



Electrónica Aplicada a Internet de las Cosas 2021

Clase 4:	2
Práctico Obligatorio N°1	2
Materiales Necesarios	2
Ejercicio 1: Mi primer programa con ESP8266	3
Ejercicio 2: Manejo de salidas digitales y conexiones	4
Ejercicio 3: Manejo del ADC	5
Ejercicio 4: Comunicación I2C	6
Ejercicio 5: WiFiScan, búsqueda de redes disponibles	8
Ejercicio 6: Crear Servidor Web local con ESP8266	9
Formato de entrega	11
Anexo 1: Circuito de programación ESP01 con Arduino	12





Clase 4:

Para realizar el práctico de forma no presencial deberá contar con una pc con acceso a internet y una red wifi de la cual deberemos conocer su contraseña de acceso. En la PC deberá tener el software instalado (IDE Arduino) y configurarlo de acuerdo a lo mostrado en la clase n°3. Si aún no lo has hecho, éste es el momento.



Además deberá contar con los componentes electrónicos que se detallan en el siguiente apartado

Para poder realizar la entrega deberá contar con un dispositivo para grabar video (celular, cámara digital, cámara web, etc) y una cuenta de gmail o youtube.

Práctico Obligatorio N°1

Lea atentamente el todo el documento antes de comenzar a realizar el práctico.

Cuente con un lugar tranquilo para trabajar.

Intente dentro de sus posibilidades contar con una mesa de trabajo despejada.

No realice modificaciones al circuito mientras el mismo se encuentra en funcionamiento.

Verifique las conexiones antes de conectar la placa a su PC o fuente de alimentación.

Siempre corrobore que no existan cortocircuitos. Pueden ser muy peligrosos.

Materiales Necesarios

- Módulo con ESP8266 o ESP32 (NodeMCU, Wemos D1, ESPduino, ESP 01, etc) con su respectiva alimentación en caso de ser necesaria
- Cable USB o adaptador para programar
- Resistencia de 100 ohms.
- Pulsador (deseable pero no indispensable)
- Led (color a elección)
- Protoboard
- Cables dupont o cables unifilares para realizar interconexiones.
- Potenciómetro de 10k Ohm lineal (deseable)
- Pantalla o sensor con I2C (opcional-deseable)



Ejercicio 1: Mi primer programa con ESP8266

Objetivo:

- ★ Familiarizarse con la placa de desarrollo.
- ★ Configurar el IDE Arduino para subir programas en nuestra placa de desarrollo.
- ★ Subir programas de prueba y realizar modificaciones en ellos.

Consigna

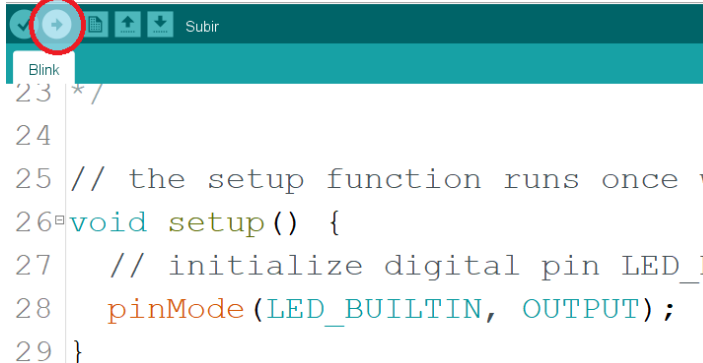
Realizar un programa que pueda encender y apagar el led integrado en la placa con un intervalo de 5 segundos.

Pasos a seguir:

1. Con el IDE Arduino abierto en la PC verificar la configuración para nuestra placa de acuerdo a lo expuesto en la clase 3. Seleccione la placa que posee.
 - a. En caso de que cuente con ESP01 seleccionar “*Generic module 8266*”. Para armar el circuito de programación vea el apartado al final del documento.
2. Diríjase a Archivo>> Ejemplos>> 01 Basic>> Blink
Se abrirá una nueva ventana en la que encontrará un código de ejemplo.
3. Conecte su placa de desarrollo. Seleccione el puerto en el que se encuentra conectada su placa de desarrollo.
4. Presione el botón con el símbolo de una flecha.

Blink Arduino 1.8.16 (Windows Store 1.8.51.0)

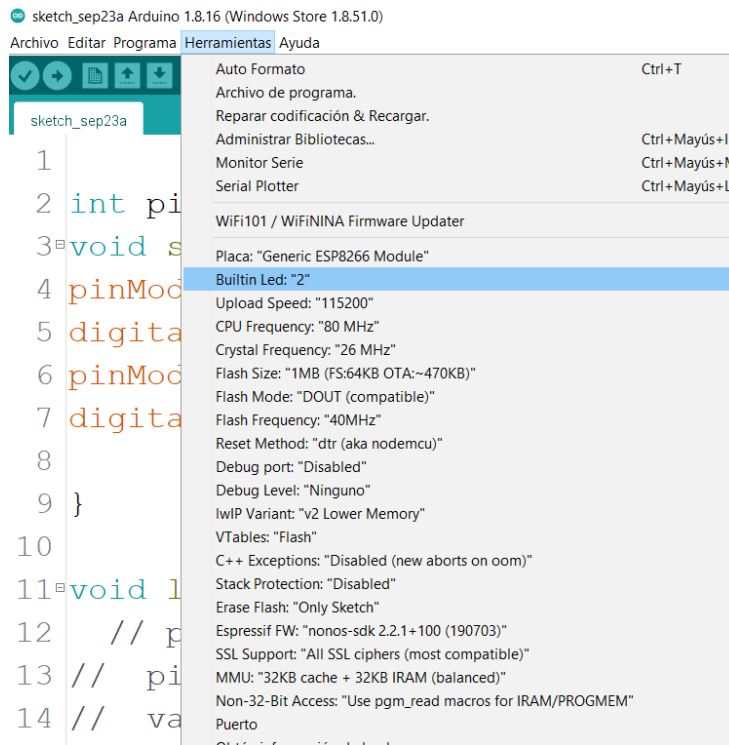
Archivo Editar Programa Herramientas Ayuda



```
23 */
24
25 // the setup function runs once w
26 void setup() {
27   // initialize digital pin LED_B
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
```

5. El programa se compilará y luego será subido a la placa. Se mostrará un mensaje en la parte inferior del IDE con la leyenda “*Subido*”. En ese momento verá que la placa comienza a realizar el parpadeo. (si esto no sucediera, presione el botón de reset sobre la placa o circuito accesorio de programación).
6. Observe que la función **delay()** es la responsable del retardo entre una instrucción y la siguiente. La función **digitalWrite(pin,mode)** nos permite elegir si la salida estará en alto o en bajo (0V o 3,3V). La función **pinMode(pin)** brinda la información acerca de si ese pin será una entrada o una salida.
7. Modifique el programa para que el retardo sea de 5 segundos en lugar de 1 segundo. **Recuerde que deberá volver a subir el programa a su placa.**

Nota: Si el LED no parpadea, debe cambiar el pin de salida (LED_BUILTIN) dentro del sketch (en el caso de ESP01 será el pin 1). En algunas placas puede diferir, verificar el funcionamiento de su placa. (Herramientas>>Builtin Led>>).



En caso de realizarlo con NodeMCU puede comprobar que existen dos opciones disponibles para activar los dos leds integrados a la placa, si lo desea puede probar el funcionamiento de ambos. Si posee otra placa de desarrollo puede verificar si existe otro led integrado.

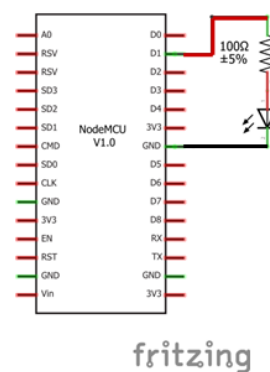
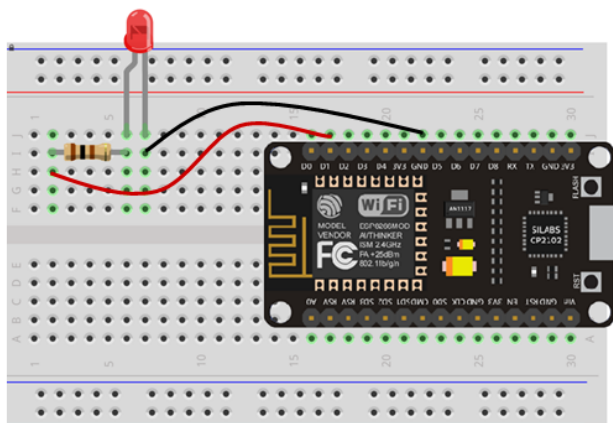
Ejercicio 2: Manejo de salidas digitales y conexiones

Consigna

Haga parpadear un led 5 veces con su placa de desarrollo.

Pasos a seguir

1. Conectar un led con una resistencia en una salida disponible de su placa. Utilice una protoboard en caso de ser necesario.





2. Escriba el programa o modifique el ejemplo del ejercicio anterior.
 3. Suba el programa a la placa y verifique su funcionamiento.
- Conectar un led en una salida digital y variar su intensidad de mas bajo a mas alto mediante el uso de una señal PWM. Utilizar funciones de tiempo para poder ver el cambio de intensidad.

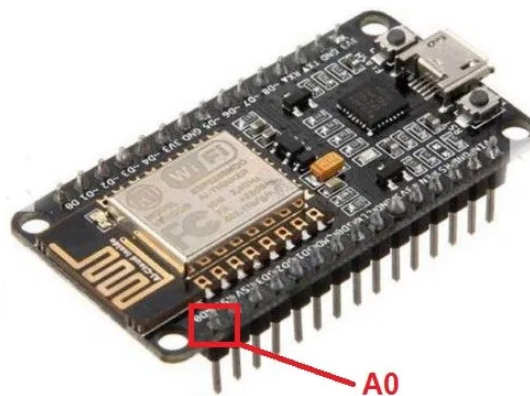
Ejercicio 3: Manejo del ADC

Objetivo: comprobar el funcionamiento del ADC en el puerto A0.

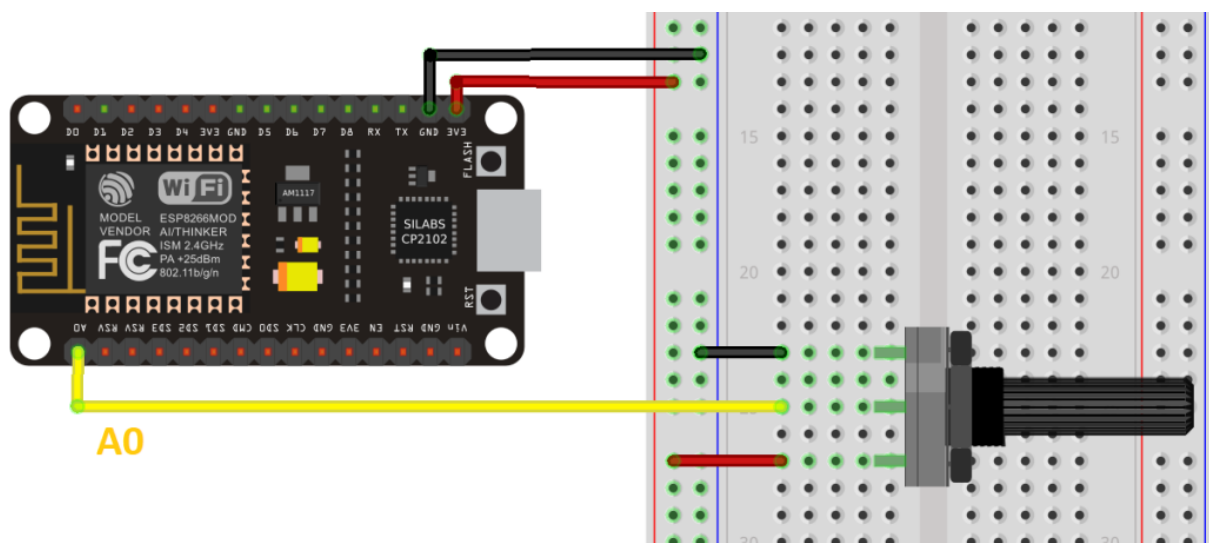
Consigna: mostrar por el puerto serie los distintos valores leídos del ADC en el puerto A0.

Pasos a Seguir:

Identificar el pin A0



Realizar la conexión de un de la siguiente manera

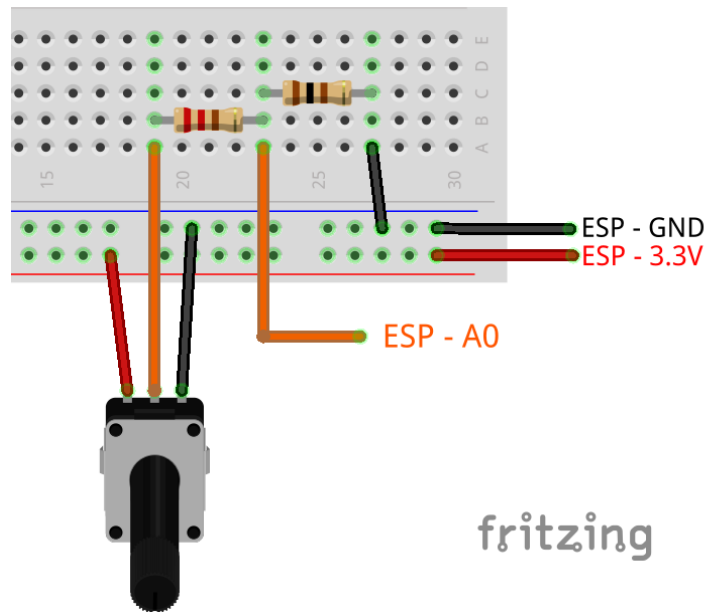




Recordar que el ADC del ESP8266 tiene un rango de 0 a 1V. El NodeMCU tiene un divisor resistivo que amplía este rango de 0V a 3,3V aproximadamente.

En caso de utilizar otra placa verificar el rango máximo del ADC.

En caso de que el rango máximo sea 1V se puede utilizar el siguiente divisor resistivo



Recordamos la fórmula del divisor resistivo:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Utilizando una resistencia **R1=220 Ohm** y una **R2=100 Ohm**, la salida Vout es 1V.

Ayuda: Para leer el valor del ADC utilizar la función **analogRead(A0)**.

Ejercicio 4: Comunicación I2C

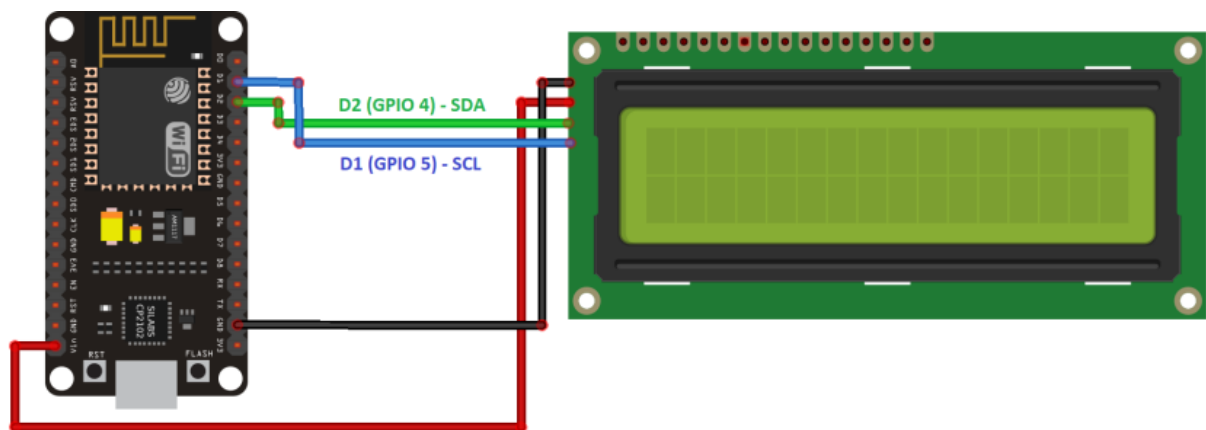
Objetivo: realizar una comunicación por I2C.

Consigna: conectar una pantalla LCD con módulo I2C e imprimir algún texto en la misma.

Pasos a seguir:

Realizar la siguiente conexión.

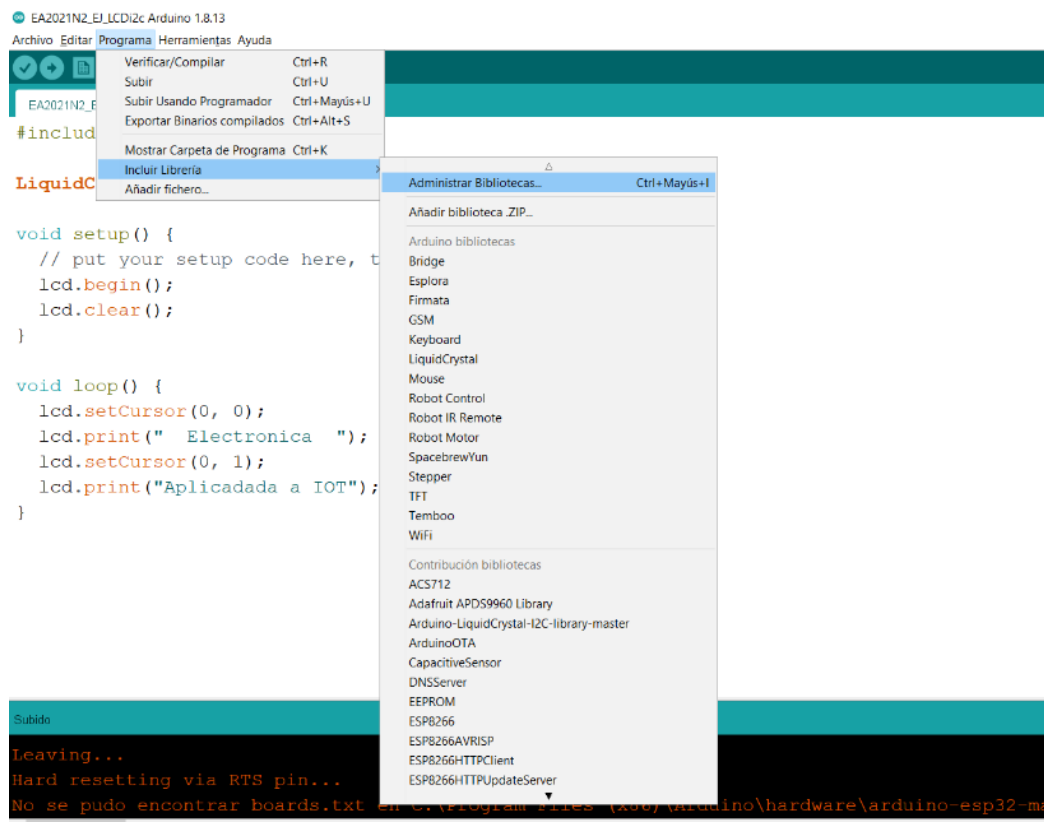
Recordar que los puertos D1 y D2 son SCL y SDA respectivamente para la comunicación I2C.



NOTA: Al alimentar por USB la placa NodeMCU en el puerto Vin tenemos los 5V del USB, el cual utilizamos para poder alimentar el módulo I2C junto con la pantalla LCD.

Para utilizar este módulo es necesario instalar la librería **LiquidCrystal_I2C.h** la cual se encuentra en la carpeta "Librerías" en la carpeta compartida del Drive.

Para instalar esta librería primero descargamos el archivo **Arduino-LiquidCrystal-I2C-library-master.rar** en una ubicación conocida de nuestra PC. Luego vamos a Programa -> Incluir Librería -> Añadir Biblioteca ZIP



Buscamos el archivo descargado y hacemos click en Abrir.



Al iniciar el programa, debemos incluir la librería y se debería marcar con letras color naranja como vemos en la imagen.

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Una vez que creamos el objeto indicando el número del dispositivo, la cantidad de columnas y filas, ya podemos utilizar.

Programa de Ejemplo:

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  // put your setup code here, to run once:
  lcd.begin();
  lcd.clear();
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print(" Electronica ");
  lcd.setCursor(0, 1);
  lcd.print("Aplicadada a IOT");
}
```

Para poder identificar la dirección I2C de su dispositivo puede utilizar el siguiente programa que realiza un barrido del puerto I2C y nos devuelve la dirección de los dispositivos conectados. En nuestro caso la dirección es 0X27 en Hexadecimal.

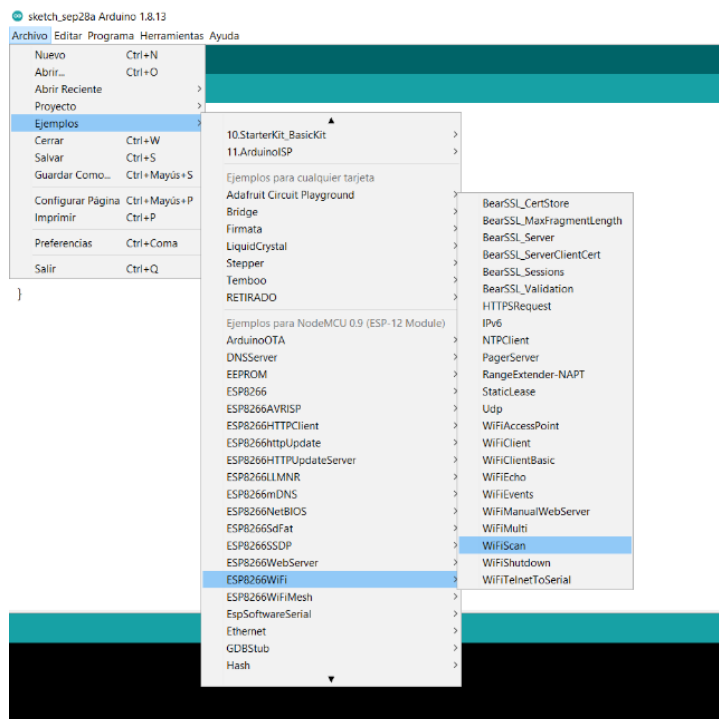
Código <https://gist.github.com/jmas/01280e8c1aae8060c90c029e59647cd9>

Ejercicio 5: WiFiScan, búsqueda de redes disponibles

Objetivos:

Escanear las redes wifi disponibles con nuestra placa de desarrollo y comprobar el correcto funcionamiento del módulo.

- Buscar el ejemplo WiFiScan, cargar el programa y comprobar las redes WiFi disponibles.



Ejercicio 6: Crear Servidor Web local con ESP8266

Este ejercicio tiene como objetivo crear un servidor web con nuestro ESP8266 y leer su respuesta desde un navegador.

- Buscar el ejemplo **WiFiManualWebServer**
 - Modificar las líneas 13 y 14 con el SSID y Contraseña de una red WiFi conocida

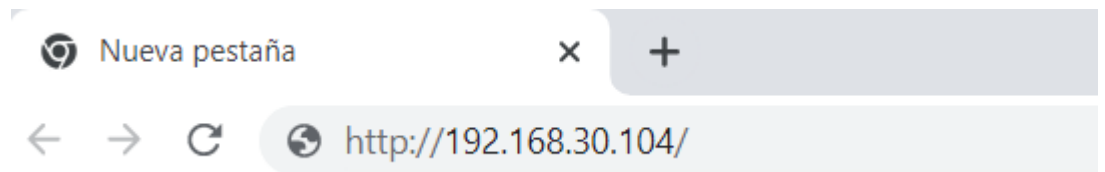
```
13 #define STASSID "your-ssid"  
14 #define STAPSK  "your-password"
```

- Cargar el programa en la placa
- Verificar que la placa se conecte a la red e identificar el número de IP

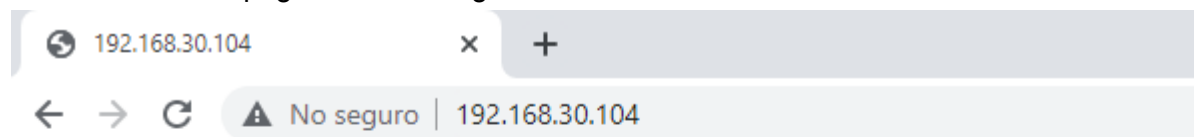


```
COM17
11:41:02.332 -> {11??|?1?|??▲▲?▲1?▲c|??□?r?c?□"??p ??n$nn??▲c□p??cd □s$po?□□▲?▲1□?
11:41:02.417 ->
11:41:02.417 -> Connecting to Comunicaciones
11:41:02.897 -> .....
11:41:10.148 -> WiFi connected
11:41:10.148 -> Server started
11:41:10.148 -> 192.168.30.104
```

- Luego abrir un navegador (por ejemplo Google Chrome) y tipear el IP correspondiente en la barra de direcciones



- Debería abrir una página como la siguiente



GPIO is now low

Click [here](#) to switch LED GPIO on, or [here](#) to switch LED GPIO off.

Haciendo click en los hipervínculos “here” debería encender y apagar el led que tiene integrada la placa. Verificar si el led de la placa utilizada se enciende por alto o por bajo.

Ahora modificar el código con los siguientes cambios:

- El texto de la página deberá aparecer en español, agregar las modificaciones que crean necesarias.
- Realizar la modificación correspondiente para que la salida digital que se enciende al hacer click sea una distinta a la que tiene conectado el led integrado. Colocar un led en la nueva salida para encenderlo y apagarlo con los hipervínculos.



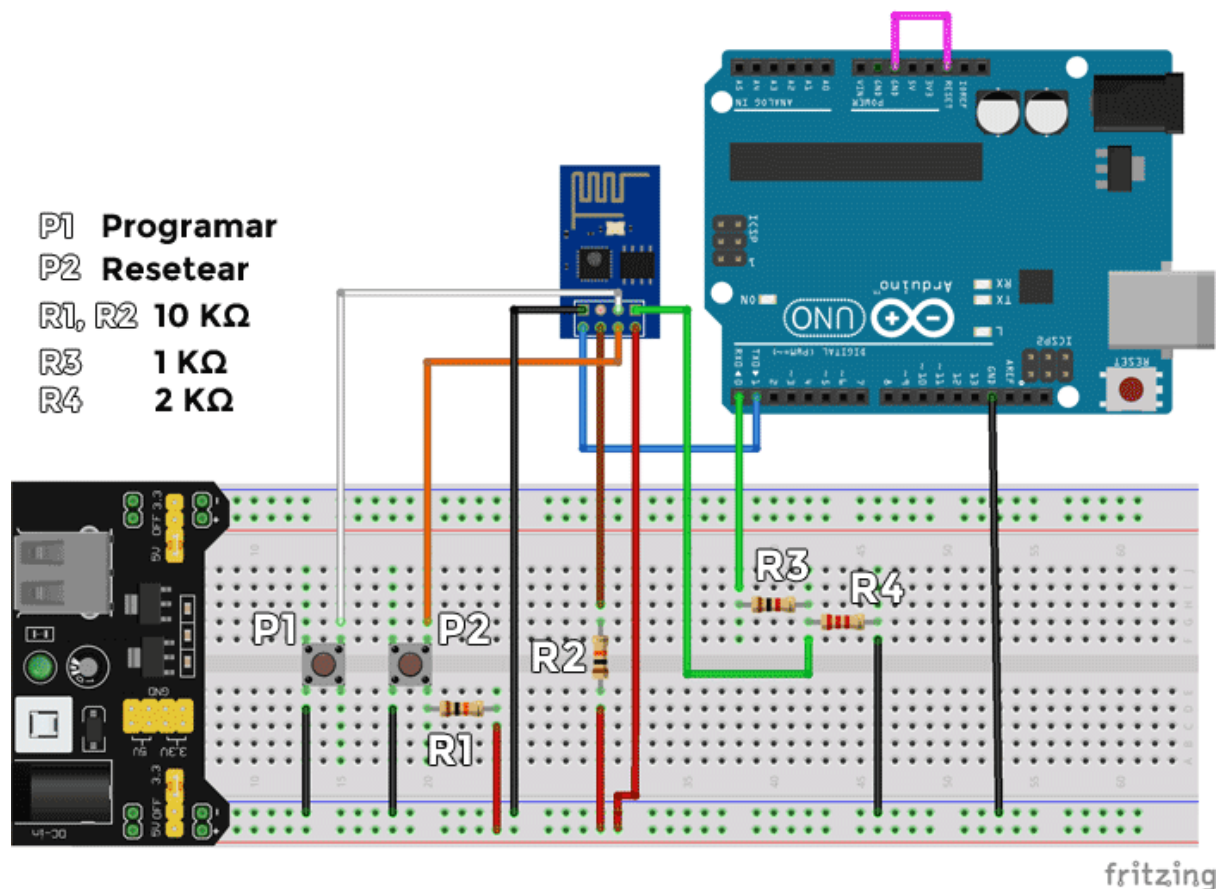
Formato de entrega

Aquellos estudiantes que realicen el práctico de forma **virtual** deberán subir un video a youtube y compartirnos el link a través del siguiente formulario:

https://bit.ly/TP1_EAIOT



Anexo 1: Circuito de programación ESP01 con Arduino



- El TX del Arduino conectado al TX del ESP-01
- El RX del Arduino conectado a la entrada del divisor de tensión
- La salida del divisor de tensión al RX del ESP-01
- Se hace un puente entre el pin GND y el pin RESET en el Arduino UNO (esto aísla al procesador de los pines GPIO, convirtiéndolo en una pasarela)
- Conectamos todas las GND entre sí
- GND de Arduino
- GND del ESP-01
- La tierra del divisor de tensión
- GND de la fuente de alimentación
- El pin CH_PD del ESP-01 a 3,3 V con una resistencia de 10 KΩ en serie
- El pin RESET del ESP-01 a 3,3 V con una resistencia de 10 KΩ en serie
- Para habilitar el modo de programación UART, conectamos el pin GPIO0
- Por último quedaría el pin GPIO2 que, como esta placa tiene una resistencia pull-up, lo podemos dejar al aire