



# NodeMCU

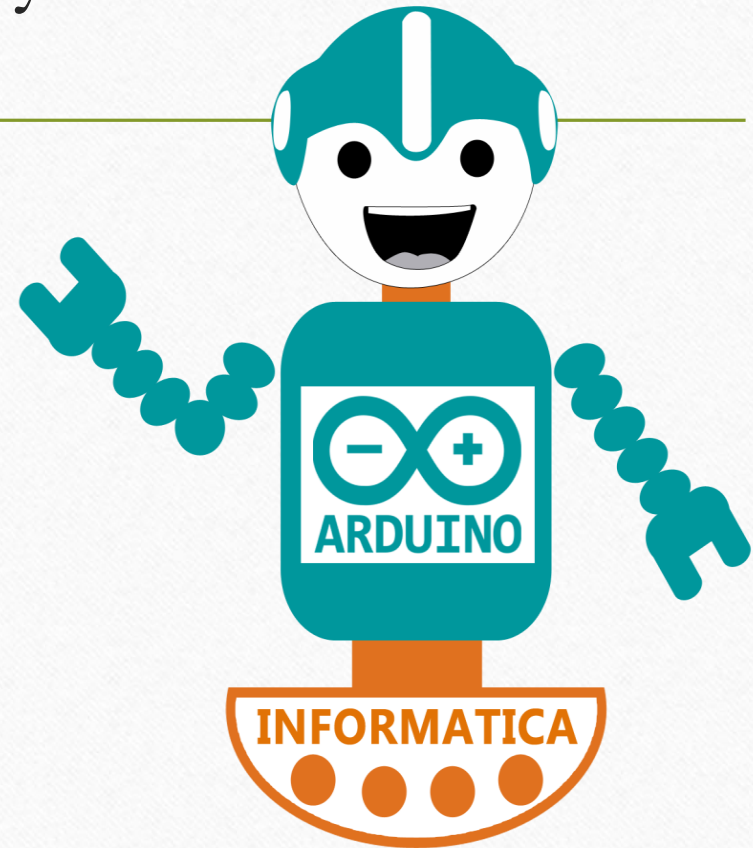
Tutora: Angela Jazmín Miranda Flores



# Fechas de presentación de proyectos

---

- 1ra presentación 16 de noviembre de 2017
  - Presentación de la ficha técnica del proyecto.
- 2da presentación 7 de diciembre de 2017
  - Proyecto desarrollado al 80%
  - Informe del proyecto en formato digital.
- Presentación final 15 de diciembre de 2017
  - Presentación del proyecto terminado.
  - 3 copias del informe para los jurados.

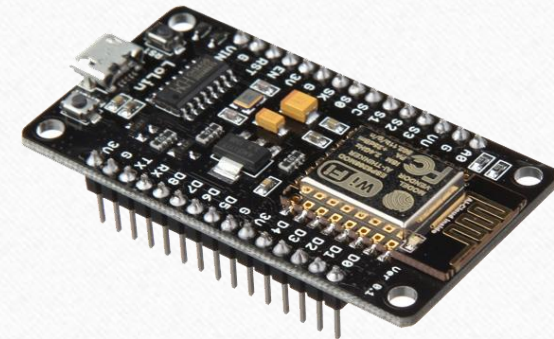




# CARACTERÍSTICAS

---

- NodeMCU es la placa de desarrollo basada en el ESP8266, utiliza el chip ESP-12E.
- Incorpora una MCU de 32-bit de bajo consumo.
- Módulo WiFi de 2.4 GHz.
- 1 entrada analógica.
- 13 pines de entrada y salida GPIO.



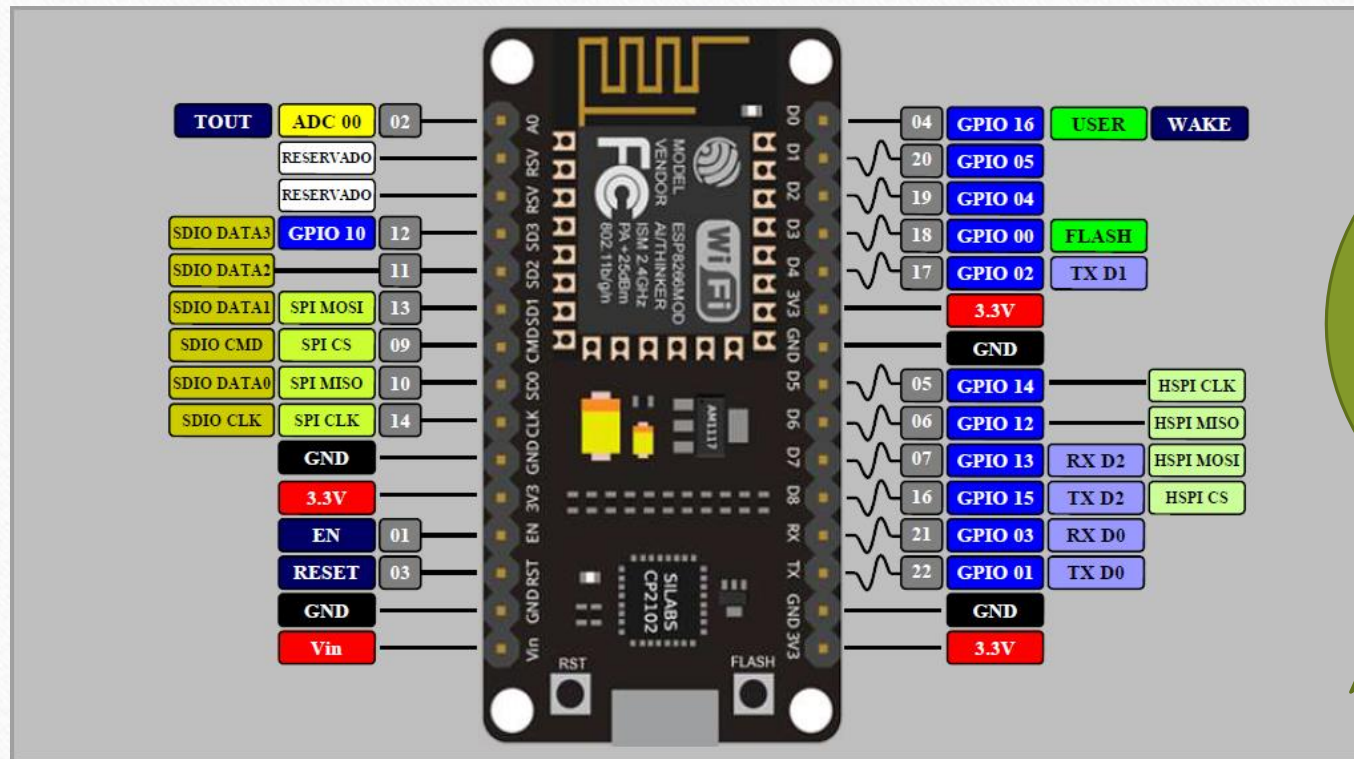
# CARACTERÍSTICAS

- Voltaje de entrada (USB): 5V
- Voltaje de salida en los pines: 3.3V
- Corriente nominal por pin: 12mA
- Frecuencia de procesador: 80MHz (160MHz max.)
- 4MB Flash
- Consumo de corriente en stand-by @80MHz: 80mA
- Velocidad de transmisión optima en baudios: 115200



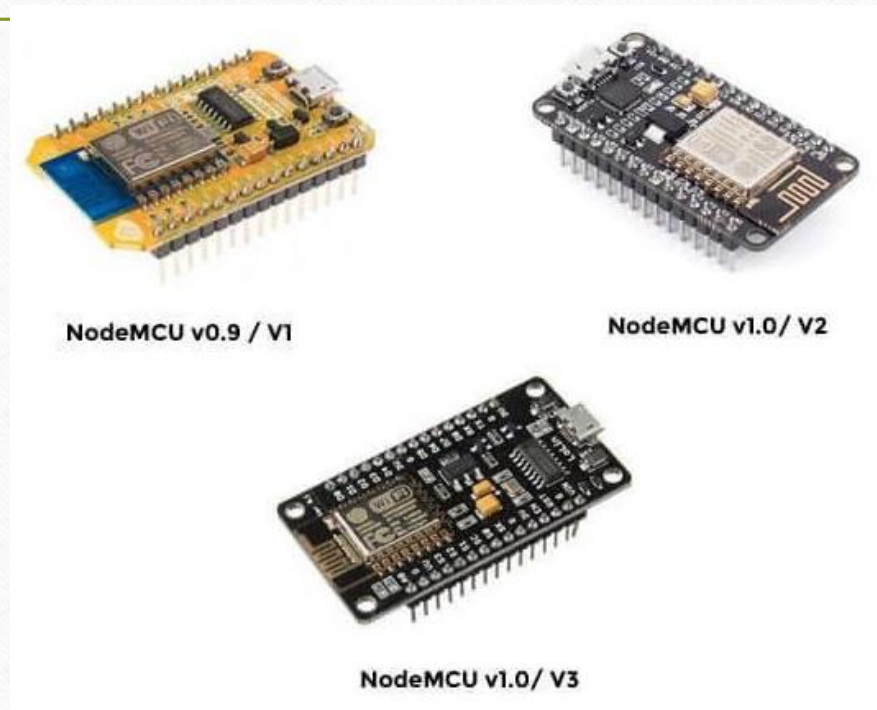


# PINOUT DE AMICA



NOTA.- Los pines D0( GPPIO16) Y D4 (GPIO2) controlan los led's integrados del NodeMCU.

# VERSIONES

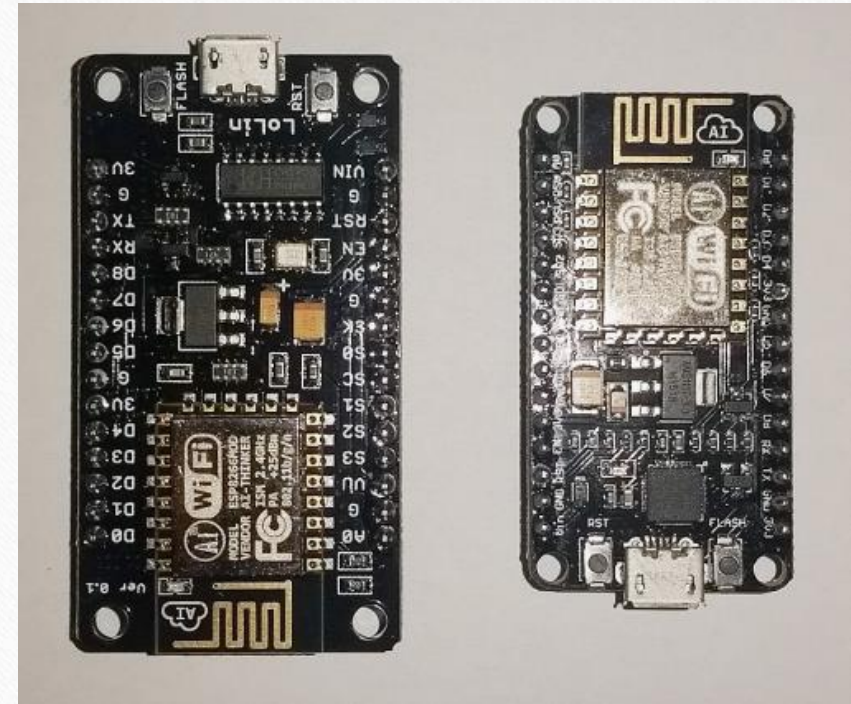




# DIFERENCIA ENTRE AMICA Y LOLIN

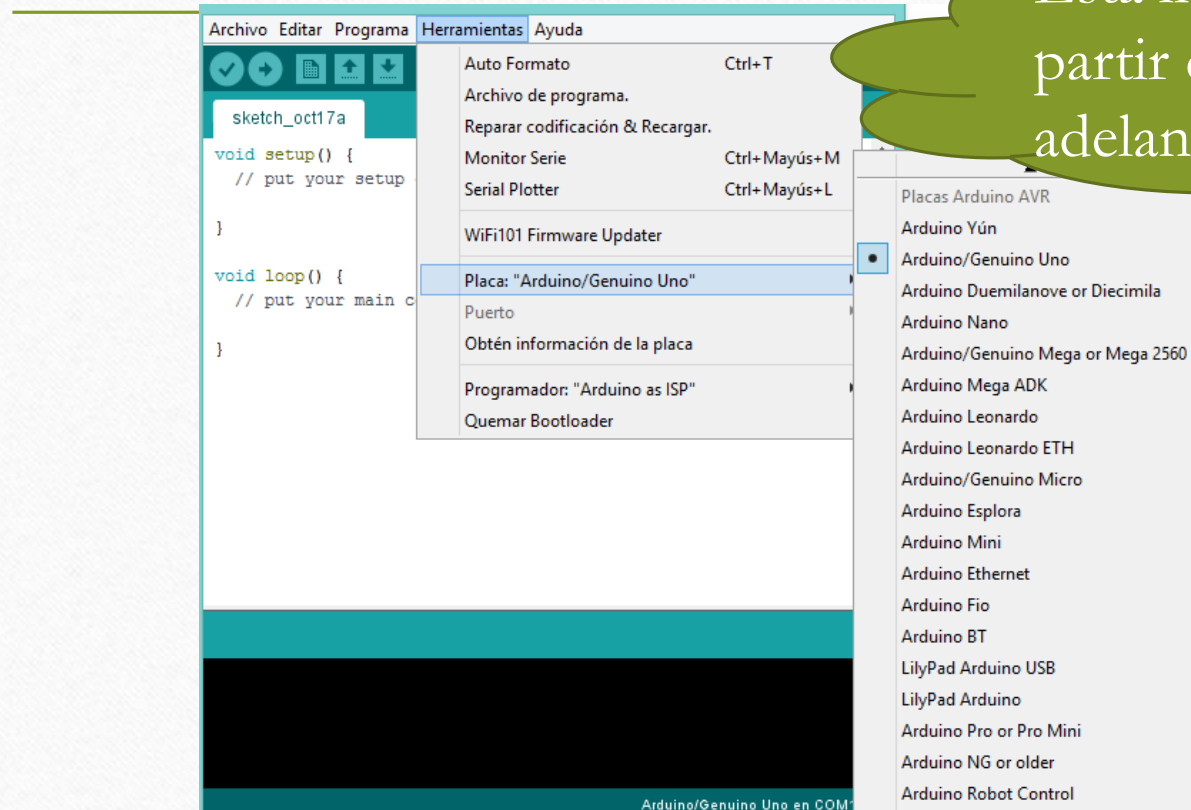
---

- ✓ Tamaño
- ✓ Fabricantes
- ✓ 2 pines de energía extra en lolin
- ✓ Amica es la versión oficial del NodeMCU v2.



# Instalación

1. Verificamos si la placa esta instalada

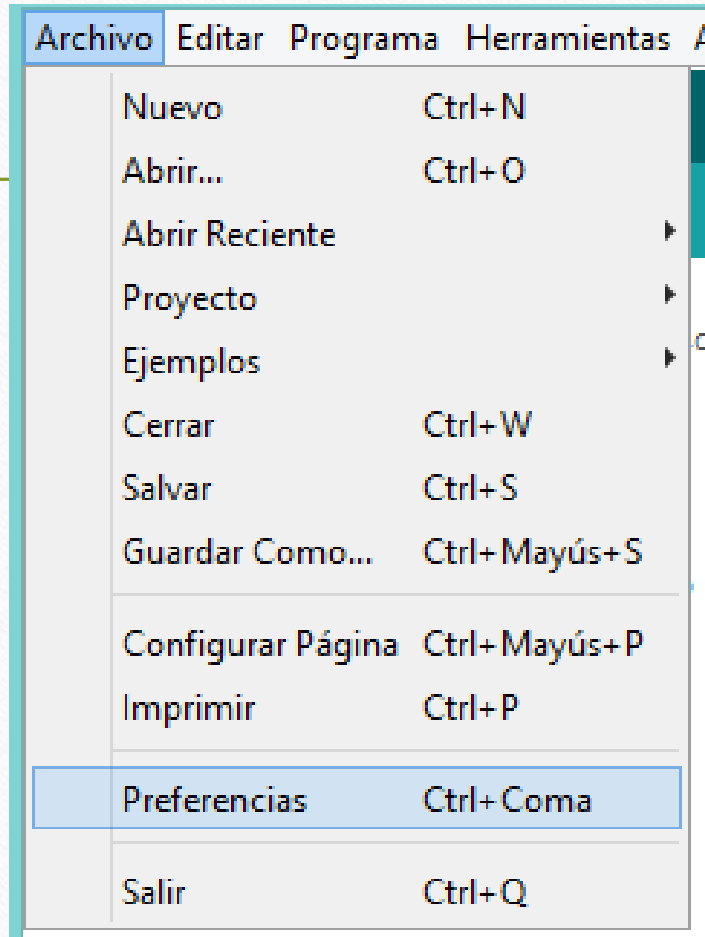


Esta instalación es compatible a partir de la versión 1.6 en adelante.



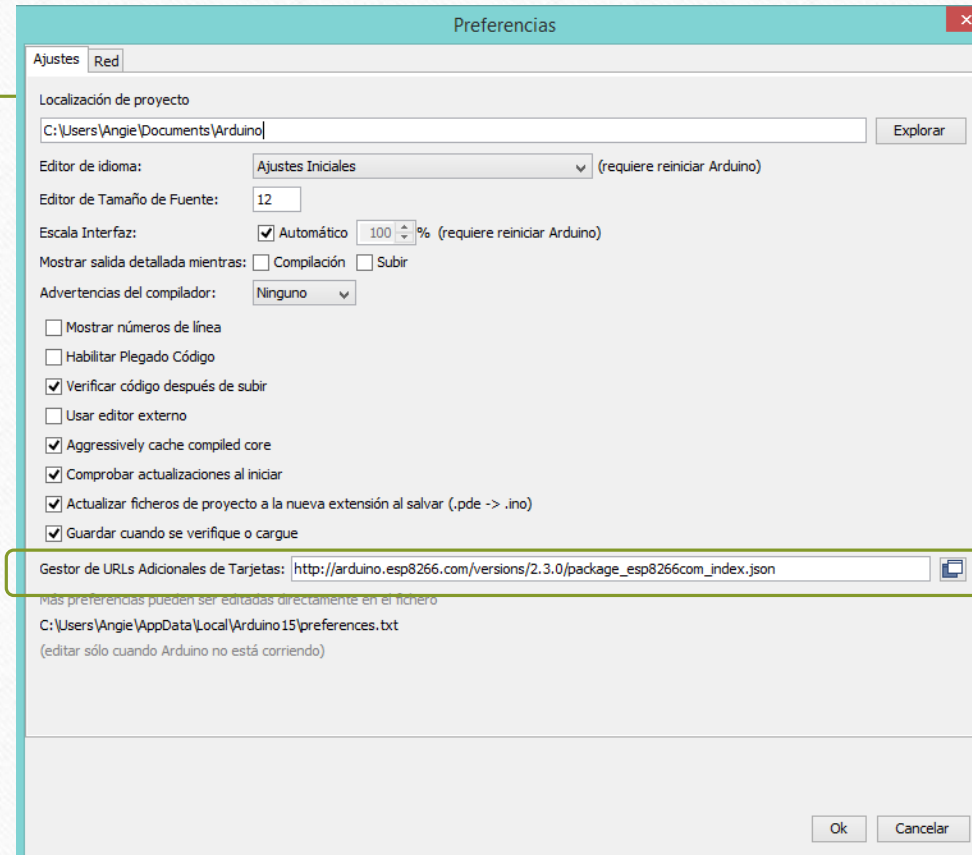


2. debemos agregar la placa a nuestro IDE



En preferencias agregamos el siguiente link:

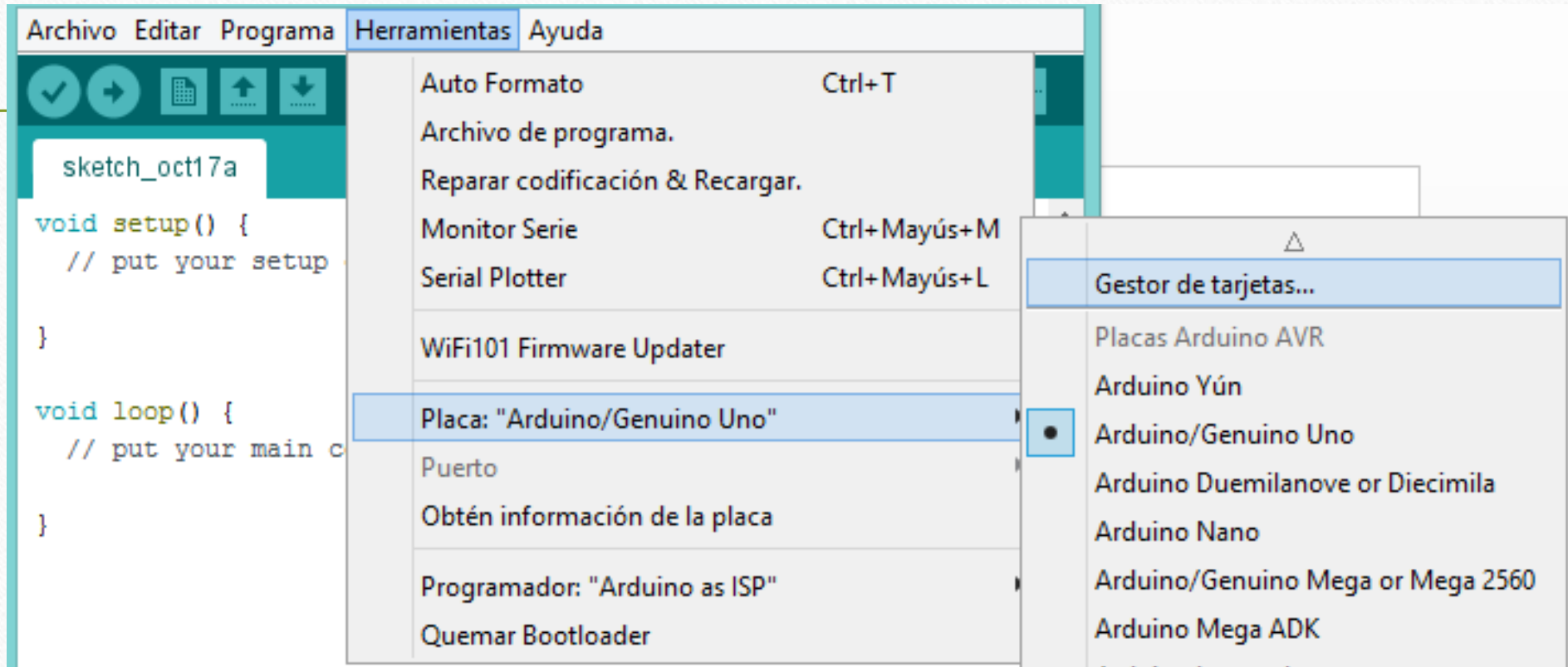
[http://arduino.esp8266.com/versions/2.3.0/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json)



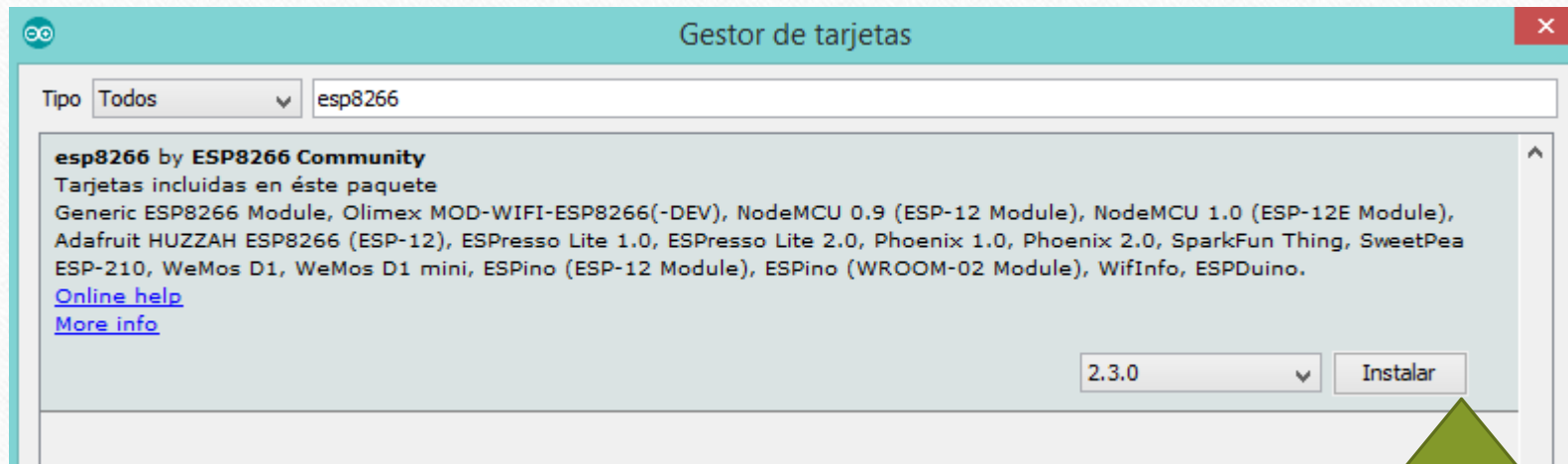
Copiamos el  
link y le  
damos click a  
OK



### 3. Entramos a nuestro gestor de tarjetas.



## 4. Buscamos la tarjeta “ESP8266”



Elegimos la ultima versión disponible y le damos a instalar



5. Cuando se instale la tarjeta nos mostrara lo siguiente:



# Ejercicios de aplicación

---



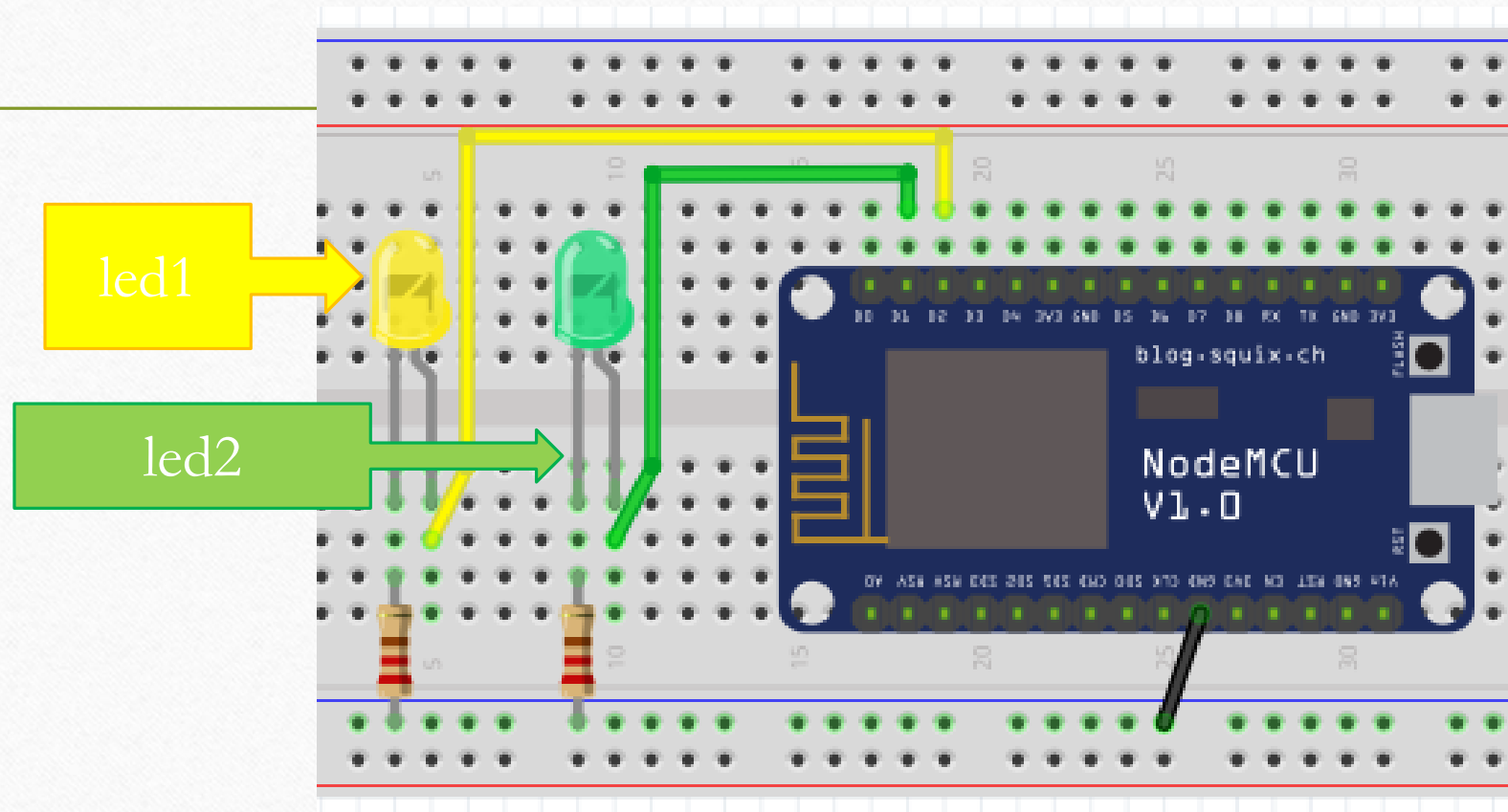


# 1. Encendido y apagado de dos led's con el NodeMCU Amica

```
#include <ESP8266WiFi.h>
#define led1 D2 // definimos el pin digital 2
#define led2 5 // definimos el GPIO 5
const char* ssid = "Angie";//nombre de red wifi
const char* password= "@ngie17Miranda";//contraseña de red WiFi
void setup() {
    pinMode(led1,OUTPUT); //pin digital 2 de salida
    pinMode(led2,OUTPUT); //pin GPIO 5 de salida
    Serial.begin(115200); //velocidad de comunicacion optima para el node mcu
    Serial.println();
    Serial.println("conectando...");
    Serial.println(ssid);
    WiFi.begin(ssid,password); //inicia la conexion a la red WiFi
    //verificacion de que se conecto a la red WiFi
    while(WiFi.status() !=WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.print("");
    Serial.println("WiFi conectado");
    Serial.println("Direccion ip: ");
    Serial.println(WiFi.localIP()); // direccion IP del node mcu
}
```

```
void loop() {
    digitalWrite(led1,HIGH);
    delay(500);
    digitalWrite(led1,LOW);
    delay(300);
    digitalWrite(led2,HIGH);
    delay(600);
    digitalWrite(led2,LOW);
    delay(400);
}
```

# Conexión





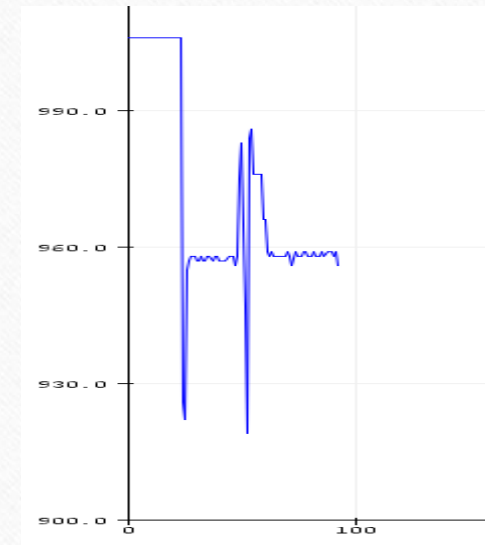
## 2. Lectura analógica con el NodeMCU Amica

```
#include <ESP8266WiFi.h>
#define pot A0 // definimos el pin analogico
const char* ssid = "Angie";//nombre de red wifi
const char* password= "@ngie17Miranda";//contraseña de red WiFi
void setup() {
    Serial.begin(115200); //velocidad de comunicacion optima para el node mcu
    Serial.println();
    Serial.println("conectando...");
    Serial.println(ssid);
    WiFi.begin(ssid,password); //inicia la conexion a la red WiFi
    //verificacion de que se conecto a la red WiFi
    while(WiFi.status() !=WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.print("");
    Serial.println("WiFi conectado");
    Serial.println("Direccion ip: ");
    Serial.println(WiFi.localIP()); // direccion IP del node mcu
}
```

# Lectura analógica con el NodeMCU Amica

---

```
void loop() {  
  Serial.println(analogRead(pot)); // realizar la lectura analógica  
  delay(1000);  
}
```





# Conexión

