



## **Prototipo para una estación meteorológica mediante la tarjeta**

### **ESP8266 desplegado en 000WEBHOSTING**

#### *Prototype for a weather station using the ESP8266 card deployed at 000WEBHOSTING*

Tapuyo Vega Anderson Isaac<sup>1</sup>

Jacome Molina Steven Christopher<sup>2</sup>

Ing. Bonilla Agualongo Paul Agustín, MSc.<sup>3</sup>



0000-0003-1978-3079

<sup>1</sup> Instituto Superior Tecnológico Tsa'chila, Ecuador [andersontapuyovega@tsachila.edu.ec](mailto:andersontapuyovega@tsachila.edu.ec)

<sup>2</sup> Instituto Superior Tecnológico Tsa'chila, Ecuador [stevenjacomemolina@tsachila.edu.ec](mailto:stevenjacomemolina@tsachila.edu.ec)

<sup>3</sup> Instituto Superior Tecnológico Tsa'chila, Ecuador [paulbonilla@tsachila.edu.ec](mailto:paulbonilla@tsachila.edu.ec)

**Recepción:** 22 de febrero de 2022

**Aceptación:** 12 de marzo de 2022

**Publicación:** 29 de abril de 2022

**Citación/como citar este artículo:** Tapuyo, A., Jácome, S. y Bonilla, P. (2022). Prototipo para una estación meteorológica mediante la tarjeta ESP8266 desplegado en 000WEBHOSTING. Ideas y Voces, 2(1), 66-93.



## **Resumen**

El tema diseño de un prototipo de control mediante tecnología IOT enfocado a sistemas domóticos muestra un sistema prototipo desarrollado previo a la investigación de sistemas similares y protocolos de comunicación basados en IOT. Se utilizó tarjetas de desarrollo de hardware libre, componentes electrónicos de control y potencia integrados todos en una placa de circuito impreso (pcb), adicionalmente se hizo uso de software de desarrollo libre para la creación de una página web para el control de cargas y monitoreo de sensores. La validación de este trabajo se realizó mediante pruebas de funcionamiento del prototipo como de la página web.

## **Palabras clave**

IOT, hardware libre, placa de circuito impreso, software

## **Abstract**

The theme design of a control prototype through IOT technology focused on home automation systems shows a prototype system developed prior to the investigation of similar systems and communication protocols based on IOT. Free hardware development cards were used, control and power electronic components all integrated into a printed circuit board (pcb), additionally free development software was used to create a web page for load control and monitoring. of sensors. The validation of this work was carried out through functional tests of the prototype and of the web page.

## **Keywords**

IOT, free hardware, Printed circuit board, software.

## Introducción

Hoy en día la internet es una herramienta muy usada e indispensable por estudiantes, profesionales, entre otros, desde su creación ha revolucionado la forma de comunicar e informar, lo que ha dado paso a la mejora en investigación en todos los ámbitos. Sin embargo, en los últimos años ha dado un paso agigantado con el surgimiento de la Internet de las cosas (IoT), este tipo de tecnología combina el envío y recepción de información con objetos físicos. Con esto se pretende mejorar la calidad de vida de las personas proponiendo soluciones a tareas repetitivas del hogar, además aportando comodidad, bienestar, eficiencia energética y seguridad.

A lo largo de los años de la creación de la carrera de Tecnología Superior en Electrónica (TSE) se han desarrollado diferentes prototipos domóticos como trabajos de integración curricular que en algunos casos han quedado rezagados y sin uso por falta de validación o factor económico.

El presente trabajo procura formar parte de un proyecto con el cual se pueda obtener un producto con las respectivas pruebas y validaciones, de esta manera el resultado cumplirá las funciones mínimas de un sistema domótico el cual pretende estar actualizado con las tecnologías que se rigen en el mercado como la domótica y los sistemas IoT.

Con el pasar de los años ha mejorado la calidad de vida de las personas debido al desarrollo tecnológico. Actualmente la mayoría de tareas se realizan por medio de un dispositivo móvil, por ejemplo: transacciones bancarias, estudios, pedidos o compras en línea, videoconferencias, entre otras. Se produjo un aumento de dispositivos tecnológicos a consecuencia del estado actual de distanciamiento social por la pandemia de Covid19 y sus variantes. Un escenario que no ha cambiado en su totalidad es la vivienda, este espacio alberga tanto a personas como a bienes materiales, y permite la convivencia entre ellos. (Varvariche Catalin, 2021).

Un sistema domótico se basa en la adquisición de información para posteriormente ser procesada y emitir órdenes. La domótica ofrece una amplia gama de soluciones a todo tipo de vivienda contribuyendo a la comodidad, bienestar y seguridad.

Actualmente existen módulos domóticos en el mercado como es el caso del IPX800v4, este dispone de entradas digitales, entradas analógicas de 10bits de resolución, y salidas a relé. Posee su propia interfaz gráfica para el control y monitoreo de cargas.

Los tecnólogos egresados de la carrera de TSE han desarrollado prototipos domóticos, pero estos en algunos casos han quedado sin uso

## **Metodología**

### **Ubicación**

Parte del desarrollo del prototipo se hizo en los laboratorios de la carrera de Tecnología Superior en electrónica, del Instituto Superior Tecnológico Tsáchila, con una duración de 6 meses.

Para el desarrollo del proyecto se han ejecutado los diferentes procesos expuestos a continuación:

### **Investigación**

Mediante hojas de datos, documentos digitales, artículos académicos y foros se investigó la información pertinente a la tecnología de la tarjeta de desarrollo utilizada.

### **Tarjeta de desarrollo**

Las tarjetas desarrollo basadas en el chip ESP8266 y ESP32 están diseñadas para implementar prototipos de IOT, ya que éstas poseen la capacidad de conectarse a la internet de manera inalámbrica además de poseer algunos pines GPIO (Entrada/Salida de Propósito General) y pines ADC (Conversor de señal Analógica/Digital).

Para seleccionar la tarjeta se tomó en cuenta la cantidad de GPIO que ofrece la misma.

En este caso fue el NodeMCU ESP8266 V1, ya que dicha tarjeta dispone de 9 pines

digitales y 1 analógico. Funciona con 3.3V esto quiere decir que emite señales de control de 3.3V y puede leer de igual manera señales entre 0V a 3.3V. El regulador AMS1117 suministrará la tensión de 3.3V necesaria para el funcionamiento del chip. Por último, dispone de un conversor USB a TTL CP2102.

### **Dispositivos de control**

Se determinó el uso de dispositivos electromecánicos como los relés y semiconductores de potencia como TRIAC's para controlar cargas. Esto debido a su robustez y factibilidad de aplicación.

### **Método para el desarrollo de PCB**

Actualmente existen muchos métodos para desarrollar o implementar una PCB, el más usado es mediante empresas industriales extranjeras especializadas en el campo que entregan PCB's de tipo profesional, pero presenta un inconveniente que es el tiempo de entrega. Por otra parte, existen empresas a nivel nacional que realizan este trabajo por el método de CNC (Control Numérico por Computadora).

### **Diseño electrónico**

Una vez investigado el método de fabricación de PCB y los componentes para la creación de la PCB se procede a dimensionar y diseñar la tarjeta electrónica.

### **Cálculos para activar relés.**

Se utilizó un transistor para activar el relé debido a que el NodeMCU emite una señal de control de 3.3V y el relé necesita 5V, por lo cual se usa un transistor que ayudará a separar esa diferencia de tensión. A continuación, se presentan las características técnicas de mayor interés de los componentes electrónicos utilizados.

### **Tabla 1.**

Características de relé SRD-05VDC-SL-C

<b>Característica</b>	<b>Valor</b>
-----------------------	--------------

Resistencia de la bobina ( $R_{relé}$ )	70 $\Omega$
Voltaje ( $V_{elé}$ )	5VDC

(AllDataSheet, 2022)

**Tabla 2.**

Características de transistor 2N3904

Característica	Valor
Ganancia ( $h_{FE}$ )	300
Corriente de colector-emisor ( $I_{C-E}$ )	200mA

(AllDataSheet, 2022)

**Tabla 3.**

Características de Diodo LED

Característica	Valor
Voltaje ( $V_{LED}$ )	2,4V
Corriente ( $I_{LED}$ )	10mA

Con la información detallada del dispositivo relé se realizaron los cálculos de corriente de consumo.

$$I_{relé} = \frac{V_{Circuito}}{R_{relé}}$$

$$I_{relé} = \frac{5V}{70\Omega}$$

$$I_{relé} = 71mA$$

Donde:

$V_{Circuito}$  = Es la tensión de operación del circuito

Se procedió a calcular la corriente que circula por el diodo LED cuando se utiliza una resistencia en serie.

$$I_{LED} = \frac{V_{Circuito} - V_{LED}}{R_{LED}}$$

$$I_{LED} = \frac{5V - 2,4V}{330\Omega}$$

$$I_{LED} = 7mA$$

De la misma manera se realiza el cálculo de corriente que circulará en colector tomando en cuenta la corriente consumida por el relé y el diodo LED.

$$I_C = I_{relé} + I_{LED}$$

$$I_C = 71mA + 7mA$$

$$I_C = 78mA$$

Donde:

$$I_C = \text{Corriente de colector}$$

Para calcular la corriente de base se tomó el peor de los casos en donde la ganancia ( $h_{FE}$ ) es inferior a la que ofrece el fabricante.

$$I_B = \frac{I_C}{h_{FE}}$$

$$I_B = \frac{78mA}{150}$$

$$I_B = 520\mu A$$

Donde:

$$I_B = \text{Corriente de base}$$

El cálculo de la resistencia de base se realizó tomando en cuenta la corriente consumida por los dispositivos mencionados anteriormente y la corriente base necesaria para permitir circular la corriente de colector calculada.

$$R_B = \frac{V_{NodeMCU} - V_{BE}}{I_B * 2}$$

$$R_B = \frac{3.3V - 0.7V}{520\mu A * 2}$$

$$R_B = 2.5K\Omega$$

$$R_B = 2.2K\Omega \text{ "Resistencia Comercial"}$$

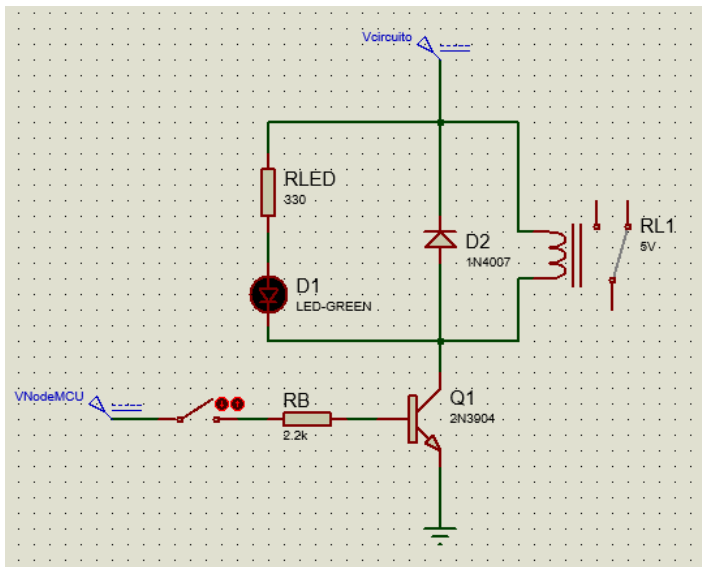
Donde:

$$V_{NodeMCU} = \text{Voltaje de control}$$

$$R_B = \text{Resistencia de base}$$

$$V_{BE} = \text{Voltaje de base-emisor}$$

Se calculó una resistencia de base ideal de  $2.5K\Omega$ , pero debido a que esta resistencia no existe en el mercado se optó por usar una de un valor inferior. Todas las conexiones se pueden apreciar en la Figura 10.



**Figura 10.** Esquema de conexiones para la activación del relé

**Fuente:** Autores

**Nota:** Es necesario utilizar un diodo de protección conectado en paralelo a la bobina del relé.

### Cálculo para activar TRIAC

Se presenta en la Figura 18 la conexión de los componentes, en donde es utilizado un OptoTRIAC (MOC3021) para aislar la tensión de control en DC con la tensión de

potencia en AC, además este dispositivo permite activar el TRIAC (BT136) por medio del terminal TRIGGER. A continuación, en la Tabla 10 y Tabla 11 se presentan las características destacables de los componentes utilizados.

**Tabla 10.**

Características del OptoTRIAC MOC3021

Característica	Valor
Voltaje	3V
Corriente	10mA

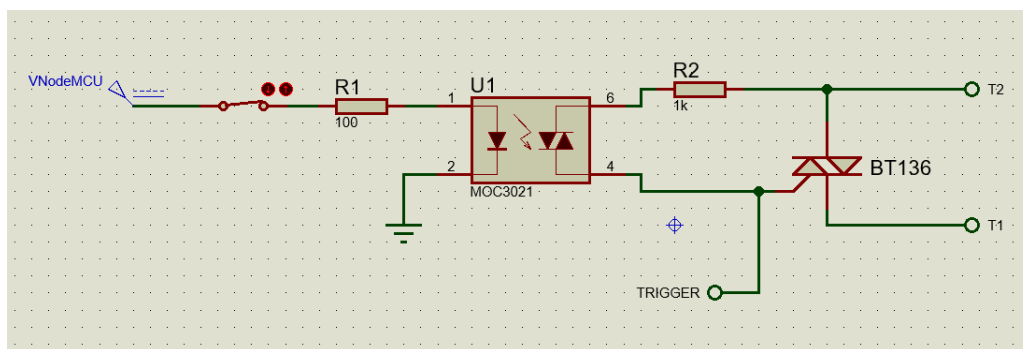
(AllDataSheet, 2022)

**Tabla 11**

Características del TRIAC BT136

Característica	Valor
Corriente de disparo	100mA
Voltaje máximo de operación	600V
Corriente máxima repetitiva	4 <sup>a</sup>

(AllDataSheet, 2022)

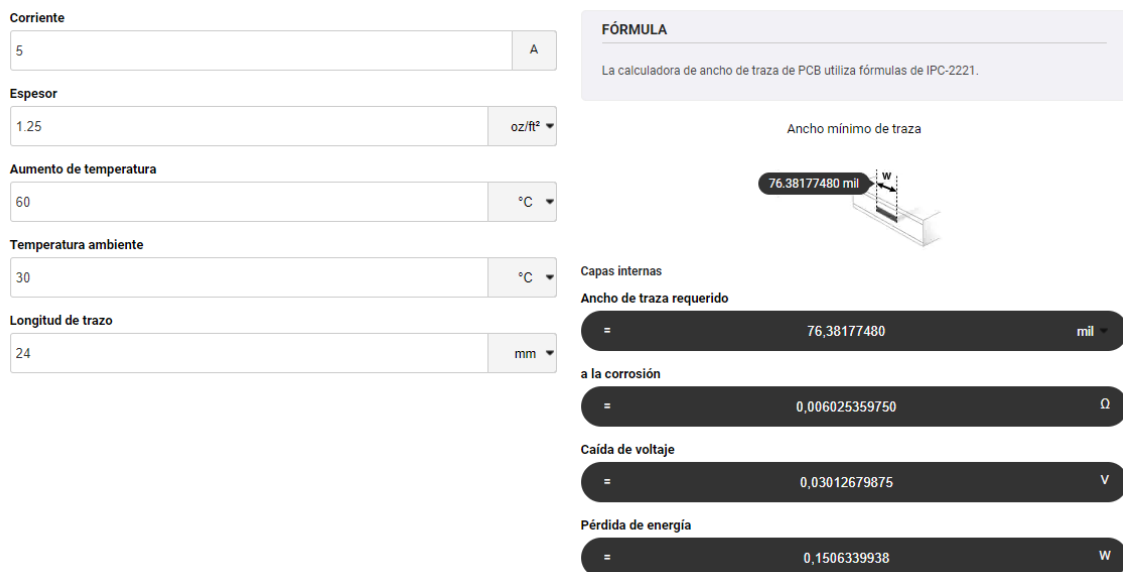


**Figura 11.** Esquema de conexiones para la activación del TRIAC.

**Fuente:** Autores

## Dimensionamiento de pistas

Para el dimensionamiento de las pistas de potencia se tomó en cuenta la corriente máxima que soportan los componentes. En el caso del relé, sus especificaciones técnicas establecen que soporta una corriente de 10A, pero para el diseño se estableció un ancho de pista de 80TH que idealmente soportarían más de 5A como se indica en la Figura 12. Para la activación de la carga por el TRIAC se utilizó un ancho de pista de 60TH la cual soporta 4A tal como se puede evidenciar en la Figura 13. Por otra parte, para el dimensionamiento de las pistas de datos o control se optó por utilizar el mínimo recomendado que es de 25TH y soporta 2.2A como se puede observar en la Figura 14.



Input	Value	Unit
Corriente	5	A
Espesor	1.25	oz/ft <sup>2</sup>
Aumento de temperatura	60	°C
Temperatura ambiente	30	°C
Longitud de trazo	24	mm

**FÓRMULA**  
La calculadora de ancho de traza de PCB utiliza fórmulas de IPC-2221.

Ancho mínimo de traza

76.38177480 mil

Capas internas

Ancho de traza requerido = 76,38177480 mil

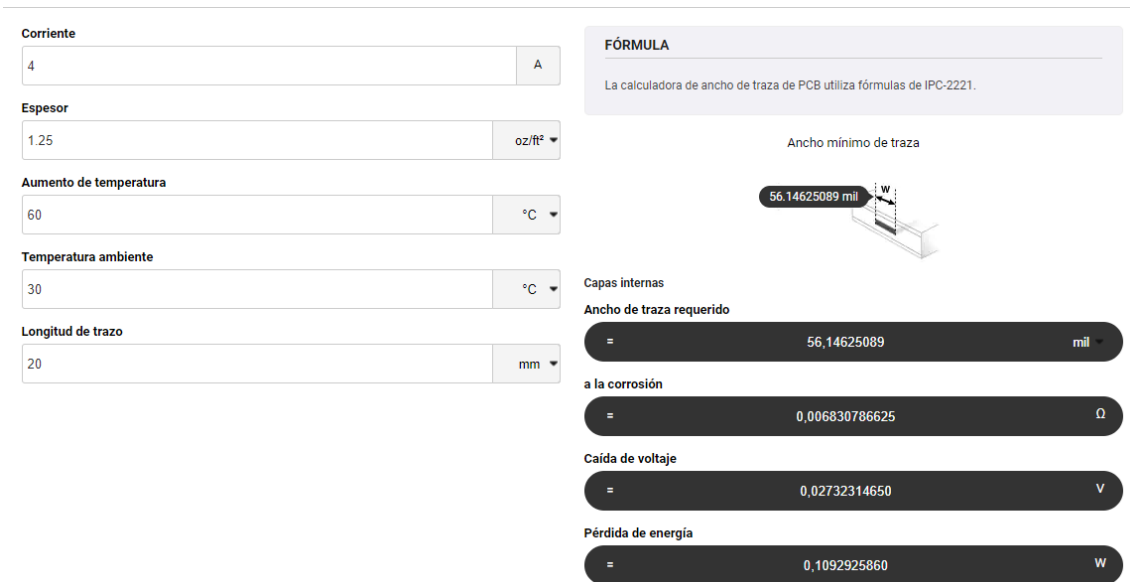
a la corrosión = 0,006025359750 Ω

Caída de voltaje = 0,03012679875 V

Pérdida de energía = 0,1506339938 W

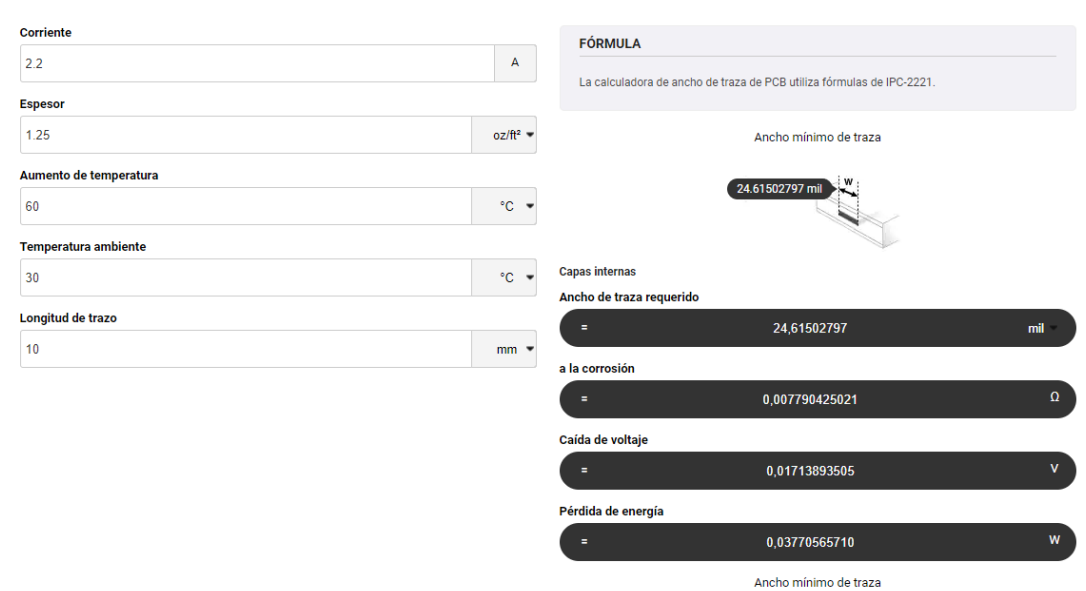
**Figura 12.** Dimensionamiento para la pista de potencia del relé

**Fuente:** Autores



**Figura 13.** Dimensionamiento para la pista de potencia del TRIAC

**Fuente:** Autores



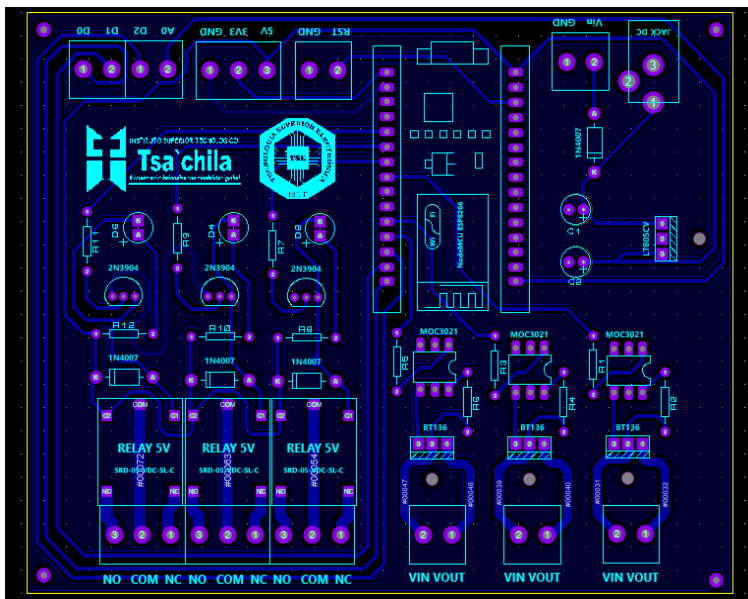
**Figura 14.** Dimensionamiento de Pistas de Control

**Fuente:** Los autores

### Diseño PCB

El diseño esquemático y el diseño PCB fueron desarrollados en el programa Proteus, algunos de los componentes se crearon ya que su empaquetado no existe, se tomó en

cuenta las dimensiones, la cantidad de pines y el ancho del terminal tal como se muestra en la Figura 15.



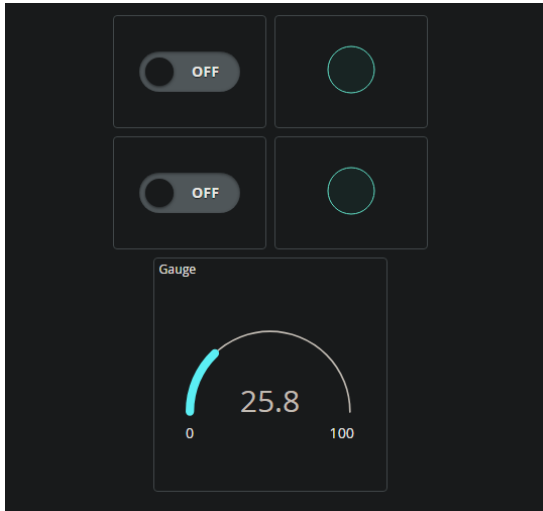
**Figura 15.** Diseño final de PCB

**Fuente:** Autores

### **Diseño de la interfaz gráfica de control en Arduino Cloud IOT**

Se diseñó una interfaz gráfica en Arduino Cloud IOT para el control de actuadores y el monitoreo de sensores, ver Figura 16.

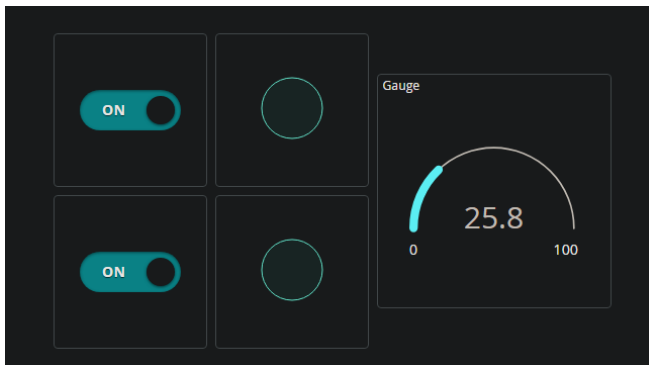
Desde la plataforma Arduino Cloud se agrega los componentes o gadget para el control y monitoreo de actuadores o sensores.



**Figura 16.** Control y monitoreo en Arduino IOT Cloud

**Fuente:** Autores

Arduino IOT Cloud envía automáticamente el estado de los switches, así como cada cambio realizado al ESP8266.



**Figura 17.** Gadgets Arduino IOT Cloud

**Fuente:** Autores

Arduino IOT Cloud automáticamente incluye las librerías necesarias para la comunicación con el ESP8266, además, declara variables y crea funciones. Solo se debe incluir el código secreto de Arduino IOT Cloud para el enlace de la placa y agregar las credenciales para la conexión a internet.

```

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[] = "6740f302-5bc5-460a-a8b1-129662895891";

const char SSID[] = SECRET_SSID; // Network SSID (name)
const char PASS[] = SECRET_PASS; // Network password (use
const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device password

void onTemporizadorChange();
void onLed1Change();
void onLed2Change();
void onPotChange();

CloudSchedule temporizador;
bool led_1;
bool led_2;
CloudTemperature pot;

```

SECRET_SSID	D@río
SECRET_PASS	Dario@HM1944.
SECRET_DEVICE_KEY	JOQO44J7AYQ7BGSSLI

Figura 18. Código creado por Arduino Cloud

**Fuente:** Autores

El NodeMCU recibe el estado de cada switch de la plataforma de Arduino. Dentro de las funciones se programa y se debe incluir las variables, ya que si la variable no se encuentra en su correspondiente función y se realiza un cambio de estado del switch el NodeMCU no detectará ni ejecutará ese cambio.

```

129 void onLed1Change() {if (led_1==true) { a1(); } else {a2();}}
130 void onLed2Change() {if (led_2==true) { b1(); } else {b2();}}
131 void onLedTpChange () {if (digitalRead(LED_BUILTIN)== LOW) {led_tp=true;} else {led_tp=false;}}
132 void onTemporizadorChange () {if (temporizador.isActive()) {digitalWrite(LED_BUILTIN, LOW);} else {digitalWrite(LED_BUILTIN, HIGH);}}

```

Figura 19. Funciones de Arduino IOT Cloud

**Fuente:** Autores

Todo el proceso es posible gracias a una API, que expone un punto final con el que un cliente puede interactuar.

Primero el cliente solicita registrar un nuevo dispositivo en Arduino IOT Cloud a través de la API, a la que el servidor (API) devuelve un UUID (Universally Unique Identifier).

Este código es responsable de generar una clave privada y almacenarla en una ranura segura que solo pueda ser leída por el elemento seguro.

### Control por voz usando Google asistente

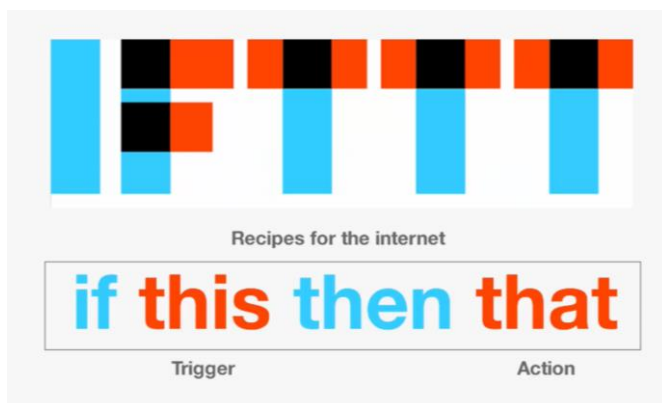
Para el control por voz se usa del servicio del asistente de Google.



**Figura 20.** Google Asistente

**Fuente:** Autores

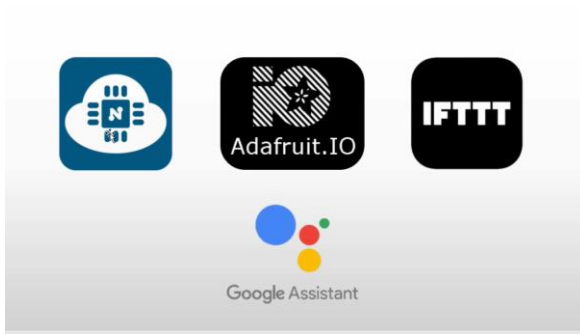
Este servicio esta enlazado a IFTTT (If This Then That), que es un servicio web que permite crear y programar acciones para automatizar tareas y realizar distintas acciones.



**Figura 21.** IFTTT (if this then that)

**Fuente:** Autores

A su vez, IFTTT se encuentra enlazado a Adafruit.io y este último al NodeMCU, encontrándose todos estos servicios directamente e indirectamente enlazados entre ellos.



**Figura 22.** NodeMCU, ADAFRUIT, IFTTT y Google Asistente

**Fuente:** Autores

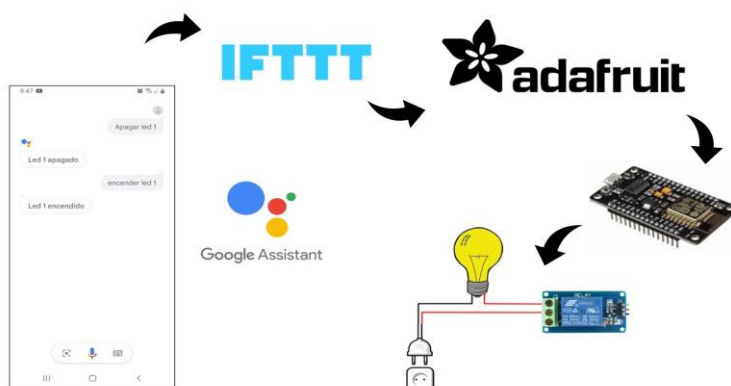
Al usar el asistente de Google para encender un led, se envía un valor de ON a Adafruit.io y este de igual manera envía el valor al NodeMCU.



**Figura 23.** Significado de IFTTT

**Fuente:** Autores

A continuación, se visualiza el proceso que se realiza para el encendido de un foco.



**Figura 24.** Proceso del control por voz

**Fuente:** Autores

### Diseño de la página web

Posteriormente se realizó una interfaz gráfica para poder ingresar a distintas redes wifi sin tener que modificar directamente el código del sistema.

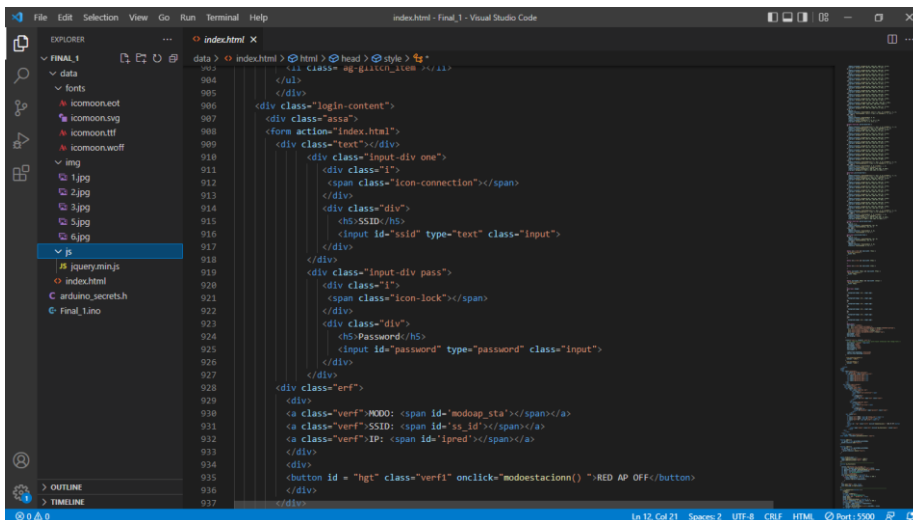
El programa que se usó para la creación de la página web es Visual Studio Code, la estructura de la página está escrita en HTML, el diseño en CSS y animaciones en JavaScript.



**Figura 25.** Logos de Visual Studio Code, CSS, HTML y JS

**Fuente:** Autores

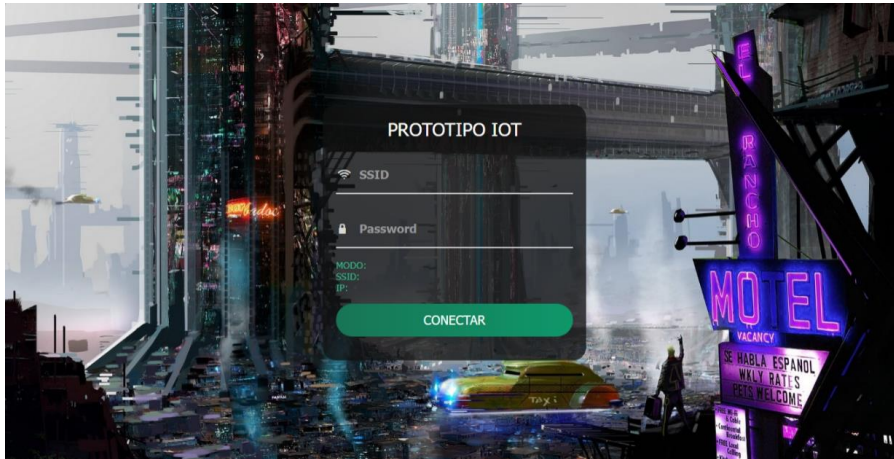
Para reducir el número de archivos se unió los archivos CSS al HTML, además, se incluyeron imágenes para el fondo y unos iconos.



**Figura 26.** Gráfica de una parte del código de la página web

**Fuente:** Autores

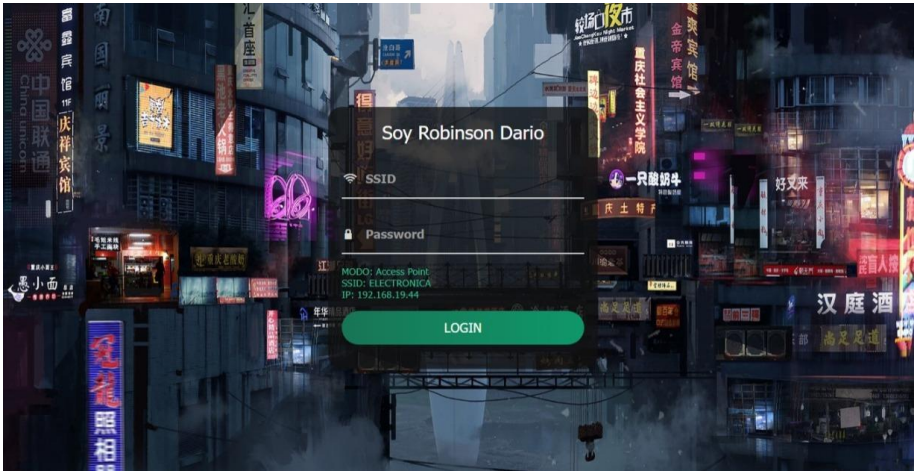
A continuación, se visualiza el resultado de la programación de la página web.



**Figura 27.** Inicio de la Interfaz Gráfica

**Fuente:** Autores

Los datos que se requiere para conectarse son dos: el nombre de la red WiFi (a la red que se quiere conectar) y la contraseña (de la red que se quiere conectar), una vez que ya se ha ingresado los parámetros se debe presionar el botón “CONECTAR” ver Figura 28. En la interfaz nos indica el modo de acceso del NodeMCU, el nombre de la red a la cual se encuentra conectada y la dirección IP, Si los datos ingresados son correctos el NodeMCU se conectará en modo Access Point y Station, y la dirección IP será de la red que se acaba de conectar, a la vez aparecerá un nuevo botón que permitirá al NodeMCU cerrar el Access Point y solo conectarse en modo Station. Cabe mencionar que una vez que el NodeMCU se encuentra conectada solo en modo Station la interfaz gráfica se cerrará. Los archivos HTML, CSS y JavaScript de la interfaz gráfica fueron tomados de modelos ya creados a los cuales se realizaron modificaciones del código para cumplir con el objetivo de la interfaz.



**Figura 28.** Interfaz gráfica para ingresar una red WiFi

**Fuente:** Autores

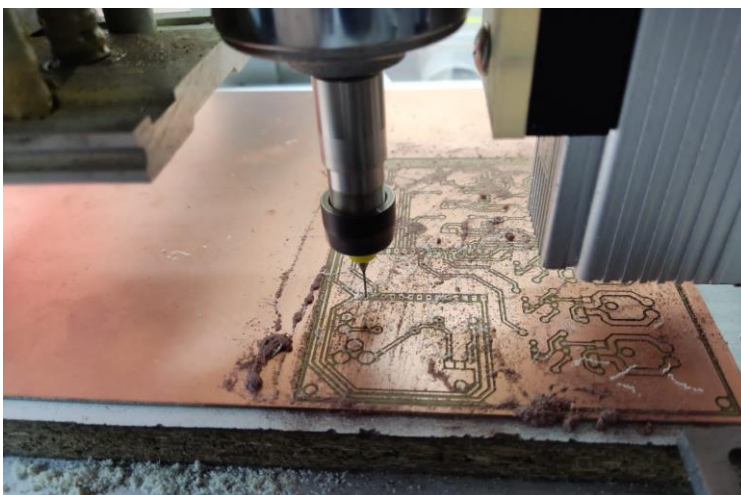
### **Implementación**

Con el diseño establecido se procedió a la fabricación de la PCB y la adquisición de los materiales.

### **Impresión CNC**

La PCB anteriormente mencionada se ha impreso por el método de CNC como se muestra en la Figura 29, en donde es utilizada una baquelita virgen con las dimensiones exactas.

Se realizan las perforaciones y las rutas de cada pista.



**Figura 29.** Fabricación la PCB

**Fuente:** Autores

### **Adquisición de materiales**

A continuación, se presentan los dispositivos utilizados en la PCB, así como el precio unitario y el precio total.

**Tabla 12.**

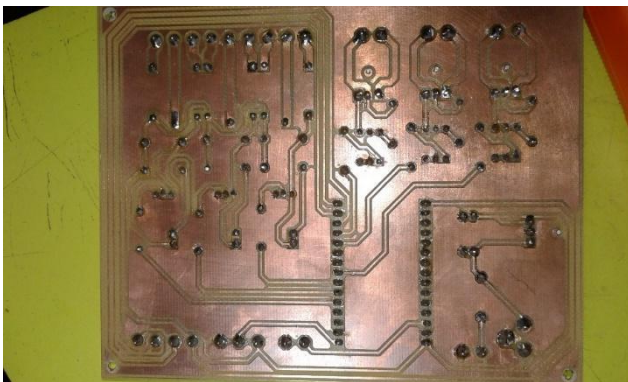
Tabla de costos

<b>Dispositivo/Equipo</b>	<b>Modelo</b>	<b>Cantida d</b>	<b>Precio unitario</b>	<b>Subtotal</b>
NodeMCU	MODESP8266	1	\$13,00	\$13,00
Transistor	2N3904	3	\$0,25	\$0,75
Relay	SRD-05-VDC- SL-C	3	\$1,50	\$4,50
Diodo	1N4007	4	\$0,25	\$1,00
Diodo LED	3mm	3	\$0,25	\$0,75
TRIAC	BT136	3	\$1,50	\$4,50
OptoTRIAC	MOC3021	3	\$1,50	\$4,50
Regulador 5VDC	L7805CV	1	\$1,00	\$1,00
Disipador de calor	TO-220	4	\$0,50	\$2,00
Aislante	TO-220	4	\$0,15	\$0,60
Capacitor electrolítico	25V/10 $\mu$ F	2	\$0,25	\$0,50
Resistencias	¼ W	12	\$0,05	\$0,60
Bornera	2 polos	7	\$0,15	\$1,05
Bornera	3 polos	4	\$0,25	\$1,00
Socket	6 pines	3	\$0,25	\$0,75
Regleta	Hembra	1	\$0,75	\$0,75

Jack	DC	1	\$0,75	\$0,75
Impresión CNC	-	1	\$20,00	\$20,00
Sensor de gas	MQ2	1	\$5,75	\$5,75
Sensor de movimiento	HC-SR501	1	\$6,00	\$6,00
Sensor de temperatura y humedad	DHT-11	1	\$4,00	\$4,00
			<b>Total</b>	<b>\$73,75</b>

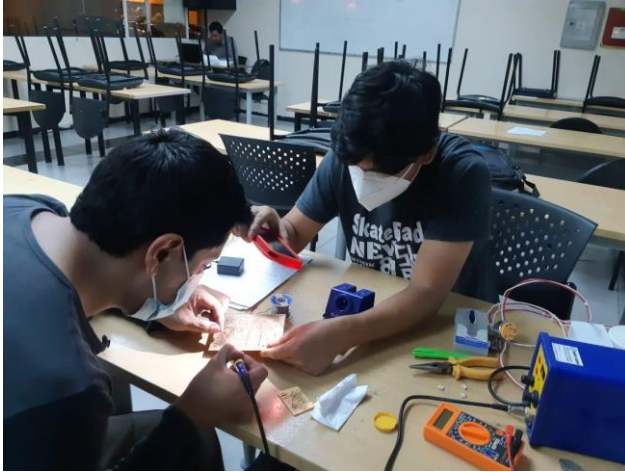
### Suelda de componentes

El diseño contiene componentes de tipo THT (Tecnología de Agujeros Pasantes), esto quiere decir que estos traspasan la PCB y los terminales de suelda quedan en la parte trasera, en la Figura 30 se puede apreciar lo mencionado. En la parte superior se encuentra la serigrafía o símbolos donde irán ubicados los componentes. La suelda fue realizada con una estación de soldadura, ver Figura 31.



**Figura 30.** Parte trasera de PCB.

**Fuente:** Autores



**Figura 31.** Proceso de solda de componentes electrónicos

**Fuente:** Autores

## **Resultados**

### **Resultados de hardware**

Al realizar las pruebas de funcionamiento se pudo detectar un inconveniente relacionado al arranque (Bootloader) del NodeMCU, el cual al momento de encender envía una señal de control por un corto periodo de tiempo generando un estado de uno lógico al pin D3 y cero lógicos al pin D4. Por lo tanto, debemos deshabilitar estos pines físicamente para que la tarjeta prototipo pueda arrancar y continuar con su trabajo, esto se solucionó retirando temporalmente el OptoTRIAC de su zócalo.

Otro punto o defecto que se debe tomar en cuenta es el método de alimentación, ya que se utilizó un conversor DC-DC lineal, dicho conversor trata de disminuir la tensión de entrada y el resto de la tensión no utilizada se transforma en calor, esto puede ser evidenciado al tener trabajando por mucho tiempo a la placa, el dissipador de calor llega a alcanzar una temperatura superior a los 70 grados centígrados.

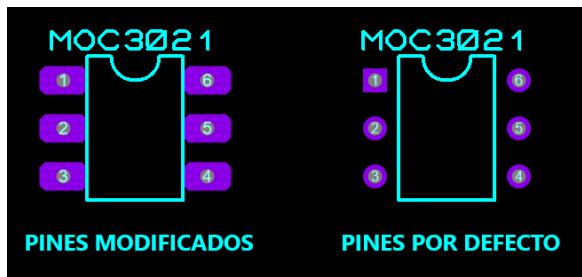
Por último, existieron fallas en el diseño de la PCB, esto debido a que no se dejó un espacio adicional para separar los relés ver Figura 32, además los pines del zócalo del

OptoTRIAC (por defecto en Proteus) tienen un área de suelda muy pequeña, se sugiere reemplazar los terminales por unos de mayor área ver Figura 33.



**Figura 32.** Error en el espacio de los relés

**Fuente:** Autores



**Figura 33.** Terminales de suelda

**Fuente:** Autores

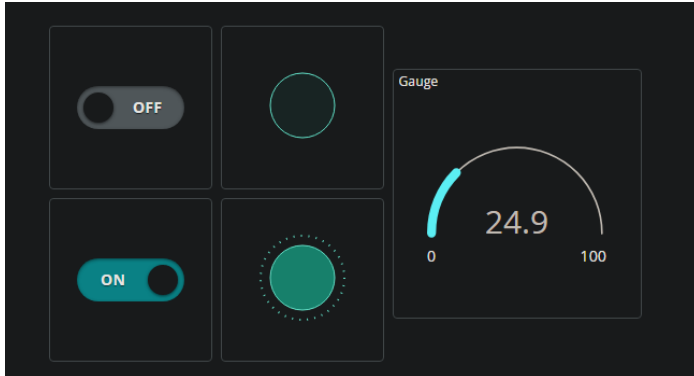
### Resultados de software

Arduino Cloud IOT es una plataforma de paga, esto quiere decir que el uso del 100% de sus servicios tiene cierto costo, pero también dispone de servicios gratuitos en los que permite el control de 5 variables para utilizar tal como se mostró en la Figura 34.

### Arduino IOT Cloud

La velocidad de respuesta va depender de la velocidad de internet que el usuario disponga, mientras mejor sea la velocidad de internet el sistema funcionara correctamente, en

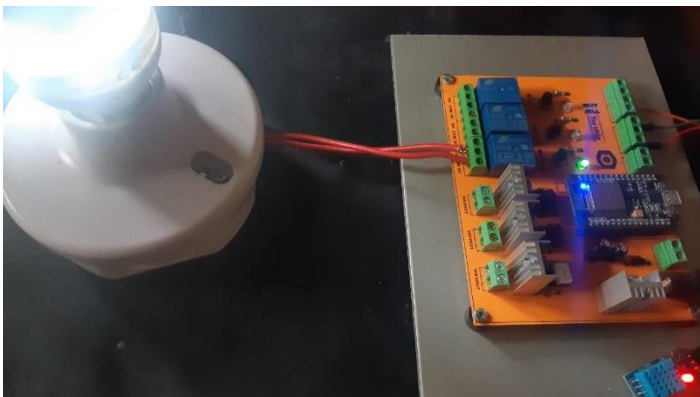
cambio, si la velocidad no es tan buena, el control por voz y el Arduino IOT Cloud llegará a desconectarse e incluso puede desconectarse el NodeMCU ocasionando que se repita el proceso de conexión. Se procede a encender el foco presionando el switch.



**Figura 34.** Switch ON en Arduino IOT Cloud

**Fuente:** Autores

Se observa en la siguiente imagen el encendido del foco a través del Arduino IOT Cloud.



**Figura 35.** Foco encendido a través de Arduino IOT Cloud

**Fuente:** Autores

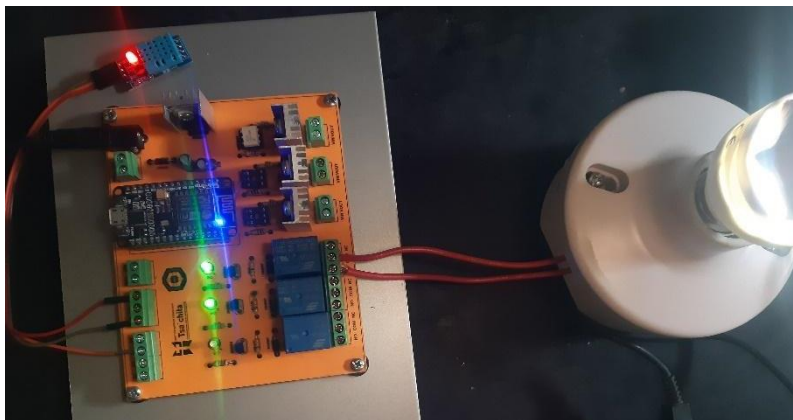
A continuación, se observa la conversación con Google Asistente.



**Figura 36.** Encendido de focos a través de Google Asistente

**Fuente:** Autores

En la siguiente imagen se muestra el resultado del encendido de un foco usando Google Asistente.



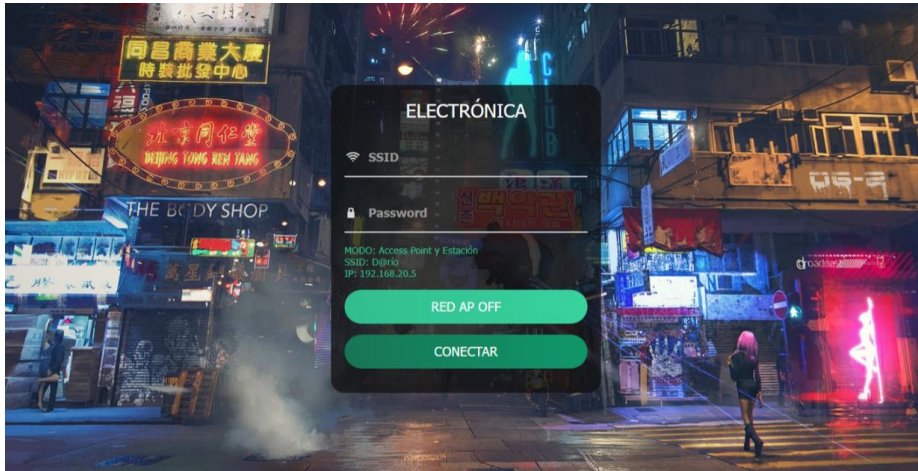
**Figura 37.** Resultado del uso de Google Asistente

**Fuente:** Autores

### Interfaz Gráfica

La visualización de la interfaz gráfica presenta lentitud, esto se debe a la poca memoria que posee el NodeMCU y la gran cantidad de archivos que se utilizaron.

Otro punto importante es el reinicio del NodeMCU, ya que para cambiar de red a una nueva se debe reiniciar, por lo que tomará un tiempo adicional hasta que el NodeMCU se conecte o levante el Access Point.



**Figura 35.** Resultado de la página web

**Fuente:** Autores

### **Conclusiones**

Se realizó la investigación pertinente a los dispositivos que integraron la tarjeta prototipo, sus características técnicas y dimensiones, ya que estas fueron necesarias para el diseño y dimensionamiento de la tarjeta prototipo.

Para el uso adecuado del NodeMCU se debe tomar en cuenta los pines de la tarjeta de desarrollo que pueden usarse sin interferir en el arranque D3 y D4 o en la transmisión de datos TX y RX.

Los resultados de la programación del software funcionan de acuerdo a lo esperado, ya que cumplen con su objetivo, mediante la aplicación móvil y los comandos de voz se pueden controlar cargas y verificar el estado de sensores, se presentan ligeros problemas del tiempo de respuesta, pero esto es producido por la latencia y el ancho de banda de la red en la que está conectado el dispositivo.

El uso de un regulador de tensión lineal no es conveniente, debido a que este dispositivo incrementa su temperatura, para evitar este problema se podría ocupar una pequeña fuente conmutada.

Los costos de implementación de la tarjeta prototipo son elevados, ya que algunos componentes no se encuentran fácilmente en las tiendas electrónicas locales y es necesario realizar pedidos para adquirirlos.

En vista de que la tarjeta fue desarrollada como prototipo se recomienda no exceder un tiempo de 24 horas de uso por el recalentamiento del regulador de tensión lo que podría ocasionar daños en la tarjeta.

Finalmente se puede concluir que la tarjeta prototipo trabaja activando cargas como iluminarias de 120V AC por medio de la interfaz gráfica desarrollada o por comandos de voz.

### **Bibliografía**

330ohms. (14 de Octubre de 2013). Obtenido de <https://blog.330ohms.com/2013/10/14/que-son-los-actuadores-electronicos/>

Aitana. (16 de Octubre de 2018). Obtenido de <https://blog.aitana.es/2018/10/16/visual-studio-code/>

AllDataSheet. (2022). Obtenido de SONGLE RELAY: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1131944/SONGLERELAY/SRD05VDCSLC.html>

AllDataSheet. (2022). Obtenido de STMicroelectronics: <https://pdf1.alldatasheet.com/datasheet-pdf/view/155897/STMICROELECTRONICS/2N3904.html>

AllDataSheet. (2022). Obtenido de Freescale: <https://pdf1.alldatasheet.com/datasheet-pdf/view/5039/MOTOROLA/MOC3021.html>

AllDataSheet. (2022). Obtenido de NXP: <https://pdf1.alldatasheet.com/datasheet-pdf/view/16751/PHILIPS/BT136.html>

Aprendiendo Arduino. (24 de Marzo de 2015). Obtenido de <https://aprendiendoarduino.wordpress.com/2015/03/24/entorno-de-programacion/>

Arduino. (2022). Obtenido de <https://arduino.cl/programacion/>

Brico Geek. (2018). Obtenido de <https://tienda.bricogeek.com/wifi/1033-nodemcu-v3-wifi-esp8266-ch340.html>

Chávez Jiménez, S. C., Beltrán Segura, D. A., Montiel Sandoval, J. R., & Ríos Malaver, N. A. (2021). Optimización de medición de agua mediante la implementación de sistemas IoT. Bogotá: Fundación Universitaria.

Estrada Corona, A. (2014). Protocolos TCP/IP de internet. Revista Digital Universitaria, 2-4.

Fernández, Y. (3 de Agosto de 2020). XATAKA. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Fonseca Yupa, J. D., & Soria Badillo, D. A. (2020). Diseño e implementación de control doméstico (con sistemas embebidos) para conectarse con aplicaciones adaptivas basados en IoT. Quito: Universidad Politécnica Salesiana.

Herrera Chávez, D. W. (2020). Diseño e implementación de un prototipo de seguridad diseño e implementación de un prototipo de seguridad de dispositivos móviles con Android. Quito : Escuela Politécnica Nacional.

Jiménez Montero, F. (7 de Agosto de 2020). Google Sites. Obtenido de <https://sites.google.com/site/ejdlcdomotica/Home/tema-1-introduccion-a-la-domotica#TOC-DOM-TICA>

Marmolejo, R. (2017). HetPro. Obtenido de <https://hetpro-store.com/TUTORIALES/microcontrolador/>

Narváez Narváez , K. S., & Contreras Pérez, V. A. (2020). Diseño de un prototipo de red de sensores IoT utilizando tecnología Lora WAN para el monitoreo de parámetro ambientales en interiores y exteriores. Guayaquil: Universidad Politécnica Salesiana.

Palma Castro, C. D., & Rodríguez Osorio, S. (2011). Tarjetas de Desarrollo: Herramientas para el diseño. Bogotá: Escuela Tecnológica Instituto Técnico Central.

Peterson, Z. (5 de Octubre de 2020). Altium. Obtenido de <https://resources.altium.com/es/p/what-is-a-pcb>

Red Hat. (8 de Enero de 2019). Obtenido de <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>

Söderby, K. (8 de Abril de 2022). Arduino. Obtenido de <https://docs.arduino.cc/cloud/iot-cloud/tutorials/iot-cloud-getting-started>

Stoer, M. (30 de Diciembre de 2021). Github. Obtenido de <https://github.com/nodemcu/nodemcu-firmware>

Vazquez, A. (2 de Octubre de 2015). Scribd. Obtenido de <https://es.scribd.com/doc/29597239/DEFINICION-DE-PROTOTIPO>