



UNIVERSIDAD DE  
GUADALAJARA  
Red Universitaria de Jalisco



**Nombre:**

Camarena Sandoval José Luis

**Código:** 217552686

**Asignatura:** Programación para internet

**Sección:** D03

**Maestro:** Michel Emanuel López Franco

**Ciclo escolar:** 2024a

Trabajo 11

**“NodeMCU”**

Abril 30 de 2024

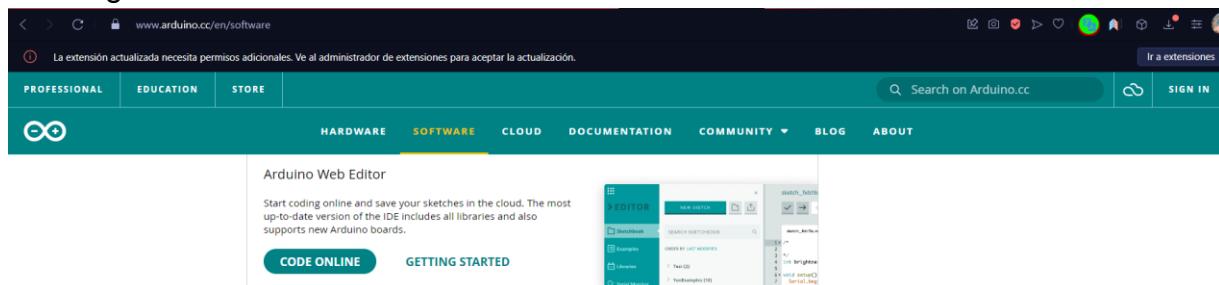


1

## Trabajo 11

### Reporte:

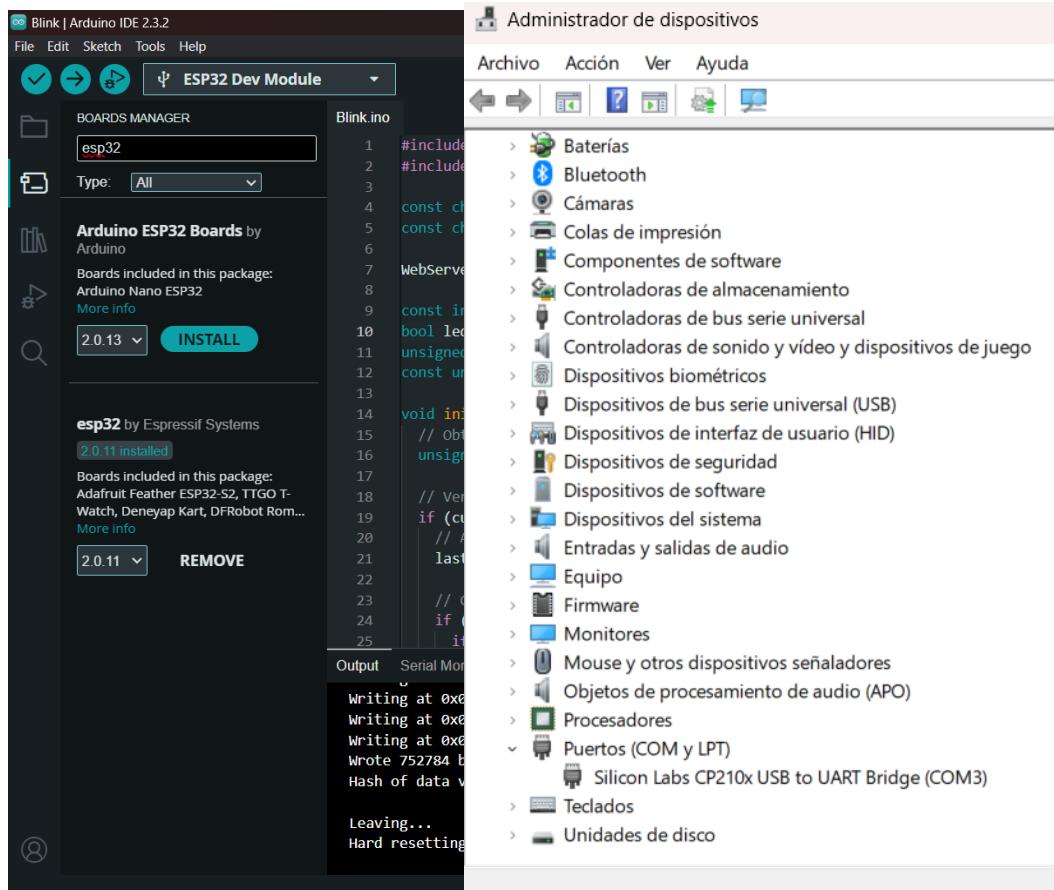
En este caso, realizamos un sitio web dentro del nodemcu que prenda y apague un led, mostrare evidencia paso a paso del trabajo realizado, empezando con la descarga del Arduino IDE.



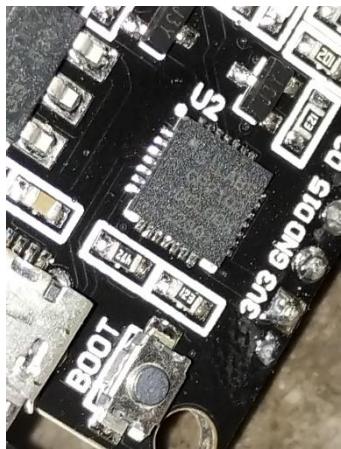
### Downloads

This screenshot shows the download page for the Arduino IDE 2.3.2. It features a large "Arduino IDE 2.3.2" logo with a teal gradient. Below it, a brief description states: "The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger." A link to the "Arduino IDE 2.0 documentation" is provided. To the right, a "DOWNLOAD OPTIONS" section lists: Windows (Win 10 and newer, 64 bits), Windows (MSI installer), Windows (ZIP file), Linux (AppImage 64 bits (X86-64)), Linux (ZIP file 64 bits (X86-64)), macOS (Intel, 10.15: "Catalina" or newer, 64 bits), and macOS (Apple Silicon, 11: "Big Sur" or newer, 64 bits). A "Release Notes" link is also present. A "Help" button is located in the bottom right corner.

Descargamos el módulo de esp32 en la barra lateral de nuestra IDE Arduino, y procedemos a conectar nuestro esp32, se debe de ver algo como en la pantalla en la parte derecha, sin embargo, si el administrador de dispositivos no reconoce el microprocesador esp32, debemos de instalar el driver manualmente.



En esta parte del esp32, podemos ver el tipo de driver que necesitaremos para poder instalar el driver.



Aquí podemos instalar el driver que necesita nuestra computadora.

The screenshot shows the Silicon Labs website with the URL [www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers](http://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers). The page title is "Download and Install VCP Drivers". It states "Downloads for Windows, Macintosh, Linux and Android below." and includes a note: "\*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at [www.kernel.org](http://www.kernel.org)." Below this, there is a section titled "Software Downloads" with two tabs: "Software (11)" and "Software · 11". The "Software · 11" tab is selected, showing a list of drivers:

Driver	Version	Last Updated
CP210x Universal Windows Driver	v11.3.0	6/24/2023
CP210x VCP Mac OSX Driver	v6.0.2	10/26/2021
CP210x VCP Windows	v6.7	9/3/2020
CP210x Windows Drivers	v6.7.6	9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6	9/3/2020

At the bottom of the list, there is a link "Show 6 more Software". A blue "Contact Us" button is located on the right side of the page.

En la parte superior en file podremos encontrar códigos ejemplo para probar la conexión a nuestro esp32, en este caso yo usé el código Blink.

The screenshot shows the Arduino IDE interface with the title "Blink | Arduino IDE 2.3.2". The sketch name is "Blink.ino" and it is connected to an "ESP32-WROOM-DA M...". The code in the editor is:

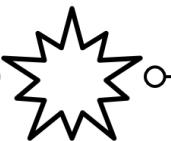
```
13 // modified 8 May 2014
14 // by Scott Fitzgerald
15 // modified 2 Sep 2016
16 // by Arturo Guadalupi
17 // modified 8 Sep 2016
18 // by Colby Newman
19
20 // This example code is in the public domain.
21
22 // https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(2, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(2, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
```

The "Output" window shows the serial communication log:

```
Writing at 0x0003f178... (87 %)
Writing at 0x00044807... (100 %)
Wrote 237152 bytes (130413 compressed) at 0x00010000 in 2.1 seconds (effective 907.3 kbit/s)...
Hash of data verified.

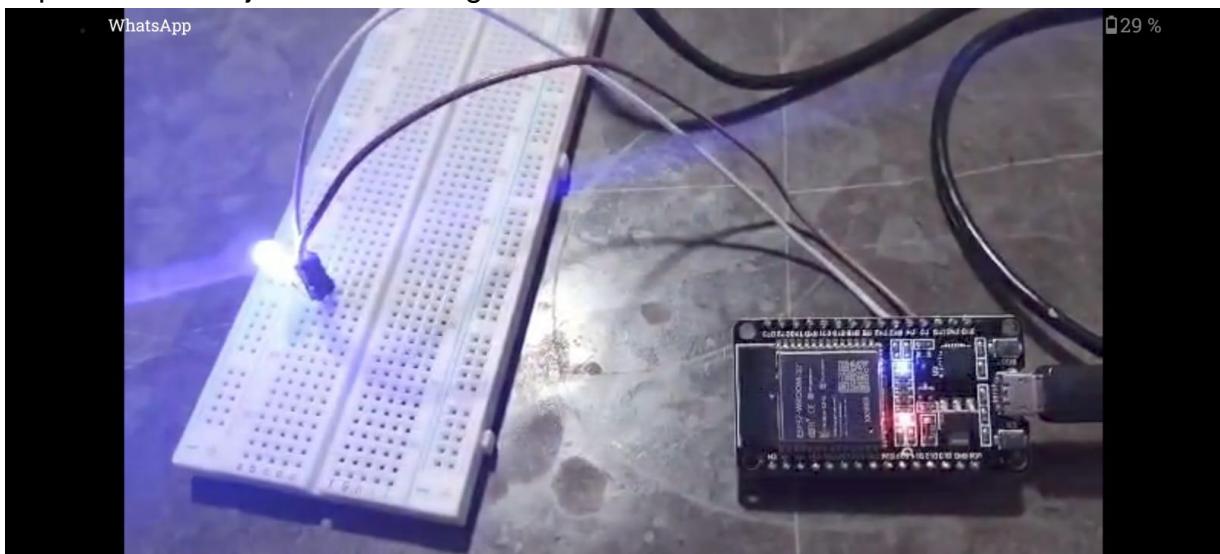
Leaving...
Hard resetting via RTS pin...
```

At the bottom right of the IDE, it says "Ln 34, Col 3 ESP32-WROOM-DA Module on COM3".

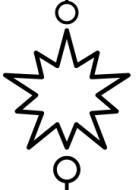
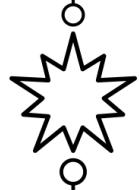


4

Aquí vemos en ejecución el código anterior.



Aquí está nuestro código ejecutándose para nuestro trabajo 11.



Screenshot of the Arduino IDE showing the 'Blink' sketch for an ESP32 Dev Module. The code includes WiFi and WebServer libraries and logic to handle LED requests via HTTP. The serial monitor shows the upload progress and a final message about hard resetting via RTS.

```
File Edit Sketch Tools Help
Blink | Arduino IDE 2.3.2
ESP32 Dev Module
Blink.ino
1 #include <WiFi.h>
2 #include <WebServer.h>
3
4 const char* ssid = "totalplay-53CS_EXT";
5 const char* password = "Camarena75#";
6
7 WebServer server(80);
8
9 const int ledPin = 2;
10 bool ledState = LOW;
11 unsigned long lastRequestTime = 0; // Variable para almacenar el tiempo de la última solicitud recibida
12 const unsigned long requestInterval = 1000; // Intervalo mínimo entre solicitudes en milisegundos (1 segundo)
13
14 void inicio() {
15     // Obtener el tiempo actual
16     unsigned long currentTime = millis();
17
18     // Verificar si ha pasado suficiente tiempo desde la última solicitud
19     if (currentTime - lastRequestTime > requestInterval) {
20         // Actualizar el tiempo de la última solicitud
21         lastRequestTime = currentTime;
22
23         // Cambiar el estado del LED según la solicitud recibida
24         if (server.hasArg("led1")) {
25             if (server.arg("led1") == "on" && ledState == LOW) {
```

Output Serial Monitor

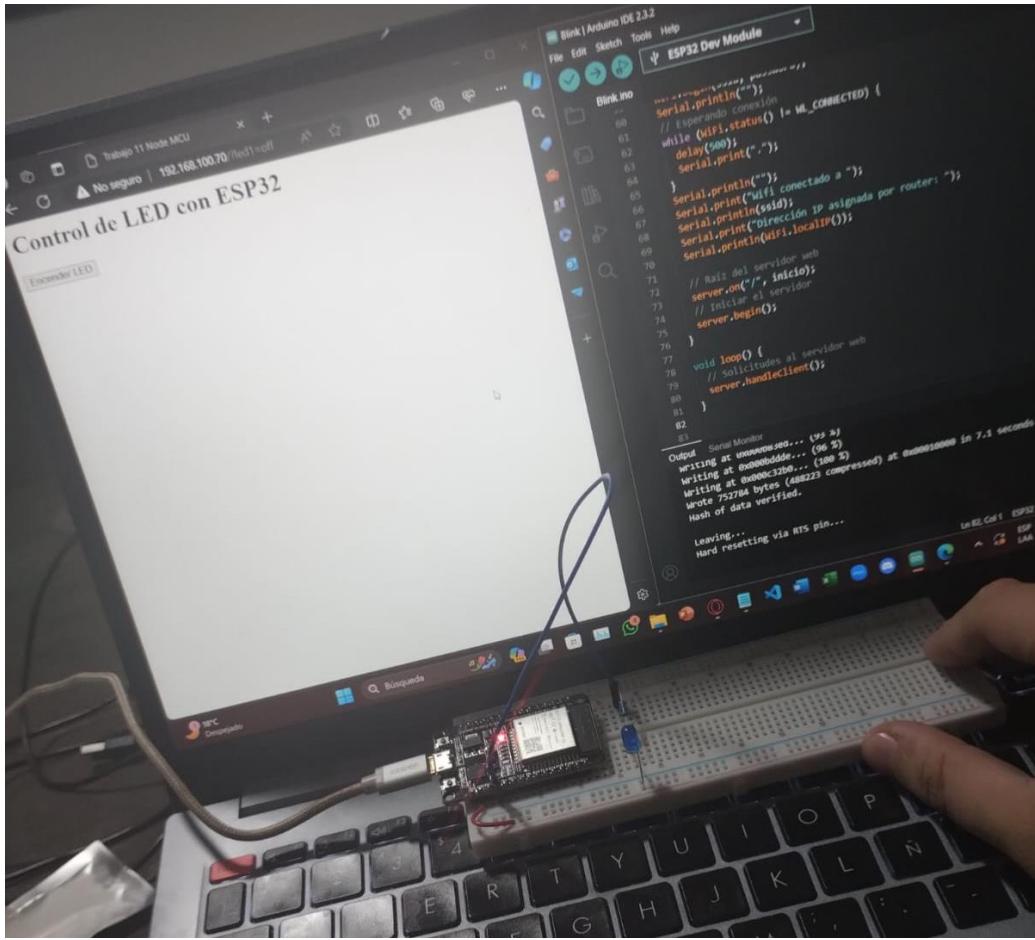
```
Writing at 0x00b83ed... (93 %)
Writing at 0x000bddde... (96 %)
Writing at 0x00c32b0... (100 %)
Wrote 752784 bytes (488223 compressed) at 0x00010000 in 7.1 seconds (effective 843.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

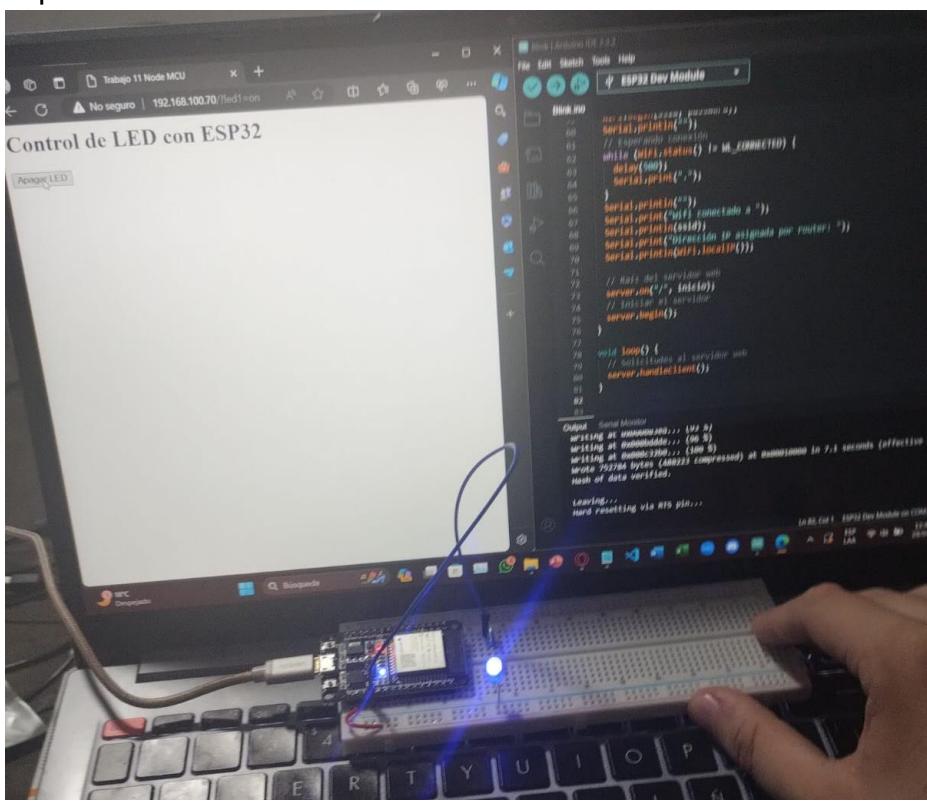
Aquí está nuestro sitio web alojado de forma local.



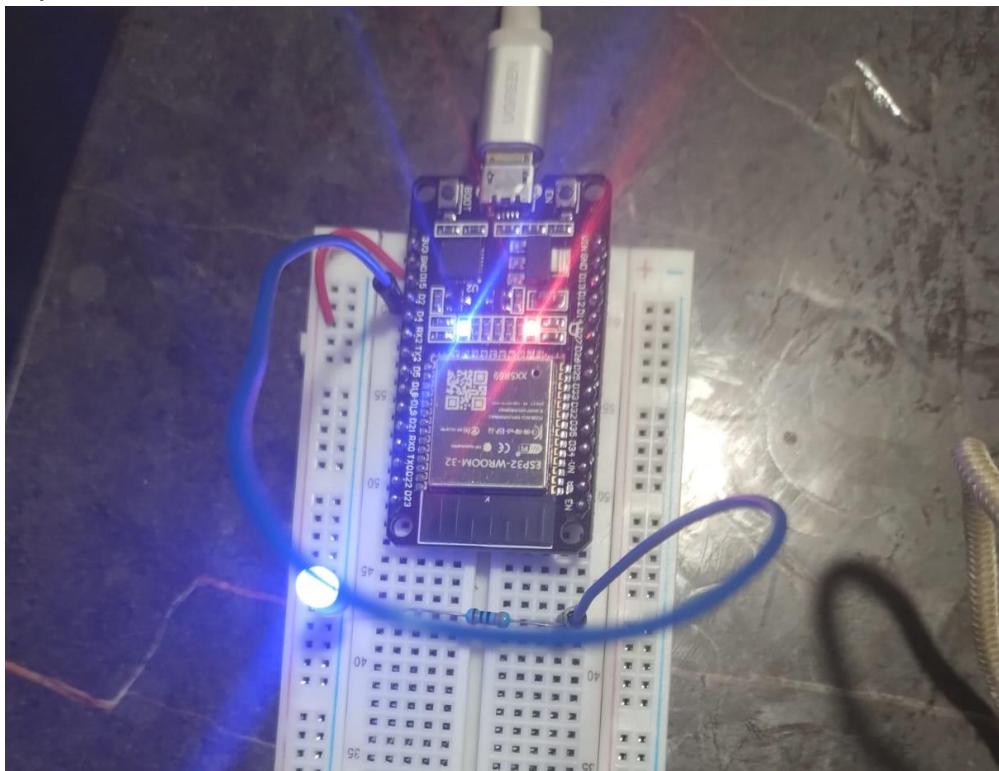
Aquí está la ejecución de nuestro código en el protoboard. En esta foto el LED se encuentra apagado.

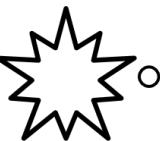


Aquí vemos cuando se encuentra encendido el LED.



Aquí está mi circuito final.





7

## Referencias

Programador Novato. (2022, 13 septiembre). *Instalar ESP32 en Arduino IDE*. YouTube.

Recuperado 28 de abril de 2024, de <https://www.youtube.com/watch?v=INW9gTfIUY>

Editronikx. (2023, 13 julio). *CONTROL de LED por WIFI con boton en HTML y WEBSERVER / curso IOT con ESP32 y ARDUINO (No5)* [Vídeo]. YouTube.

Recuperado 28 de abril de 2024, de [https://www.youtube.com/watch?v=c\\_9oP1bCjW4CP210x USB to UART Bridge VCP Drivers - Silicon Labs](https://www.youtube.com/watch?v=c_9oP1bCjW4CP210x%20USB%20to%20UART%20Bridge%20VCP%20Drivers%20-%20Silicon%20Labs). (s. f.-b). Silicon Labs. Recuperado 28 de abril de 2024, de <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

*Software*. (s. f.). Arduino. Recuperado 28 de abril de 2024, de <https://www.arduino.cc/en/software>