

**PROYECTO FINAL**  
**SISTEMAS EN TIEMPO REAL**

**Presentado por:**

Tomás Jiménez Álvarez, *1002655777*  
Anamaría López Bonilla, *1002547269*  
Laura Cossette Ríos Sarmiento, *1002656401*

**Profesores:**

Gustavo Adolfo Osorio Londoño  
Victor Alejandro Patiño Martínez

2023-1S



**Universidad Nacional de Colombia**  
**Facultad de Ingeniería y Arquitectura**  
**Departamento de Ingeniería Eléctrica, Electrónica y Computación**



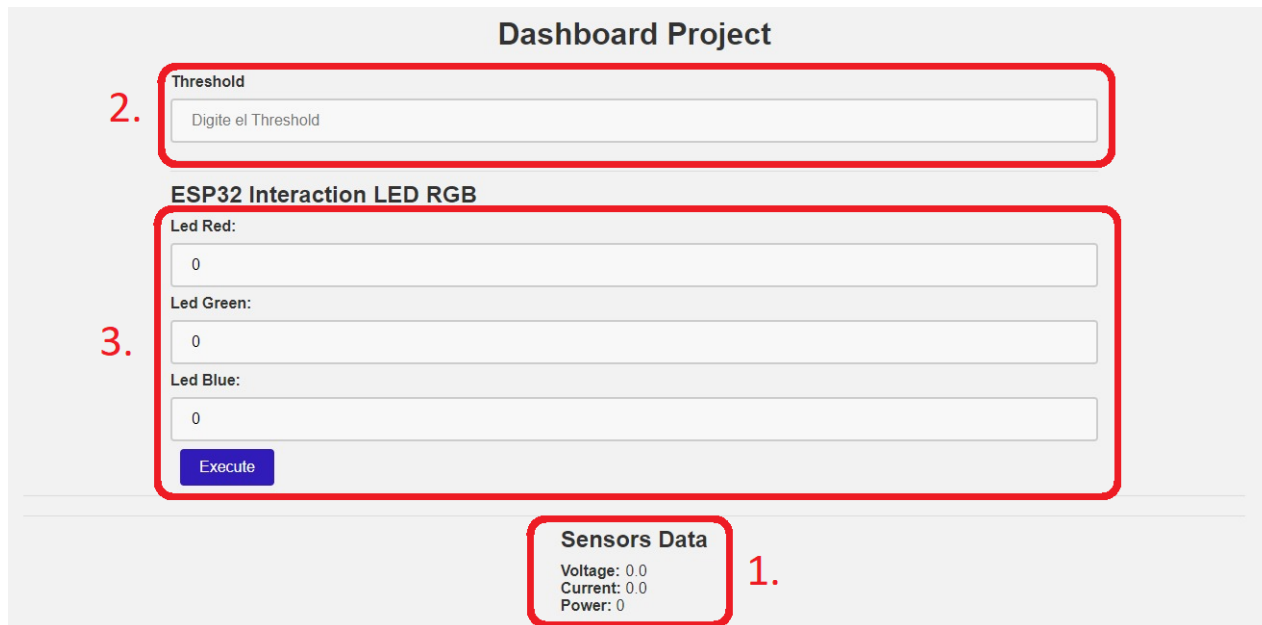
## **2020**

### **CONTENIDO**

<b>1. Explicación en lenguaje natural</b>	<b>3</b>
<b>2. Diagrama de bloques</b>	<b>4</b>
<b>3. Diagrama de flujo</b>	<b>4</b>
Main, Servidor y WiFi	4
Funcionamiento del ADC	5
Funcionamiento del LEDC	5
<b>4. Sensor de corriente</b>	<b>6</b>
<b>5. Sensor de tensión</b>	<b>6</b>
<b>6. Cálculo de potencia y obtención del límite (threshold)</b>	<b>6</b>

# 1. Explicación en lenguaje natural

Inicialmente, el usuario se debe conectar al WiFi del microcontrolador y esto lo redirecciona a una página web. La interfaz de la página está dividida en tres secciones, como se muestra en la siguiente figura.



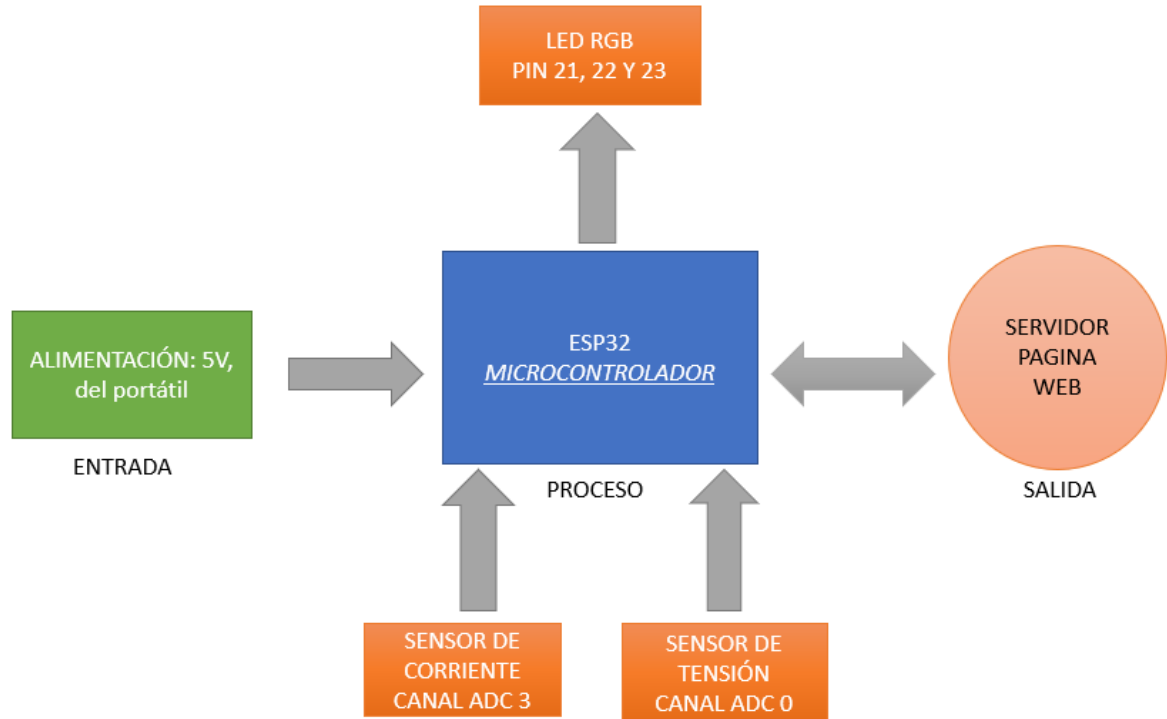
**Sección 1:** Acá se pueden visualizar los valores medidos por los sensores. Primero, está la tensión del circuito, luego la corriente y, con estos dos valores, se calcula la potencia.

**Sección 2:** Esta sección únicamente cuenta con un cuadro de texto en el cual se le estipula un valor límite a la potencia. Si esta llega a sobrepasar el límite, se activará un Led RGB.

**Sección 3:** Para que el Led RGB pueda encenderse, se le debe asignar un espectro de color, controlando la intensidad del color rojo, verde y azul. La presente sección existe para poder definir estos valores. Una vez seleccionados estos parámetros, se deben mandar al microcontrolador por medio del botón “Execute”.

## 2. Diagrama de bloques

En la siguiente imagen se muestra el diagrama de bloques correspondiente al proyecto final.



## 3. Diagrama de flujo

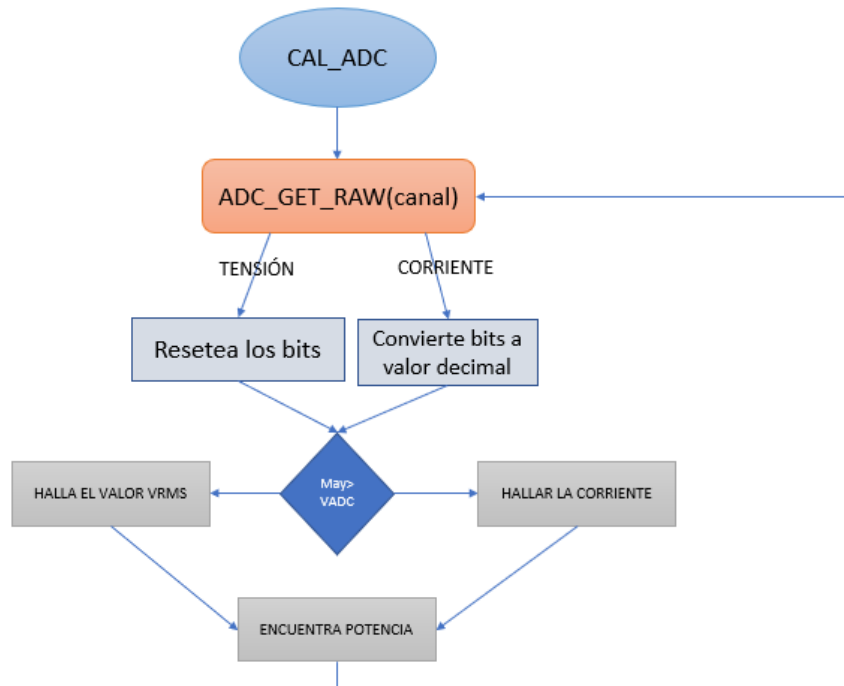
A continuación, se mostrarán los diagramas de flujo que describen el funcionamiento del código.

### Main, Servidor y WiFi

Estos diagramas se encuentran agregados en los anexos, pero también los puede visualizar haciendo [click aquí](#).

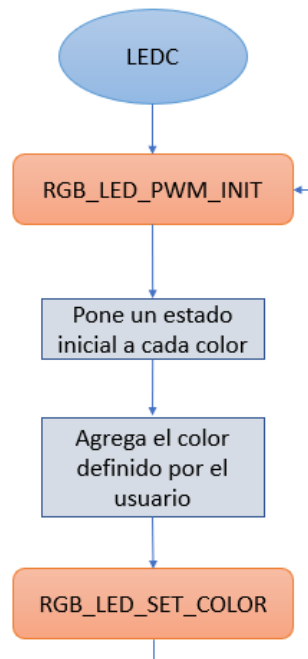
## Funcionamiento del ADC

Esta función es la encargada de tomar la información que se recibe de los sensores e interpretarla.



## Funcionamiento del LEDC

Esta función es la que se encarga de poner el color del LED RGB según lo que el usuario ingrese.



## 4. Sensor de corriente

El sensor de corriente es un módulo ZMCT 103C lo cual tiene un amplificador operacional y una bobina que es la que genera la corriente. El módulo tiene 4 pines, dos pines a tierra, uno de señal y otro que es por donde se alimenta el sensor.

La configuración con el microcontrolador es por medio de un canal ADC, para este sensor se utiliza el canal 3 que es por donde recibe la señal que está le genera. Al obtener esta señal, se hace la conversión analógica - digital por lo que con este valor se puede aplicar la ley de OHM para obtener la corriente. Sin embargo, este sensor cuenta con una bobina que tiene una relación 1:1000 por lo que dentro de la ley de ohm se debe de tener en cuenta.

La resistencia implementada es valor 1000 (resistencia interna) por la ganancia que se puede manipular por medio del potenciómetro. Para que nuestro sistema funcione se requiere una ganancia de 22 lo cual la resistencia puesta es 2220. Ya que como se está implementando con un bombillo que requiere de 100W para su funcionamiento, la corriente que se requiere al menos es de 0.83 ya que el bombillo se alimenta con aproximadamente 120V.

En la parte del código en donde está programado previamente lo nombrado anteriormente está en la función CAL\_ADC, en dónde se puede observar su diagrama de flujo.

## 5. Sensor de tensión

El sensor de voltaje ZMP T101B se encarga de monitorear el voltaje de una red de corriente alterna de hasta 250 voltios. Posee un potenciómetro de ajuste de múltiples vueltas que se puede utilizar para ajustar la salida analógica.

El módulo está integrado por un transformador, por el que se conecta el voltaje alterno con una bornera por el lado primario del transformador y, en el lado secundario del transformador, se encuentra un divisor de tensión y un circuito con amplificador operacional (OPAMP LM358) para adicionar un desplazamiento (offset) a la salida análoga. También cuenta con cuatro pines de los cuales tres se conectan al microcontrolador. Uno, se conecta a la alimentación de 5V del microcontrolador, otro a la tierra del mismo, y la salida analógica que se conectará al pin que ya ha sido designado.

El valor de la tensión se halla en dos pasos. Primero, se hace la lectura de lo que entra por el adc. Luego, se resta un valor de referencia para que la medición empiece en 0V. Así pues, se procede a encontrar el valor mayor de la tensión (hallando así el valor pico). Finalmente, se toma este valor pico y se le encuentra el valor RMS dividiendo el valor pico por raíz cuadrada de dos multiplicada por 2,3969. Todo este procedimiento se hace en la función CAL\_ADC.



## 6. Cálculo de potencia y obtención del límite (threshold)

La potencia se da por el resultado dado por el sensor de tensión y por el tensión corriente. Puesto que ya se sabe que la potencia es el producto entre estos dos parámetros.

El threshold es un valor dado por el usuario en donde indica el límite de potencia para encender un LED. En la programación, por medio de la función `parseFloat` es como se obtiene el valor en String y luego se convierte a numérico con la función `parseFloat`.

Ya luego este valor se compara con el valor dado por los sensores y si es mayor entonces entra a la función LEDC explicada más adelante.

## 7. Código completo

Finalmente, se agrega el GitHub donde se encuentra todo el código del proyecto explicado anteriormente.

Link del GitHub:

[https://github.com/TJA1303/ESP32\\_API\\_WEB\\_SENSORS\\_ZMPT101B\\_ZMCT103C\\_LEDs\\_THRESHOLD/blob/main/main/APIs/LED\\_RGB/rgb\\_led.c](https://github.com/TJA1303/ESP32_API_WEB_SENSORS_ZMPT101B_ZMCT103C_LEDs_THRESHOLD/blob/main/main/APIs/LED_RGB/rgb_led.c)