

Guia completo do NodeMCU – ESP12 – Usando Arduino IDE (2)

12-16 minutes

Nesse tutorial NodeMCU – ESP12 – Usando Arduino IDE (2), você verá :

- [Porque programar NodeMCU com Arduino IDE ?](#)
- [Procedimentos para instalar NodeMCU na Arduino IDE](#)
- [Programando o NodeMCU ESP12 com a Arduino IDE](#)
- [NodeMCU ESP-12 acessando o relógio da WEB](#)

Porque programar NodeMCU com Arduino IDE ?

No primeiro Tutorial sobre NodeMCU ESP-12 , eu já havia informado que a grande vantagem dessa placa **NodeMCU** é que ela funciona como se fosse um Arduino. Você conecta o cabo USB, e com a IDE Arduino você carrega seus programas na placa. Nem precisa pressionar os botões !

Mas porque usar Arduino IDE ? A ESPRESSIF (fabricante do ESP8266) disponibiliza no site, alguns firmwares para serem usados no módulo ESP8266 (Microcontrolador da placa NodeMCU).

- [ESP8266 SDK](#)
- [ESP8266 AT](#)
- [ESP8266 Touch](#)
- [ESP8266 NOW](#)

Esses firmwares são específicos para o módulo ESP8266. Usam muitos recursos do Chip, mas são complicados de usá-los. Para programá-los são necessários conhecimentos de programação orientada a objetos e da Linguagem C++. Além disso, a documentação não é bem clara para iniciantes em programação de Microcontroladores. Se você já tem esses pré-requisitos, sugiro que estude os links acima.

O grande sucesso do Arduino no mundo inteiro, foi por causa da grande facilidade em usá-lo e programá-lo. Todo mundo sabe disso. E sabiamente, um dos funcionários da ESPRESSIF, Ivan Grokhotkov iniciou a adaptação da Arduino IDE para o ESP8266 em 2016. Hoje já são inúmeros contribuidores, e posso dizer que a Arduino IDE já esta madura para o ESP8266. Muitas bibliotecas já foram adaptadas e muitas aplicações já estão disponíveis. Contando com a grande facilidade de programação ! Por isso, vamos usar a IDE do Arduino.

Esse link abaixo é muito importante, pois é o Repositório para **ESP8266/Arduino**, onde estão as Bibliotecas, códigos de exemplos e todos outros arquivos necessários para usar a Arduino IDE com o ESP8266.

[GitHub ESP8266 Arduino](#)

Um comentário eu devo fazer sobre o firmware LUA. Na verdade, o módulo NodeMCU foi criado por um idealista que adaptou o firmware do ESP8266 para rodar a linguagem LUA. Mas infelizmente, o projeto NodeMCU LUA não foi muito difundido, apesar dos esforços feitos até hoje .

- [NodeMCU LUA](#)

Procedimentos para instalar NodeMCU na Arduino IDE

Existem alguns procedimentos para a instalação do NodeMCU na Arduino IDE. Mas usarei o mais simples.

Primeiro passo é atualizar a Arduino IDE. Algumas IDEs antigas não suportam o NodeMCU ESP8266, por isso é necessário que atualize para a versão mais recente. Atualmente a versão mais nova é a 1.8.5.

Baixe a Arduino IDE e instale no seu computador Windows. Existem versões para MacOS e Linux também.

- [Download Arduino IDE](#)

Inicie o programa Arduino IDE. Clique na barra superior em **Arquivos** e depois em **Preferências** .



Dentro da janela de **Preferências** , copie o link abaixo no campo **URLs adicionais para Gerenciadores de Placas** e depois clique em **OK**.

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

Preferências

Configurações Rede

Local do Sketchbook:

C:\Users\jgust\Documents\Arduino

Idioma do editor: Padrão do Sistema (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: Nenhum

☐ 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

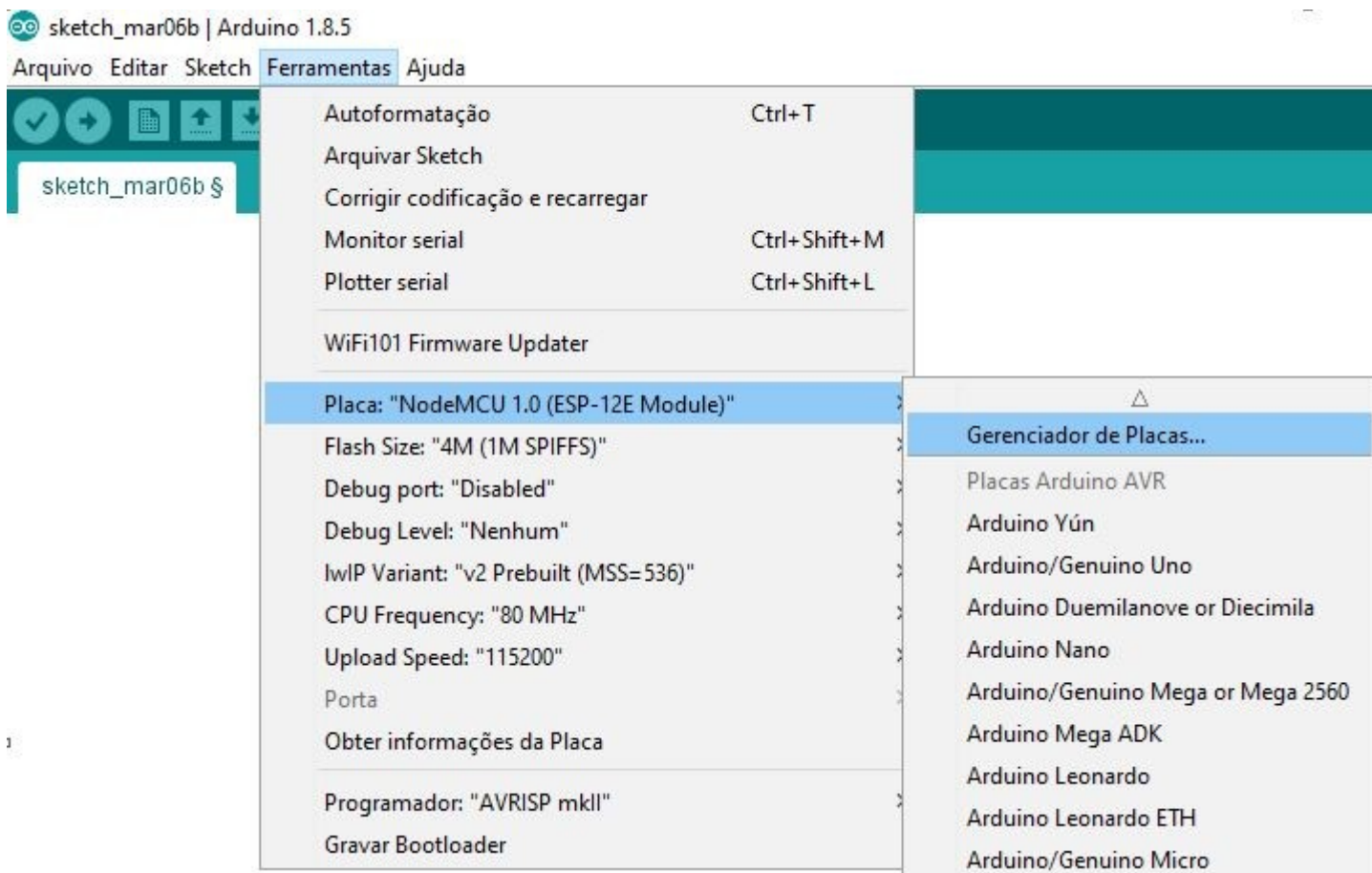
Mais preferências podem ser editadas diretamente no arquivo

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

(editar apenas quando o Arduino não estiver em execução)

Na barra superior novamente, clique em **Ferramentas** e depois em **Gerenciamento de Placas**.

Obs: na minha IDE, o ESP8266 já estava instalado.



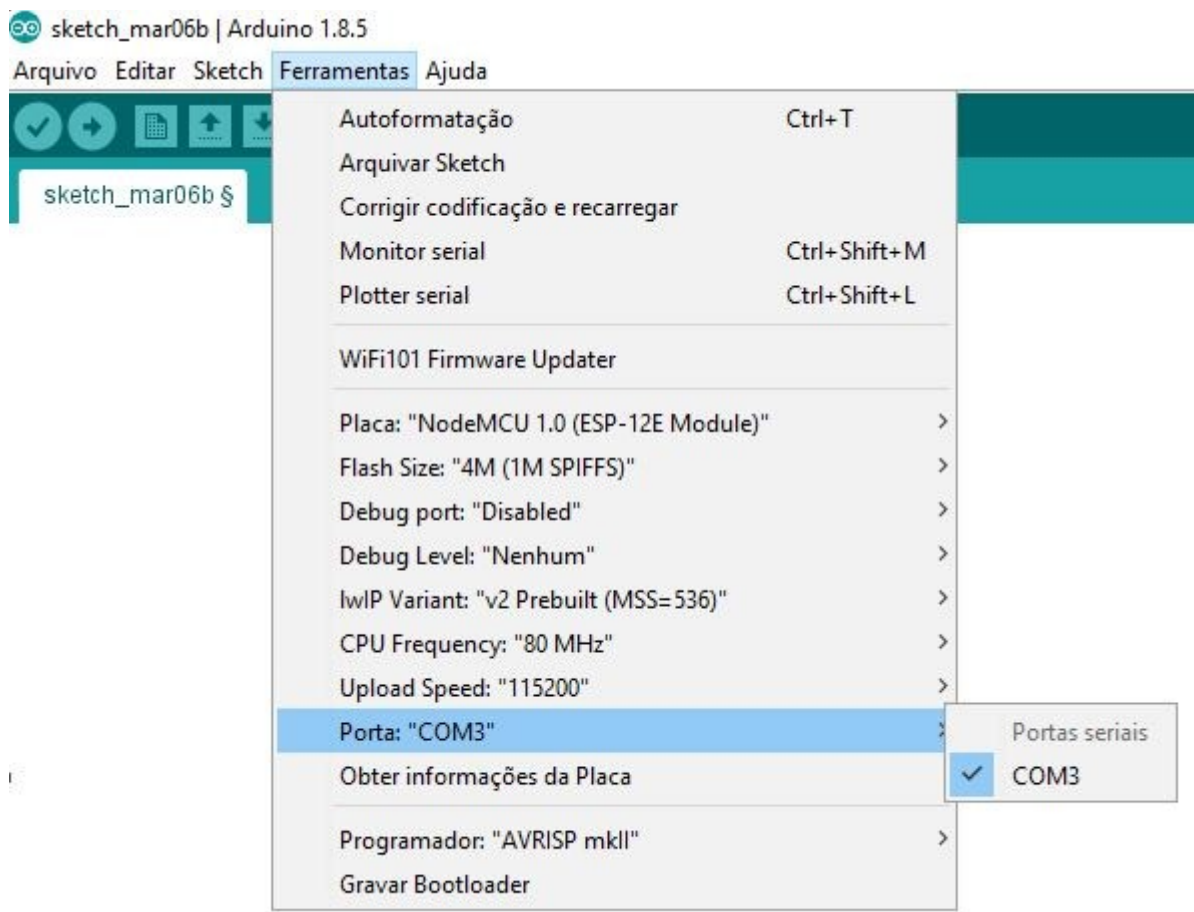
Na janela do **Gerenciador de Placas**, refine a sua pesquisa com : **ESP8266** . Clique em **More Info** e depois em **Instalar**. Lembre-se que o seu computador deve estar conectado na Internet. Após alguns segundos aparecerá **INSTALLED**. Isto é, instalado. Feche essa janela.



Para a escolher o modelo da sua placa, isto é **NodeMCU ESP12** , clique novamente em **Ferramentas** e depois em **Gerenciamento de Placas**. Na janela das Placas, escolha **NodeMCU 1.0 (ESP12E Module)** . Não altere os outros parâmetros da Placa NodeMCU. Somente altere a porta COM da interface serial-USB. No meu caso a porta é a COM3 . O procedimento para identificar qual porta COM é usada no Windows está no Tutorial :

[NodeMCU – ESP12 – Introdução \(1\)](#)

Saiba que a velocidade padrão da Console da IDE Arduino para o ESP8266 é de 115200 Bps.



IMPORTANTE : Após tudo configurado com sucesso, feche e abra o programa Arduino IDE novamente, para que todas as configurações sejam efetivadas. Se não fizer isso, poderá ter problemas ao usá-la.

Programando o NodeMCU ESP12 com a Arduino IDE

Vamos começar os testes do ESP8266 com a Arduino IDE do modo mais simples. Usando exemplos do site **Arduino_ESP8266**. Escolhi o Sketch **Scan WIFI**, pois com ele, podemos testar o circuito WIFI do ESP8266, sem ter que montar nada além do NodeMCU. Recomendo que insira a Placa NodeMCU em um **Protoboard**, pois assim seus pinos ficarão protegidos (evitando um possível curto-circuito).

- Conecte o NodeMCU com o cabo USB na porta USB do seu Computador PC,
- Abra a Arduino IDE e verifique se a porta COM do NodeMCU foi reconhecida ,
- Copie o programa abaixo na Arduino IDE, compile-o e grave-o no ESP8266 (clique no botão **Carregar**).

OBS: Quando o programa estiver sendo carregado no ESP8266, o led azul ficará piscando.

Botões Arduino IDE



Sketch para pesquisar Redes WIFI (próximas do NodeMCU) :

Sketch ESP8266 ScanWIFI.ino

```
/* Pesquisa de Redes WIFI usando o NodeMCU-ESP12
   Arduino IDE 1.8.5 - ESP8266
   Gustavo Murta 07/mar/2018
   baseado em
   https://arduino-esp8266.readthedocs.io/en/2.4.0/esp8266wifi/scan-class.html
   Blog Eletrogate http://blog.eletrogate.com/nodemcu-esp12-usando-arduino-ide-2/
*/
#include "ESP8266WiFi.h"

void setup()
{
  Serial.begin(115200);          // configura monitor serial 115200 Bps
  Serial.println();             // imprime uma linha
  WiFi.mode(WIFI_STA);         // configura rede no modo estacao
  WiFi.disconnect();           // desconecta rede WIFI
  delay(100);                   // atraso de 100 milisegundos
}

void prinScanResult(int networksFound)
{
  Serial.printf("\n");          // imprime uma linha
  Serial.printf("%d redes encontradas\n", networksFound); // imprime numero de
  redes encontradas
  for (int i = 0; i < networksFound; i++) // contagem das
  redes encontradas
  {
    Serial.printf("%d: %s, Ch:%d (%ddBm) %s\n", i + 1, WiFi.SSID(i).c_str(),
    WiFi.channel(i), WiFi.RSSI(i), WiFi.encryptionType(i) == ENC_TYPE_NONE ?
    "aberta" : "");
  }
}
```



```

void loop()
{
  WiFi.scanNetworksAsync(prinScanResult);    // imprime o resultado
  delay(500);                                // atraso de 0,5 segundos
}

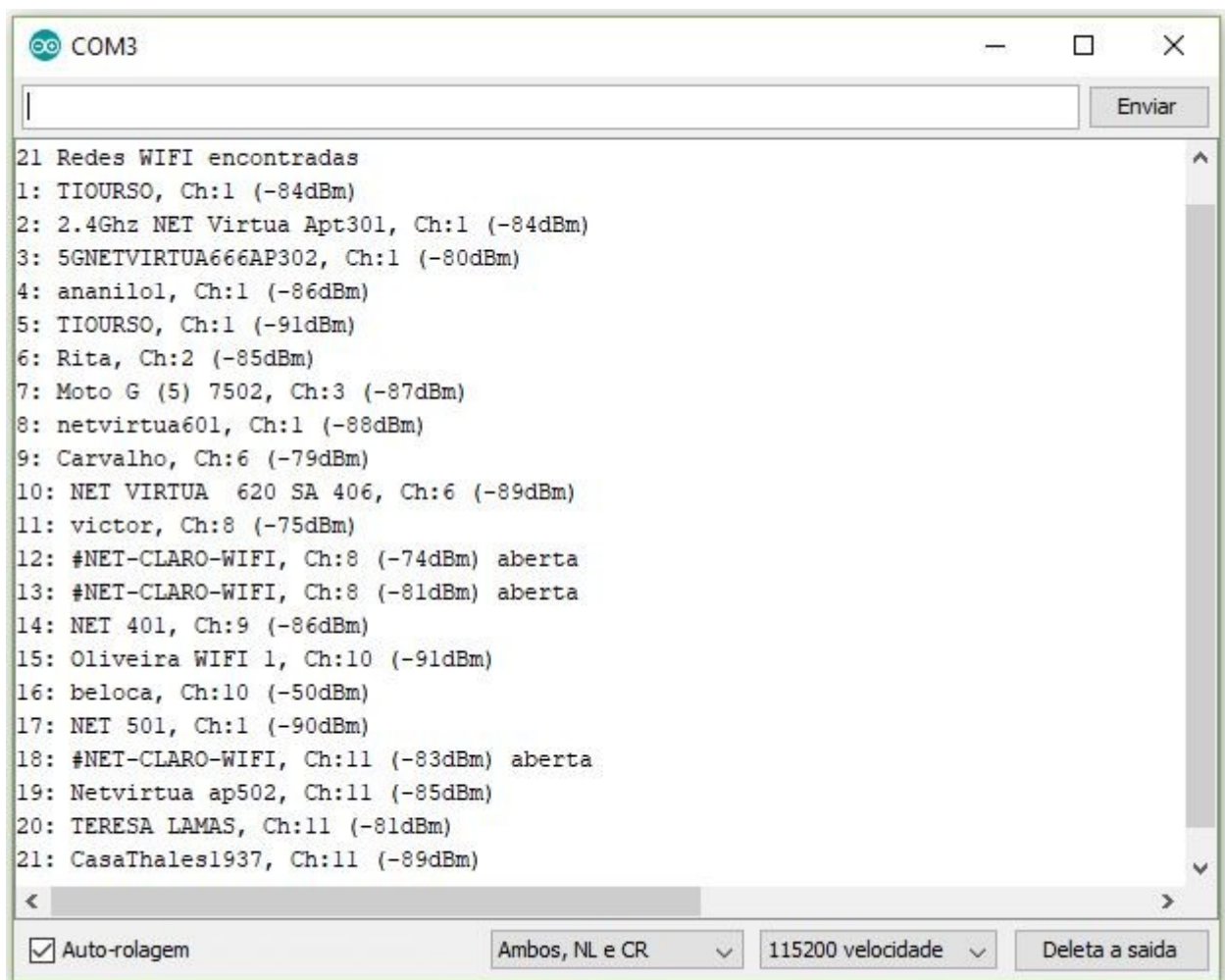
```

Console da Arduino IDE – ESP8266 (115200 Bps) :

(clique no botão Monitor serial e altere a velocidade para 115200 Bps na barra inferior da janela)

Exemplo – redes encontradas em minha residência .

Nomes das Redes, canal WIFI usado e o nível de transmissão em dBms.



Para entender alguns detalhes do uso do ESP8266 com a Arduino IDE, recomendo que leia essa documentação:

[Documentação de Referência – ESP8266](#)

Informações sobre as Bibliotecas já instaladas na IDE e mais algumas não instaladas, poderá encontrar no Link:

[Bibliotecas ESP8266 – Arduino IDE](#)

Sobre a Biblioteca ESP8266WIFI :

[Biblioteca ESP8266WiFi](#)

Sobre o primeiro Sketch a ser rodado no NodeMCU ESP12 , veja as funções do SCAN WIFI :

[Biblioteca Arduino WIFI – classe SCAN](#)

NodeMCU ESP-12 acessando o relógio da WEB

Mais um exemplo para você testar o WIFI do NodeMCU ESP-12, acessando a rede WIFI do seu roteador. Eu escolhi um dos exemplos do GitHub Arduino Libraries , um sketch para o ESP8266 acessar o relógio do servidor NTP do Brasil.

[GitHub Arduino-Libraries](#)

NTP é um protocolo para sincronização dos relógios dos computadores baseado no protocolo UDP da Internet. Ao acessar o servidor NTP, o computador ou Microcontrolador poderá acertar o relógio. O horário padrão dos servidores NTP é o UTC (Tempo Universal Coordenado). Portando ao acessá-lo, é necessário ajustar a hora com o desvio da sua região.

[Informações sobre NTP](#)

Esses são os desvios de horário das regiões do Brasil :

Por exemplo = horário de Brasília (inverno) UTC – 3 Horas



Imagem : Fusos Horários no Brasil – Wikipedia

No Brasil , os relógios podem ser sincronizados através de alguns servidores que estão no site <https://ntp.br/>

Servidores NTP.br

a.st1.ntp.br 200.160.7.186 e 2001:12ff:0:7::186

b.st1.ntp.br 201.49.148.135

c.st1.ntp.br 200.186.125.195

d.st1.ntp.br 200.20.186.76

a.ntp.br 200.160.0.8 e 2001:12ff::8

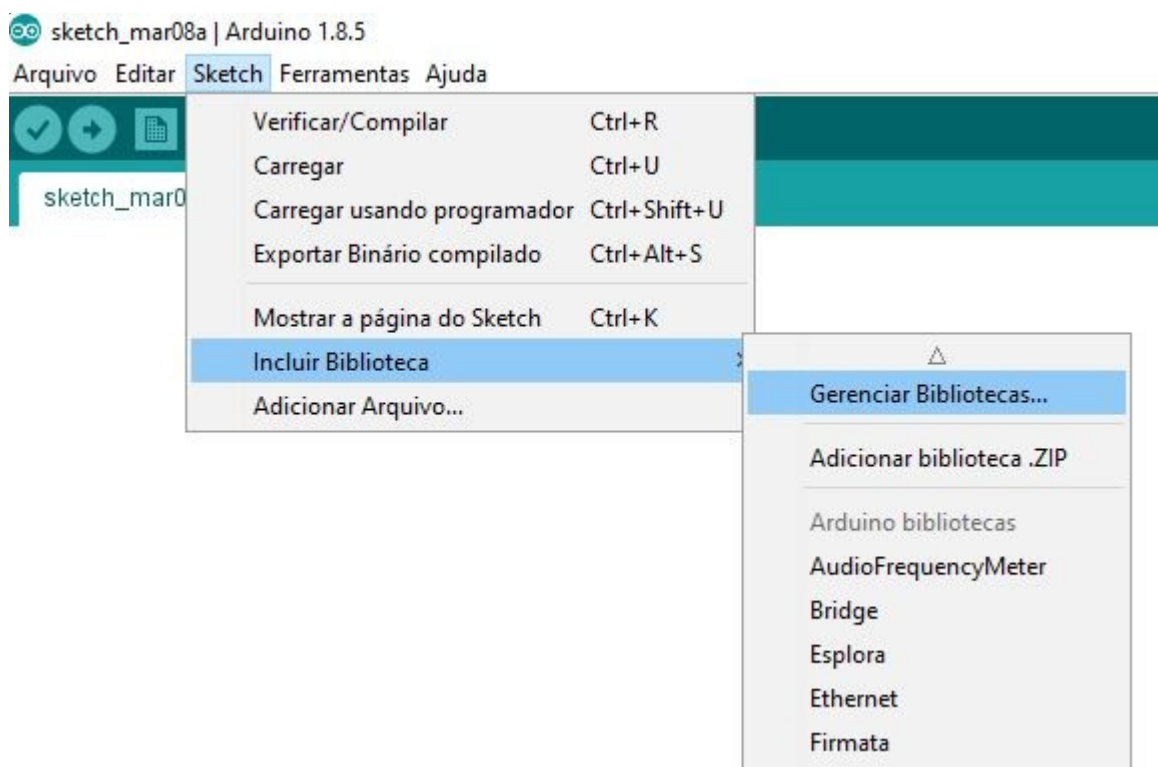
b.ntp.br 200.189.40.8

c.ntp.br 200.192.232.8

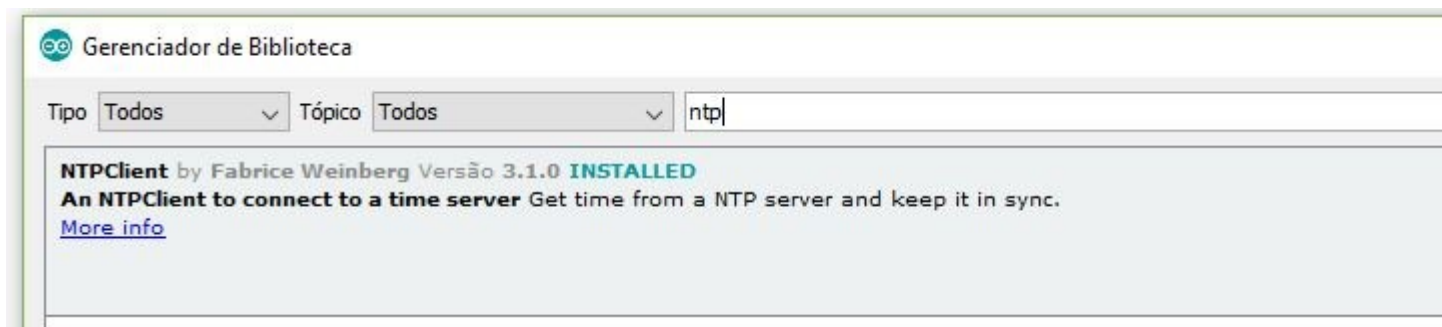
gps.ntp.br 200.160.7.193 e 2001:12ff:0:7::193

Para rodar o Sketch, será necessário instalar uma Biblioteca que não esta instalada na Arduino IDE. Boa oportunidade para aprender como instalar uma nova Biblioteca !

Conecte o NodeMCU ESP-12 no seu computador. Abra a Arduino IDE e veja se a configuração da placa esta correta. Para instalar a nova Biblioteca, clique em **Sketch > Incluir Biblioteca > Gerenciar Bibliotecas .**



Após abrir a janela do **Gerenciador de Biblioteca**, refine a busca digitando **ntp**. Na biblioteca **NTPClient**, clique em **More Info** e depois em **Instalar**. Após alguns segundos, ela será automaticamente instalada. Na imagem abaixo essa biblioteca já esta instalada. Lembre-se que o seu computador precisa estar conectado na internet. Após a instalação da Biblioteca, é necessário que feche e abra novamente o programa Arduino IDE.



Tudo pronto para você carregar o exemplo. Fiz algumas alterações no exemplo da Biblioteca **NTPClient** do **GitHub Arduino Libraries**, alterando o nome para o servidor NTP do Brasil e ajustando o relógio para o horário de Brasília (UTC – 3).

[Exemplo Advanced NTPClient](#)

Esse é o Sketch já configurado para o Brasil .

IMPORTANTE : No sketch abaixo, você deverá alterar o nome do seu roteador WIFI (**ssid**) e a senha do roteador (**password**) também. Digite entre as aspas, exemplo = “eletrogate”.

[Sketch ESP8266 NTPclock.ino](#)

```
/*  NodeMCU ESP-12 acessando o relógio da WEB
    Arduino IDE 1.8.5 - ESP8266
    Gustavo Murta 08/mar/2018
    baseado em
    https://github.com/arduino-
    libraries/NTPClient/blob/master/examples/Advanced/Advanced.ino
    Blog Eletrogate http://blog.eletrogate.com/nodemcu-esp12-usando-arduino-ide-
2/
*/

#include <NTPClient.h>
// Biblioteca https://github.com/arduino-libraries/NTPClient
#include <ESP8266WiFi.h>
// Biblioteca
https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi
#include <WiFiUDP.h>
// Biblioteca
https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi

const char *ssid      = "roteador WIFI";          // nome do seu roteador WIFI
(SSID)
const char *password = "senha do WIFI";          // senha do roteador WIFI

WiFiUDP ntpUDP;
// Definindo o Servidor de relógio NTP Brasil, ajusta relógio UTC -3 horas,
intervalo de atualizacao em milisegundos
NTPClient timeClient(ntpUDP, "gps.ntp.br", -3 * 3600, 60000);

void setup()
{
  Serial.begin(115200);                          // print no Serial Monitor da IDE
  WiFi.begin(ssid, password);                     // acessando a rede WIFI

  while ( WiFi.status() != WL_CONNECTED )        // aguardando a conexão WEB
  {
    delay ( 500 );                                // aguarda 0,5 segundos
    Serial.print ( "." );
  }
}
```

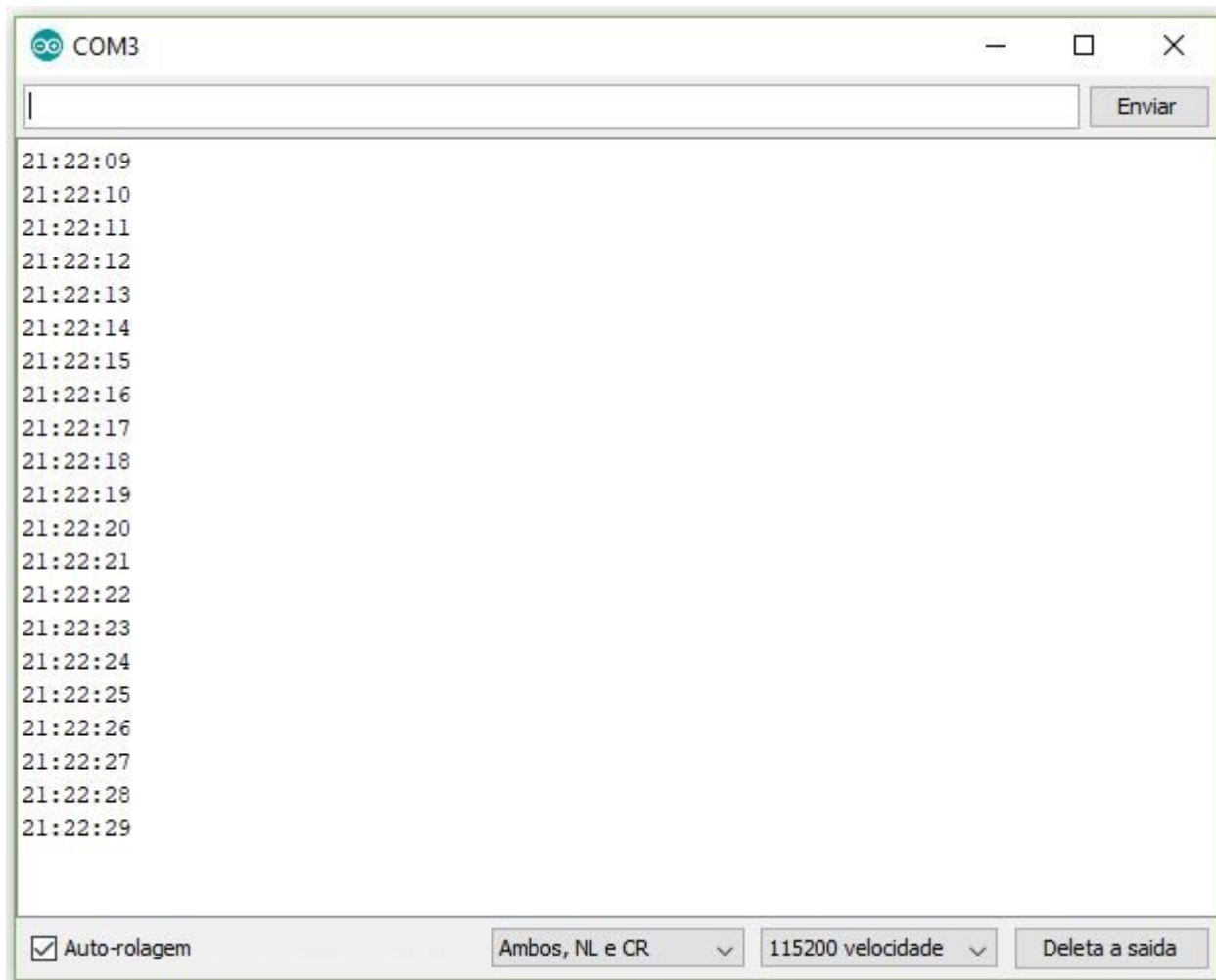
```

    }
    timeClient.begin();
}

void loop()
{
    timeClient.update();                // atualiza o relógio
    Serial.println(timeClient.getFormattedTime()); // print do relógio da WEB
    delay(1000);                       // atraso de um segundo
}

```

Após compilar e carregar o programa no NodeMCU (clique no botão **Carregar**), abra a janela da console (Serial Monitor) da IDE. Clique no botão Monitor serial e altere a velocidade para 115200 Bps na barra inferior da janela. O horário atual deverá aparecer na tela, a cada segundo .



Se tiver alguma dúvida, deixe um comentário !

Tutoriais sobre NodeMCU – ESP12 :

[***NodeMCU – ESP12 – Introdução \(1\)***](#)

[***NodeMCU – ESP12 – Alarme Residencial IOT \(3\)***](#)

Faça sua avaliação.