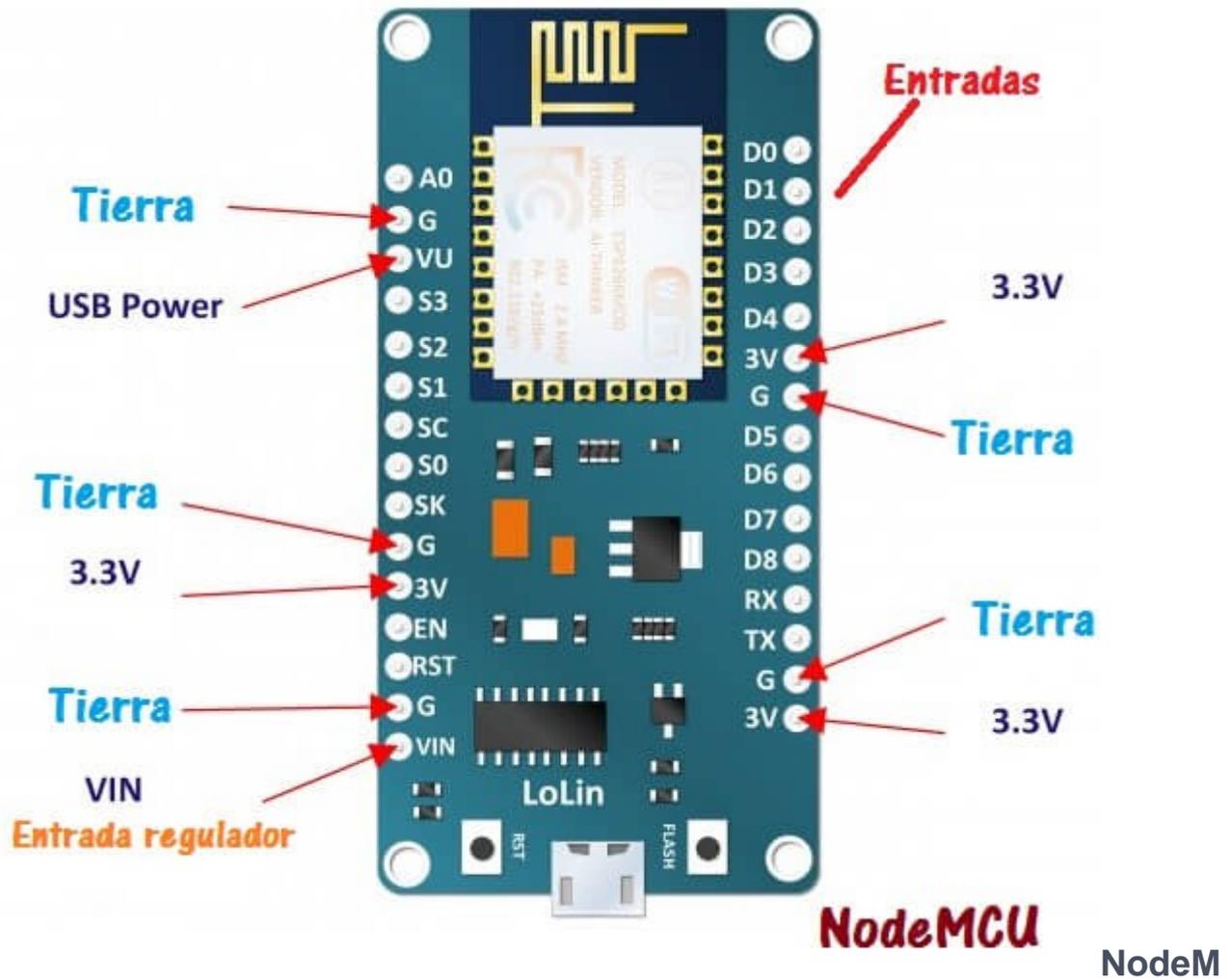


# Teoría Básica del NodeMCU



CU el kit de desarrollo para el IoT

## ¿ Qué es NodeMCU ?

NodeMCU es una placa de desarrollo , programable en el entorno de arduino (con raíces de lenguaje C ). Es una placa cuya finalidad es dar soporte físico ( alimentación, puertos, acceso a terminales RX, Tx, etc ) al microcontrolador ESP 8266.

Hay muchas otras placas con mayor o menor ventaja, como la ESP 01 ya tratada en la práctica:

Control de luz con esp8266

Dentro de esta placa, el chip que ejecuta las funciones de programación se llama ESP8266, que es un chip tipo SoC ( System on a Chip ). Dentro de ese SOC tenemos el microcontrolador, que en este caso es un Tensilica L106 de 32-bit

### ¿ Y qué tiene básicamente un ESP 8266 ?



Para entrar

en detalle de esta SOC, deberíamos visitar su hoja de características, que una vez revisada tomamos los siguientes aspectos mas relevantes

1. Soporte de protocolo 802.11n (2.4 GHz), con velocidades hasta 72.2 Mbps
2. Puede funcionar en varios de los canales de la banda 2,4
3. Potencia TX entre los 14 y 20 dBm
4. Sensibilidad hasta los -91 dmm ( en 802.11 b
5. Tensión de trabajo 2.5V ~ 3.6V
6. Corriente de salida de 80 mA
7. Seguridad WPA/WPA2
8. Encriptación WEP/TKIP/AES
9. Amplio abanico de usos ( automatización, camaras IP, etc)
10. Varios puertos digitales ( GPIO )

## **CPU, Memory, and Flash**

El procesador de 32 bits Tensilica L106 usado en el ESP8266 alcanza un consumo de energía extra bajo y una velocidad máxima de reloj de 160 MHz.

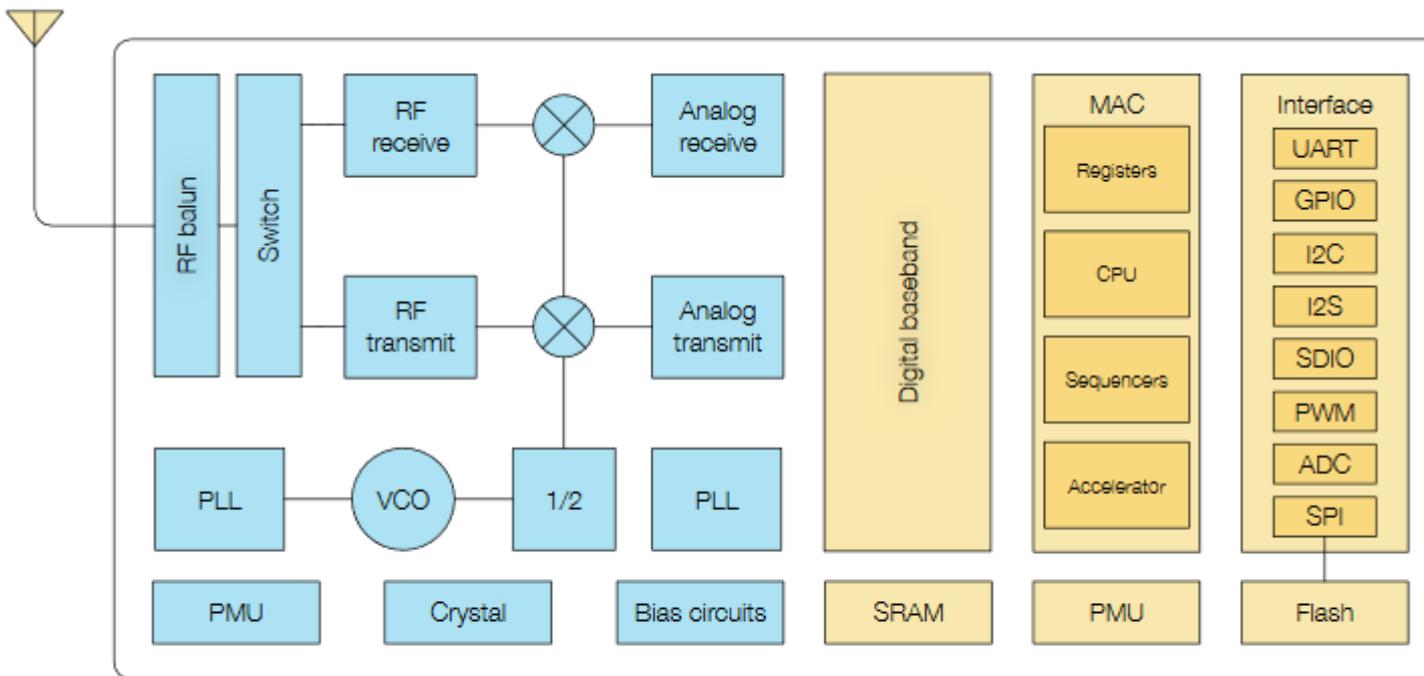
La CPU incluye las interfaces RAM/ROM programables (iBus)

El tamaño de la memoria RAM < 50 kB

No hay ROM programable en el SoC. Por lo tanto, el programa de usuario debe almacenarse en un flash SPI externo.

El Flash externo ESP8266EX utiliza el flash SPI externo para almacenar programas de usuario, y soporta teóricamente hasta 16 MB de capacidad de memoria.

Mostramos a modo de curiosidad , los diferentes bloques operativos que contiene el ESP8266

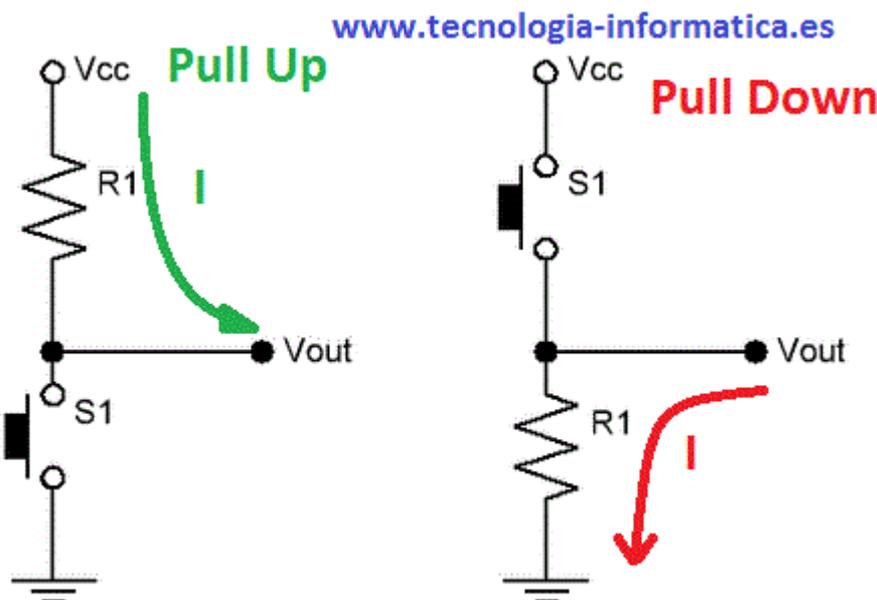


## Diagrama de bloques funcional del esp8266

### Reloj de alta frecuencia

El reloj de alta frecuencia del ESP8266EX se utiliza para accionar tanto los mezcladores de transmisión como los de recepción. Este reloj se genera a partir de un oscilador de cristal interno y un cristal externo. La frecuencia del cristal oscila entre 24 MHz y 52 MHz.

### Entrada/salidas de propósito general (GPIO) ESP8266EX



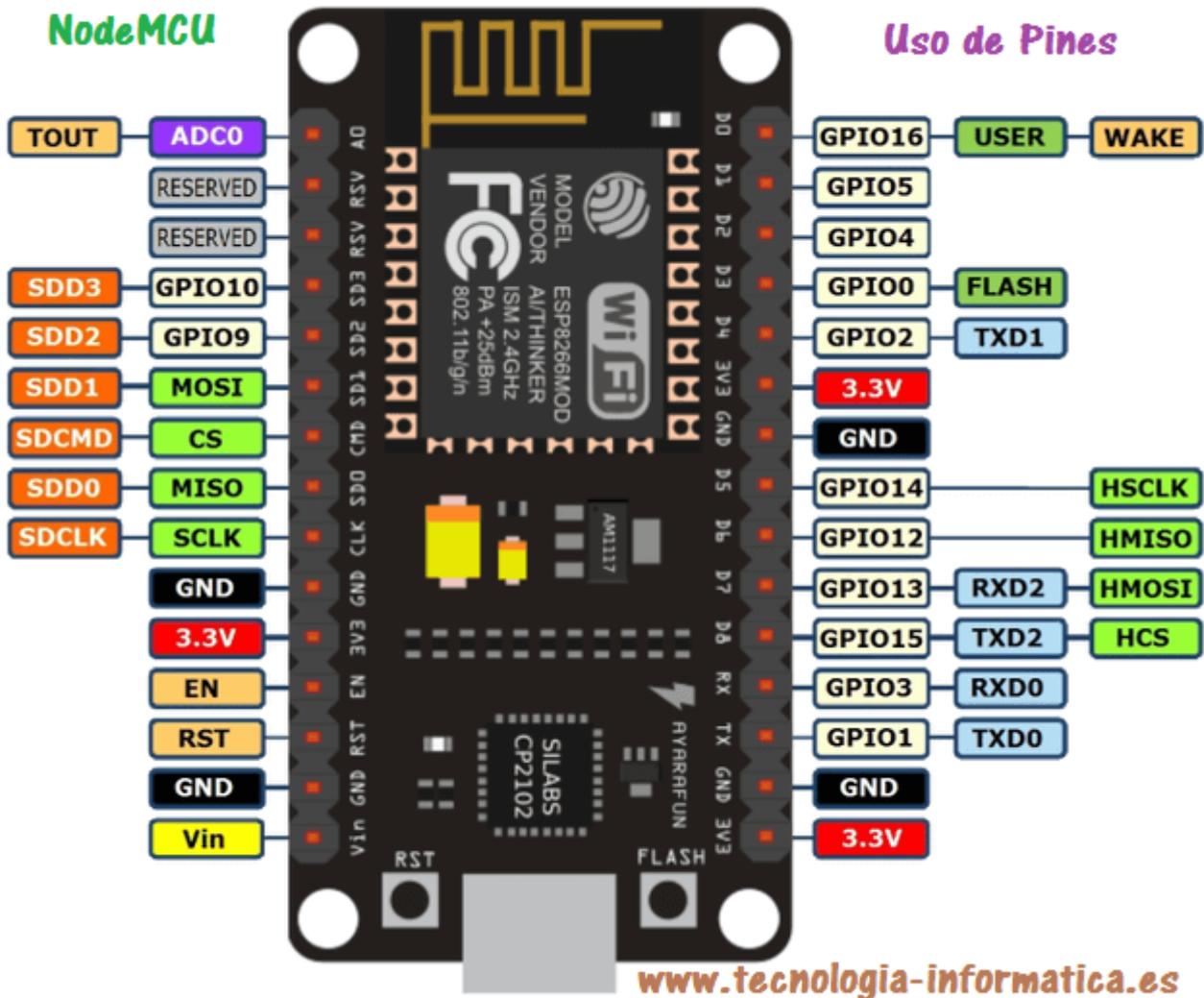
El chip tiene 17 pines GPIO que pueden asignarse como entradas o salidas, y que además pueden

varias funciones mediante la programación de los registros correspondientes.

Cada GPIO PAD puede configurarse con pull-up o pull-down interno (XPD\_DCDC sólo puede configurarse con pull-down interno, otros GPIO PAD sólo pueden configurarse con pull-up interno), o con alta impedancia.

El concepto de pull-up o pull-down ya se ha visto en el módulo de electrónica , y para recordarlo, adjuntamos la imagen.

Esta placa de desarrollo tiene pines que pueden ser utilizados para diferentes propósitos. Para la finalidad de este módulo vamos a usar por defecto , aquellos puertos que sean puros. Se adjunta una imagen donde puede verse lo versátil de los pines



Existen muchas otras características, para lo cual, y en un entorno de desarrollo más profundo, es conveniente estudiarlas dentro de la [datasheet](#) del fabricante.

### Ponemos nuestro NodeMCU en marcha

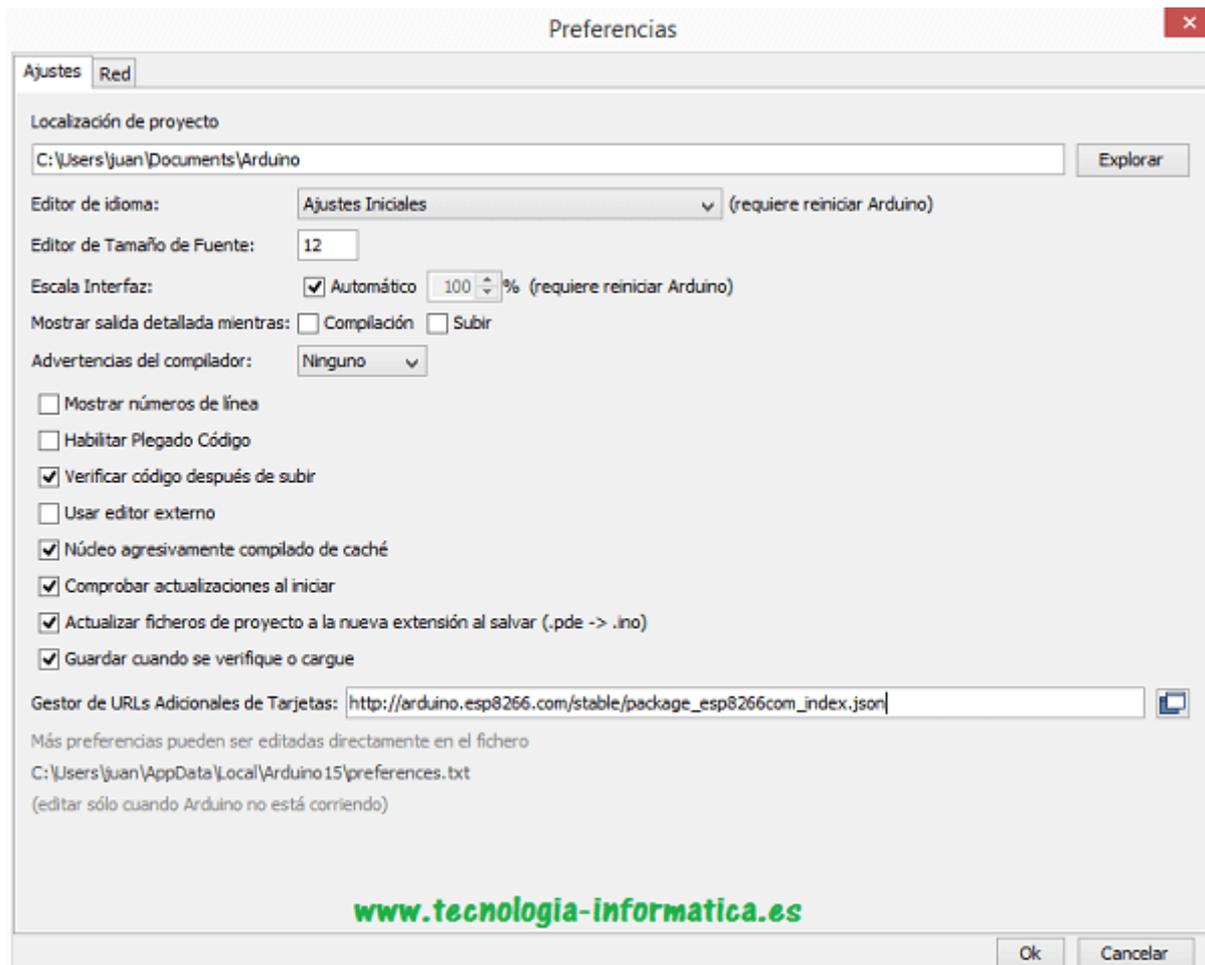
Para programar esta placa, al igual que lo hacemos con la placa de arduino uno, debemos de hacer los mismos paso, pero añadiendo algunos pasos extras, dado que el entorno de programación de

1º Descarga e instala el [IDE de Arduino](#)

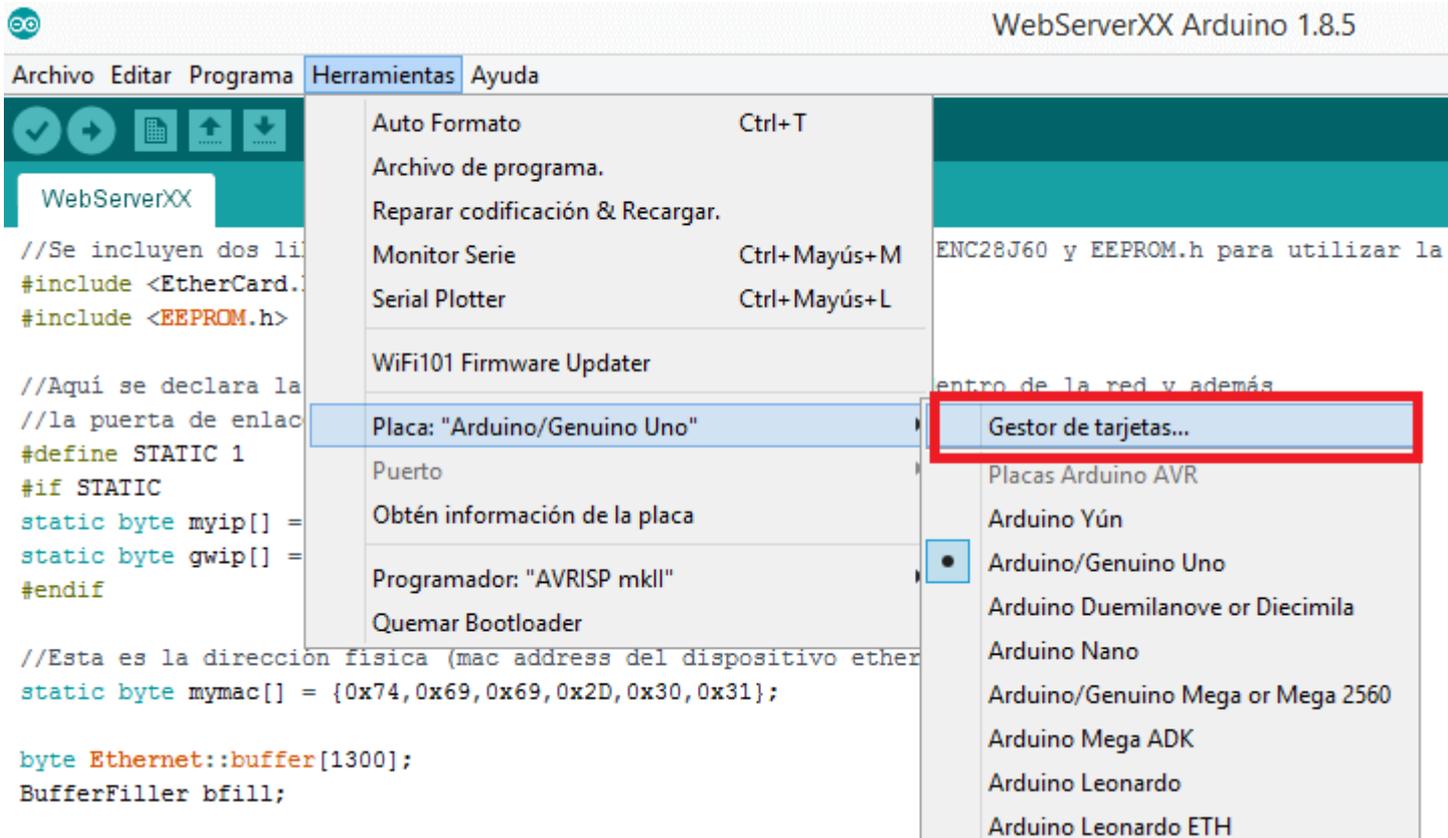
2º Luego, necesita instalar el complemento ESP8266 para el IDE de Arduino. Para ello, vaya a Archivo > Preferencias.

3º Introduzca

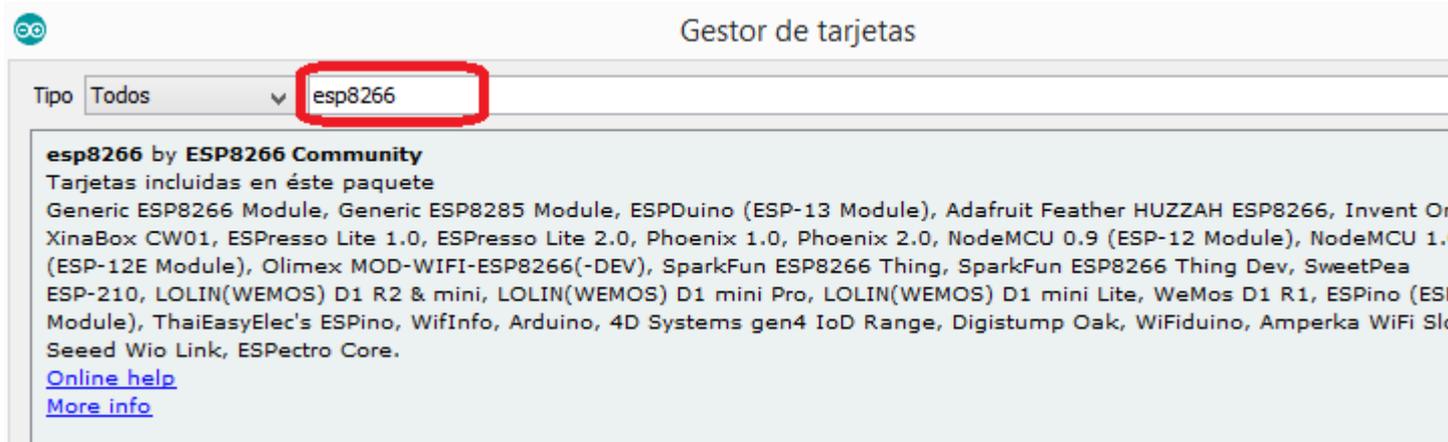
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) en el campo "Additional Board Manager URLs" como se muestra en la siguiente figura. A continuación, haga clic en el botón "Aceptar".



4º Vaya a Herramientas > Placas > Gestor de tarjetas.



5º Escriba esp8266 y aparece "esp8266 by ESP8266 Community", como se muestra en la siguiente figura. Pinchar e instalar. Tarda un poco en instalar todo el paquete. Una vez que aparece **INSTALLED**, podemos cerrar.



6º Vaya a Herramientas > Placas y elija su placa ESP8266. Luego, reabra su IDE de Arduino.

Ahora, en placas, veremos todas aquellas que tienen el ESP8266 como SOC.

7º Elegimos la placa Node Mcu 1.0

8º Ahora vamos a meter nuestro primer programa, usando el Led que trae la misma placa. Ese Led está conectado con una resistencia pull-up y por eso se enciende si el pin está a nivel bajo

```
#define LED_PLACA 2
void setup() {
  pinMode(LED_PLACA, OUTPUT);
}
void loop() {
  digitalWrite(LED_PLACA, LOW);
  delay(200); // Esperar 200 milisegundos
  // Apagar el LED
  digitalWrite(LED_PLACA, HIGH);
  delay(2000); // Esperar 2 segundos
}
```

Tenemos nuestra placa preparada para pasar a la primera práctica.

2º Ahora toca conectar la placa a la red wifi

Conectamos nuestra nodeMCU a una red wifi. Para esta segunda práctica tutorada, tenemos que configurar nuestro IDE y ajustar el programa para que el programa sea cargado sin problemas. Detallamos

Este es el programa que tenemos que cargar

```
#include <ESP8266WiFi.h>

// Sustituir con datos de vuestra red
const char* ssid = "playadonana";
const char* password = "play*****";
void setup()
{
  // Establecemos conexión serie a 115200 baudios/segundo. Esto nos va a generar un problema. Lo dejamos para que comprobéis el mensaje de consola.
  // mas abajo mostramos la solución
  Serial.begin(115200);
  delay(10);
  // demora de 10 ms
  WiFi.mode(WIFI_STA);
```

```

WiFi.begin(ssid, password);
Serial.print("Conectando a:\t");
Serial.println(ssid);
// Esperar a que nos conectemos
while (WiFi.status() != WL_CONNECTED)
{
// en estas lineas , se espera 0,2 segundos a probar conexión y en caso contrario, se muestra un .
delay(200);
Serial.print('.');
}
// Una vez conectado, se muestra la ip asignada a nuestro NODEMCU
Serial.println();
Serial.print("Conectado a:\t");
Serial.println(WiFi.SSID());
Serial.print("IP address:\t");
Serial.println(WiFi.localIP());
}
void loop()
{
}

```

Abrimos la consola desde Herramientas -> Monitor serie

Vemos los símbolos extraños ?

### **Problema de símbolos extraños al cargar el programa en el Node MCU**

Nuestra placa , aparte de tener el MCU esp8266, tiene otros chips que tienen su función, como por ejemplo, el adaptador de tensión de 5 a 3,3, la memoria externa, etc. Uno muy importante es el chip que permite la conexión entre el MCU y el software. Ese chip para que funcione , hay que instalar el driver CH340G. Además es muy importante que el chip trabaje con los requisitos del fabricante.

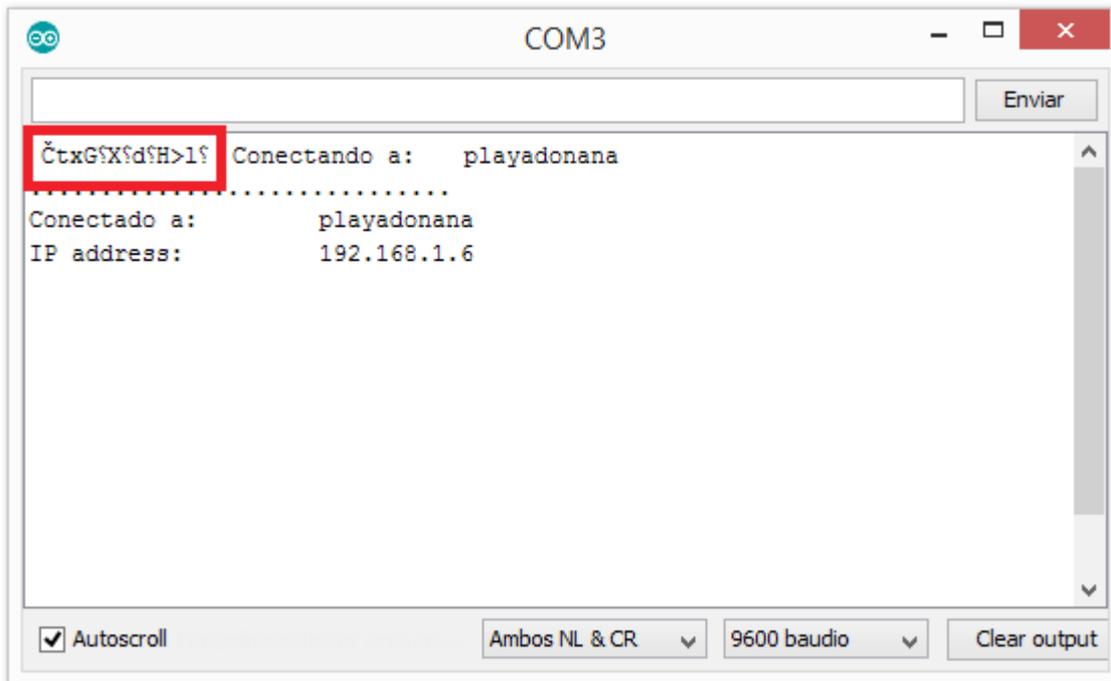
La velocidad con la que el IDE ( programa de arduino ) manda los datos a ese chip , viene medidos en baudios ( símbolos de información por segundos ).

Hemos comprobado que se se excede de 9600, la grabación genera una serie de errores, que hemos representado en la siguiente imagen.

La solución es cambiar la línea donde pone

`Serial.begin(115200);` por `Serial.begin(9600);`

Vemos ahora la pantalla de comunicaciones serie. Hemos marcado en rojo los símbolos que salen de forma errónea si la comunicación serie en baudios no es la correcta



Para terminar esta primera parte. añadir estas líneas para que nos genere más información de la red

```
Serial.print("hostname:\t");  
Serial.println(WiFi.hostname());  
Serial.print("subnetMask:\t");  
Serial.println(WiFi.subnetMask());  
Serial.print("Puerta de enlace:\t");  
Serial.println(WiFi.gatewayIP());  
Serial.print("IP DNS:\t");  
Serial.println(WiFi.dnsIP());  
Serial.print("macAddress:\t");  
Serial.println(WiFi.macAddress());
```



Puedes bajar el archivo en [Conexión WIFI de Node MCU](#)

Una vez visto la más básico, veremos los siguientes proyectos, con utilidad en la domótica

1. [Servidor web utilizando el NodeMCU](#)

<https://programarfacil.com/esp8266/como-programar-nodemcu-ide-arduino/>

Biblio

<https://randomnerdtutorials.com/esp8266-web-server-spiffs-nodemcu/>

<https://randomnerdtutorials.com/esp8266-web-server/>

[https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/#Versiones\\_de\\_NodeMCU](https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/#Versiones_de_NodeMCU)

<https://www.prometec.net/esp8266-con-arduino-ide/>

[http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html)

## **Enlaces del NodeMCU ( ESP 8266 )**

1. [Teoría Básica del NodeMCU](#)
2. [Conectarse a tu red local \(LAN\) desde Internet](#)
3. [Servidor web con NodeMCU \( ESP8266 \)](#)