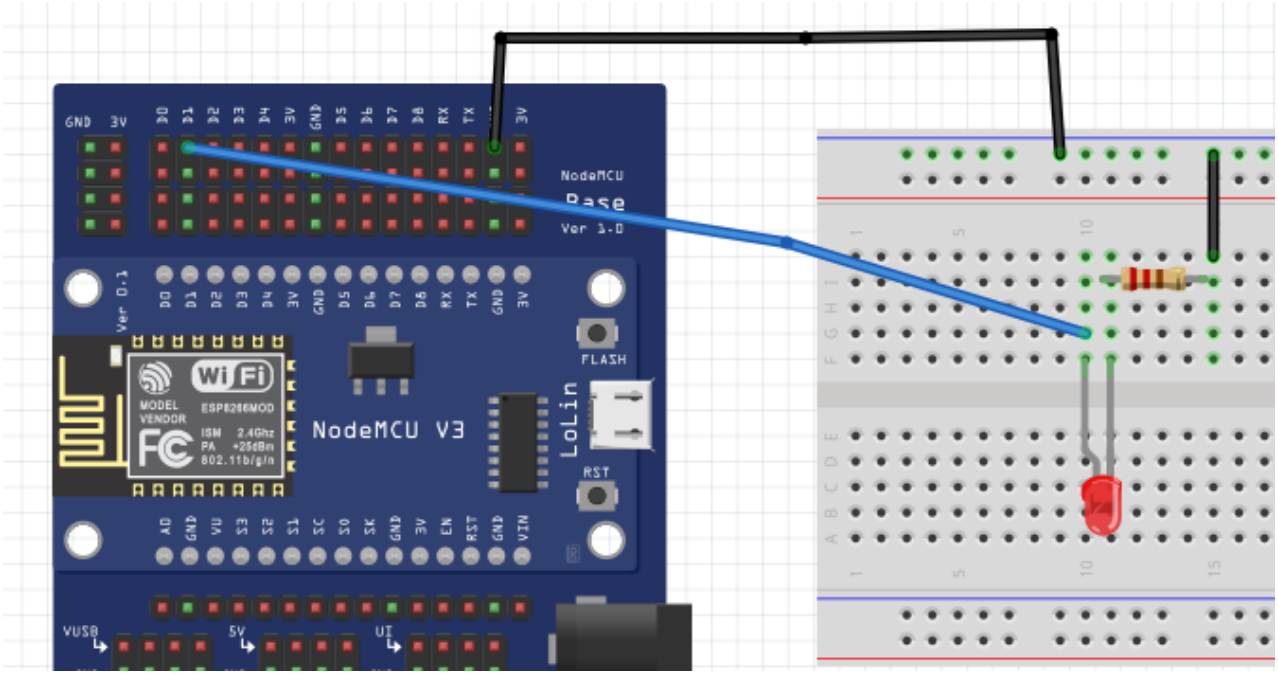
NodeMCU: 05_Punto de acceso_y_web_server.

agrportfolioeducativo.blogspot.com/2020/03/nodemcu-05punto-de-acceso.html



El código que convierte el NodeMCU en punto de acceso es el siguiente:

```
#include <ESP8266WiFi.h> // Incluye la biblioteca Wifi

const char *ssid = "Probando"; // Nombre de la red que emito
const char *password = "Probando"; // Contraseña para conectarse

void setup() {
Serial.begin(115200);
delay(10);
Serial.println('\n');

WiFi.softAP(ssid, password); // Emitiendo como Punto de acceso
Serial.print("Access Point \"");
Serial.print(ssid);
Serial.println("\" started");

Serial.print("IP address:\t");
```

```
  Serial.println(WiFi.softAPIP()); // Desde qué IP emito
}

void loop() { }
```

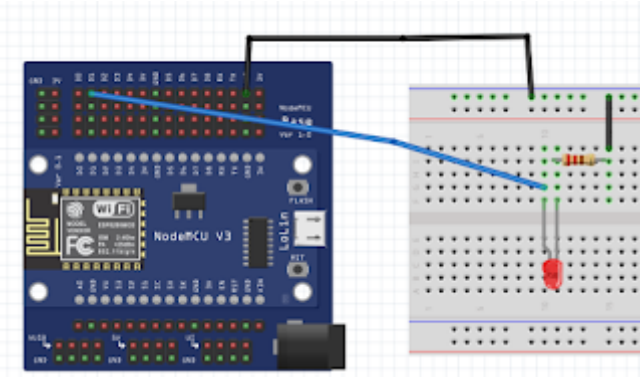
view raw NODEMCU_Punto_acceso.ino hosted with ❤ by GitHub

En el programa, lo importante es la orden **WiFi.softAP(ssid, password);** ue convierte nuestro NodeMCU en un punto de acceso.

= = = = = = = = = =

En el siguiente programa, profundizamos en este concepto. No sólo convierto nuestro NodeMCU en punto de acceso sino que a través de un puerto emite contenido en HTML. Este contenido puede ser captado desde un navegador en un móvil, por ejemplo, y ser usado para mandar información de vuelta al AP, pudiendo cambiar un estado.

En el siguiente ejemplo, modificación del ejemplo que se cita en la bibliografía de la web https://www.esploradores.com/, puede actuarse desde el navegador encendiendo u apagando un LED.




```
/*
NodeMCU Access Point - Servidor Web por Dani No www.esploradores.com
Crea un servidor Web en modo Access Point que permite encender y apagar un LED conectado a la salida D4 (GPIO02) del módulo NodeMCU.
Este código de ejemplo es de público dominio.
*/

#include <ESP8266WiFi.h> //Incluye la librería ESP8266WiFi

const char ssid[] = "DeAurelio"; //Definimos la SSDI de nuestro servidor WiFi -nombre de red-
```

```
const char password[] = "12345678"; //Definimos la contraseña de nuestro servidor
WiFiServer server(80); //Definimos el puerto de comunicaciones

int PinLED = D1; //Definimos el pin de salida - GPIO2 / D1
int estado = LOW; //Definimos la variable que va a recoger el estado del LED

void setup() {
Serial.begin(115200);

pinMode(PinLED, OUTPUT); //Inicializamos el GPIO2 como salida
digitalWrite(PinLED, LOW); //Dejamos inicialmente el GPIO2 apagado

server.begin(); //inicializamos el servidor
WiFi.mode(WIFI_AP);
WiFi.softAP(ssid, password); //Red con clave, en el canal 1 y visible
//WiFi.softAP(ssid, password,3,1); //Red con clave, en el canal 3 y visible
//WiFi.softAP(ssid); //Red abierta

Serial.println();

Serial.print("Direccion IP Access Point - por defecto: "); //Imprime la dirección IP
Serial.println(WiFi.softAPIP());
Serial.print("Direccion MAC Access Point: "); //Imprime la dirección MAC
Serial.println(WiFi.softAPmacAddress());

//IPAddress local_ip(192, 168, 1, 1); //Modifica la dirección IP
//IPAddress gateway(192, 168, 1, 1);
//IPAddress subnet(255, 255, 255, 0);
//WiFi.softAPConfig(local_ip, gateway, subnet);
//Serial.println();
//Serial.print("Access Point - Nueva direccion IP: ");
//Serial.println(WiFi.softAPIP());
```

```
}

void loop()
{
// Comprueba si el cliente ha conectado
WiFiClient client = server.available();
if (!client) {
return;
}

// Espera hasta que el cliente envía alguna petición
Serial.println("nuevo cliente");
while(!client.available()){
delay(1);
}

// Imprime el número de clientes conectados
Serial.printf("Clientes conectados al Access Point: %d",
WiFi.softAPgetStationNum());

// Lee la petición
String peticion = client.readStringUntil('r');
Serial.println(peticion);
client.flush();

// Comprueba la petición
if (peticion.indexOf('/LED=ON') != -1) {
estado = HIGH;
}
if (peticion.indexOf('/LED=OFF') != -1){
estado = LOW;
```

```
}

//Enciende o apaga el LED en función de la petición
digitalWrite(PinLED, estado);

// Envía la página HTML de respuesta al cliente
client.println("HTTP/1.1 200 OK");
client.println(""); //No olvidar esta línea de separación
client.println("<!DOCTYPE HTML>");
client.println("<meta charset='UTF-8'>");
client.println("<html>");

//Imprime el estado del led
client.print("<h1>El LED está ahora: ");
if(estado == HIGH) {
client.print("ENCENDIDO</h1>");
} else {
client.print("APAGADO</h1>");
}

//Se crean enlaces para modificar el estado del LED
client.println("Presiona <a href='/LED=ON'>AQUÍ</a> para encender el LED<br>");
client.println("Presiona <a href='/LED=OFF'>AQUÍ</a> para apagar el LED<br>
<br>");

//Se crean cajas de comprobación (checkbox) para modificar el estado del LED
client.println("<input type='checkbox' onClick=location.href='/LED=ON'> ENCENDER
</input><br>");
client.println("<input type='checkbox' onClick=location.href='/LED=OFF'> APAGAR
</input><br><br>");

//Se crean botones para modificar el estado del LED
```

```
    client.println("<button type='button' onClick=location.href='/LED=ON'> ENCENDER </button>");
    client.println("<button type='button' onClick=location.href='/LED=OFF'> APAGAR </button><br><br>");

    //Se crean botones con estilos para modificar el estado del LED
    client.println("<button type='button' onClick=location.href='/LED=ON' style='margin:auto; background-color:green; color:#A9F5A9; padding:5px; border:2px solid black; width:200;'><h2> ENCENDER</h2> </button>");
    client.println("<button type='button' onClick=location.href='/LED=OFF' style='margin:auto; background-color:red; color:#F6D8CE; padding:5px; border:2px solid black; width:200;'><h2> APAGAR</h2> </button><br><br>");

    //Se crea un único botón para modificar el estado del LED
    if(estado == HIGH) {
    client.print("<button type='button' onClick=location.href='/LED=OFF'> APAGAR </button><br><br>");
    } else {
    client.print("<button type='button' onClick=location.href='/LED=ON'> ENCENDER </button><br><br>");
    }
    client.println("</html>");
    delay(1);
    Serial.println("Petición finalizada"); // Se finaliza la petición al cliente. Se inicaliza la espera de una nueva petición.

    //Desconexión de los clientes
    //WiFi.softAPdisconnect();
    }
```

view raw NodeMCU_punto_acceso_webserver_03.ino hosted with ❤ by GitHub

.

Bibliografía

=========

https://www.esploradores.com/access-point-servidor-web-nodemcu/

https://tttapa.github.io/ESP8266/Chap01%20-%20ESP8266.html