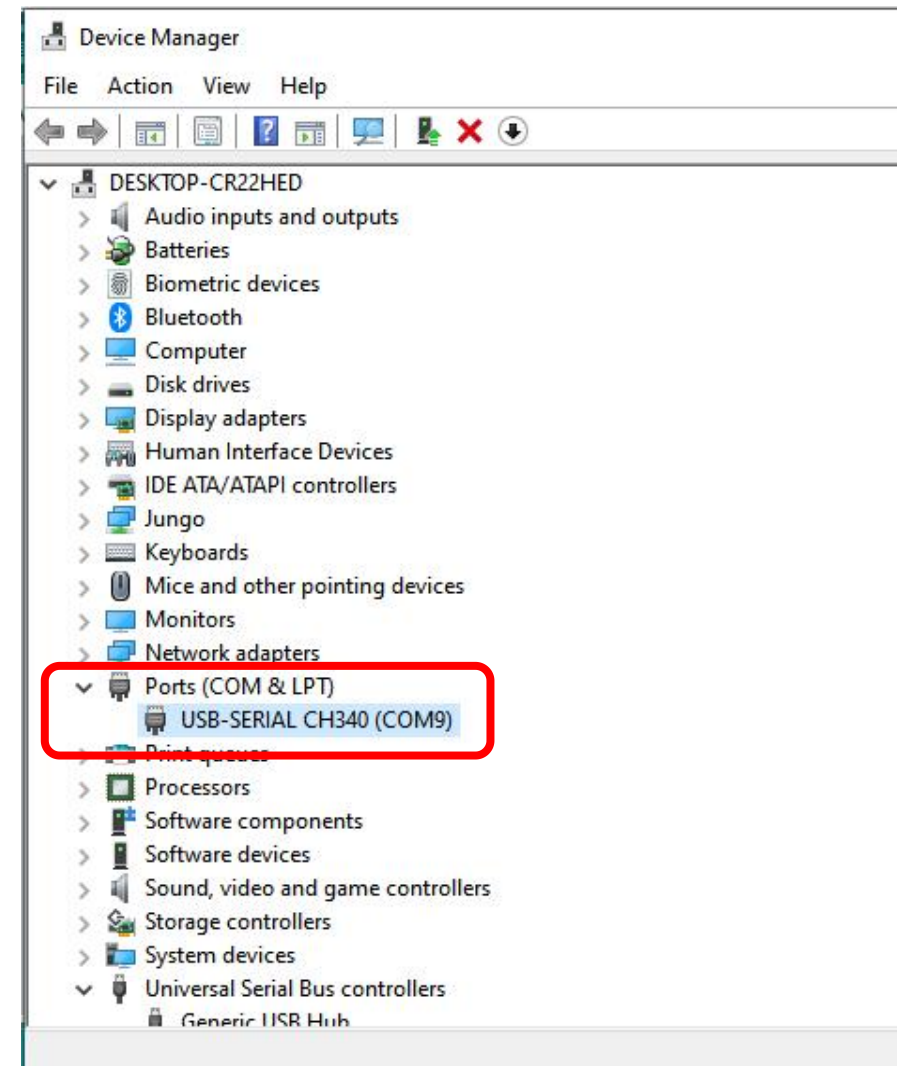
NodeMCU v3 CH340

PINOUT



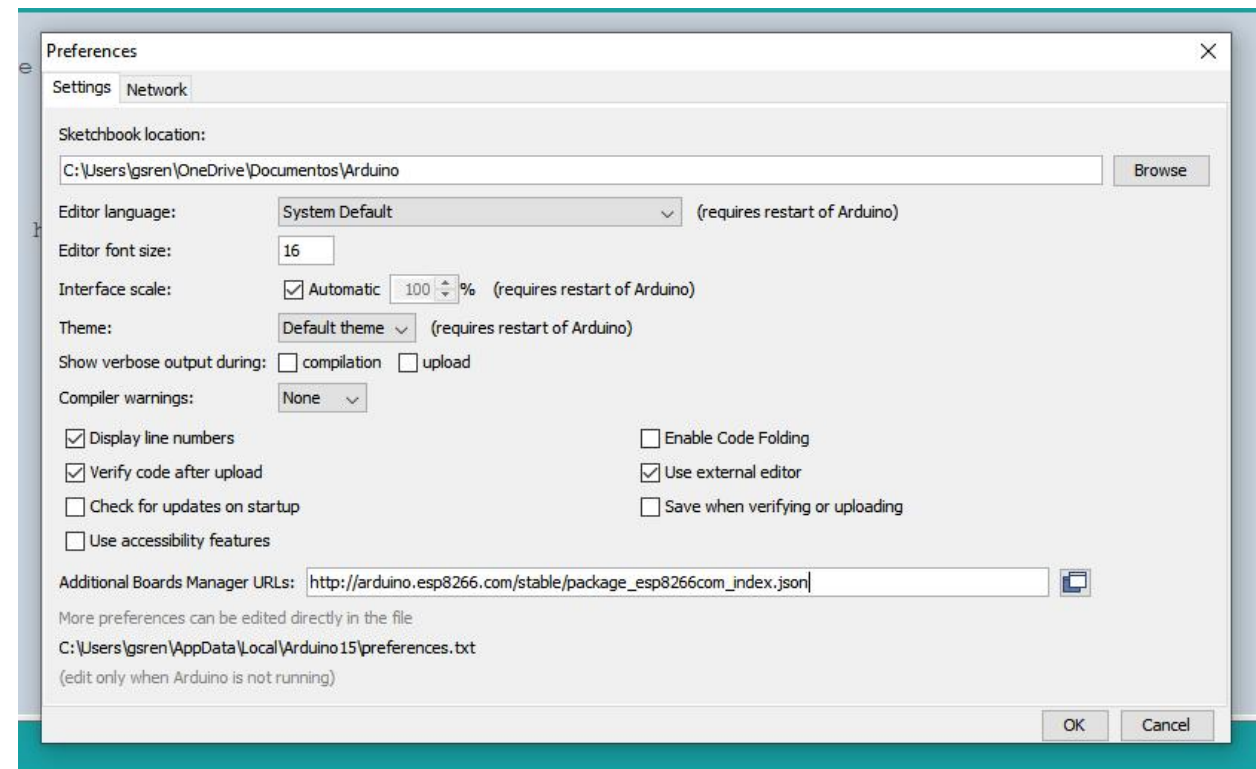
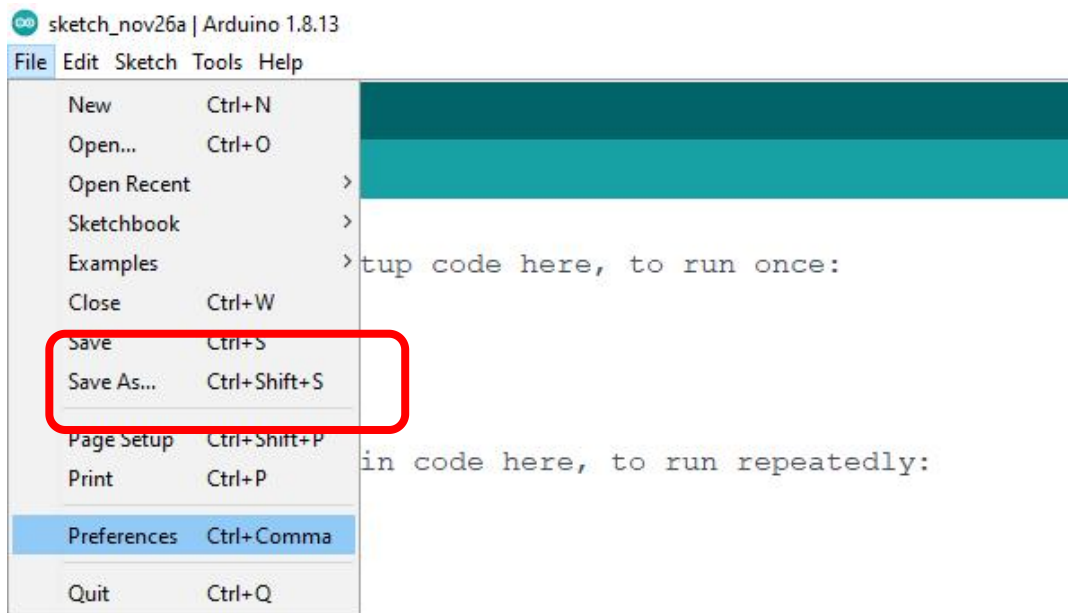
Instalando o driver CH340

- Conecte o ESP8266 na porta USB e veja se o pc reconheceu a porta USB.
- Se não estiver mostrando a porta COM, instale o driver CH340.



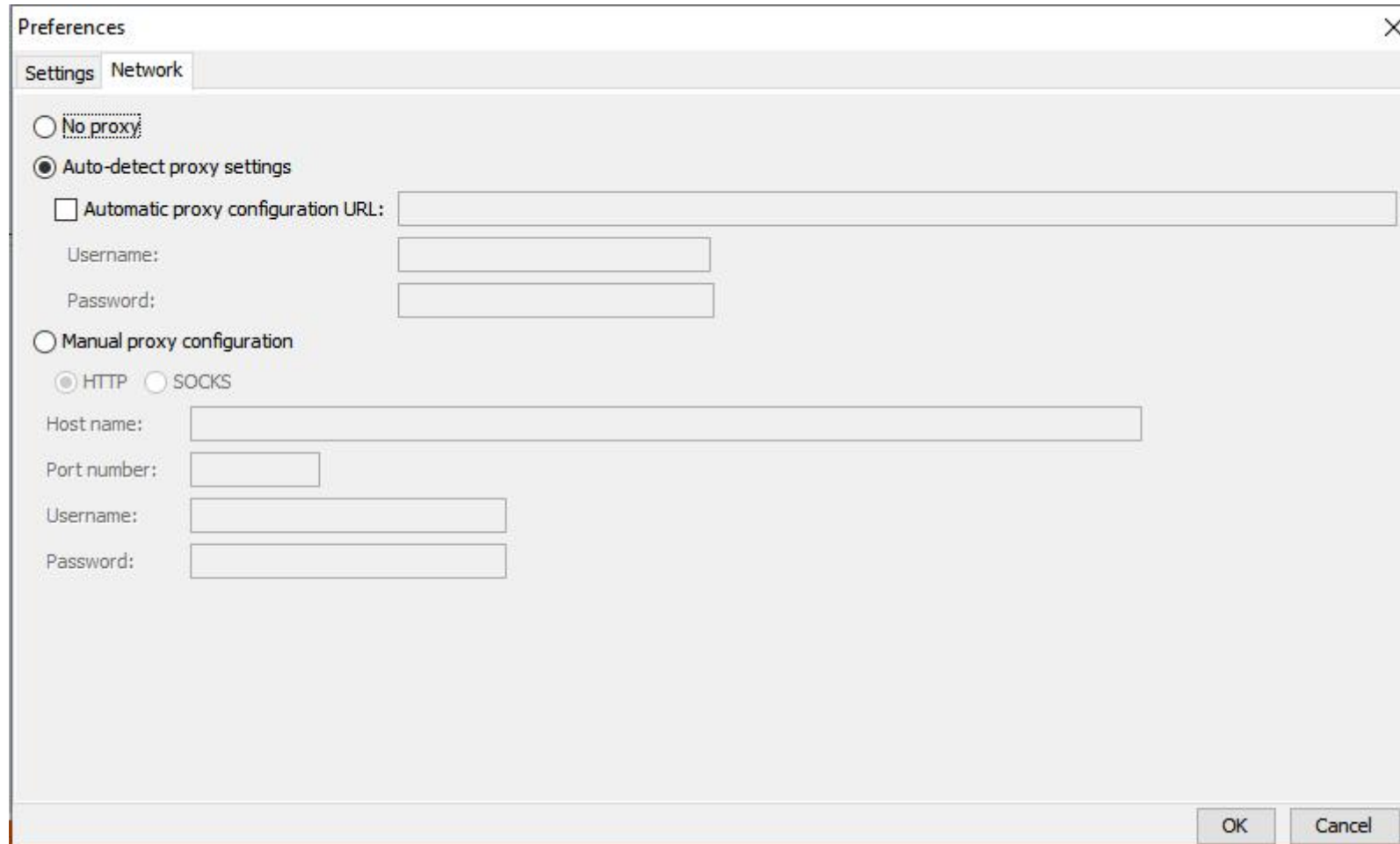
Instalando o ESP8266 no Arduino IDE

- Abra o Arduino IDE.
- Cole o link na area de preferências:
 - http://arduino.esp8266.com/stable/package_esp8266com_index.json



Instalando o ESP8266 no Arduino IDE

- Na aba Preferencias > Network configure o proxy como auto-detect.



The screenshot shows the 'Preferences' dialog box in the Arduino IDE, with the 'Network' tab selected. The 'Auto-detect proxy settings' option is chosen, and the 'Automatic proxy configuration URL' checkbox is checked. The 'Manual proxy configuration' section is also visible, showing options for HTTP and SOCKS proxies.

Preferences

Settings Network

☐ No proxy

☒ Auto-detect proxy settings

☐ Automatic proxy configuration URL:

Username:

Password:

☐ Manual proxy configuration

☒ HTTP ☐ SOCKS

Host name:

Port number:

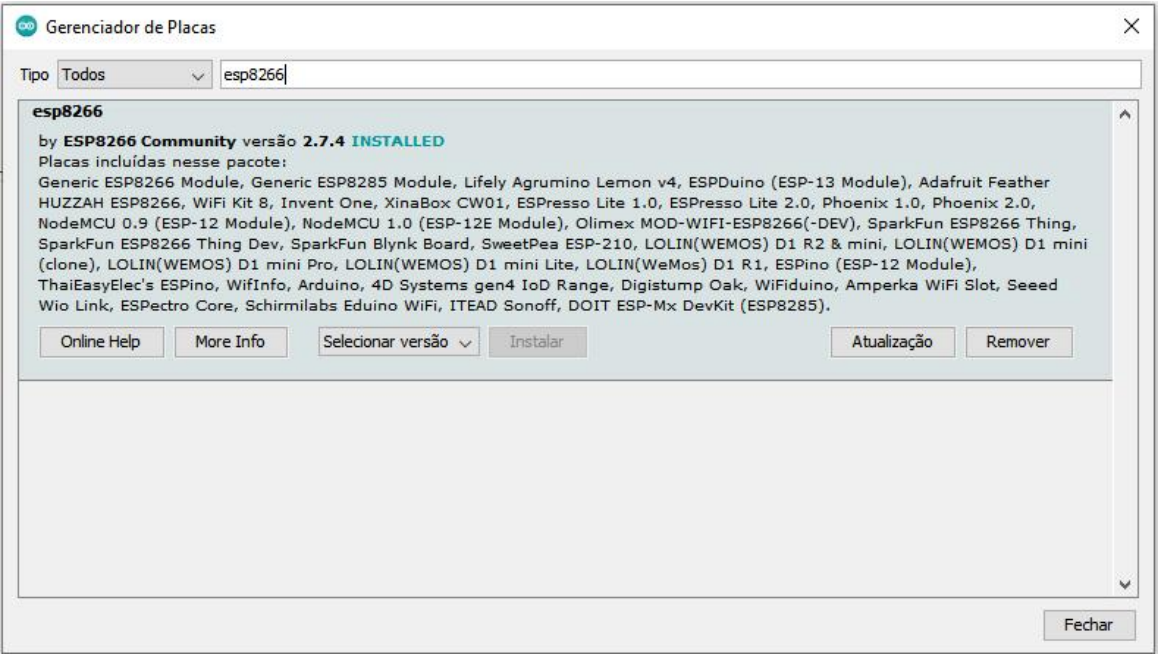
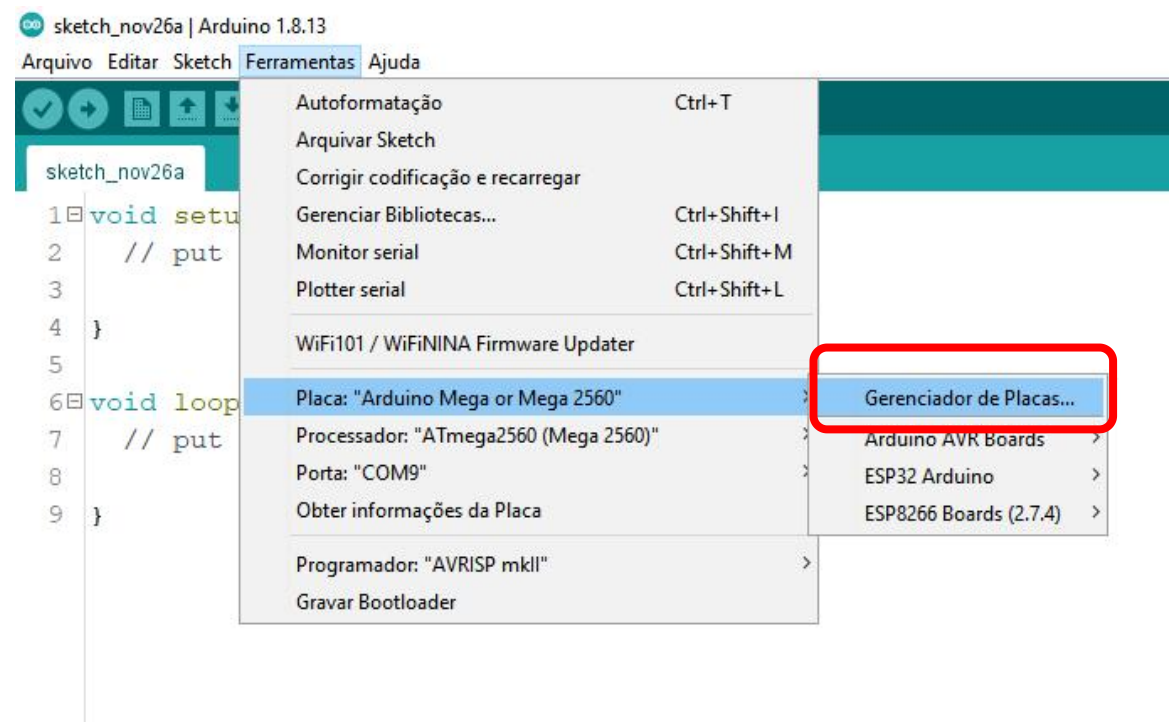
Username:

Password:

OK Cancel

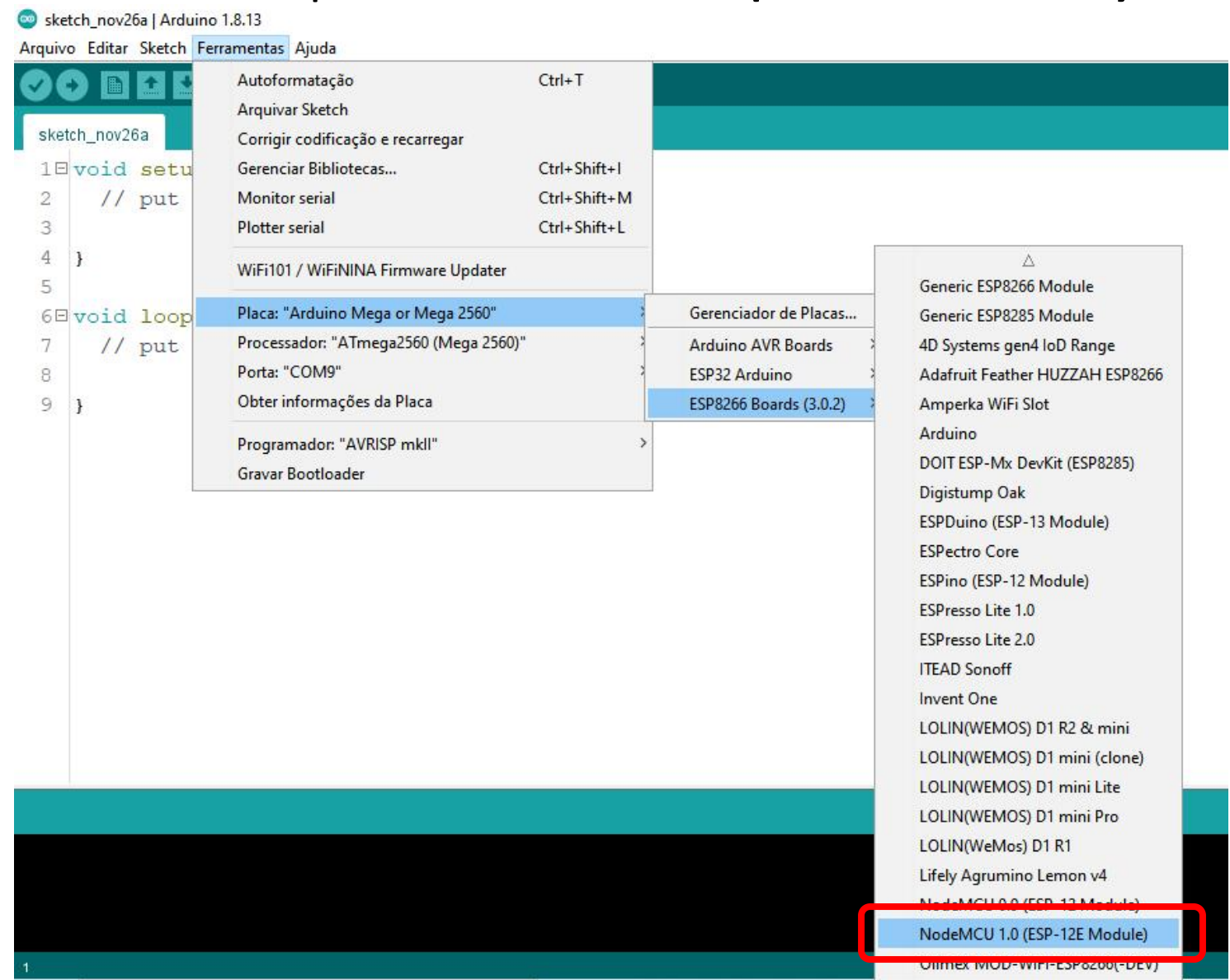
Instalando o SDK (software development kit) no Arduino IDE

- Na aba Ferramentas > placa > Gerenciador de Placas.
- Digite esp8266 na busca e instale a biblioteca.



Instalando o SDK (software development kit) no Arduino IDE

- Selecione a placa **NodeMCU 1.0 (ESP-12E Module)**



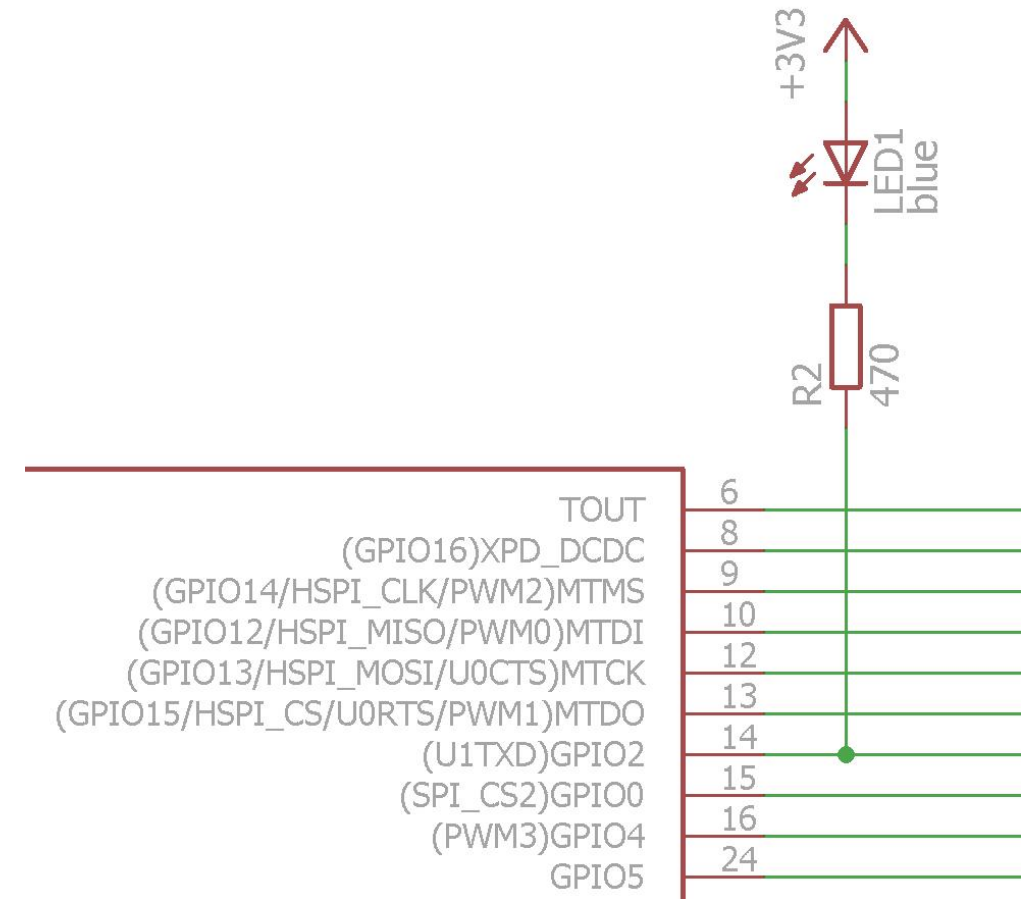
- Configuração

Autoformatação	Ctrl+T
Arquivar Sketch	
Corrigir codificação e recarregar	
Gerenciar Bibliotecas...	Ctrl+ Shift+I
Monitor serial	Ctrl+ Shift+M
Plotter serial	Ctrl+ Shift+L
WiFi101 / Wi-FiNINA Firmware Updater	
Placa: "NodeMCU 1.0 (ESP-12E Module)"	>
Builtin Led: "2"	>
Upload Speed: "115200"	>
CPU Frequency: "80 MHz"	>
Flash Size: "4MB (FS:2MB OTA:~1019KB)"	>
Debug port: "Disabled"	>
Debug Level: "Nenhum"	>
IwIP Variant: "v2 Lower Memory"	>
VTables: "Flash"	>
C++ Exceptions: "Disabled (new aborts on oom)"	>
Stack Protection: "Disabled"	>
Erase Flash: "Only Sketch"	>
SSL Support: "All SSL ciphers (most compatible)"	>
MMU: "32KB cache + 32KB IRAM (balanced)"	>
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"	>
Porta: "COM9"	>
Obter informações da Placa	>
Programador	>
Gravar Bootloader	

Primeiro programa: Blink

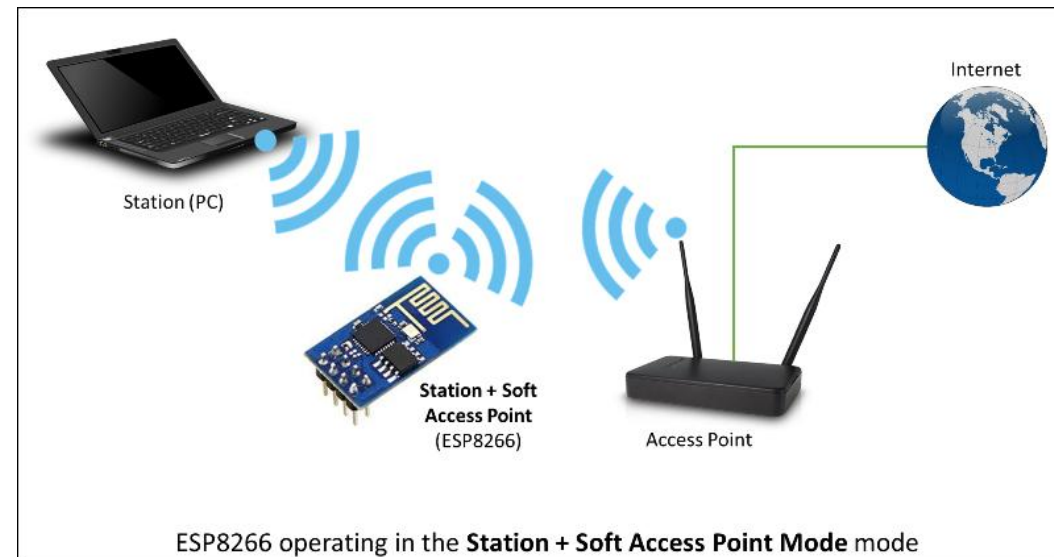
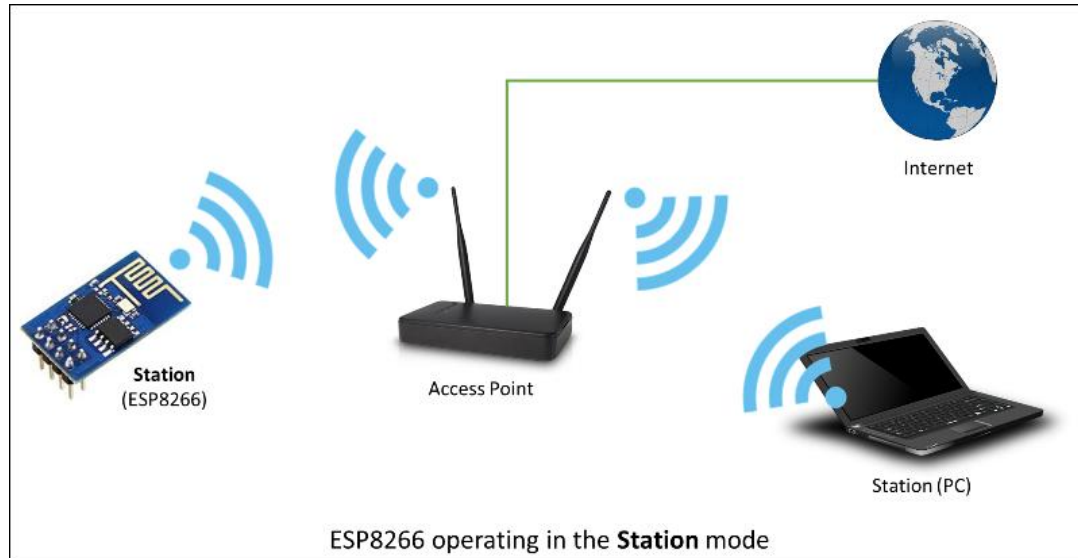
- O led da placa pode estar conectado ao GPIO 2 ou ao 16.
- O led é acionado com sinal baixo (LOW).

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
void loop() {  
  digitalWrite(LED_BUILTIN, LOW); // liga o led  
  delay(1000);  
  digitalWrite(LED_BUILTIN, HIGH); // desliga o led  
  delay(2000);  
}
```



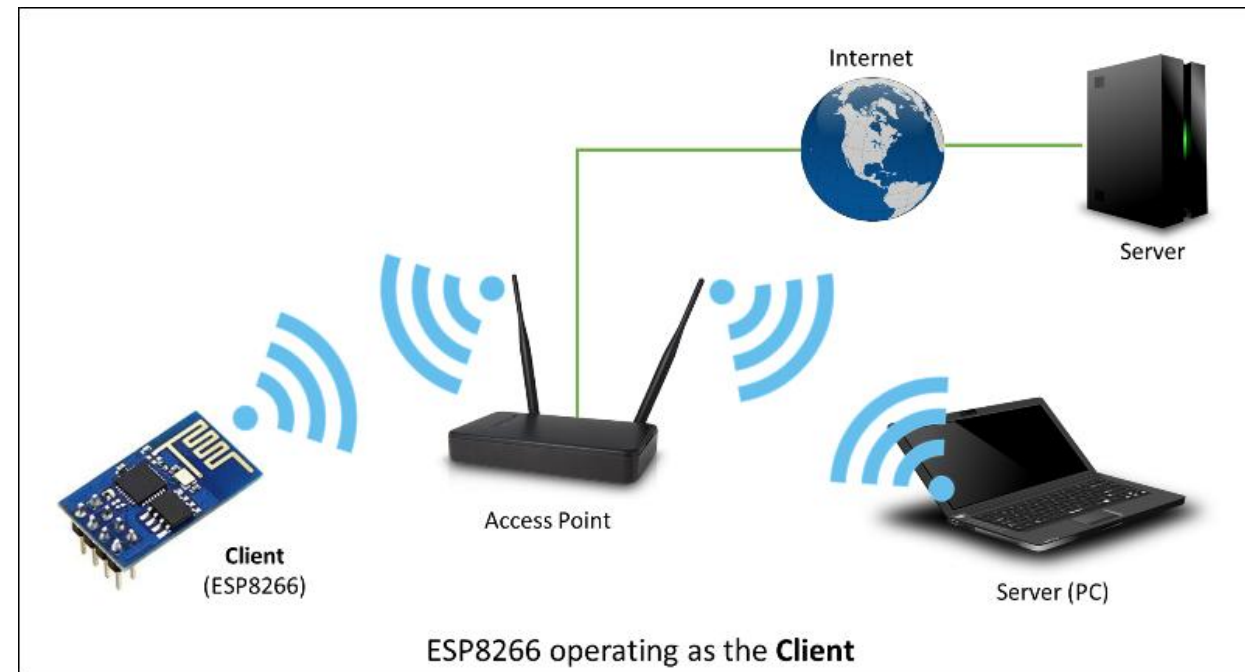
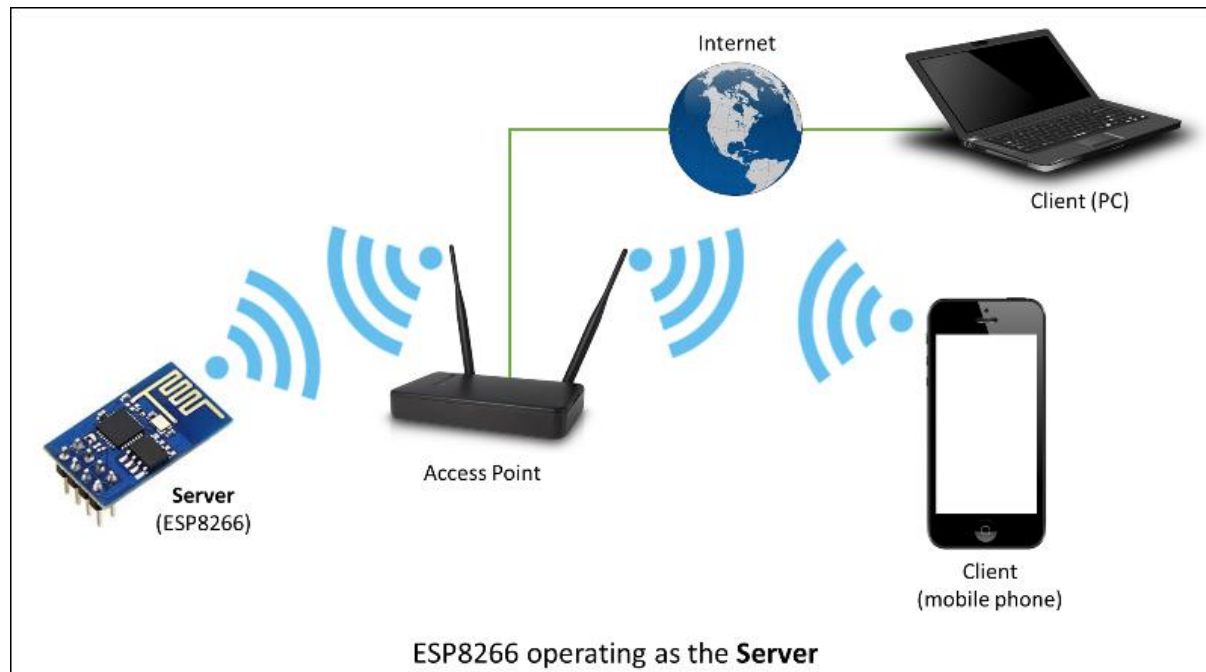
Conectando ao Wi-Fi

- O ESP8266 pode conectar à internet como Estação, Ponto de Acesso ou ambos.



Conectando ao Wi-Fi

- O ESP8266 também pode ser configurado como cliente ou servidor.



Conectando ao Wi-Fi

- Utilizando a biblioteca <ESP8266WiFi.h>
- A biblioteca é dividida em classes. Cada classe possui uma função específica.

B BufferDataSource BufferedStreamDataSource	E ESP8266WiFiAPClass ESP8266WiFiClass ESP8266WiFiGenericClass ESP8266WiFiMulti ESP8266WiFiScanClass ESP8266WiFiSTAClass	S SList SSLContext	WiFiClient WiFiClientSecure WiFiEventHandlerOpaque WiFiEventModeChange WiFiEventSoftAPModeProbeRequestReceived WiFiEventSoftAPModeStationConnected WiFiEventSoftAPModeStationDisconnected WiFiEventStationModeAuthModeChanged WiFiEventStationModeConnected	WiFiEventStationModeDisconnected WiFiEventStationModeGotIP WiFiServer WiFiUDP
C ClientContext		U UdpContext		
D DataSource	P ProgmemStream	W WifiAPlist_t		

Atividades

1. Escaneando as redes disponíveis.
2. Conectando à rede.
3. Controle de led Web