

Práctica 1

NodeMCU y Aplicación Móvil

Introducción

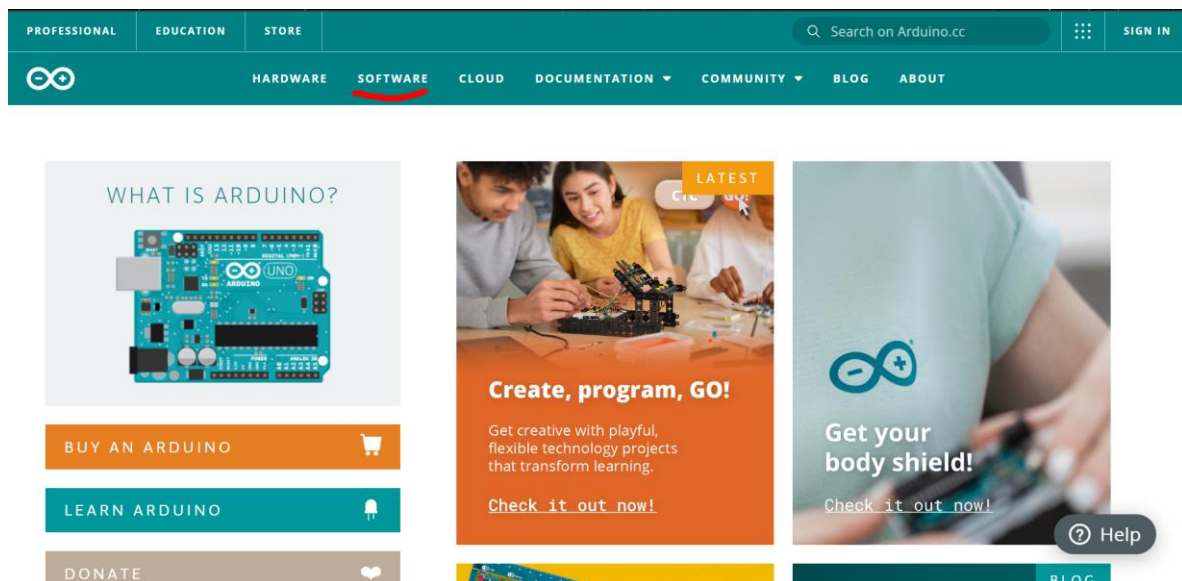
Las aplicaciones de IoT requieren conexiones Internet de gran confiabilidad para intercambiar paquetes de datos compactos, pero para los cuales resulta crítico que el tiempo de tránsito entre el sensor y el servidor, o entre el servidor y el actuador, sea lo más reducido posible (lo cual da nombre al concepto de 'baja latencia'); con estos requerimientos, muchas de las aplicaciones no necesitan almacenar información en bases de datos relacionales y estructuradas, en vez de esto requieren un sistema que opere en tiempo real interconectando a múltiples dispositivos que pueden compartir fuentes de datos y recursos volátiles de almacenamiento.

Para esta práctica, usaremos una aplicación móvil la cual controlará, por medio de Wi-Fi, el prendido y apagado de un led que estará la tarjeta NodeMCU.

Instalación de Arduino

Para el uso de estas aplicaciones, se va a tener que descargar Arduino como software ya que ahí se van a correr los programas para el uso de NodeMCU.

Como primer paso, se necesitará entrar a la página de <https://www.arduino.cc/> y ubicar la pestaña de “Software”.



Dentro de la pestaña de “Software” se encuentra la sección de “Downloads” o “Descargas”. Aquí habrá varias opciones para instalar el programa según el sistema operativo de su agrado. Para estas prácticas, todo se correrá en Windows. Para esto, la primera opción es la que nos permitirá instalar el programa en Windows 7 para adelante. Damos clic en la opción y procedemos a descargarlo.

Downloads



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- [Windows](#) Win 7 and newer
 - [Windows](#) ZIP file

[Windows app](#) Win 8.1 or 10 [Get](#)
- [Linux](#) 32 bits

[Linux](#) 64 bits

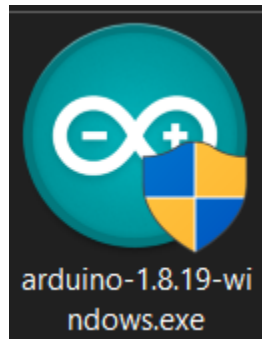
[Linux](#) ARM 32 bits

[Linux](#) ARM 64 bits
- [Mac OS X](#) 10.10 or newer

[Release Notes](#)

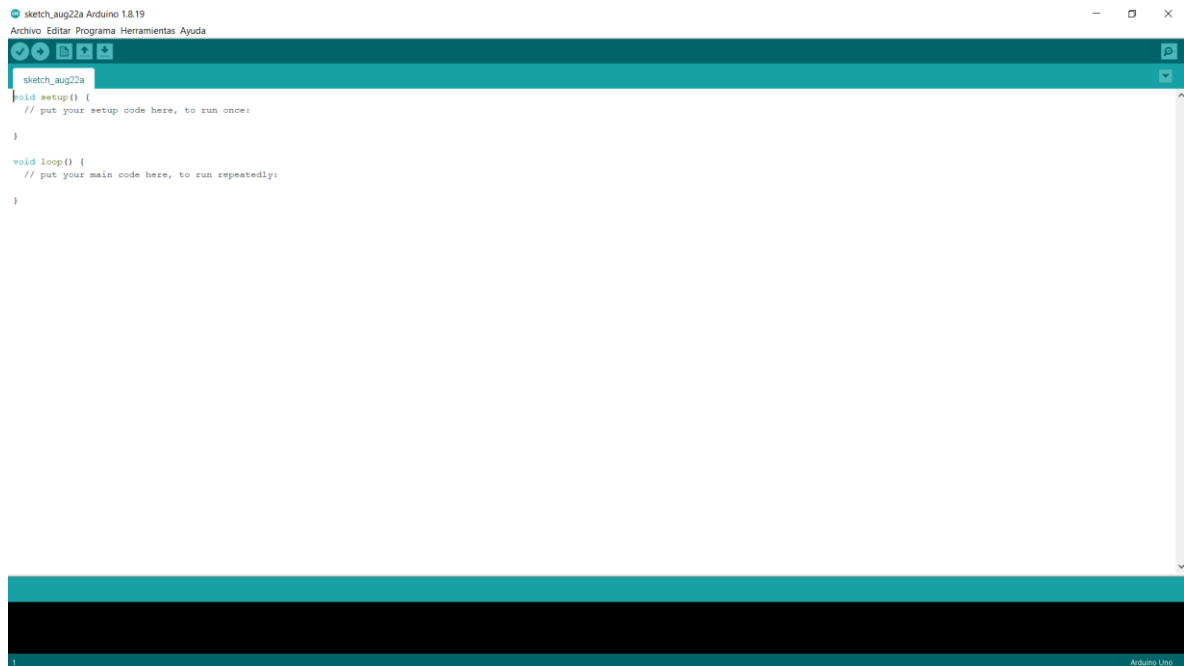
[Checksums \(sha512\)](#)

Ya con el software descargado, se ejecuta el instalador y se da aceptar en todo lo que nos pide. Seleccionamos donde instalarlo y se tendrá que permitir instalar los Drivers que permitan aceptar la conexión de los dispositivos externos.



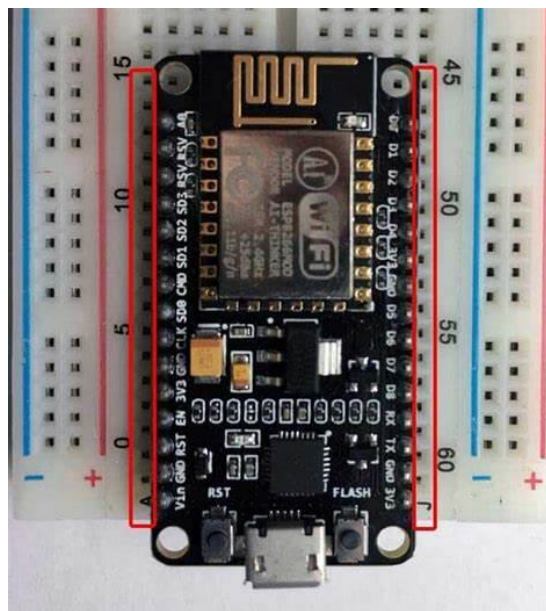
La versión que se descargué dependerá del momento de la descarga, pero por lo regular se instalará la versión más actual que haya en el momento. Esto no cambiará nada el uso del software.

Una vez instalado podemos ubicar el acceso directo en el escritorio, si así lo quisimos al momento de la instalación. Lo ejecutamos, damos permisos si así lo pide y podemos entrar a la interfaz gráfica del programa, lista para escribir código y comenzar a arrancar los programas de interés.



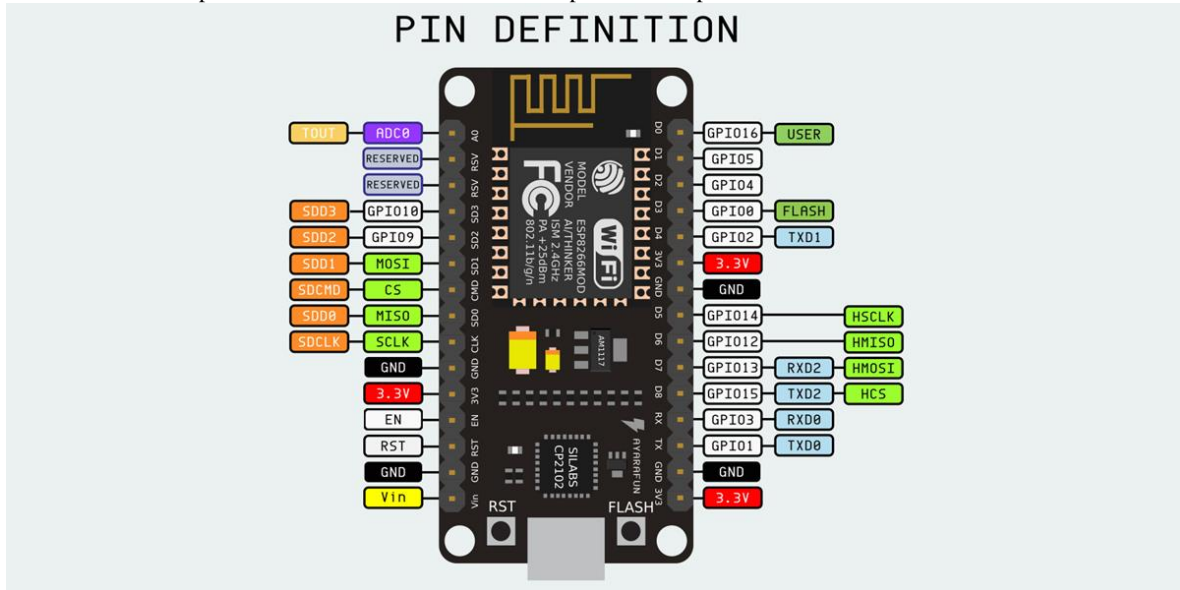
Configuración de la tarjeta NodeMCU

NodeMCU es una placa de desarrollo en el ESP12E el cuál es probablemente el módulo más popular que integra el System on Chip (SoC) ESP8266. Sin embargo, pese a la popularidad de las placas NodeMCU, también existe mucha confusión respecto a la terminología (Lua, Lolin, versiones) y, en ocasiones, se mezclan o incluso se usan como sinónimos. En este caso la versión a usar será la V2 fabricada por Amica y se usará debido al tamaño que tiene, ya que mide alrededor de 2.7 cm y es posible conectarla a un protoboard dejando espacio para conectar más cosas en ella, como se muestra en la siguiente figura, donde se observa que sobra una franja de pines donde podremos conectar los demás sensores.

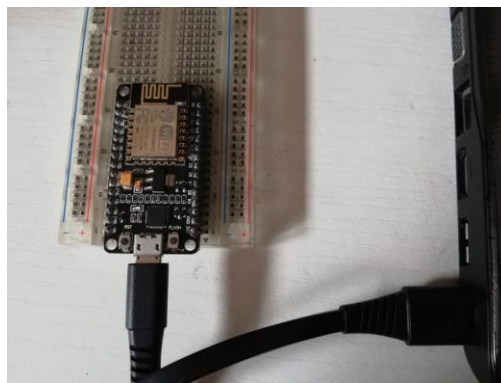


El NodeMCU cuenta con distintos pines los cuales anexaré en la siguiente imagen, pero es de importancia saber que puede recibir programas por medio de una conexión micro USB. Por medio de esta se podrá conectar a la computadora para recibir y almacenar los códigos que vayamos generando en Arduino y poder replicar las prácticas a realizar. Pero para tener una conexión entre el software de Arduino y el NodeMCU, tendremos que verificar si los drivers están bien instalados e instalar los paquetes necesarios para seleccionar la tarjeta al momento de abrir Arduino.

En la siguiente imagen anexo las conexiones que se tienen en la tarjeta NodeMCU y para que es cada una, eso ya más adelante en las prácticas se va a entender un poco más pero se anexa como información adicional.



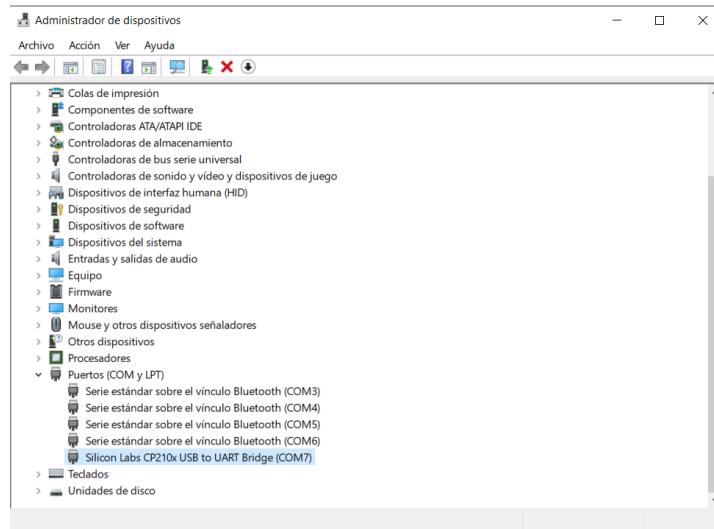
Para cerciorarnos si tenemos instalado de manera correcta el NodeMCU, lo primero que se tiene que hacer es conectar por medio de un cable la tarjeta con un puerto USB de la computadora. También es de importancia estar conectados a una red de internet ya que por lo regular la computadora descarga de manera automática los Drivers.



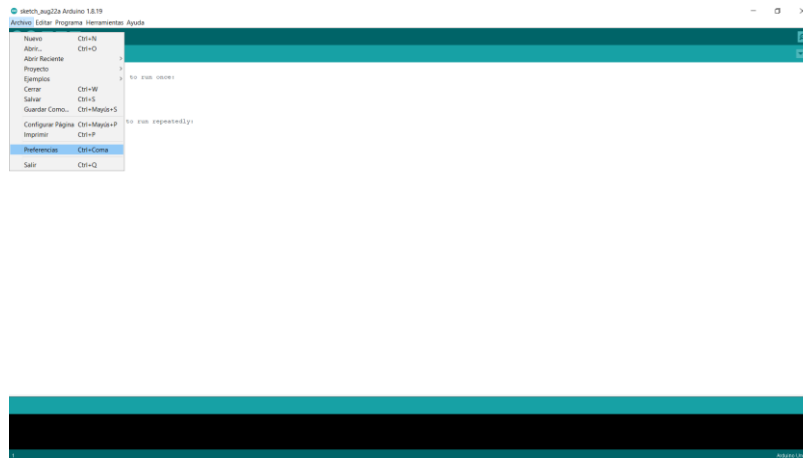
Una vez hecho esto, buscamos en la barra de búsqueda de Windows “Administrador de dispositivos” y nos dirigimos a la sección de “Puertos COM y LPT” para ver si encontramos un dispositivo que diga “**Silicon Labs CP210x USB to UART Bridge (COM N)**” donde N dependerá el puerto que la computadora le asigne a la tarjeta conectada. En caso de que no sea así, tendremos que descargar el Driver por nuestra cuenta.

Algunos lugares donde pueden descargarse son en:

- CH340 (el más común en la versión v1.0/V2)
 - <https://www.wch.cn/product/CH340.html>
- CP1202 (es el más utilizado en la versión v0.9/V1)
 - <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>



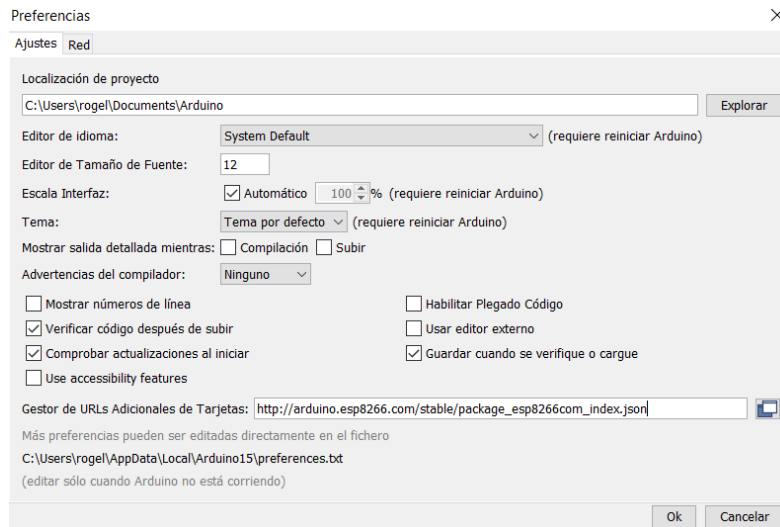
Una vez que tenemos identificado que nuestra computadora ya identifica la tarjeta NodeMCU, procedemos a abrir Arduino para instalar los paquetes que permitirán la conexión con la tarjeta. Abrimos Arduino y vamos a “Preferencias”.



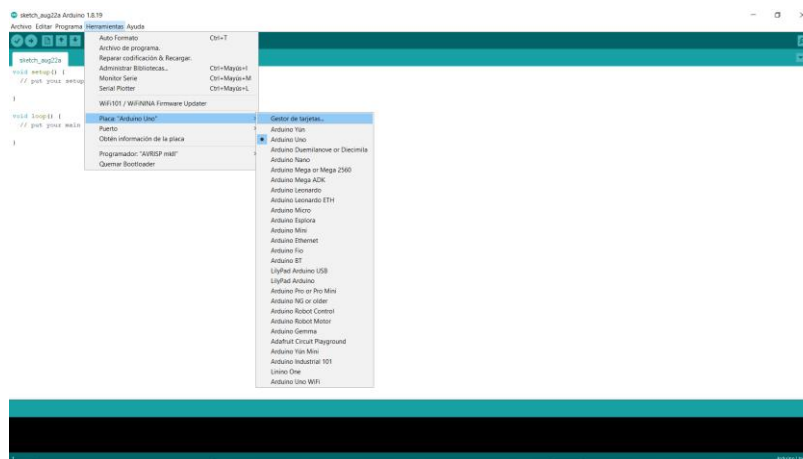
Una vez ahí, ubicamos la sección en donde dice “Gestor de URLs adicionales de Tarjetas” y pegamos la siguiente dirección:

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

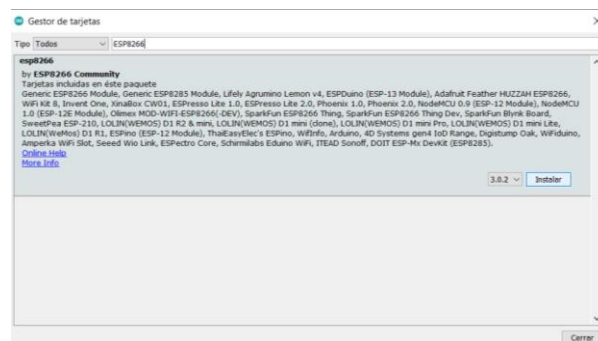
En esta URL, se encuentran descripciones de varios tipos de tarjetas. Es una base de datos con la información de las diferentes tarjetas que no se encuentran predeterminadas en Arduino.



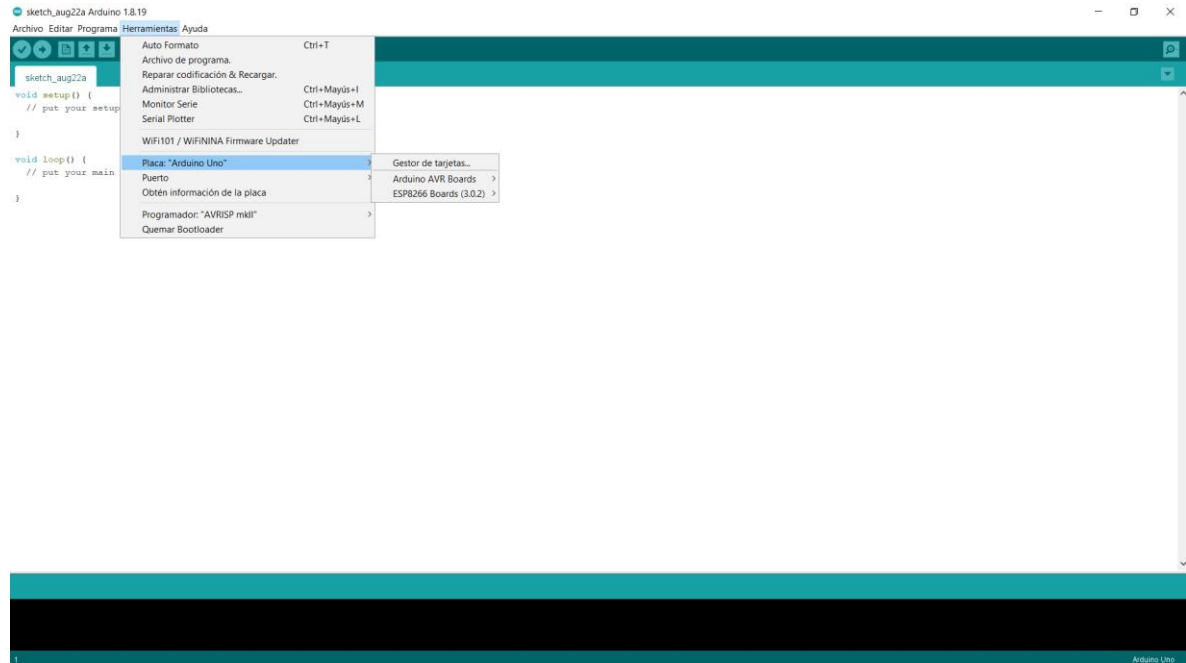
Una vez hecho esto, damos Ok y nos dirigimos a la pestaña de “Herramientas” para buscar la opción de “Placa: n” y hacer clic en la opción “Gestor de tarjetas...”



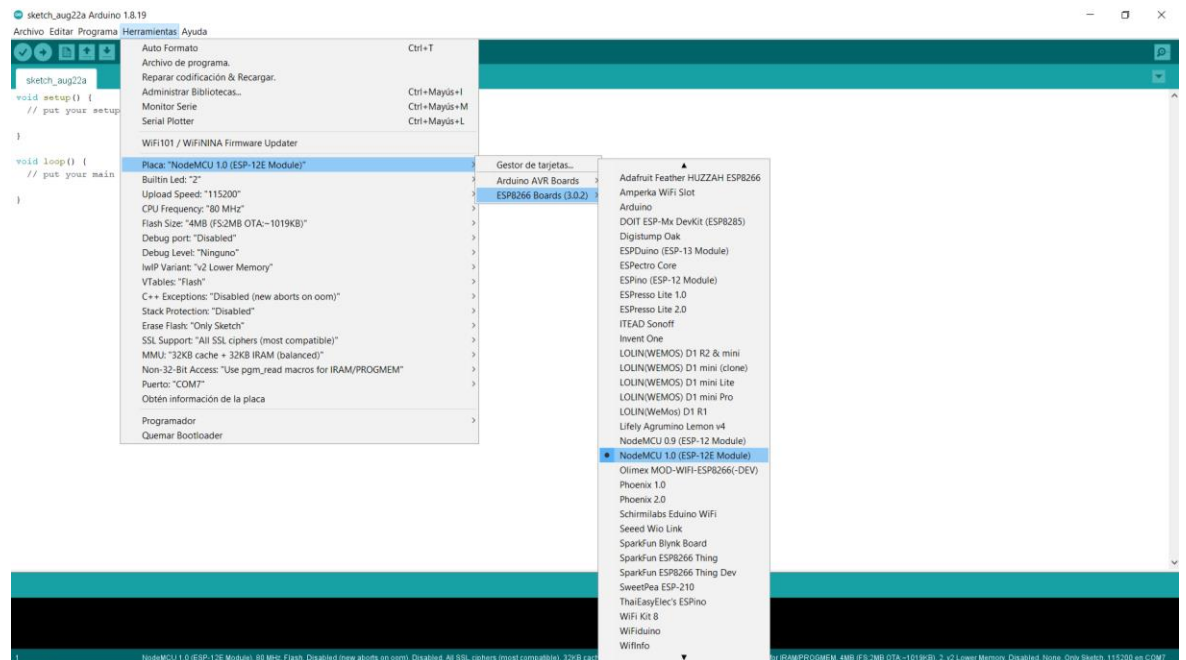
Dentro de esta opción, buscamos el nombre del SoC “ESP8266” y nos debería de salir la opción de instalarlo.



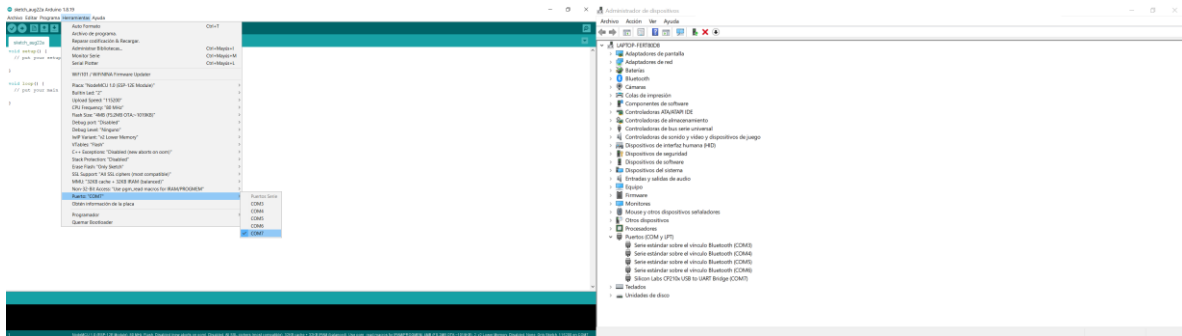
Después de instalarlo, nos dirigimos a la pestaña de “Herramientas”, nuevamente, y en donde solo aparecían placas de Arduino, ahora saldrá la opción “ESP8266 Boards (3.0.2)”.



En esta opción debemos buscar la tarjeta “NodeMCU 1.0 (ESP-12E Module)” ya que es el SoC con el que cuenta la NodeMCU de Amica.



También debemos de seleccionar el puerto, en este caso el puerto nos lo indica la ventana de “Administrador de dispositivos”. En el caso de la imagen como ejemplo, es el puerto 7. Esta varía dependiendo de los dispositivos conectados a tu computador. Por lo que sugiero revisar el puerto antes de seleccionarlo.

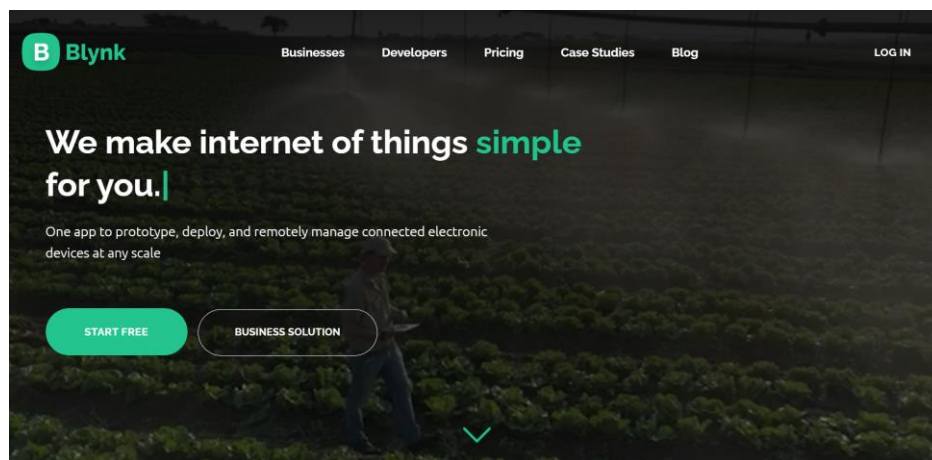


Con todo esto hecho, la tarjeta esta lista para correr los códigos que vayamos creando.

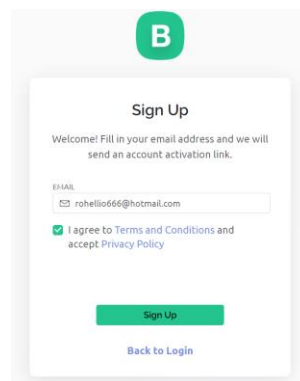
Aplicación móvil

Para conectar la aplicación móvil con nuestra tarjeta NodeMCU, usaremos la aplicación Blynk, una aplicación para crear prototipos, desplegar y gestionar remotamente dispositivos electrónicos conectados a cualquier escala. La página de la aplicación es:

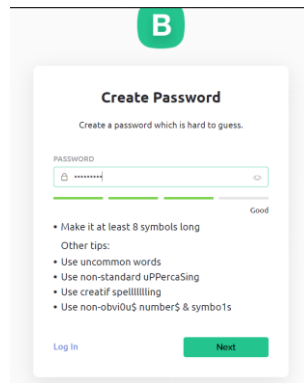
<https://blynk.io/>



Dentro de esta, vamos a dar clic en “START FREE” y vamos a crear una cuenta con algún correo que usemos. Y daremos clic en “Sign Up”.

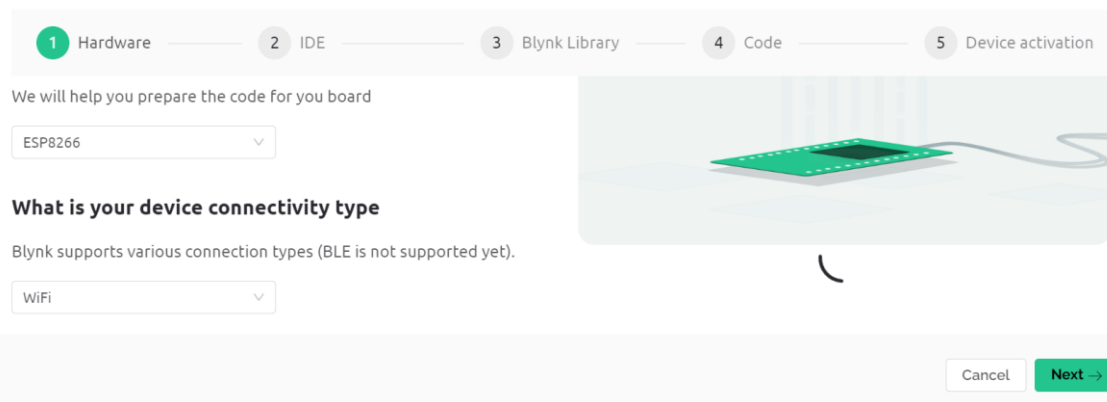


Con esto listo, nos llegará un correo al e-mail introducido con las instrucciones para crear una contraseña.



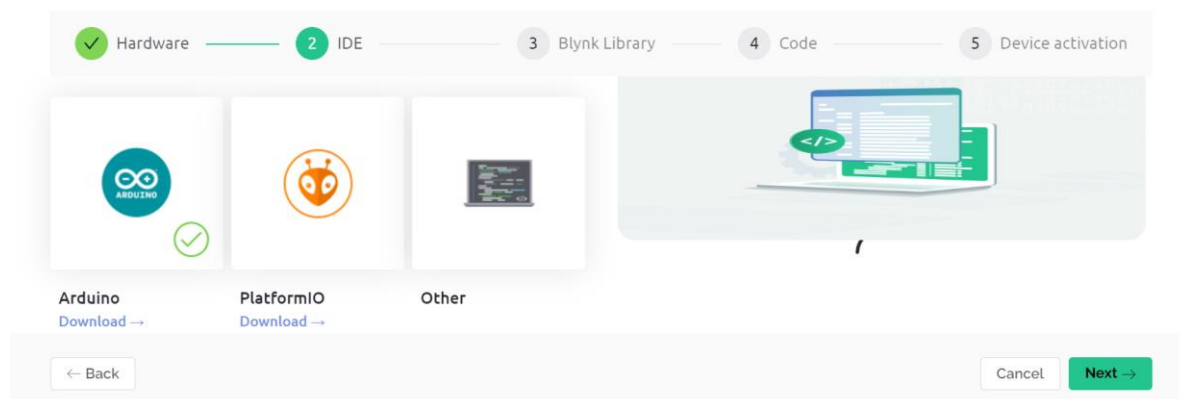
Con esto hecho, nos pedirá un nombre y aceptamos. Aparecerá la página de Blynk con algunos consejos para empezar. Los leemos y damos clic en “Let’s Go” hasta llegar a la parte en donde seleccionaremos el hardware a usar. En este caso, el hardware de uso es el **ESP8266**. También seleccionaremos el tipo de conexión. Haciéndola WiFi.

Quickstart



Damos clic en “Next” y seleccionamos Arduino como nuestra plataforma de desarrollo.

Quickstart



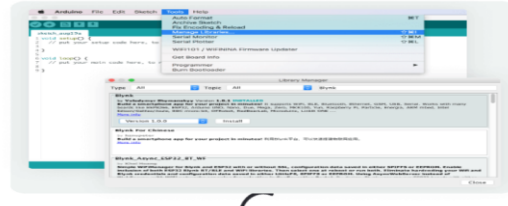
Damos “Next” y aparecerá información para descargar la librería de Blynk en Arduino. Seguimos los pasos que menciona.

Quickstart

✓ Hardware — ✓ IDE — 3 Blynk Library — 4 Code — 5 Device activation

Install Blynk Library for Arduino

1. Go to Arduino -Tools - Manage Libraries...
2. Search for Blynk there.
3. Choose the latest version and press Install.



← Back

Cancel

Next →

Una vez descargada la última versión de la librería (al menos para estas prácticas se está usando la versión 1.1.0), damos clic en “Next” y aparecerá una ventana en donde nos pedirá el nombre de la red a la que queremos conectar nuestro dispositivo y la contraseña. En mi caso, usaré la red de mi casa y su contraseña para este paso.

Quickstart

✓ Hardware — ✓ IDE — ✓ Blynk Library — 4 Code — 5 Device activation

Here is a code for your device

1. Enter your Wi-Fi network SSID (name) and password to connect your device.
* We never store or send this information anywhere. It's only used to generate the firmware code. You can leave these fields empty and manually add WiFi credentials in your sketch.

INFINITUM5322_2.4

N3dT2tE6MP

2. Copy code from the right panel (or download it as a file).
3. Create a new sketch in your IDE and paste the code.
4. In IDE check that you are using correct board and port settings

```
/* ***** Copy code Download As File ***** */

This is a simple demo of sending and receiving some data.
Be sure to check out other examples!
***** /

// Template ID, Device Name and Auth Token are provided by the Blynk Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLHc2GsJgQ"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "QZUgShh0x28hptrrgIQeCbwCXeic5Sn4"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
```

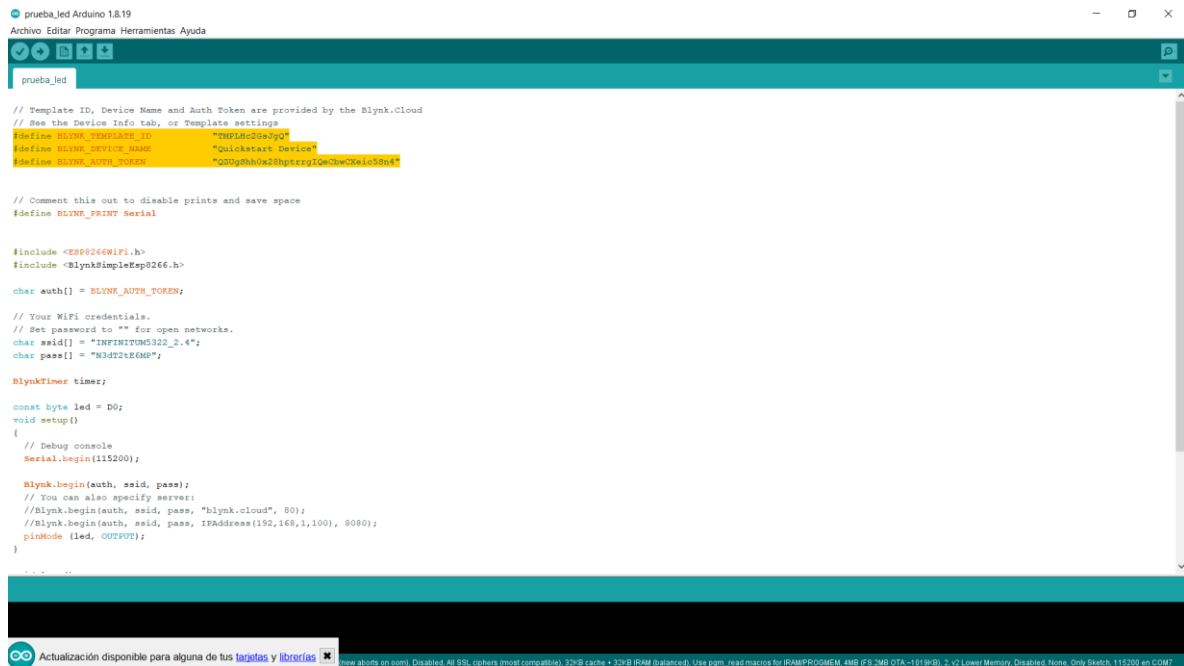
← Back

Cancel

Next →

Con esto, un código se generará en el lado derecho de la misma ventana. Lo copiamos haciendo clic en “Copy code” y seguimos los pasos que menciona debajo de donde introducimos la red y la contraseña.

Esto es abrir Arduino, pegar el código, cerciorarnos que seleccionamos la tarjeta y el puerto correcto y cargamos el programa a la tarjeta.



```
// prueba_led Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

prueba_led

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLc0e3g0"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "QD0g9hh0x23hptrrgTQcBwC6a1c58n4"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "INFINITUM5322_2.4";
char pass[] = "W3d72tEGMS";

BlynkTimer timer;

const byte led = D0;
void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  pinMode(led, OUTPUT);
}

// ...

Actualización disponible para alguna de tus librerías y librerías
```

Nos cercioramos también de introducir la contraseña y el nombre de la red a la que queremos conectar la tarjeta:



```
// prueba_led Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

prueba_led

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLc0e3g0"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "QD0g9hh0x23hptrrgTQcBwC6a1c58n4"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "INFINITUM5322_2.4";
char pass[] = "W3d72tEGMS";

BlynkTimer timer;

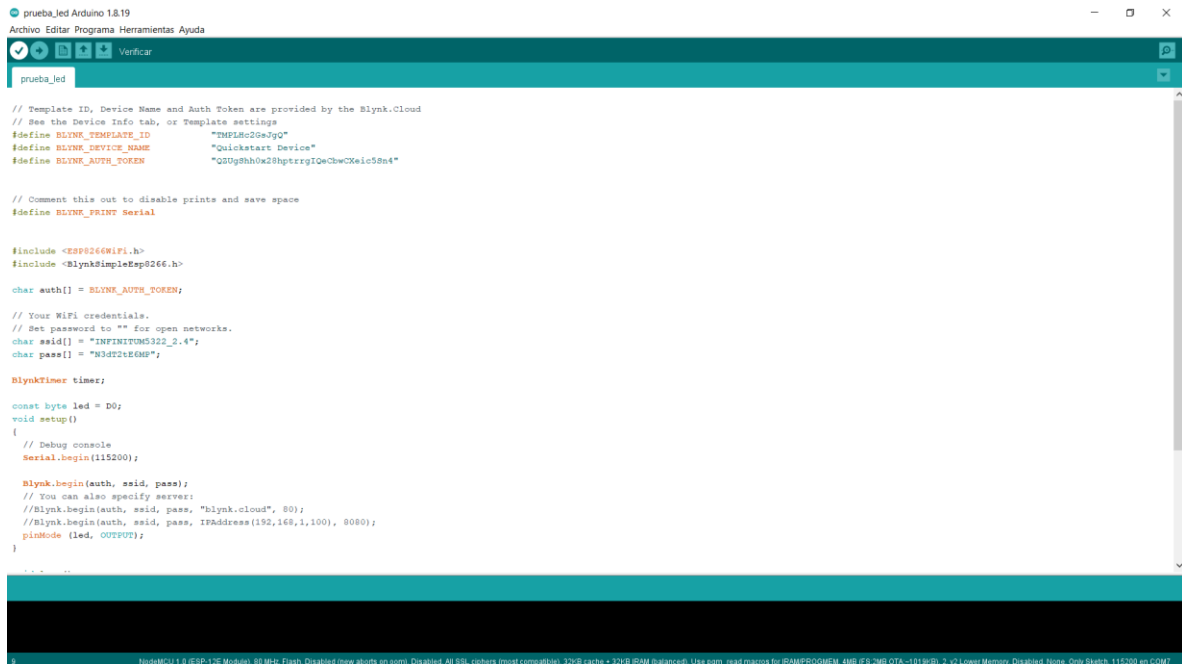
const byte led = D0;
void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  pinMode(led, OUTPUT);
}

// ...

NodeMCU v1.0 (ESP-12E Module), 80 MHz Flash, Disabled (new stock on com), Disabled, All SBL chips (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAMPROGMEM, 4MB (FS 2MB OTA -191KB), 2.2V Lower Memory, Disabled, None, Only Sketch, 115200 en COM7
```

Para cargar el programa podemos primero verificar si está funcionando en el ícono de una palomita debajo de la pestaña “Archivo”:



```
// prueba_led Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
prueba_led

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPL0c0e3g0"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "Q30g8hh0x28pstrrg1QeCbwCkeic58n4"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "INFINITUM322_2.4";
char pass[] = "W3d72tE6H8";

BlynkTimer timer;

const byte led = D0;
void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  pinMode(led, OUTPUT);
}

// ...
```

Y después dar clic en el botón de al lado que dice “Subir”



```
// prueba_led Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
prueba_led

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPL0c0e3g0"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "Q30g8hh0x28pstrrg1QeCbwCkeic58n4"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "INFINITUM322_2.4";
char pass[] = "W3d72tE6H8";

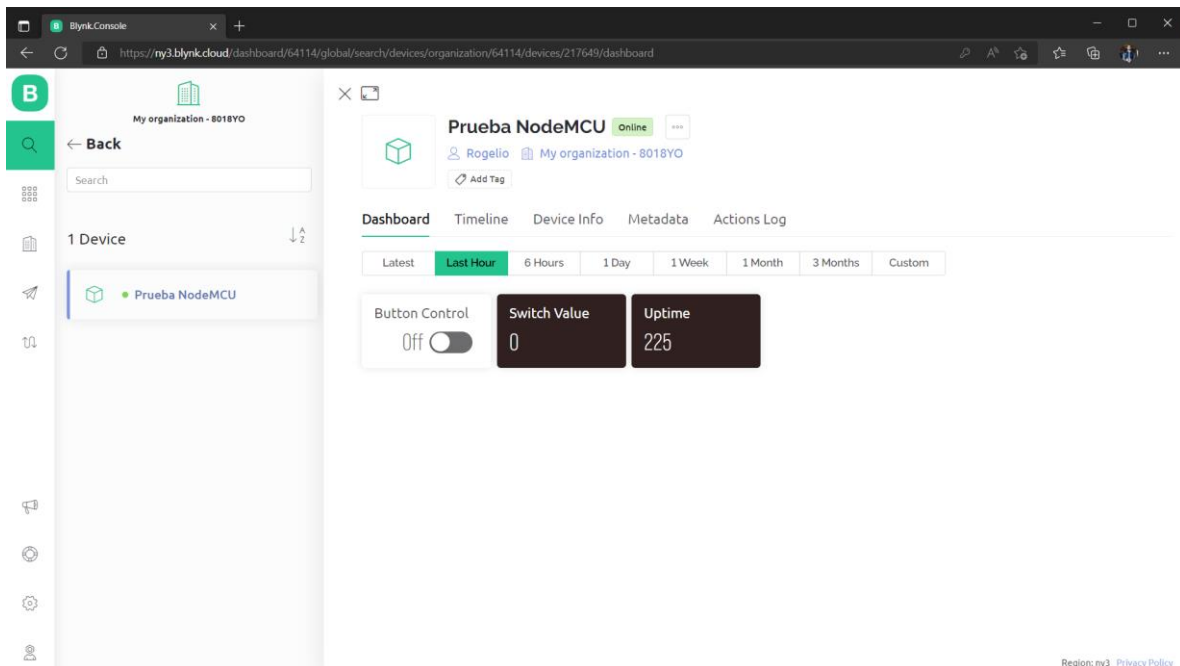
BlynkTimer timer;

const byte led = D0;
void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  pinMode(led, OUTPUT);
}

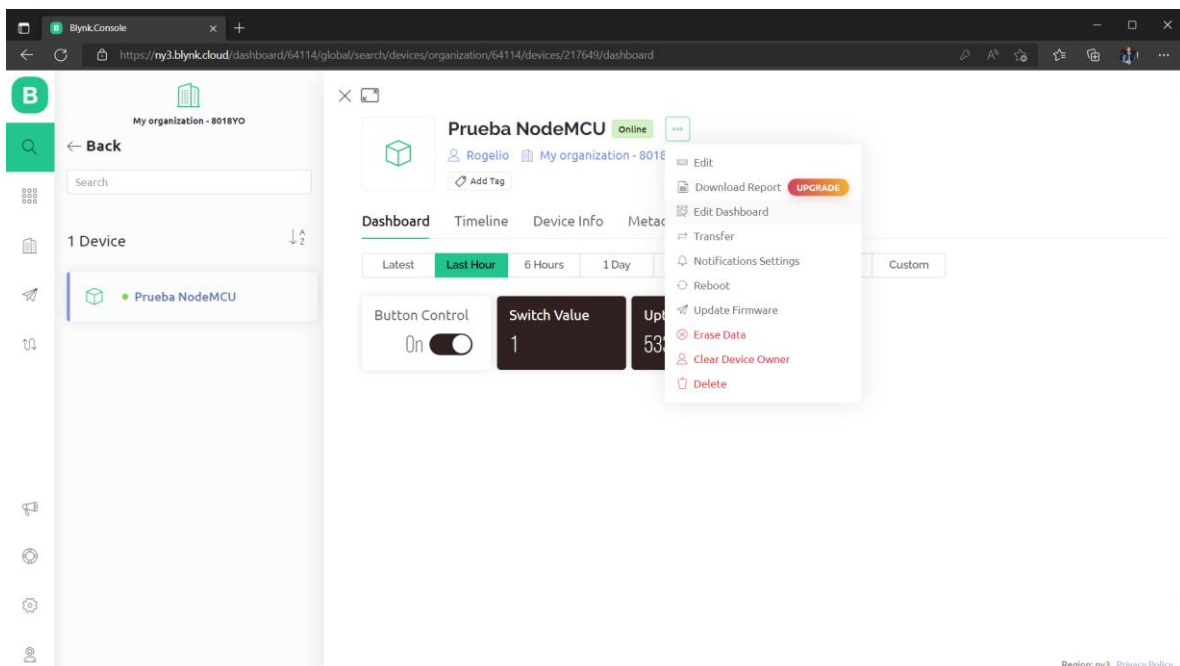
// ...
```

Después damos en “Next”. En la siguiente ventana, deberíamos tener un mensaje diciendo que se conectó correctamente nuestro dispositivo a la red. Se generará una plantilla en automático en nuestro espacio Blynk. Aquí habrá un switch que cada que lo piquemos, deberá de mandar un 0 o 1 dependiendo el estado del switch. Habrá otro contador diciendo cuánto tiempo lleva prendido nuestro dispositivo.

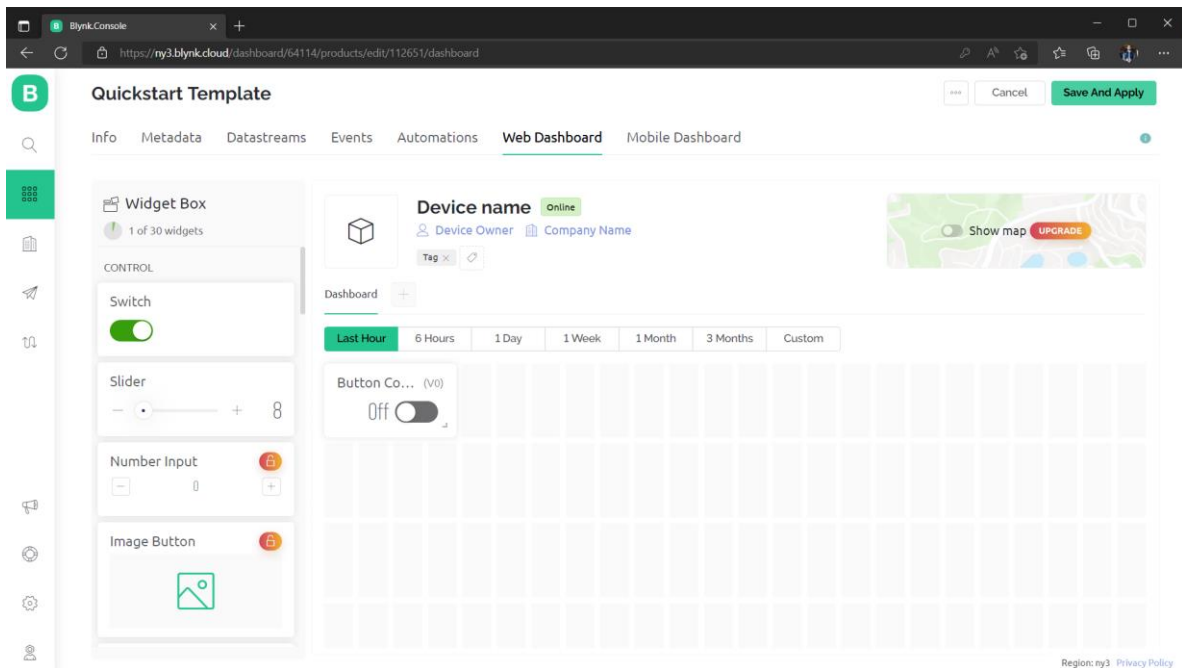


Una vez conectado el dispositivo, pasaremos a correr un código para usar un dispositivo móvil con el cual encenderemos el led físico de la tarjeta. Para esto, debemos saber que podemos editar o crear nuestras propias plantillas. Para eso, lo primero que haremos es generar un script que controle el encendido del led. Este código lleva el nombre de “prueba_led.ino” y lo cargaremos a la tarjeta.

En la interfaz de Blynk, editaremos el dashboard que se generó al momento de conectar nuestro dispositivo en línea. Damos clic en los “...” a lado de donde está el nombre del dispositivo y escogemos la opción “Edit Dashboard”.



Después de esto, podremos seleccionar que interruptores o medidores queremos. En este caso, borraremos las dos ventanas que arrojan valores de salida y solo dejamos el switch de encendido y apagado.



Damos clic en “Save And Apply” al costado superior derecho de la página y ahora si estamos listos para encender y apagar el led D0 que viene integrado a la tarjeta.

Si se quiere usar en un smartphone, se tiene que descargar la aplicación Blynk IoT e iniciar sesión. Al iniciar sesión, nos aparecerá un mensaje de felicitaciones diciendo que pudimos poner el dispositivo en línea y el switch para apagar y prender el led. Con esto ya podremos usar un dispositivo móvil para controlar el led.

