



Guia de Práticas - IoT

2018/2

Prof. Otávio

E-mail: schocair@gmail.com.br

Internet das Coisas (IoT)

- Atuação em dispositivos (HM)
- Monitoramento de dispositivos (HM)
- Comunicação entre máquinas (M2M)

Internet das Coisas (IoT)

- Indústria 4.0
- Comércio
- Residência (Domótica)

Micrcontroladores

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Fonte:

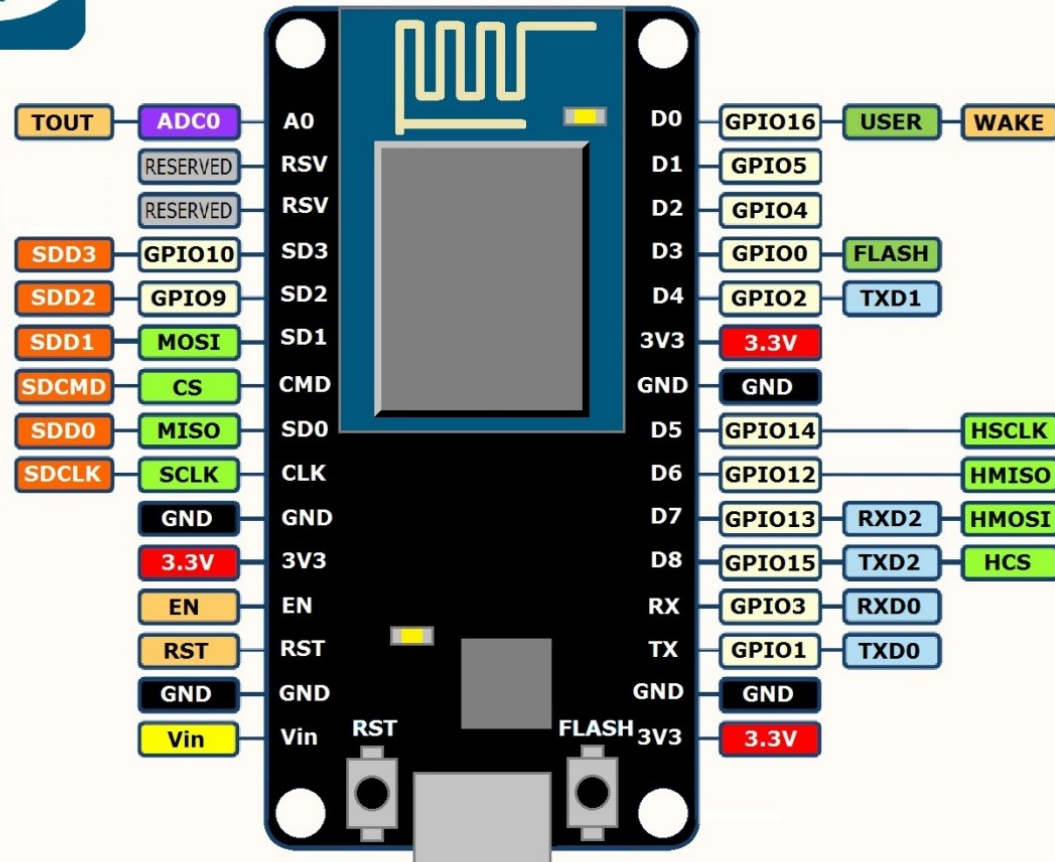
<http://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32/>

Pinagem ESP8266 NodeMCU



NodeMCU ESP-12 development kit V1.0

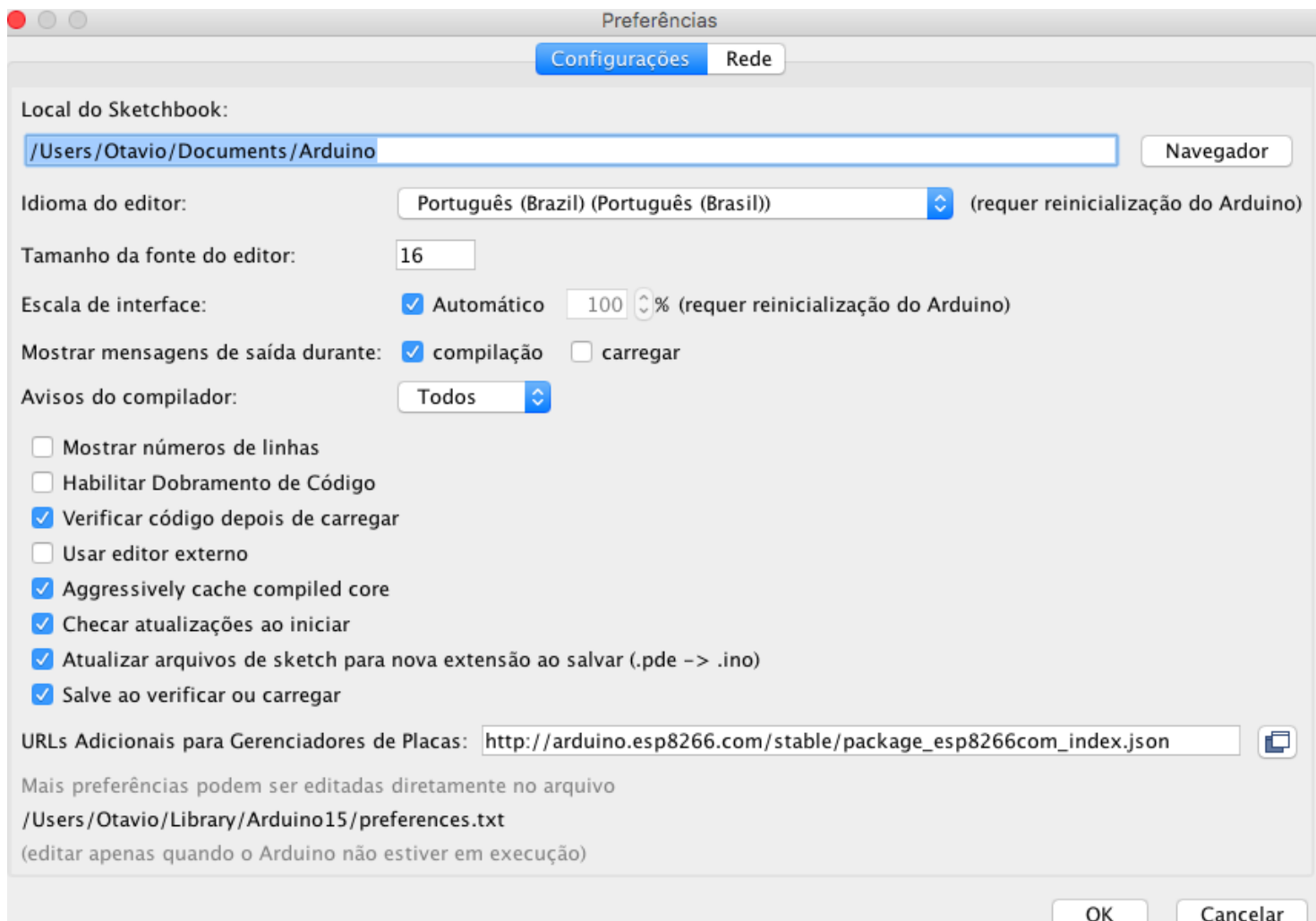
PIN DEFINITION



Configuração da IDE Arduino

- Na aba “Arduino” escolher “Preferências” inserir o link abaixo:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
Seguido do botão “OK”
- Depois na aba “Ferramentas” escolher “Placa” depois “Gerenciar placas” procurar por “esp8266” sem seguida instalar.

Configuração da IDE Arduino

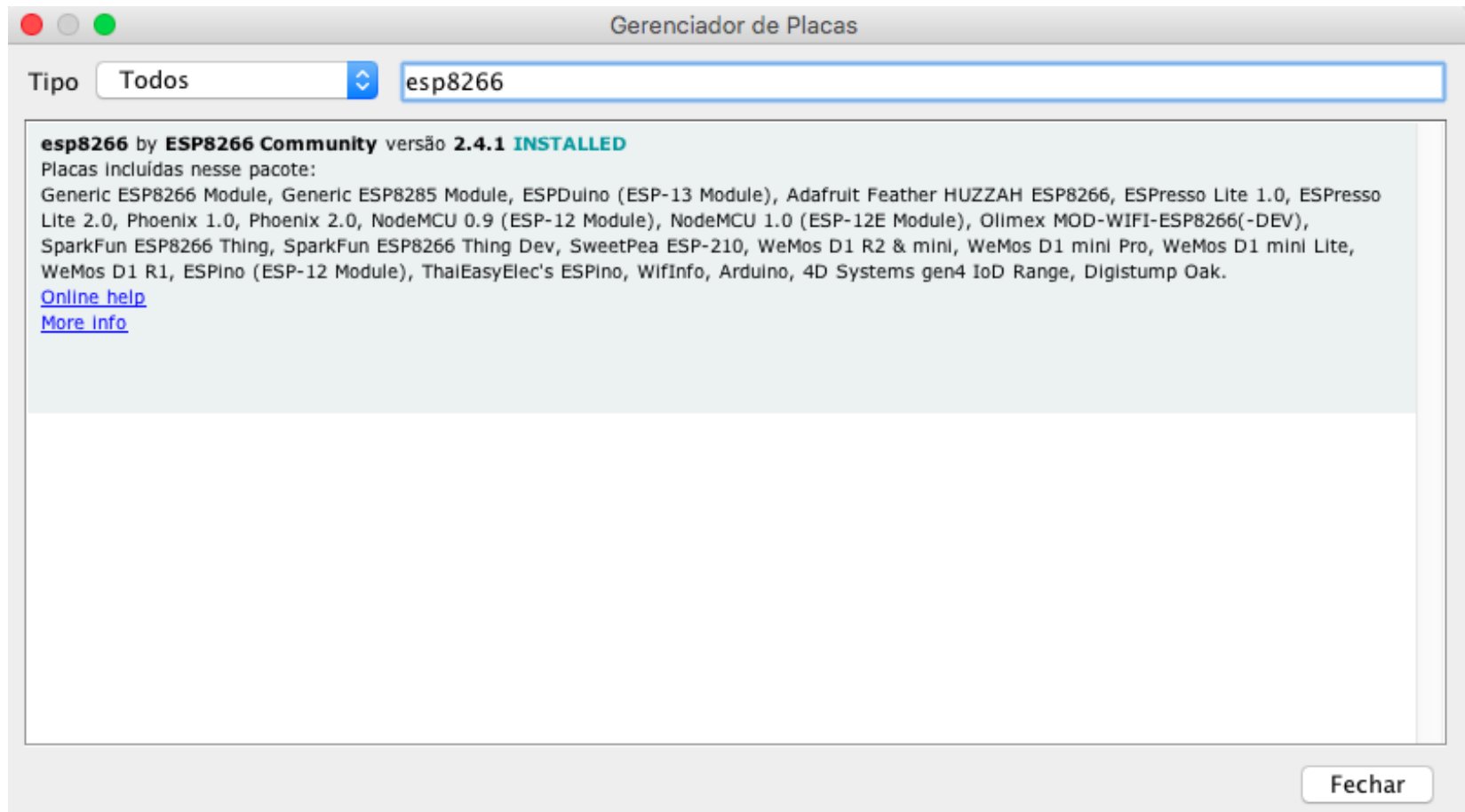


The image shows the 'Preferências' (Preferences) window of the Arduino IDE. The window has two tabs: 'Configurações' (selected) and 'Rede'. The 'Configurações' tab contains the following settings:

- Local do Sketchbook:** A text field containing '/Users/Otavio/Documents/Arduino' and a 'Navegador' button.
- Idioma do editor:** A dropdown menu set to 'Português (Brasil) (Português (Brasil))' with a note '(requer reinicialização do Arduino)'.
- Tamanho da fonte do editor:** A text field containing '16'.
- Escala de interface:** A checkbox for 'Automático' is checked, followed by a percentage field set to '100%' and a note '(requer reinicialização do Arduino)'.
- Mostrar mensagens de saída durante:** A checkbox for 'compilação' is checked, and a checkbox for 'carregar' is unchecked.
- Avisos do compilador:** A dropdown menu set to 'Todos'.
- Options:**
 - ☐ 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:** A text field containing 'http://arduino.esp8266.com/stable/package_esp8266com_index.json' and a file icon button.

At the bottom, there is a note: 'Mais preferências podem ser editadas diretamente no arquivo /Users/Otavio/Library/Arduino15/preferences.txt (editar apenas quando o Arduino não estiver em execução)'. The window has 'OK' and 'Cancelar' buttons at the bottom right.

Configuração da IDE Arduino



Clicar em “More info”, caso não apareça o botão de instalação e/ou atualização

Biblioteca WiFi – 1ª prática

```
#include <ESP8266WiFi.h> // Inclui a biblioteca
```

```
const char* SSID = "TESTE"; // SSID Nome da rede
```

```
const char* PASSWORD = "teste123"; // Senha
```

```
const char* myHostname = "Maq1"; // Nome do host
```

```
//Variáveis e objetos globais
```

```
WiFiClient espClient; // Cria o objeto espClient
```

```
WiFi.begin(SSID, PASSWORD) // Inicia a rede WiFi
```

```
WiFi.hostname(myHostname) // Informa o Host
```

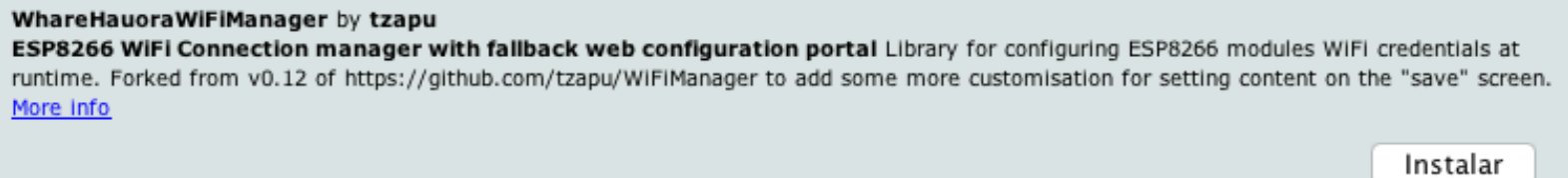
```
WiFi.status() != WL_CONNECTED // Testa conexão
```

```
WiFi.localIP() // Informa o IP obtido via DHCP
```

Biblioteca WiFiManager

2ª prática

Instalar a biblioteca WiFiManager by tzapu
 Na aba Sketch escolher “Gerenciador de biblioteca” em seguida selecionar a biblioteca (clicar em “More info” para aparecer o botão de instalação, conforme figura:



```
#include <ESP8266WiFi.h> // bibliotecas necessárias
#include <WiFiManager.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
```

Biblioteca OTA – 3ª prática

OTA (Over The Air) – gravação via WiFi

Para utilização deste recurso é necessário:

- Instalação do Phyton na máquina cliente;
- A máquina cliente com o código precisa está na mesma rede do esp8266.
- Para instalar a biblioteca OTA, o procedimento é idêntico ao WiFiManager:

ArduinoOTA Built-In by Ivan Grokhotkov and Miguel Angel Ajo Versão 1.0.0 **INSTALLED**

Enables Over The Air upgrades, via wifi and espota.py UDP request/TCP download. With this library you can enable your sketch to be upgraded over network. Includes mdns announces to get discovered by the arduino IDE.

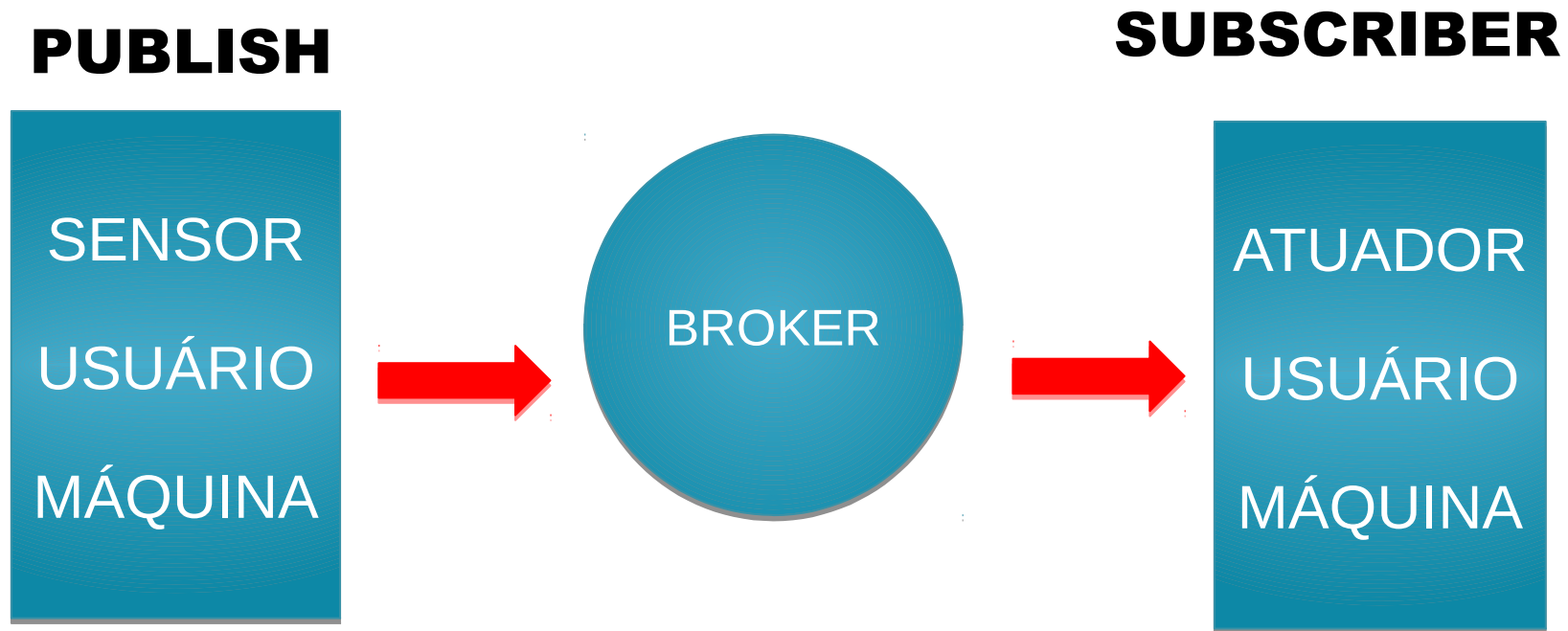
[More Info](#)

```
#include <ESP8266mDNS.h>
#include <ArduinoOTA.h>
```

Protocolo MQTT

- MQTT - Message Queuing Telemetry Transport;
- Desenvolvido pela IBM década de 90;
- Mais utilizado em IoT;
- Baixo overhead comparado ao http.

MQTT - Diagrama



BROKER – Servidor

PUBLISH – Publica no tópico

SUBSCRIBER – Inscreve no tópico

Protocolo MQTT - Publish

- Retain true – informa o último estado quando usuário ou máquina no momento que se conecta ao broker;
- QoS nível 0 (default) – enviada apenas uma vez;
- QoS nível 1 – enviada uma vez, mas os receptores podem receber duplicados;
- QoS nível 2 (mais lento) – envio com garantia de entrega.

Biblioteca MQTT – 4ª prática

É necessário instalar a biblioteca “PubSubClient” da mesma forma que as bibliotecas das práticas anteriores:

PubSubClient by Nick O'Leary

A client library for MQTT messaging. MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.

[More Info](#)

PubSubClientTools by Simon Christmann

```
#include <PubSubClient.h> // Importa a  
Biblioteca PubSubClient
```

OBRIGADO!