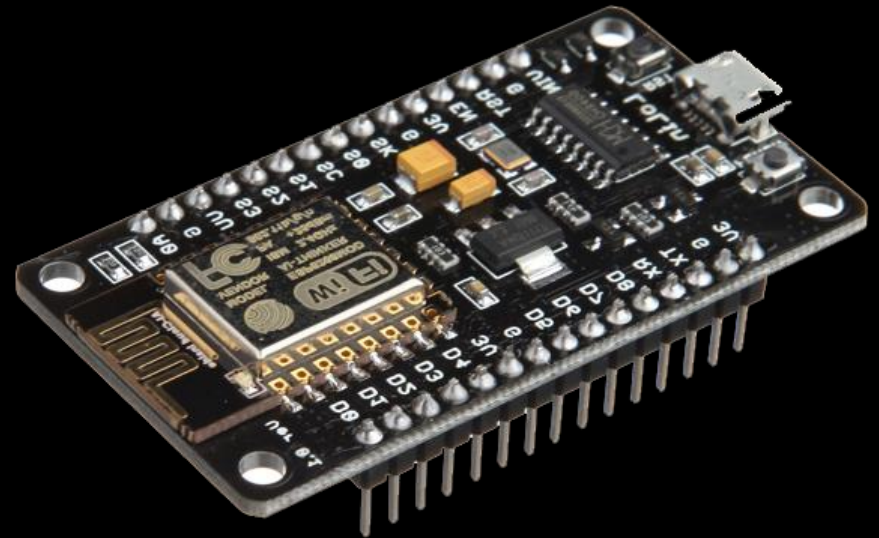


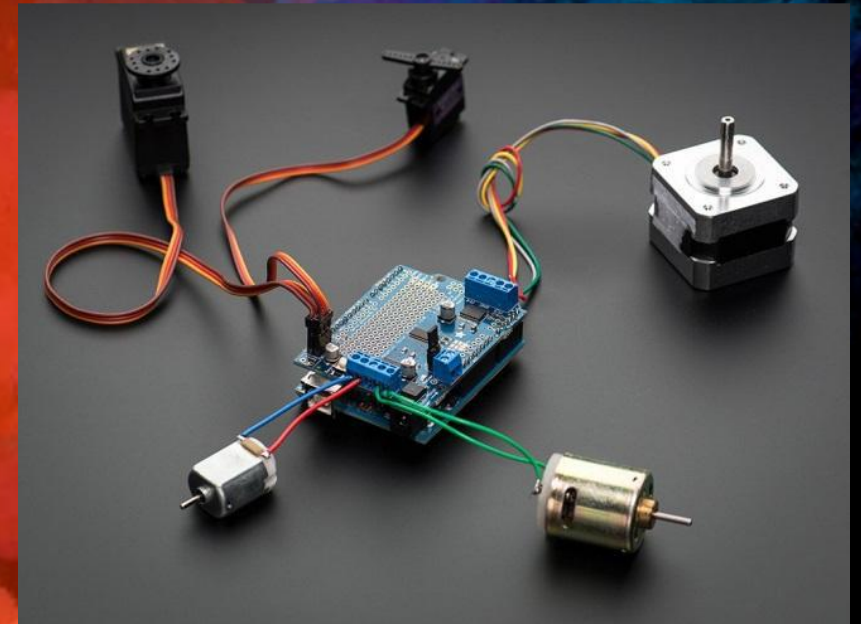
The MAKERS INFORMATICA

ACTUADORES Y
SERVIDOR WEB

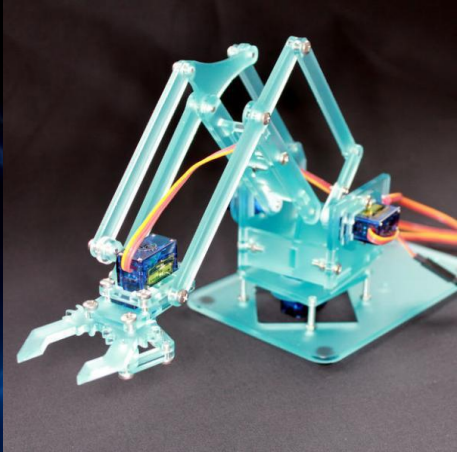


¿QUÉ SON LOS ACTUADORES?

- Un actuador es un dispositivo capaz de transformar energía hidráulica , neumática o eléctrica en la activación de un proceso el cual genera un efecto sobre elementos externos.
- Node mcu y otros autómatas no disponen de potencia suficiente para mover actuadores. De hecho la función de un procesador no es **ejecutar acciones sino mandar a ejecutar acciones** y para ello existen los drivers que realizan el “trabajo pesado”.

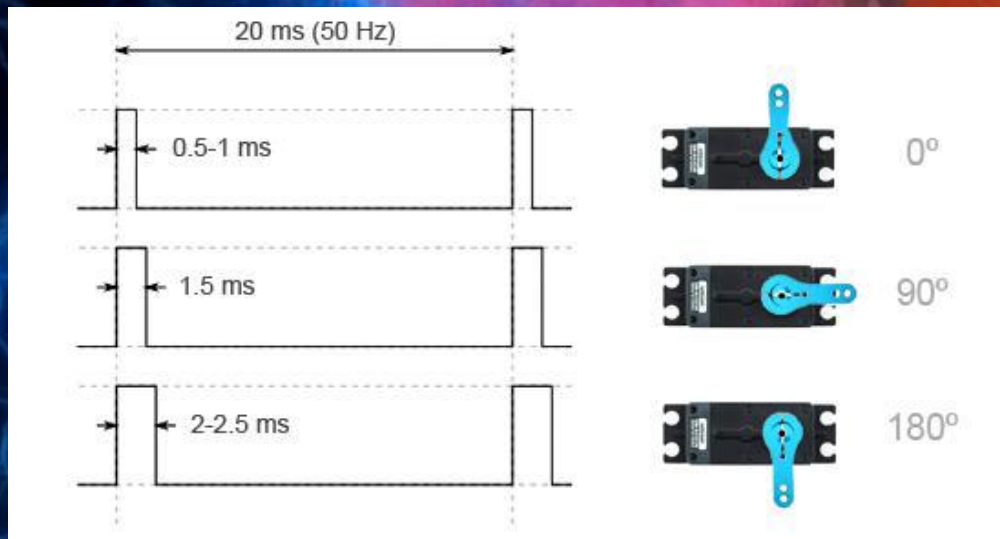


SERVOMOTORES



- El servomotor o servo es un actuador similar a un motor de corriente continua pero este tiene la capacidad de colocarse en una posición determinada (una ángulo) y permanecer fija en esta.
- Puede aguantar cierta cantidad de peso a través del torque (máxima fuerza rotacional) indicado en sus características, generalmente se mide en Kg/cm.
- Para poder manejar servos con Arduino es necesario incluir la librería: **SERVO.H**

SERVOMOTORES



- Para controlar un servo se utiliza el PWM. La mayoría de los servos trabaja en una frecuencia de 50Hz(20ms).
- Cuando se manda un pulso la anchura de este determina la posición angular del servo, esta anchura varia según el servo pero normalmente es entre 0.5ms a 2.5ms.
- La librería Servo nos ayuda a conseguir la frecuencia adecuada. Esta librería solo soporta hasta 12 servos .

PARTES DE UN SERVOMOTOR

Esta formado por un motor de corriente continua, una caja reductora, un juego de engranes, un potenciómetro y un circuito de control.



TIPOS DE SERVOMOTORES

Servomotor SG90

Engranés de plástico

Voltaje de operación: 4.8 a 7.2V

Torque: 1,2 Kg*cm

Velocidad: 0.1 seg/60°(4.8V) y
0.08 seg/60°(6V)

Angulo de giro: de 0 a 180° con
1° de precisión



Servomotor MG90S

Engranés metálicos

Voltaje de operación: 4.8 a 7.2V

Torque: 1.8 Kg*cm

Velocidad: 0.1 seg/60°(4.8V) y
0.08 seg/60°(6V)

Angulo de giro: 180°



Servomotor MG995

Engranés metálicos

Voltaje de operación:

4.8 a 7.2V

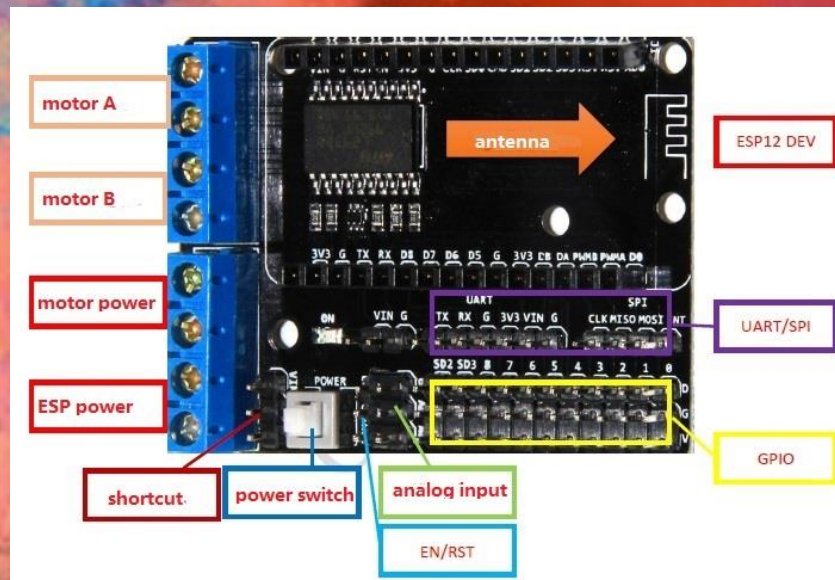
Torque: 11,2 Kg a 6V

Angulo de giro: 180°



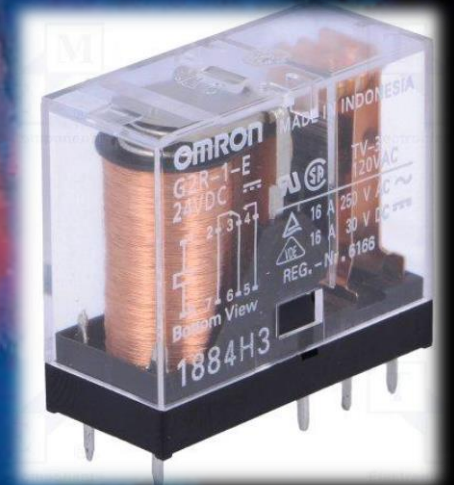
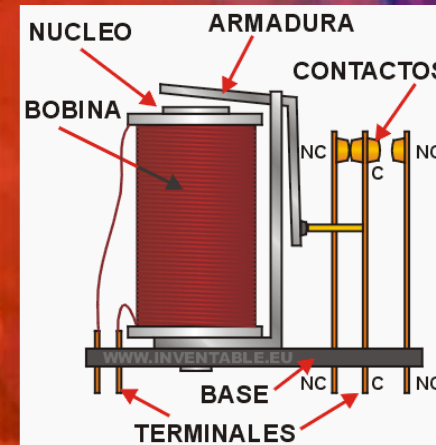
RECOMENDACIONES

- En caso de usar mas de 1 servomotores es recomendable utilizar una alimentación externa en caso de no hacerlo podríamos dañar nuestro Node mcu.
- No olvidar colocar los GND en común.
- Para el manejo de varios servomotores se puede optar por las Shields para servomotores.



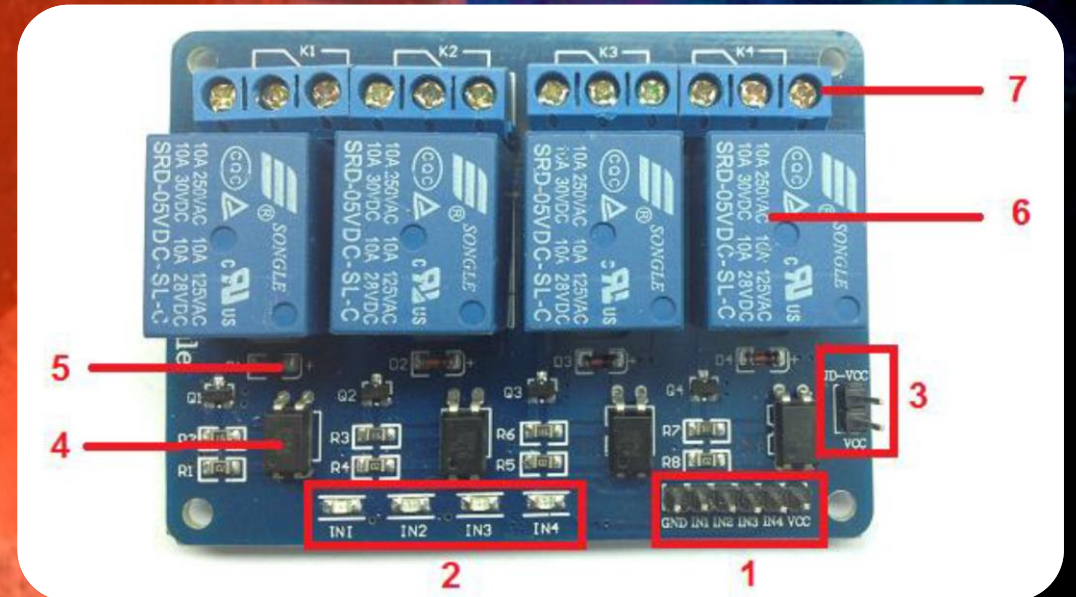
MODULO RELÉ

- Un relé o relevador es un interruptor que podemos activar mediante una señal eléctrica.
- Es un pequeño electro-imán (una bobina) que cuando lo excitamos con una señal eléctrica mueve su condición de un contacto eléctrico de conectado a desconectado o viceversa.
- Normalmente se usa un relé cuando se requiera conmutar grandes picos de tensión o intensidad como por ejemplo: arrancar motores de corriente alterna.
- Un relé puede tener las siguientes configuraciones: Normalmente abiertos NA o normalmente cerrados NC.



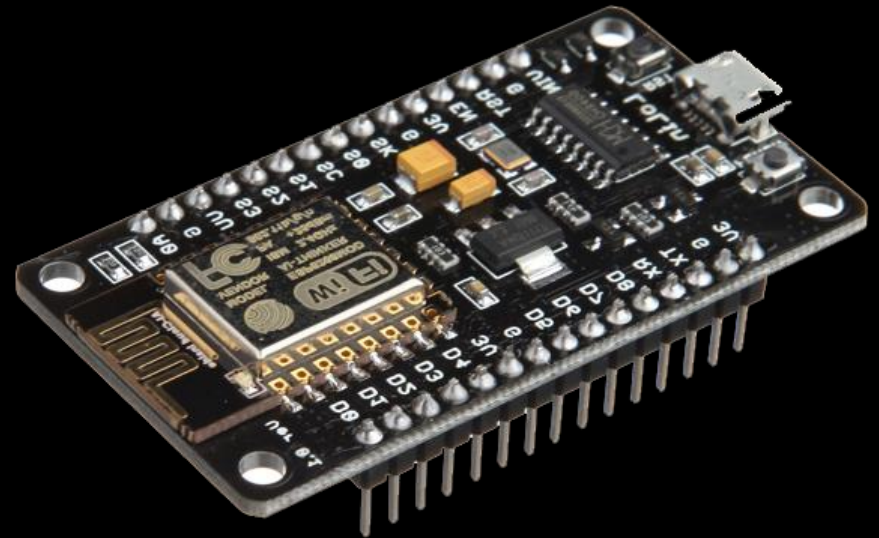
PARTES DEL MODULO RELÉ

1. La placa tiene varios pines de entradas (EN1-EN4) y alimentación (GND y VCC).
2. Un led por cada relé que indican el estado de las entradas.
3. Un jumper selector para la alimentación de los relés.
4. Un opto acoplador por cada relé (tipo FL817C) .
5. Un diodo de protección por cada relé.
6. Un relé marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10Am en una tensión de 250V.
7. Tres borneras por cada relé (Común, Normal Cerrado y Normal Abierto) para las salidas.



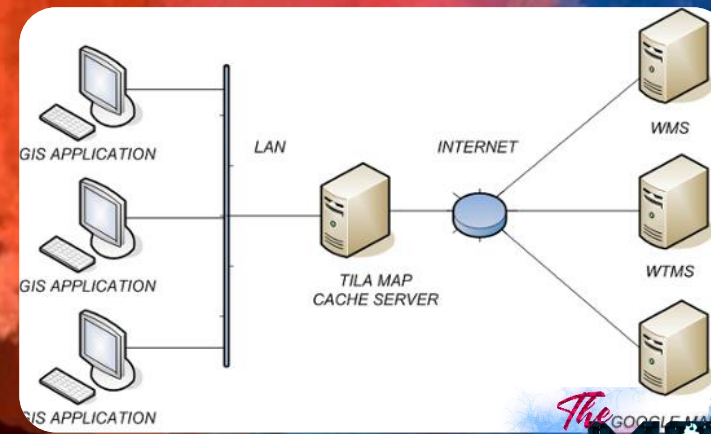
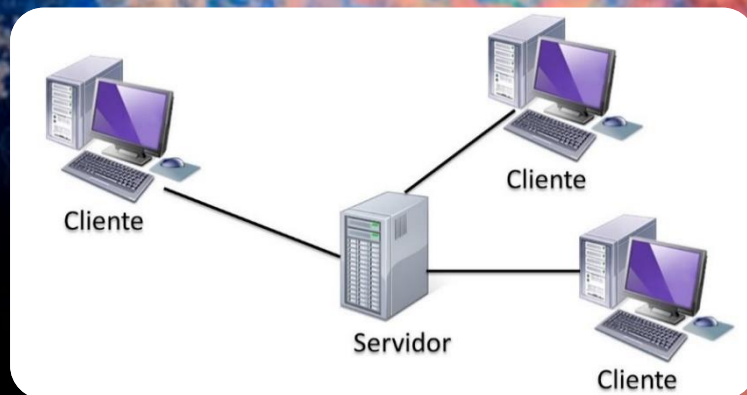
The MAKERS INFORMATICA

SERVIDOR WEB



¿QUÉ ES UN SERVIDOR?

Un **servidor** es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras dedicadas a las cuales se les conoce individualmente como «el servidor». En la mayoría de los casos una misma computadora puede proveer múltiples servicios y tener varios servidores en funcionamiento.



PROTOCOLO HTTP

HyperText Transfer Protocol (Protocolo de Hipertransferencia de texto) es un protocolo de comunicación que permite la transferencia de datos en la WWW (World Wide Web).

Utiliza el puerto de red 80, este protocolo puede ser utilizado sin conexión y sin estado.

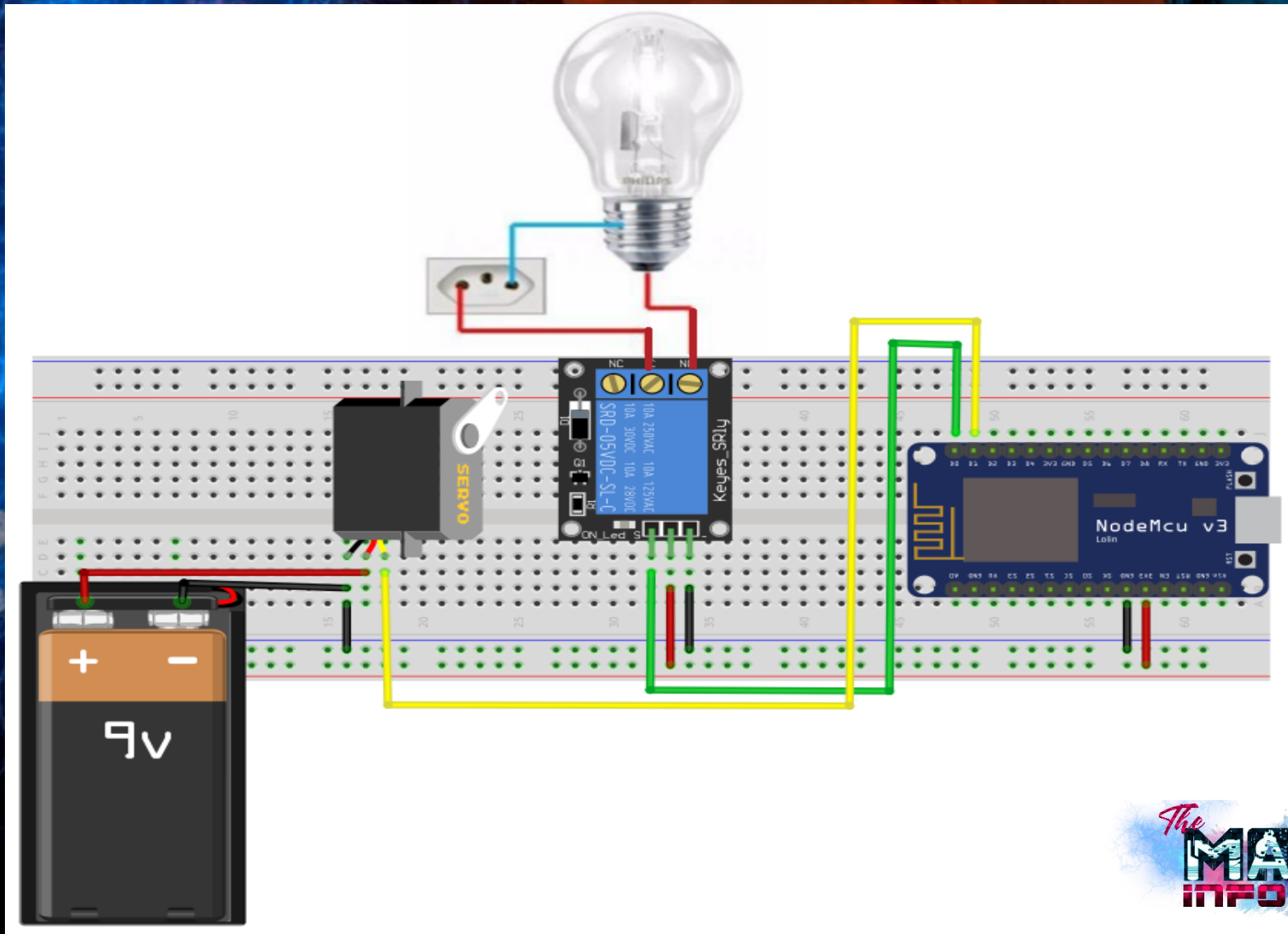
Métodos de petición y envío de datos

- GET: Transmite la información a través de la URL.
- POST: Envía los datos para que sean procesados solo por el recurso identificado.

FUNCIONAMIENTO DEL SERVIDOR WEB



CONEXIÓN



CÓDIGO DE NUESTRO SERVIDOR

```
#include <ESP8266WiFi.h>
#include <Servo.h>
Servo serv;
#define servo 5 //definimos el servomotor
#define rele 16 // definimos el pin para el rele
int value = LOW;
const char* ssid = "MAKERS";//nombre de red wifi
const char* password= "M@kersInformatic@";//contraseña de red WiFi

WiFiServer server(80);//puerto por el que se conecta al servidor del nodemcu
int selector;
```


CÓDIGO DE NUESTRO SERVIDOR

```
void setup() {  
  pinMode(rele, OUTPUT);  
  serv.attach(servo);  
  Serial.begin(115200); //velocidad de comunicacion optima para el node mcu  
  Serial.println();  
  Serial.println("conectando a...");  
  Serial.println(ssid);  
  WiFi.begin(ssid,password); //inicia la conexion a la red WiFi  
  //verificacion de que se conecto a la red WiFi  
  while(WiFi.status() !=WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.print("");  
  Serial.println("WiFi conectado");  
  Serial.println("Direccion ip: ");  
  Serial.println(WiFi.localIP()); // direccion IP del node mcu  
  //inicializamos el servidor  
  server.begin();  
  Serial.println("servidor iniciado");  
}
```


CÓDIGO DE NUESTRO SERVIDOR

```
void loop() {  
  
    //comprobamos si hay un cliente disponible  
    //NOTA: si existe un cliente se realiza una operacion caso contrario no  
    WiFiClient cliente= server.available();  
    if(!cliente){  
        return;  
    }  
    //nueva peticion del cliente  
    Serial.println("nuevo cliente");  
    while(!cliente.available()){  
        delay(1);  
    }  
  
    // Leo la primera linea de la petición del cliente  
    String request = cliente.readStringUntil('\r'); // Leo hasta retorno de carro  
    Serial.println(request); //Imprimo la petición  
    cliente.flush(); //Limpio el buffer
```


CÓDIGO DE NUESTRO SERVIDOR

```
// Interpreta lo que ha recibido
if (request.indexOf("/RELE=ON") != -1) {
    digitalWrite(rele, HIGH);
    value = HIGH;
}
if (request.indexOf("/RELE=OFF") != -1) {
    digitalWrite(rele, LOW);
    value = LOW;
}

if (request.indexOf("/GARAJE=ON") != -1) {
    abrirg(); //funcion para mover el servo y simular que se abre una puerta
}
if (request.indexOf("/GARAJE=OFF") != -1) {
    cerrarg();//funcion para mover el servo y simular que se cierra una puerta
}
```

Busca la posición de la acción
que le consultamos.
Request.indexOf("acción")

CÓDIGO DE NUESTRO SERVIDOR

```
// Devuelvo la respuesta al cliente -> Todo ha ido bien, el mensaje ha sido interpretado correctamente
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println(""); // do not forget this one

// A partir de aquí se crea la página HTML, el cliente.println es el formato que maneja el node mcu para interpretar el código html
cliente.println("<!DOCTYPE HTML>");
cliente.println("<html>");
  cliente.println("<center>");
cliente.print("EL RELE ESTA: ");
if(value == HIGH) {
  cliente.print("ON");

} else {
  cliente.print("OFF");
}
```


CÓDIGO DE NUESTRO SERVIDOR

```
// Los botones con enlace
cliente.println("<br><br>");
cliente.println("RELE  ");
cliente.println("<a href=\"\"/RELE=ON\"\"><button>Encender </button></a>");
cliente.println("<a href=\"\"/RELE=OFF\"\"><button>Apagar </button></a><br />");
cliente.println("<br><br>");

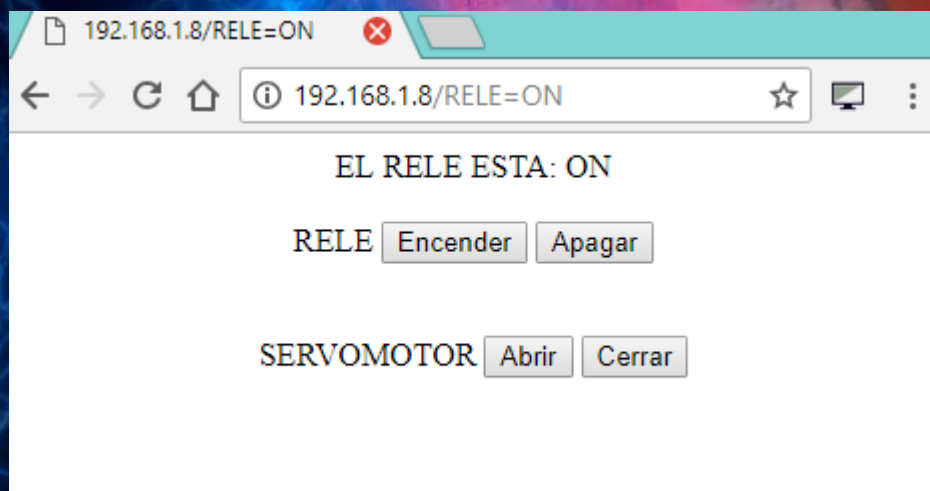
cliente.println("SERVOMOTOR  ");
cliente.println("<a href=\"\"/GARAJE=ON\"\"><button>Abrir </button></a>");
cliente.println("<a href=\"\"/GARAJE=OFF\"\"><button>Cerrar </button></a><br />");
cliente.println("<br><br>");
cliente.println("</center>");
cliente.println("</html>");

delay(1);
Serial.println("Cliente desconectado"); // Nos desconectamos del cliente
Serial.println("");
}
```


CÓDIGO DE NUESTRO SERVIDOR

```
//funcion para mover el servo y simular que se abre una puerta
void abring(){
    for(int i=0;i<=105;i++){
        int a=160-i;
        serv.write(a);
        delay(70);
    }
}
//funcion para mover el servo y simular que se cierra una puerta
void cerrarg(){
    for(int i=55;i<=160;i++){
        serv.write(i);
        delay(90);
    }
}
```


SALIDA DEL SERVIDOR



```
COM
.....WiFi conectado
Direccion ip:
192.168.1.8
servidor iniciado
nuevo cliente
GET /RELE=ON HTTP/1.1
Cliente desconectado

nuevo cliente
GET /favicon.ico HTTP/1.1
Cliente desconectado

nuevo cliente
GET /RELE=OFF HTTP/1.1
Cliente desconectado

nuevo cliente
GET /favicon.ico HTTP/1.1
Cliente desconectado
```

Dirección de
nuestro servidor