

# ***Sesión II***

Programación en NodeMcu



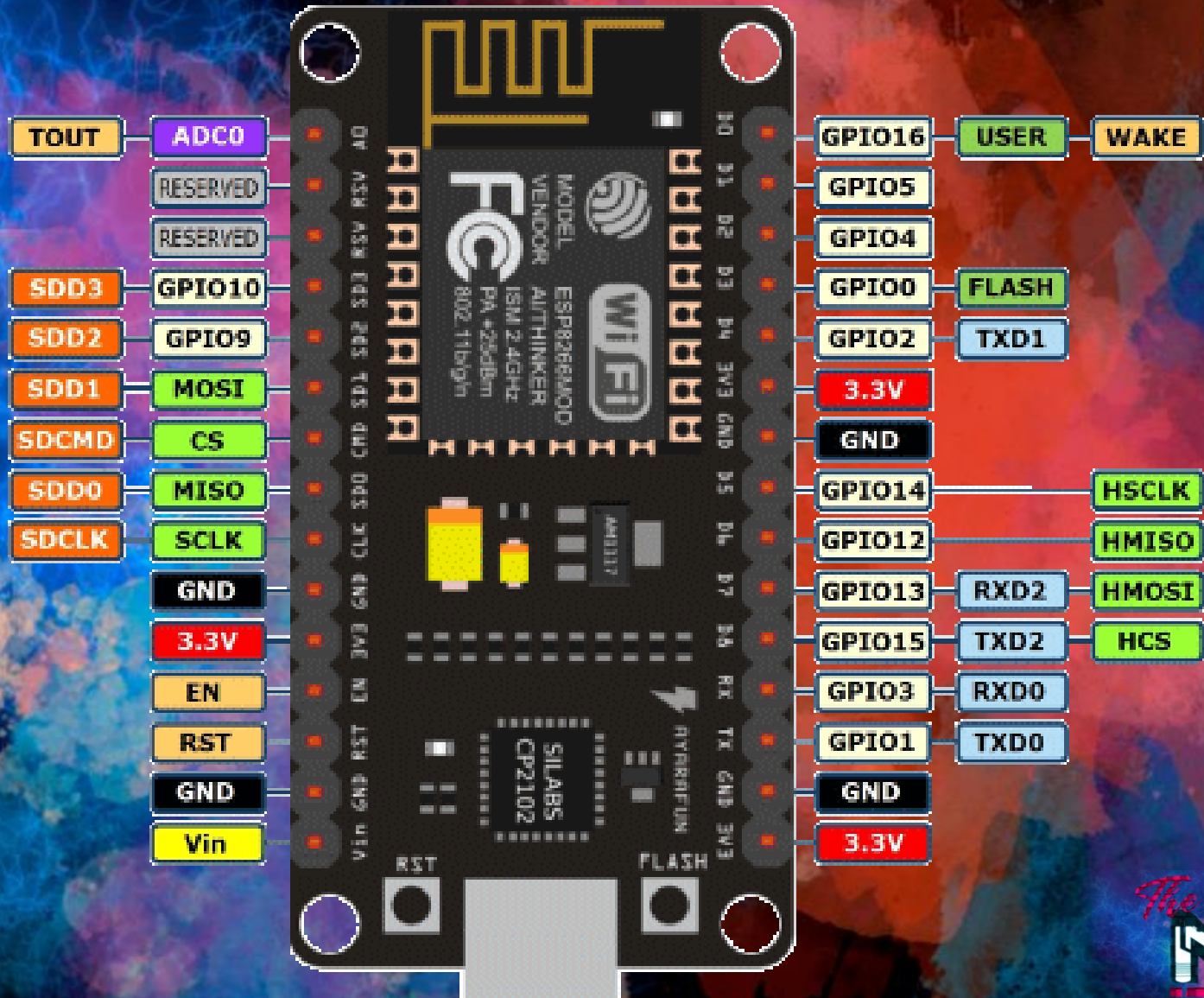


# ***Temas de avance***

- Pines básicos NodeMcu
- Entorno de Desarrollo Integrado
- Instalación y configuración del IDE Arduino para ESP8266.
- Preguntas



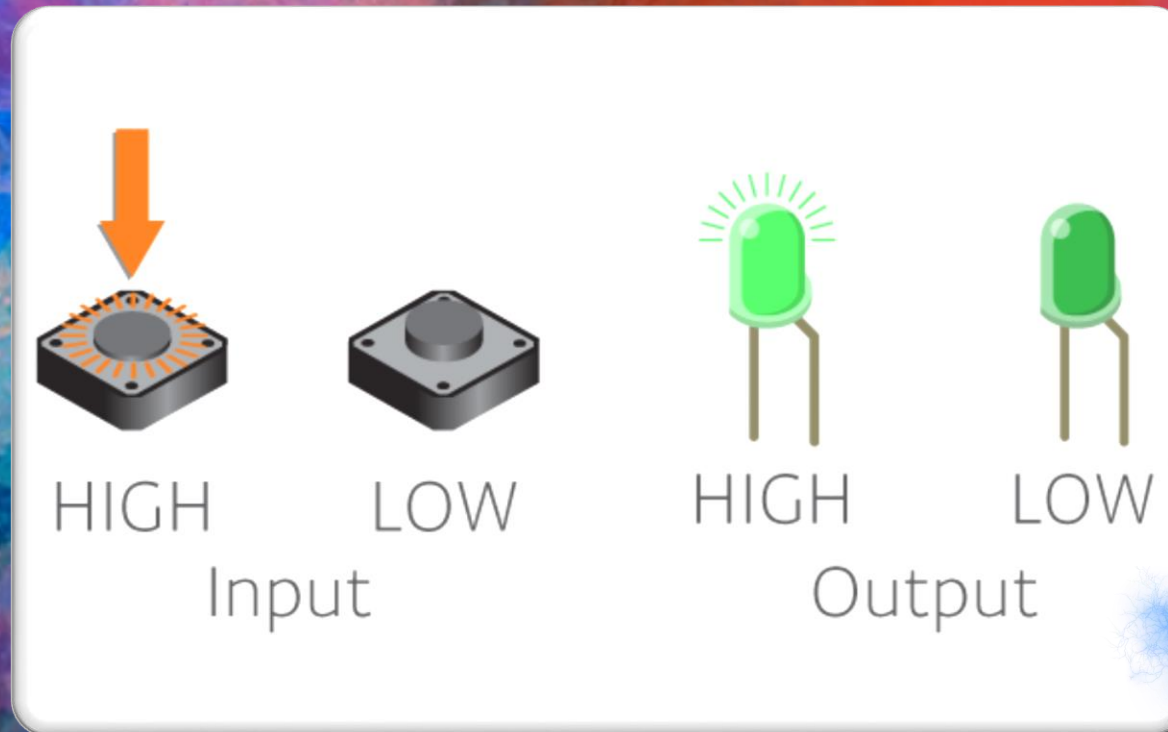
# MAPA DE PINES NODE MCU





# ***PINES DIGITALES***

- Funcionan tanto como entrada y salida digital.
- La señal digital puede estar o ENCENDIDA O APAGADA (HIGH o LOW).
- Los pines 0 y 1 se pueden utilizar para cargar el programa en la placa, ya que estos son de transmisión y recepción de datos (Tx y Rx respectivamente).





# ***PINES ANALÓGICOS***

## **Salidas Analógicas**

- Estos pines se denominan PWM (Pulse Width Modulation).
- En realidad son pines digitales que imitan salidas analógicas modificando la separación entre los diferentes pulsos de la señal.
- La señal PWM puede dar diversos valores hasta 255 incluyendo el 0, se utilizan por ejemplo para variar la intensidad de un led o hacer funcionar un Servomotor.
- Hay que hacer notar que estos pines funcionan como salidas o entradas digitales o como salidas analógicas.



0



64



128



192



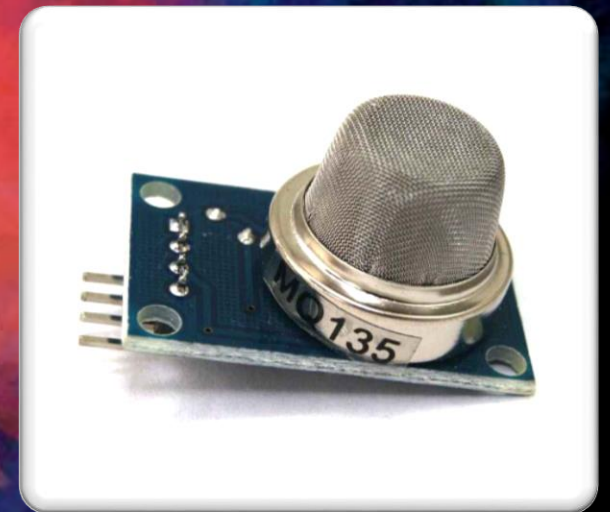
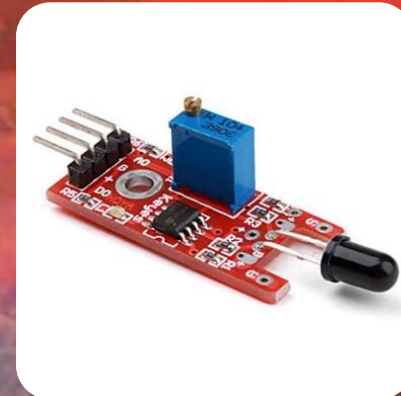
255



# ***PINES ANALÓGICOS***

## **Entradas Analógicas**

- Se utilizan para que entre una señal de un sensor analógico, tipo un potenciómetro o un sensor de temperatura, que dan un valor variable entre 0 y 1023. También se pueden utilizar como pines digitales





# ¿QUÉ ES UN IDE?

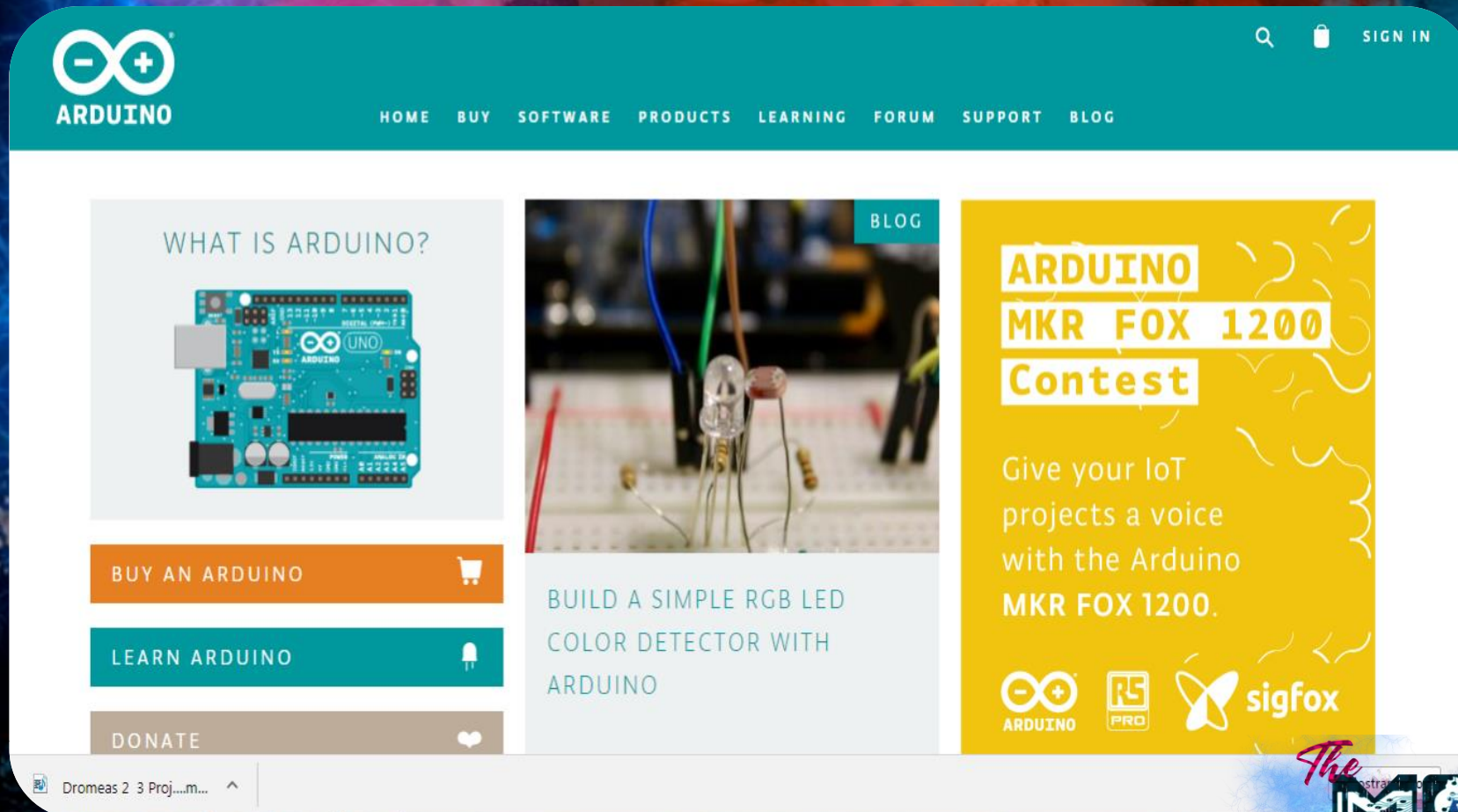
- Un IDE (Entorno de Desarrollo Integrado) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes.
- Un IDE es un editor de código, un compilador, un depurador y un constructor de interfaz grafica y en el caso del IDE de Arduino incorpora herramientas para cargar el programa ya compilado en la memoria flash del hardware.





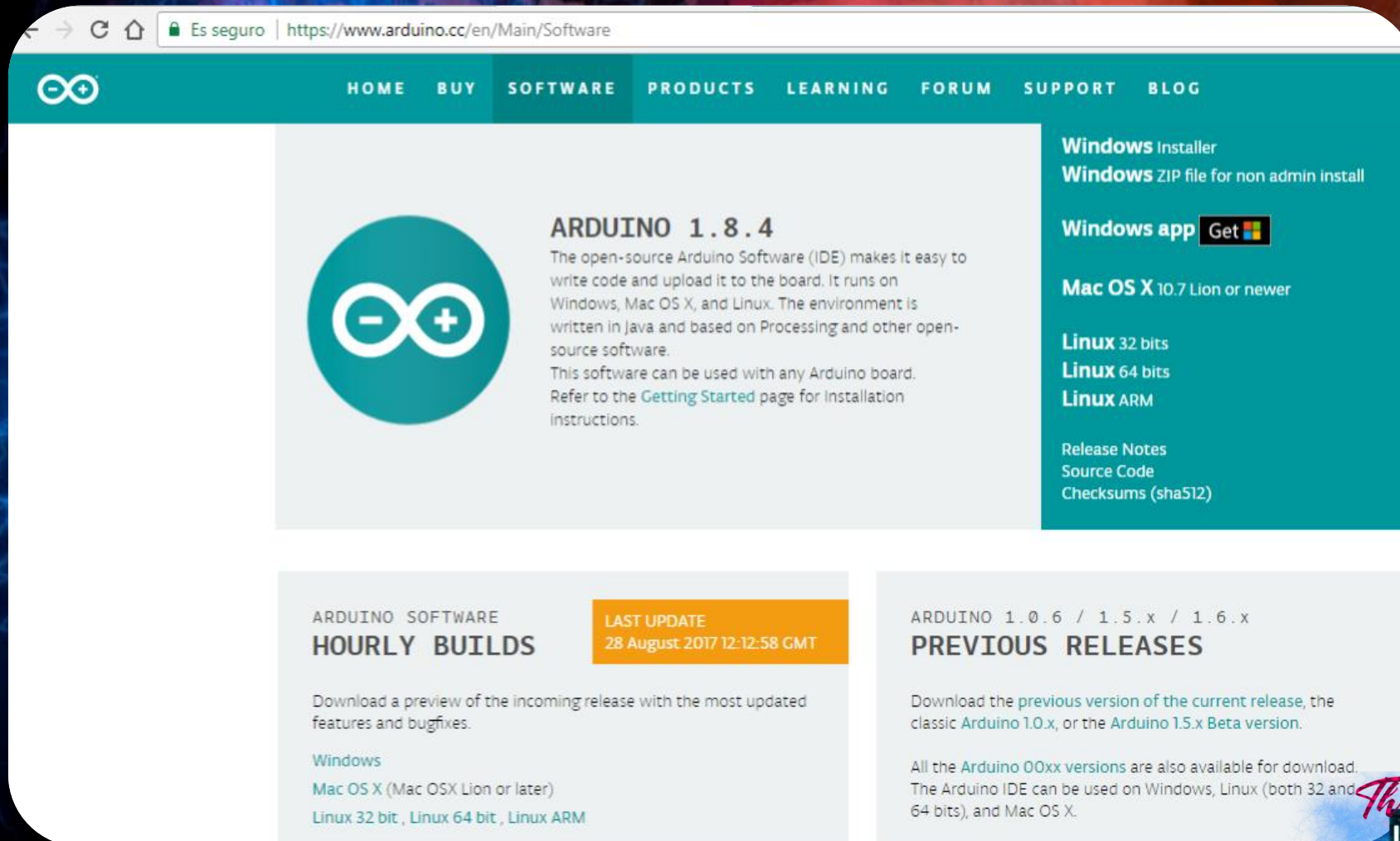
# DESCARGA E INSTALACIÓN

- Descargamos el Arduino IDE de su propia pagina web [www.arduino.cc](http://www.arduino.cc)





- Nos dirigimos a la pestaña Software para descargar el IDE.



The screenshot shows the Arduino website's 'Software' page. The browser address bar indicates the URL is <https://www.arduino.cc/en/Main/Software>. The page features a teal header with navigation links: HOME, BUY, SOFTWARE (active), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The main content area is divided into three sections. The top section highlights 'ARDUINO 1.8.4', describing it as open-source software for Windows, Mac OS X, and Linux, written in Java and based on Processing. It includes a large Arduino logo and a description of its capabilities. To the right of this section, there are links for 'Windows Installer', 'Windows ZIP file for non admin install', 'Windows app' (with a 'Get' button), 'Mac OS X 10.7 Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', 'Release Notes', 'Source Code', and 'Checksums (sha512)'. The bottom left section is titled 'ARDUINO SOFTWARE HOURLY BUILDS' and includes a 'LAST UPDATE' badge for '28 August 2017 12:12:58 GMT'. It describes the availability of preview builds and lists supported operating systems: Windows, Mac OS X (Mac OSX Lion or later), and Linux (32 bit, 64 bit, ARM). The bottom right section is titled 'ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES' and provides information about downloading previous versions, including the classic 1.0.x and 1.5.x Beta versions, and mentions that all Arduino 00xx versions are also available for download.

Es seguro | <https://www.arduino.cc/en/Main/Software>

HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

**ARDUINO 1.8.4**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows Installer**  
**Windows** ZIP file for non admin install

**Windows app** [Get](#)

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

ARDUINO SOFTWARE  
**HOURLY BUILDS**

**LAST UPDATE**  
28 August 2017 12:12:58 GMT

Download a preview of the incoming release with the most updated features and bugfixes.

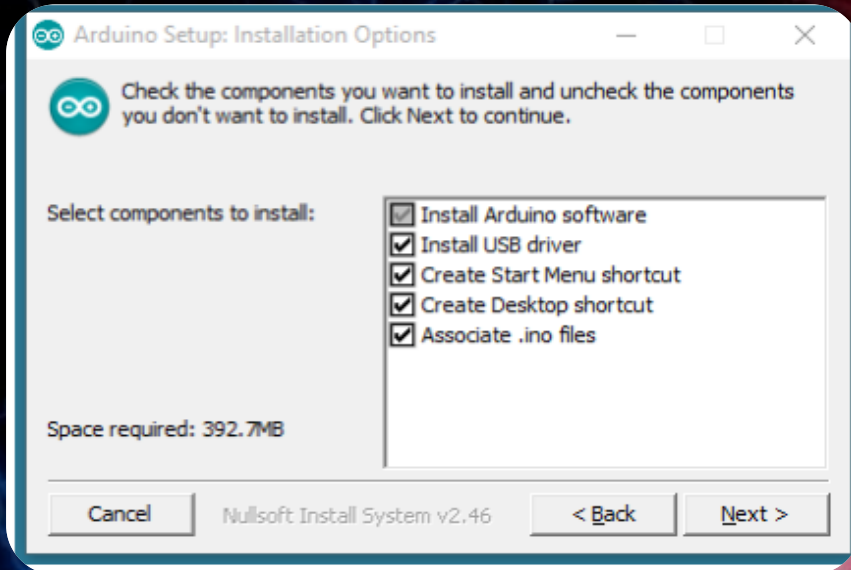
[Windows](#)  
[Mac OS X](#) (Mac OSX Lion or later)  
[Linux](#) 32 bit , [Linux](#) 64 bit , [Linux](#) ARM

ARDUINO 1.0.6 / 1.5.x / 1.6.x  
**PREVIOUS RELEASES**

Download the previous version of the current release, the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta](#) version.

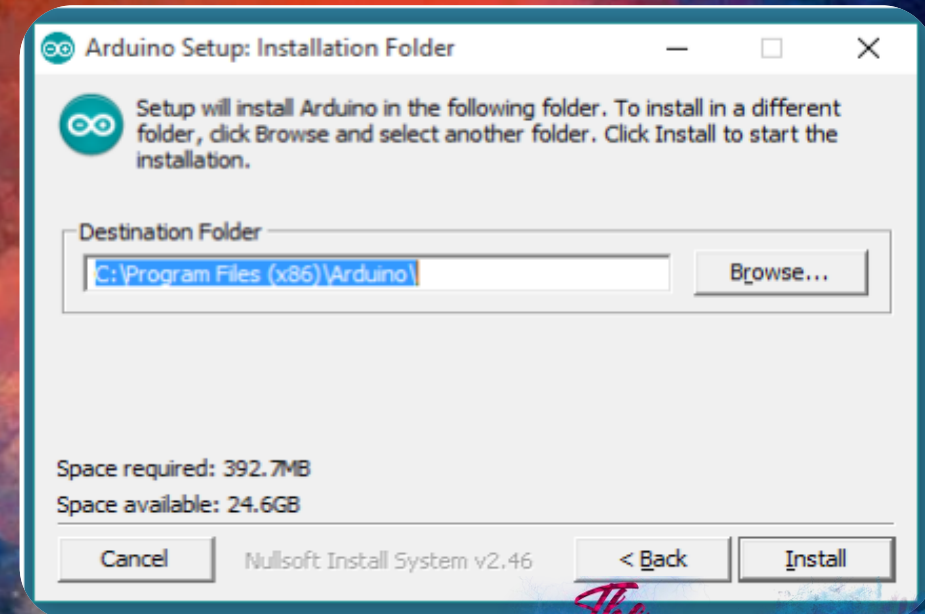
All the [Arduino 00xx](#) versions are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.





- Verificamos que todas las opciones estén marcadas.

- Seleccionamos la ruta donde queremos que se instale el IDE Arduino. Seleccionamos “Instalar” y listo.





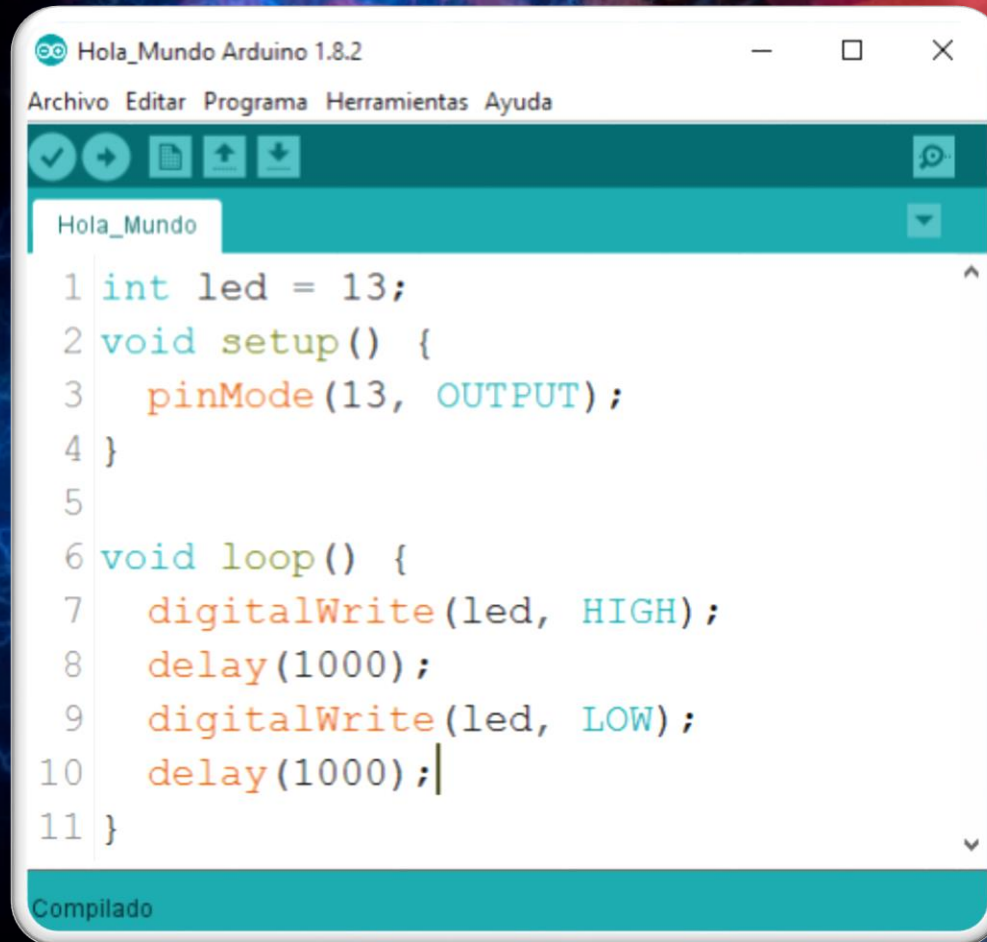
# IDE ARDUINO

- Arduino nos proporciona un **software** consistente en un entorno de desarrollo (IDE) para la programación de arduino y un bootloader cargado en el microcontrolador para facilitar el uso de Arduino. La principal característica del software y del lenguaje de programación de Arduino es su sencillez y facilidad de uso.
- La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing que es similar a C++.





# IDE ARDUINO



The screenshot shows the Arduino IDE 1.8.2 window. The title bar reads 'Hola\_Mundo Arduino 1.8.2'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar contains icons for opening, saving, uploading, and downloading. The file explorer on the left shows 'Hola\_Mundo'. The main text area contains the following C++ code:

```
1 int led = 13;
2 void setup() {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop() {
7   digitalWrite(led, HIGH);
8   delay(1000);
9   digitalWrite(led, LOW);
10  delay(1000);
11 }
```

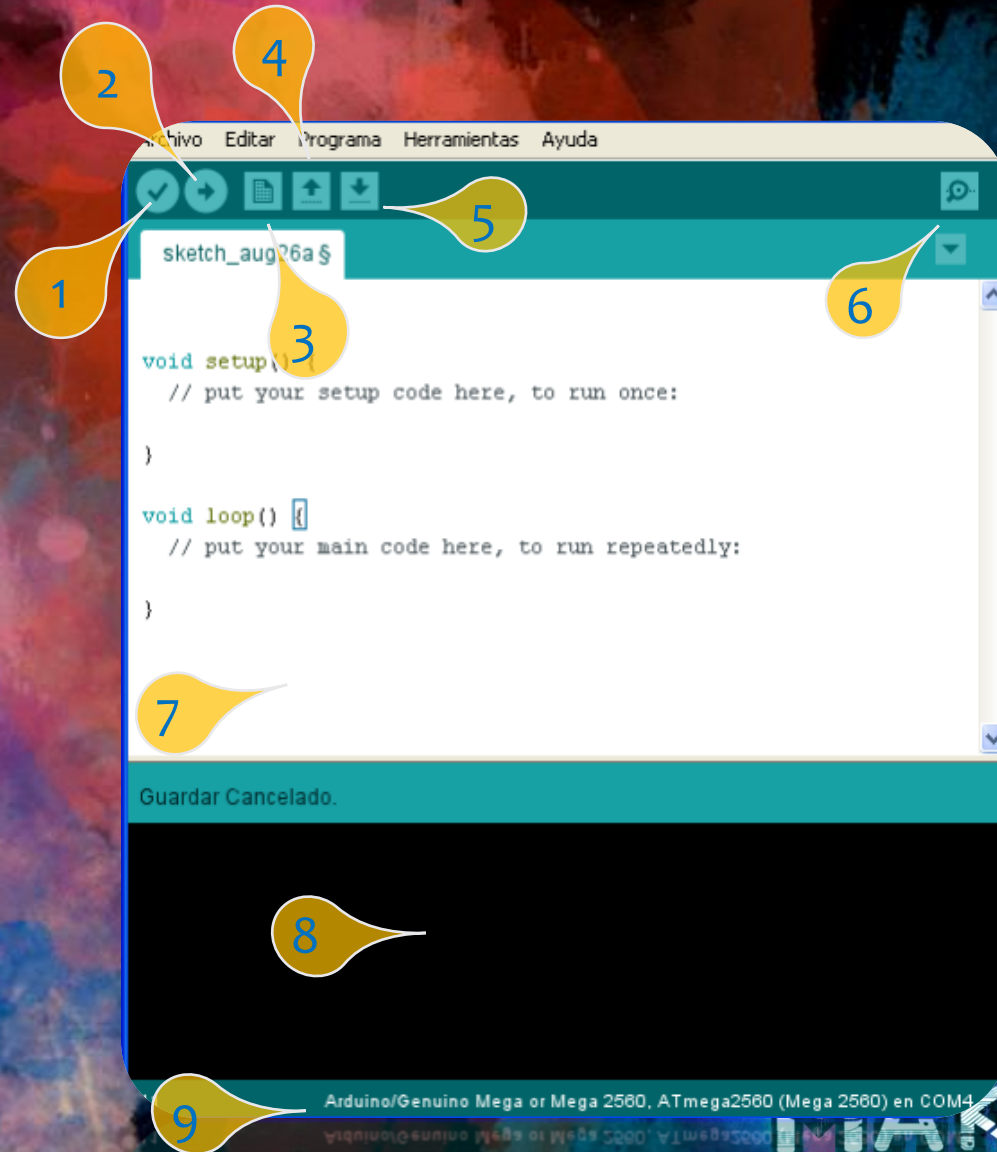
The status bar at the bottom indicates 'Compilado'.

- En el microcontrolador podemos grabar instrucciones utilizando una conexión entre una **computadora** y el **Arduino** a través de **USB**. El lenguaje Arduino, derivado de C, es mucho más amigable que el utilizado por otros microcontroladores, el Assembler.
- El lenguaje de programación multifuncional con el que trabaja la **placa Arduino** puede trabajar con muchos otros lenguajes



# IDE DE ARDUINO

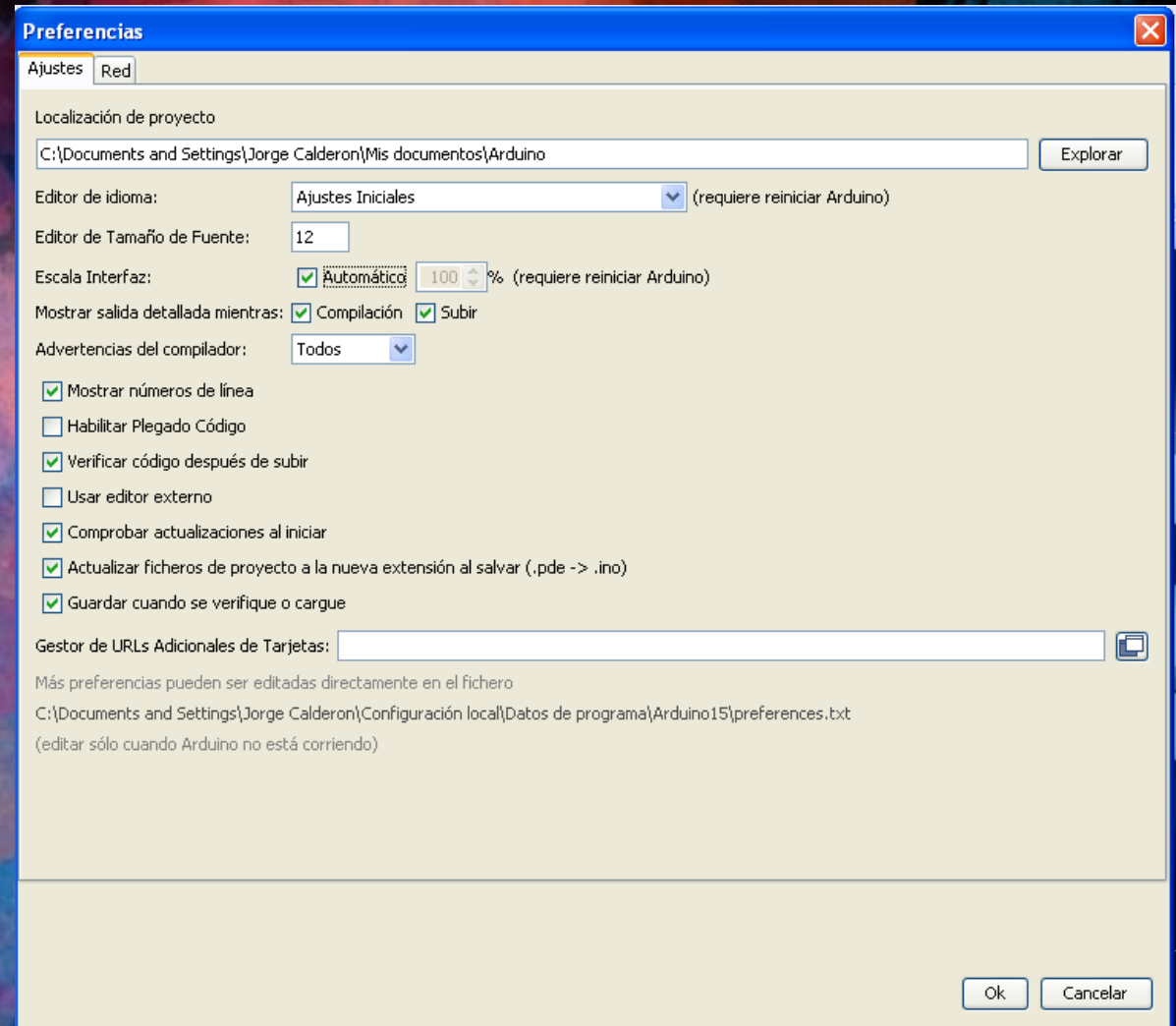
1. Verifica si tu programa esta bien escrito y puede funcionar.
2. Carga el programa a la placa de Arduino tras compilarlo.
3. Crea un nuevo programa (Sketch).
4. Abre un programa.
5. Guarda el programa.
6. Monitor Serial, abre una ventana de comunicación con la placa Arduino en la que podemos ver las respuestas que nuestro Arduino nos esta dando, siempre que tengamos el USB conectado.
7. Cuadro editor de texto.
8. Área de mensajes y la consola Arduino.
9. Información del modelo de placa y puerto donde se encuentra conectado.





# ENTORNO DE PROGRAMACIÓN Y CONFIGURACIÓN

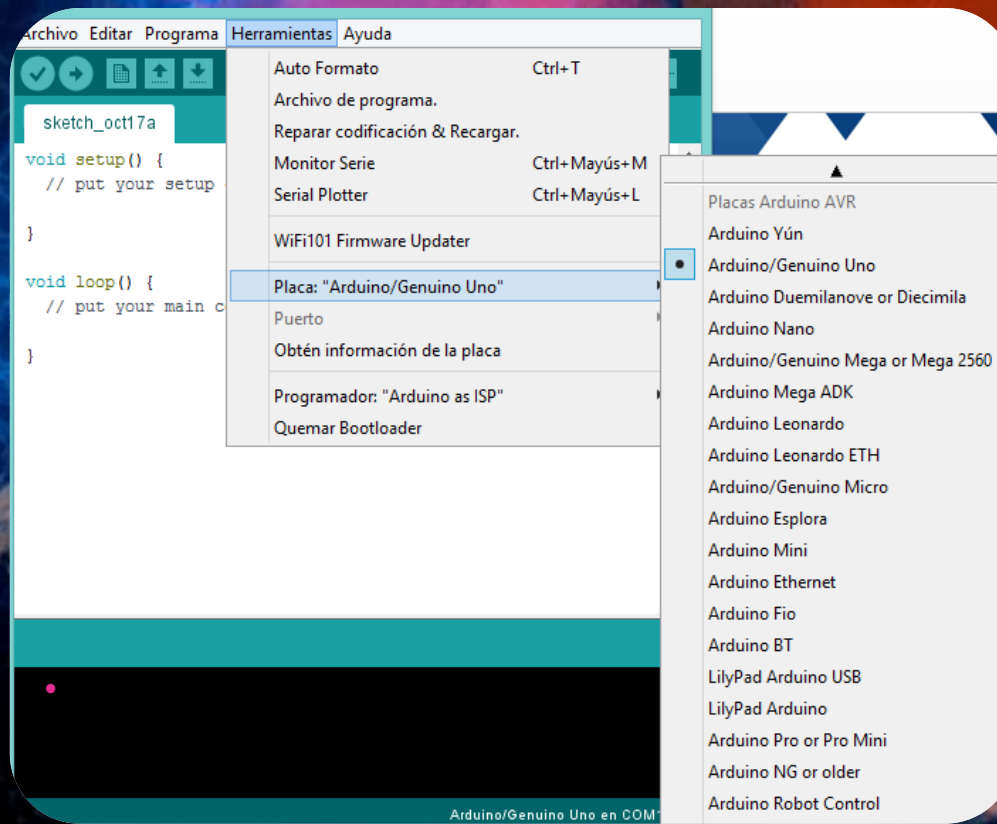
- Para configurar algunas opciones del entorno hacemos click en la parte superior izquierda en la pestaña Archivos luego a Preferencias.



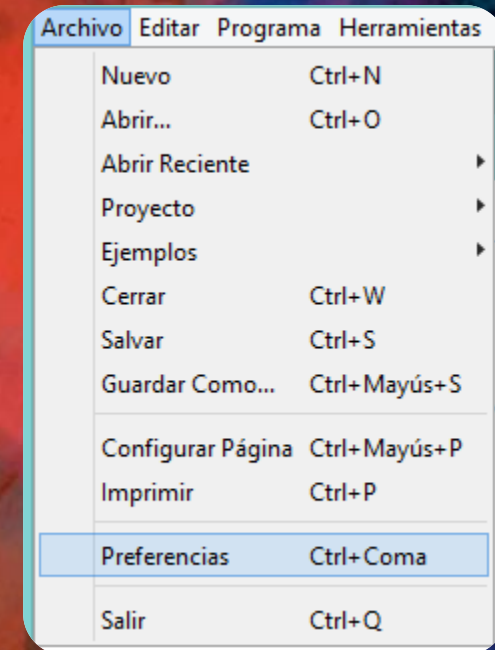


# INSTALACION DE PLACA

- Verificamos si la placa esta instalada

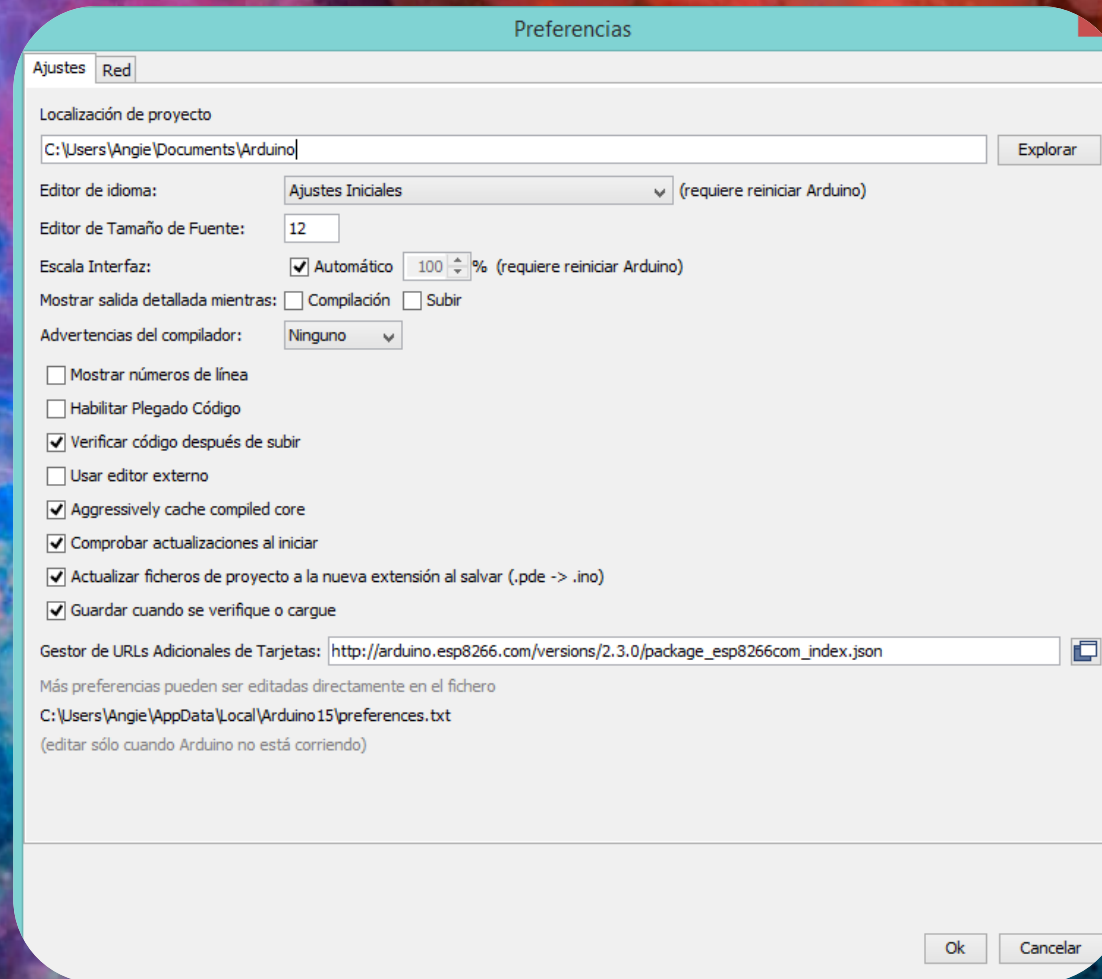


- Debemos agregar nuestra placa a nuestro IDE





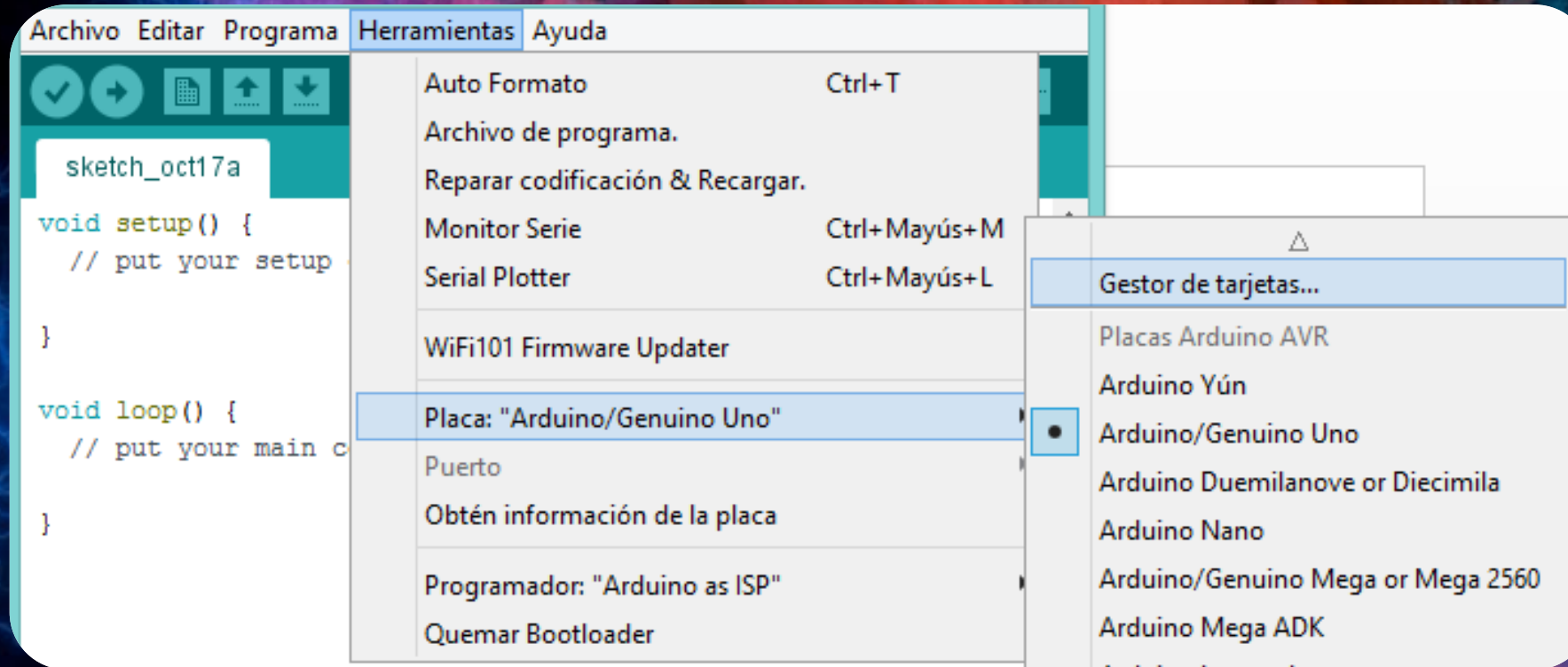
- En preferencias agregamos el siguiente link:
- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



Copiar el link  
y le damos  
click en OK

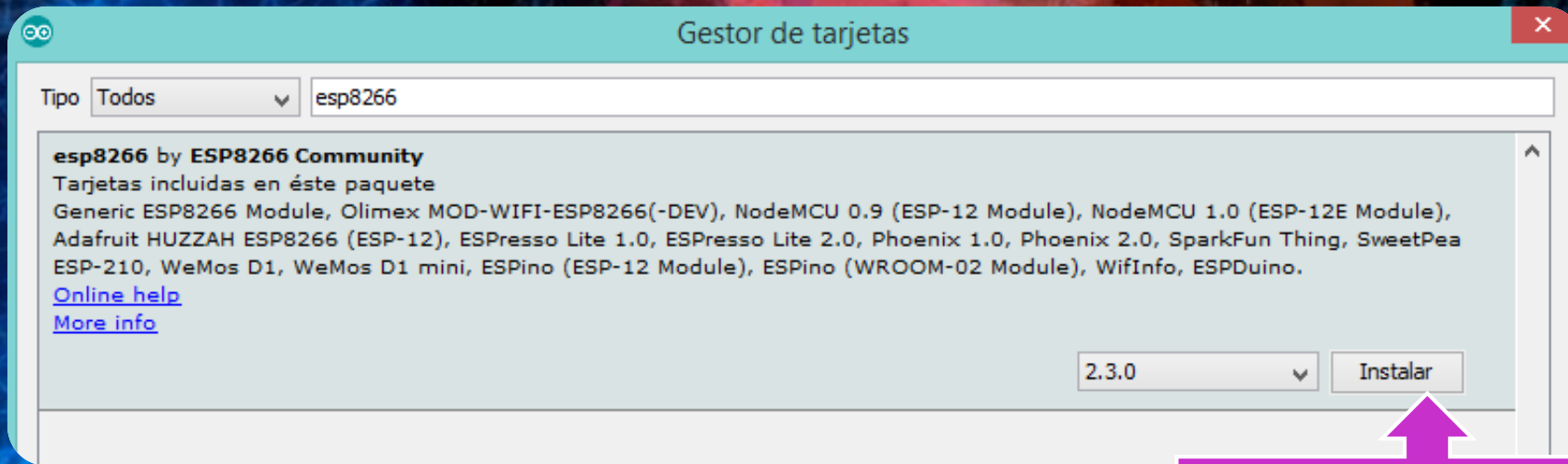


- Entramos al gestor de tarjetas





- Buscamos la tarjeta “ESP8266”



Elegimos la ultima versión disponible y le damos a instalar

- Después de instalarlo nos mostrara:





# INSTALACIÓN OFFLINE

Para realizar la instalación y configuración de IDE de Arduino para poder trabajar con nuestra Placa de Desarrollo Node Mcu debemos contar con lo siguiente:

- **Arduino15.zip**, este zip contiene todas las herramientas necesarias que necesita el IDE de Arduino para trabajar con nuestra Placa Node Mcu.
- Instalador del IDE de Arduino. (En caso de no tenerlo instalado)
- **PASO 1.-** Instalar el IDE de Arduino de manera normal y como ya se conoce.
- **PASO 2.-** Ingresar a la siguiente ubicación dentro de su equipo:

C:\Users\Nombre\_Usuario\AppData\Local\Arduino15

DONDE Nombre\_Usuario SERA EL NOMBRE POR DEFECTO QUE LE HAYAN DADO A SU USUARIO

- **PASO 3.-** Una vez en esa dirección ELIMINAR TODO el contenido que viene por defecto en esa dirección y reemplazarlo por lo que contiene el comprimido ARDUINO15.ZIP
- **PASO4.-** Abrir el IDE de Arduino de manera normal y ya tendrá instaladas las placas compatibles con ESP8266



# ***Bloques Principales.***

- **Setup()**

Es la primera función de ejecutarse dentro de un programa.

Aquí es donde establecemos algunos criterios que requieren una ejecución única.

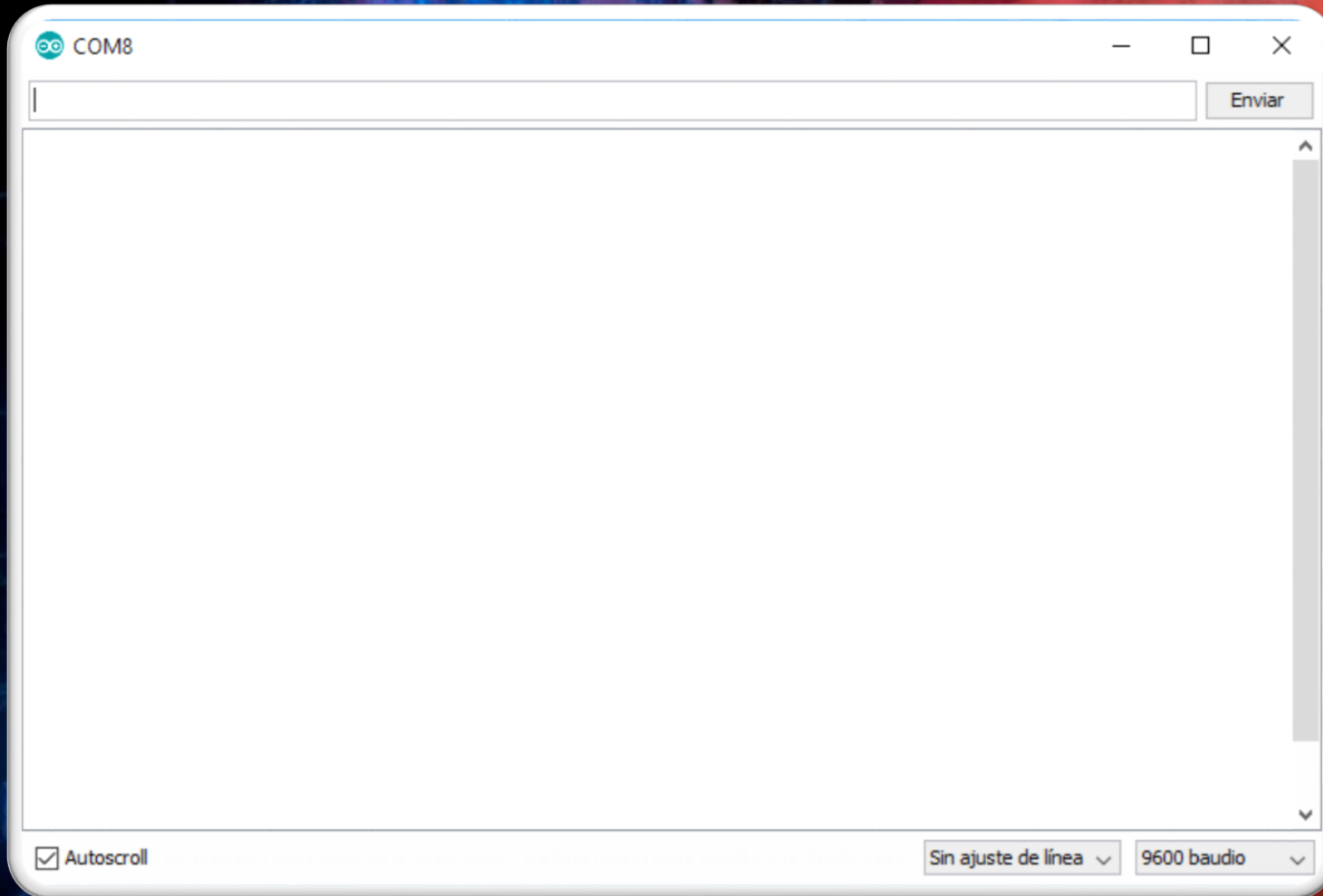
- **Loop()**

Es la función que se ejecuta un número infinito de veces.

El loop se repite de forma indefinida hasta que se apague o se reinicie el microcontrolador.



# Monitor Serial



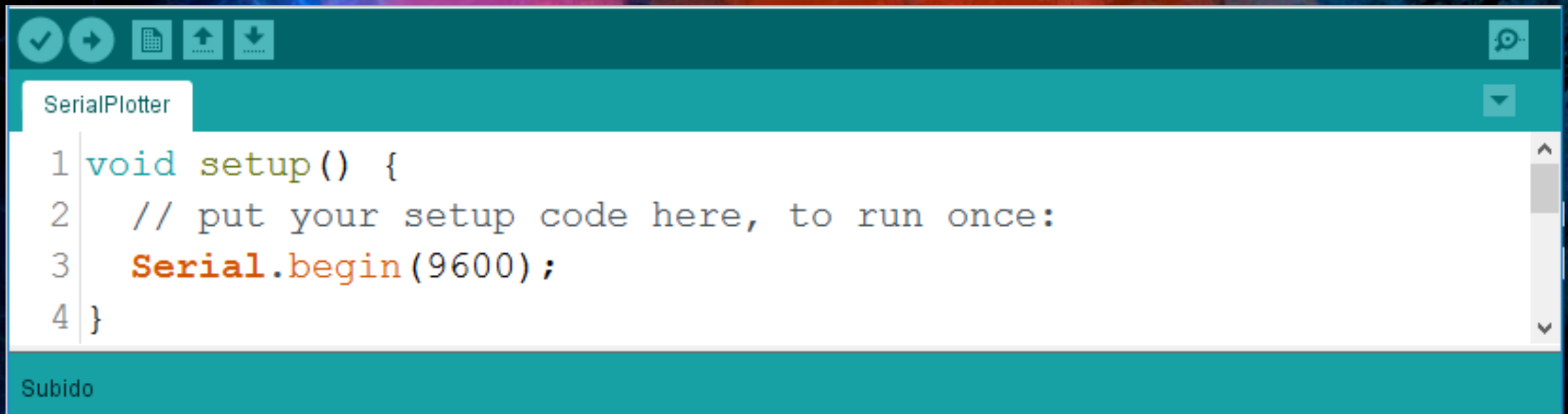
Esta herramienta nos permite enviar y visualizar los datos que se manejan a través del puerto serie.

Es la forma más simple que existe para establecer la comunicación serial.



# Monitor Serial

- Para iniciar la comunicación serial utilizando el monitor serial debemos establecer algunos comandos en el IDE y luego subirlos al microcontrolador.



```
1 void setup() {  
2   // put your setup code here, to run once:  
3   Serial.begin(9600);  
4 }
```

Subido

- ▶ El 9600 indica la cantidad de baudios que manejará el puerto serie en esta ocasión.
- ▶ Se define baudio como una unidad de medida que representa al números de símbolos por segundo en un medio de transmisión



***--Preguntas--***