

Explicación del código del archivo led.py:

Las dos primeras líneas son imports de librerías, la primera son librerías del sistema y la segunda son librerías para poder trabajar con los Gpio. Crea un objeto gpio para poder trabajar con sus métodos.

En la línea 3 encontramos un método de gpio para que no nos salten alertas.

Luego hay dos variables que guardaremos en Python en función de lo que le pasemos en el script. En una variable guardamos el pin del gpio y en la otra el estatus de encendido o apagado.

Después vemos dos Print, que sirven para que el controlador cuando el parámetro de salida out nos devuelve algo de led.py, esta salida out es la salida que va mostrar los Print.

Con el método setmode decimos en el gpio como estamos llamándolo.

El método setup dice que gpio queremos que este en salida o entrada. Al ser un led, necesita recibir voltaje para encenderse, así que estará en salida (out).

Y por último tenemos un condicional que lo que hace es que según el estatus pone el estatus en true o false, con voltaje o sin voltaje.

Explicación del código del archivo api.php:

Lo primero es guardar en dos variables el pin del gpio con el que interactuamos y el estatus de encendido o apagado.

Tenemos un condicional por el cual decimos que si no pasamos los parámetros de numero del pin del gpio y el estatus, automáticamente saltaría al else y nos daría un error.

En el caso de que los parámetros lleguen correctamente haríamos lo siguiente:

Con el comando exec ejecutaríamos un comando como en la terminal, en este caso el script led.py. Pasamos el pin gpio con el que queremos interactuar, el estatus y el parámetro de salida.

Luego tenemos dos cabeceras (header) que nos devuelven la respuesta de que todo ha ido bien con un código 200 o ha ido mal con un código 400. Luego hay otra variable en la cabecera que nos sirve para decirle al cliente web que lo que se devuelve es un archivo de tipo Json.

En el archivo Json, una de las claves es el gpio pasado por parámetro, el estatus y la salida que devuelve el script de Python. Todo esto es un Array asociativo, lo bueno de php es que con la función json_encode puede coger un Array y convertirlo en un json.

Y lo último sería imprimir el json para que en Postman viéramos la respuesta.

Explicación del código del archivo cliente.php:

Lo primero que haremos para crear la lista de números aleatorios es crear un array.

Después, declararemos cuántos números queremos en nuestra lista de números aleatorios en php y con este número creamos un bucle for.

En cada una de las interacciones del bucle vamos a crear el número aleatorio con rand y a insertarlo en la lista. Para insertar en la lista de números aleatorios en php nos apoyamos en el método .push del array.

Ya tendremos la lista de números aleatorios en php rellena. Ya solo nos quedará mostrarla por pantalla recorriendo el array.