

# INTERNET DE LAS COSAS APLICADO AL MONITOREO Y MEDICIÓN DE LA CALIDAD DEL AIRE

CRISTHIAN ANDRÉS PUELLO ROJAS  
JULIÁN ANDRÉS JEREZ NIETO



UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA  
2018

INTERNET DE LAS COSAS APLICADO AL MONITOREO  
Y MEDICIÓN DE LA CALIDAD DEL AIRE

CRISTHIAN ANDRÉS PUELLO ROJAS  
JULIÁN ANDRÉS JEREZ NIETO

*Trabajo de grado para optar al título de  
Ingeniero Electrónico*

Director  
JAIME GUILLERMO BARRERO PÉREZ  
Magister en potencia eléctrica

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA  
2018

*Dedicado a*

*A mi madre Margarita Rojas Rojas por su apoyo y voz de aliento en estos años de carrera, y lo importante que es siempre estar superándose, su motivación para culminar una gran meta.*

*A mi tía Briceida Rojas que siempre tuvo ese interés en mi bienestar y apoyo por el camino a completar una gran meta en mi vida, por sus lecciones y conocimientos en la vida profesional.*

*A mi tío y padrino Gustavo Rojas Rojas quien siempre estuvo ahí para mi cuando lo necesitaba, por sus lecciones en la vida profesional y su interés en mi bienestar en este trayecto.*

*A mis Demas familiares, A mis hermanos, Fabián, Fredy Jr., Margarita quienes siempre estuvieron pendiente de mi progreso en esta carrera a pesar de la distancia, A mi abuela María Rojas quien siempre estaba pendiente de mí.*

*Por último, pero no menos importante, a mi compañero de proyecto por su infinita paciencia todos estos años que me permiten entender lo importancia de estar en momentos cruciales y por todo el apoyo en lo otorgado en lo personal, profesional y reconocer la gran persona que es, trabajo en equipo.*

**Cristhian Andrés Puello Rojas**

*Dedicado a*

*A mi madre Derly Eumelina por brindarme todo el amor, apoyo, confianza y paciencia que me ha tenido a través de todos estos años.*

*A mi padre Pedro Pablo por enseñarme la importancia de ayudar a la sociedad, crecer en conjunto con mis amistades, formar mi carácter y hacerme ver la importancia de la familia.*

*A mi hermana Sharay por aguantar mi personalidad de viejo cascarrabias, brindarme una mano y acompañarme en mis momentos difíciles.*

*A mi Abuela Eumelina Barajas que siempre con una sonrisa me hacía olvidar que estaba lejos de casa.*

*A toda mi familia por tener Fe en mí, por el apoyo y compañía que en muchas ocasiones necesitaba.*

*A don William y doña Aracelly por hacerme sentir como un miembro más de su familia, por todo el apoyo en estos 5 años, estaré eternamente agradecido por todo lo que han hecho por mí.*

*A todas mis amistades que me acompañaron en toda la carrera, se vivieron momentos de alegría y de tristeza, y seguimos para adelante!*

*A mi compañero y amigo Cristhian Puello por acompañarme en todo el desarrollo de la carrera y por ser paciente conmigo.*

**Julián Andrés Jerez Nieto**

## *Agradecimientos*

Al profesor **Jaime Barrero**, por su asesoría, acompañamiento y colaboración; por su tiempo y enseñanza durante estos últimos años y por ser un excelente director.

A los Ingenieros **Sebastián** y **Diego** de la CDMB por su incondicional ayuda en la prestación de servicio y acompañamiento en este proyecto.

Al ingeniero **Hernando Márquez Suarez** por su total apoyo en el desarrollo del proyecto.

A nuestros **familiares**, por su apoyo incondicional en esos momentos difíciles, paciencia y enseñanzas.

A **compañeros y amigos**, quienes también estuvieron en el camino, compartiendo a lo largo de esta carrera grandes momentos y enseñanzas, buenos y malos momentos, que permitieron forjar grandes amistades.

A la **Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones (E3T)** y a la **Universidad Industrial de Santander (UIS)**

¡Gracias Infinitas!

# Índice general

<b>INTRODUCCIÓN</b>	<b>18</b>
<b>1 ALCANCES</b>	<b>20</b>
<b>2 OBJETIVOS</b>	<b>21</b>
<b>3 DESCRIPCIÓN DE HARDWARE</b>	<b>22</b>
3.1 ESP32 . . . . .	22
3.1.1 Datos técnicos y características del ESP32 . . . . .	23
3.1.2 Descripción de los pines ESP32. . . . .	25
3.2 SENSORES Y ADECUACIÓN . . . . .	26
3.2.1 Los sensores . . . . .	26
3.3 DISEÑO DEL ESQUEMA Y CIRCUITO IMPRESO . . . . .	30
3.3.1 Ajuste de los sensores . . . . .	35
3.3.1.1 Sensor de monóxido de carbono MQ7 . . . . .	35
3.3.1.2 Sensor de ozono MQ131 . . . . .	37
<b>4 DESCRIPCIÓN DE SOFTWARE</b>	<b>39</b>
4.1 SERVIDOR LOCAL: XAMPP . . . . .	39
4.1.1 Montaje de servidor y base de datos con XAMPP . . . . .	40
4.2 COMUNICACIÓN ENTRE EL SERVIDOR LOCAL Y EL MÓDULO WIFI ESP32 .	42
4.2.1 Solicitud GET HTTP . . . . .	42
4.2.2 Ingreso de datos al servidor . . . . .	43
4.2.3 Programación módulo ESP32 . . . . .	44
4.3 VISUALIZACIÓN <i>WEB</i> . . . . .	50
<b>5 PRUEBAS DE LABORATORIO Y CAMPO</b>	<b>53</b>
5.1 CARACTERIZACIÓN PUERTOS ADC . . . . .	53
5.2 PRUEBA PARA AJUSTE DE LOS SENSORES . . . . .	55

<b>6 CONCLUSIONES</b>	<b>64</b>
<b>7 RECOMENDACIONES PARA TRABAJOS FUTUROS</b>	<b>65</b>
<b>BIBLIOGRAFÍA</b>	<b>66</b>

## Lista de Apéndices

(Ver apéndices adjuntos en el CD y pueden visualizarlos en la Base de Datos de la Biblioteca UIS)

Apéndice A. Utilización de la herramienta Webplotdigitizer

Apéndice B. Utilización de la herramienta Worrpress

Apéndice C. Hoja de datos del sensor de ozono MQ131

Apéndice D. Hoja de datos del sensor de monóxido de carbono MQ7

Apéndice E. Informe de la CDMB sobre calidad del aire, enero a marzo del 2010

Apéndice F. Hoja de datos del sensor de ozono 03 42M

Apéndice G. Hoja de datos del Regulador LM317

Apéndice H. Código en arduino del proyecto

Apéndice I. Código en lenguaje php conectar del proyecto

Apéndice J. Código en lenguaje php envío de datos del proyecto

Apéndice K. Código en lenguaje php de entrada de datos del proyecto

Apéndice L. Código en lenguaje HTML con JavaScript de grafica en tiempo real del proyecto

Apéndice M. Código HTML con JavaScript de grafica indicadora por colores

Apéndice N. Archivo circuito impreso del proyecto

Apéndice Ñ. Datos de la caracterización de los puertos ADC de la ESP32

Apéndice O. Datos del sensor de ozono con su respectivo ajuste

Apéndice P. Datos de la prueba realizada de concentración de ozono en la CDMB

Apéndice Q. Datos de la prueba realizada con generador de ozono casero

Apéndice R. Datos de del sensor de monóxido de carbono a 100 ppm

Apéndice S. Archivo para la pantalla nextion

Apéndice T. Archivo del software nextion para programar la pantalla

Apéndice U. Datos de los sensores de monóxido de carbono con su respectivo ajuste

Apéndice V. Datos del sensor de ozono 03 24M De la CDMB durante 24 Horas

Apéndice W. Esquema general de componentes del prototipo

Apéndice X. Pagina web del proyecto

## Índice de figuras

Figura 1	Diagrama de bloques del ESP32 . . . . .	24
Figura 2	Descripción de pines (pinout) de ESP32 DOIT DEV KIT V1 . .	24
Figura 3	Circuito para la lectura de Rs para el sensor de ozono MQ131 .	27
Figura 4	Circuito para la lectura de Rs para el sensor de monóxido de carbono MQ7 . . . . .	27
Figura 5	Sensor de monóxido de carbono MQ7 . . . . .	27
Figura 6	Ciclos sensor de monóxido de carbono MQ7 . . . . .	28
Figura 7	Sensor de Ozono MQ131 . . . . .	29
Figura 8	Conexión eléctrica interna sensores serie MQ . . . . .	29
Figura 9	Diagrama de potencia del circuito . . . . .	30
Figura 10	Esquema circuito de regulación de 1.4 V . . . . .	31
Figura 11	Esquema circuito de conmutación de la tensión para sensor MQ7	31
Figura 12	Pantalla Nextion . . . . .	33
Figura 13	Distribución de pines ESP32 para el prototipo . . . . .	33
Figura 14	Diseño circuito impreso . . . . .	34
Figura 15	prototipo de medición de calidad del aire . . . . .	35
Figura 16	Curvas de gases para el MQ7 . . . . .	36
Figura 17	Curva CO, ppm vs (Rs/Ro) en Excel . . . . .	37
Figura 18	Curvas de gases para el MQ131 . . . . .	38
Figura 19	Curvas de tensión de salida analógica vs concentración O3 para sensor MQ131 . . . . .	38
Figura 19	Diagrama de comunicación IoT . . . . .	39
Figura 19	Interfaz de XAMMP . . . . .	40
Figura 20	Página principal del servidor local . . . . .	41
Figura 21	Acceso a base de datos . . . . .	41
Figura 22	Base de datos . . . . .	42
Figura 23	Esquema solicitud cliente-servidor . . . . .	43

Figura 24	Esquema solicitud tipo GET . . . . .	43
Figura 25	formulario de datos . . . . .	44
Figura 26	Diagrama de flujo proceso ESP32 . . . . .	45
Figura 27	Código lectura analógica de los sensores . . . . .	46
Figura 28	Código de muestreo de la señal analógica . . . . .	46
Figura 29	Ejemplo código de conexión y desconexión a una red Wifi . . . . .	46
Figura 30	Plantilla HTTPclient para solicitud tipo GET . . . . .	47
Figura 31	Plantilla almacenamiento memoria microSD . . . . .	48
Figura 32	Diseño de la visualización en NEXTION Editor . . . . .	49
Figura 33	Verificación de los datos cargados a la pantalla Nextion . . . . .	49
Figura 34	Plantilla de visualización NEXTION . . . . .	50
Figura 35	Página principal web . . . . .	51
Figura 36	Enlace Gráficas . . . . .	51
Figura 37	Enlace PPMs . . . . .	52
Figura 37	lectura ADC vs tensión suministrada . . . . .	53
Figura 38	Ensayo calibración del sensor MQ7 . . . . .	56
Figura 39	gráficas de respuesta de sensores MQ7 . . . . .	57
Figura 40	gráficas de respuesta de sensores MQ7 con corrección polinómica	59
Figura 41	Medidor de monóxido de carbono AS8700A . . . . .	59
Figura 42	Medidor de monóxido de carbono Fluke CO-220 . . . . .	60
Figura 43	Medidor de ozono 03 42M . . . . .	61
Figura 44	Prueba del sensor MQ131 con generador de ozono . . . . .	62
Figura 45	Prueba del sensor MQ131 con generador de ozono . . . . .	63
Figura 46	Respuesta del sensor MQ131 . . . . .	63

## Índice de tablas

Tabla 1	Opciones de tarjeta de desarrollo con conexión wifi. . . . .	23
Tabla 2	Características del ESP32 [3] . . . . .	23
Tabla 3	Tabla especificaciones sensor MQ7 . . . . .	28
Tabla 4	Distribución de pines módulo microSD . . . . .	32
Tabla 5	Distribución de pines pantalla nextion . . . . .	32
Tabla 6	Especificaciones del prototipo . . . . .	35
Tabla 6	Porcentaje de error puerto ADC . . . . .	54
Tabla 7	Porcentaje de error con ajuste lineal . . . . .	54
Tabla 8	Porcentaje de error con ajuste de orden 4 . . . . .	55
Tabla 9	Porcentaje de error absoluto respecto al sensor fluke CO-220 . .	57
Tabla 10	Porcentaje de error absoluto de los sensores MQ7 corregido res- pecto al sensor fluke CO-220 . . . . .	58
Tabla 11	Ficha técnica Medidor de monóxido de carbono AS8700A . . . .	60
Tabla 12	Ficha técnica Medidor de monóxido de carbono fluke CO-220 . .	60
Tabla 13	Ficha técnica Medidor de ozono O3 42M, propiedad de la CDMB	61

# RESUMEN

**Título:** Internet de las cosas aplicado a la medición y monitoreo de calidad del aire.<sup>1</sup>

**Autores:**

Julián Andrés Jerez Nieto<sup>2</sup>

Cristhian Andrés Puello Rojas<sup>2</sup>

**Palabras Clave:** Calidad del aire, internet de las cosas.

## DESCRIPCIÓN

El interés en conocer la calidad del aire nace a partir de una conciencia ambientalista para con nuestras generaciones y por ende nuestro medio ambiente, es por eso que se hace necesario conocer las concentraciones de emisiones provenientes de vehículos de combustión y de las industrias, las cuales son las principales emisoras de monóxido de carbono (CO), hidrocarburos no quemados (HC), óxidos de nitrógeno (NOx), óxidos de azufre (SOx) y dióxido de carbono (CO<sub>2</sub>) y que en grandes cantidades son perjudiciales para el ser humano. Se presenta en este proyecto el desarrollo de un equipo electrónico que mide dos variables relacionadas con la calidad del aire y las envía inalámbricamente para poner esta información a disposición de la comunidad por medio de *internet of things* ó IoT. Para lograr esto, se diseñó una página *web* montada en un servidor local con el fin de ofrecer el servicio de presentar la información de la calidad del aire de acuerdo a las necesidades de la aplicación y de manera gratuita desde un equipo de cómputo o celular en tiempo real y así conocer si es o no perjudicial el nivel medido. El dispositivo es de bajo consumo, expandible, es decir puede monitorear más variables, presenta la información de manera local en una pantalla táctil-gráfica y puede almacenar datos en una microSD de hasta 32 GB.

---

<sup>1</sup>Trabajo de Grado

<sup>2</sup>Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, Magister.

# ABSTRACT

**Title:** Internet of things applied to the measurement and monitoring of air quality<sup>1</sup>

**Authors:**

Julián Andrés Jerez Nieto<sup>2</sup>

Cristhian Andrés Puello Rojas<sup>2</sup>

**Key Words:**

Air Quality, Internet of things.

## DESCRIPTION

The interest in knowing the quality of the air is born from an environmental awareness to our generations and therefore our environment, is therefore that it is necessary to know the concentrations of emissions from combustion vehicles and the industries, which are the main stations of carbon monoxide (CO), unburned hydrocarbons (HC), nitrogen oxides (NOx), oxides of Sulphur (SOx), and carbon dioxide (CO<sub>2</sub>) and in large amounts are harmful to human beings. Is presented in this project the development of an electronic equipment which measures two variables related to the quality of the air and the send wirelessly to make this information available to the community through internet of things (IoT). To achieve this, design a website mounted on a local server to offer the service to submit information as the quality of air according to the needs of the application and for free from a team of computer or cell phone in real time and thus know whether or not is damaging the measured level. The device is low power consumption, expandable, i.e. you can monitor more variables, presents the information locally in a screen touch-graphic and can store data in a microSD up to 32 GB

---

<sup>1</sup>Bachelor Thesis

<sup>2</sup>Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, PhD.

## *INTRODUCCIÓN*

El internet de las cosas, *internet of things* o IoT por sus siglas en inglés, un concepto que nació en el Instituto de Tecnología de Massachusetts (MIT). La idea que intenta representar queda bien ilustrada por su nombre: cosas cotidianas que se conectan al internet; pero en realidad este concepto abarca muchas temáticas donde son inmensas las aplicaciones que se podrían desarrollar.

Para comprender cuál es la visión del internet de las cosas se debe también saber que sus fundamentos y estudios no son en lo absoluto nuevos, han pasado 30 años desde que se viene trabajando con la idea de hacer una interacción digital entre objetos de uso cotidiano. Ideas como el hogar inteligente, también conocido como la casa del mañana. Se trata de una revolución en las relaciones entre los objetos y las personas, incluso entre los objetos directamente, que se conectarán entre ellos, red local e internet, donde nos ofrece la posibilidad de procesar datos en tiempo real y darles un uso objetivo, o, dicho de otro modo, se acerca la digitalización del mundo físico.

El internet de las cosas amplía la utilidad de objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores; dando la posibilidad de que los equipos puedan comunicarse globalmente mediante el uso de la red. Ahora en la actualidad, donde la producción de módulos programables de comunicación es de bajo consumo y de bajo costo, se da la posibilidad de acceder al monitoreo de datos, interpretación y registro virtual de cualquier proceso ya sea industrial o doméstico en tiempo real desde cualquier ubicación.

Con el uso de estos microcontroladores que permiten la transmisión de datos obtenidos de sensores para que sean de fácil acceso a la comunidad, este proyecto se enfoca en variables asociadas a la calidad del aire [8]. La contaminación del aire urbano es un

serio problema en grandes ciudades del planeta. El intenso e incesante tráfico, unido a fábricas que no controlan sus emisiones, convierte el aire de ciudades de todo el mundo en auténticas nubes de *smog*<sup>1</sup>. Los niveles de concentración de partículas contaminantes sobrepasan en muchos casos el límite de seguridad para la salud humana marcado por la OMS [8]. Lo que la convierte en un gran problema de salud medioambiental. En lugares con mucha contaminación en el aire afecta la salud y puede acarrear grandes problemas.

Por lo tanto, uno de los fines de este proyecto es la medición y transmisión de la concentración de algunos compuestos químicos, los cuales son el monóxido de carbono y el ozono, que en grandes concentraciones o al estar expuestos un tiempo prolongado pueden ser perjudiciales para la salud. Con el fin de informar de manera oportuna (tiempo real) a la comunidad de dichos datos, se desarrolló un equipo electrónico de bajo consumo y portátil, ubicado en cualquier sitio específico, que los adquiere y transmite de forma inalámbrica para que desde una página *web* sea fácilmente consultada por la comunidad desde un equipo de cómputo o celular. El equipo cuenta además con una pantalla gráfica y además táctil para presentar la información de manera local y una Micro SD para guardar la información por un periodo de hasta aproximadamente 2 años.

Para el ajuste del sensor de monóxido de carbono, se utilizó un medidor comercial que sirvió como patrón y para el ajuste del sensor de Ozono se realizó con la cooperación de la Corporación Autónoma Regional Para La Defensa De La Meseta De Bucaramanga CDMB.

En el primer capítulo se presenta la descripción del *hardware*, en el segundo capítulo se abarcarán los temas de la programación para la plataforma web y del prototipo y en el capítulo final se muestran las pruebas de laboratorio y de campo. Finalmente, conclusiones y recomendaciones.

---

<sup>1</sup>Nube baja formada de dióxido de carbono, hollines, humos y polvo en suspensión que se forma sobre las grandes ciudades o núcleos industriales.

# *Capítulo 1*

## *ALCANCES*

- ❖ El alcance de la visualización de los datos se da en un servidor local, dado que para el desarrollo no se puede realizar con un dominio y un *hosting* gratuitos, ya que no ofrecen uno de los servicios importantes para desarrollar una plataforma IoT, como lo es la conexión de un dispositivo IoT al servidor.
- ❖ Las variables escogidas para la medición de calidad del aire a medir son la concentración de monóxido de carbono y de ozono, estas siendo variables principales de calidad del aire, pero el sistema se puede expandir fácilmente a más variables.
- ❖ Este proyecto está enfocado en la adquisición, procesamiento, transmisión y visualización de 2 variables.

## *Capítulo 2*

# **OBJETIVOS**

### **OBJETIVO GENERAL**

Implementar una plataforma de información que permita en tiempo real monitorear, transmitir y almacenar la información obtenida de algunas variables de calidad de aire por medio de un módulo Wifi.

### **OBJETIVOS ESPECÍFICOS**

- ❖ Monitorear información de la calidad del aire, proveniente de dos sensores de monóxido de carbono y uno ozono.
- ❖ Seleccionar una entre tantas opciones una tarjeta programadora de bajo costo, tamaño, consumo y procesamiento de información que interprete la información de dos sensores de monóxido de carbono y un sensor de ozono, con el fin de procesarlo, transmitirlo y almacenarlo usando comunicación wifi.
- ❖ Programar una plataforma web que se conecte a un equipo de cómputo o celular para el monitoreo en tiempo real de las variables de calidad del aire.

## *Capítulo 3*

# *DESCRIPCIÓN DE HARDWARE*

### **3.1. ESP32**

A nivel mundial existen numerosas alternativas para el desarrollo de aplicaciones que hacen uso del internet de las cosas y el interés de la comunidad está enfocada en aquellos equipos que sean de bajo costo, consumo y altas prestaciones de computo. En el mercado se encuentran sistemas de desarrollo como la Raspberry PI ZERO, Raspberry PI 3, Omega 2, ESP8266, ESP32, entre otros módulos; para el desarrollo de este proyecto, se optó por el ESP32 debido a su bajo precio, bajo consumo y gran rendimiento, ver tabla 1, tarjetas de desarrollo en el mercado actual.

La tarjeta de desarrollo seleccionada para este proyecto consta de un módulo wifi ESP32 [3], integrado con una antena y amplificador de potencia, filtros de bajo ruido, tecnología de bajo consumo, bajo costo en el mercado, pequeño volumen y con conector USB para la alimentación, es una tecnología reciente donde su procesador contiene dos núcleos; posee capacidad de transmisión Wifi y bluetooth. A su vez la ESP32 es programable en diferentes plataformas como lo son MicroPython, ESP-IDF<sup>1</sup>, JavaScript entre otros. En este proyecto, para programar el ESP32, se utilizó la plataforma Arduino<sup>2</sup>.

---

<sup>1</sup>Espressif IoT Development Framework.

<sup>2</sup>plataforma de hardware y software de código abierto.

Tarjeta de desarrollo	Almacenamiento	Consumo	Memoria RAM	precio
rasberry pi 3 modelo B+	micro sd	200mA a 800mA	1Gb	158000 CO
Esp32	Hasta 16 MB	50mA a 300mA	520 kB	36000 CO
Omega 2	16 MB	—	64MB	43000 CO

Tabla 1: Opciones de tarjeta de desarrollo con conexión wifi.

**3.1.1 Datos técnicos y características del ESP32** El ESP32 es un potente microcontrolador que posee grandes características pensadas para su implementación en proyectos IoT<sup>1</sup>, ver figura 1.

Característica	ESP32
Procesador	32 bits a 240 MHz de Tensilica doble núcleo
Memoria RAM	520 kB
Memoria Flash	Hasta 16 MB
Memoria ROM	448 kB
Tensión de alimentación	2.2 a 3.6 V
Rango de temperatura	-40 a 125.° C
Consumo de corriente	50 mA promedio hasta un máximo de 300 mA
Consumo en modo sueño profundo	2.5 uA
Coprocador de bajo consumo	Si, inferior a 150uA
Wifi	Protocolo 802.11 b/g/n(hasta +20 dBm)WEP,WPA .
Bluetooth	v4.2 BR/EDR y BLE.
UART	3
I <sup>2</sup> C	2
SPI	4
GPIO	32 (input-Output).
PWM	16
ADC	18 (12bits)
ADC con preamplificador	sí (bajo ruido) hasta 60dB
DAC	2 (8 bits)

Tabla 2: Características del ESP32 [3]

Existen diferentes sistemas de desarrollo que utilizan el microcontrolador con wifi ESP32, A continuación, se presenta una descripción del módulo que se utilizó en este

---

<sup>1</sup>Internet of Things

proyecto, la DOIT DEV KIT V1, ver figura 2.

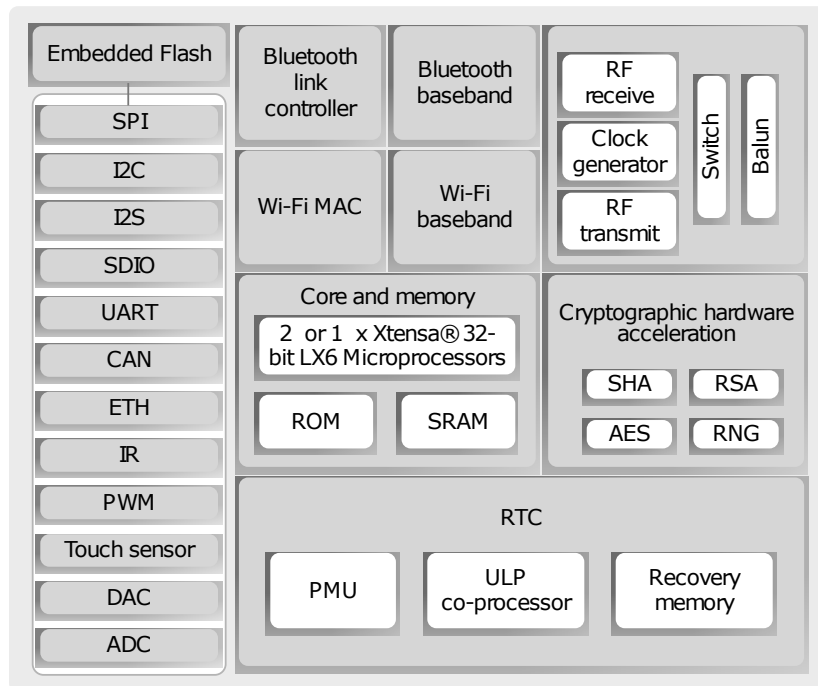


Figura 1: Diagrama de bloques del ESP32

Fuente: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)

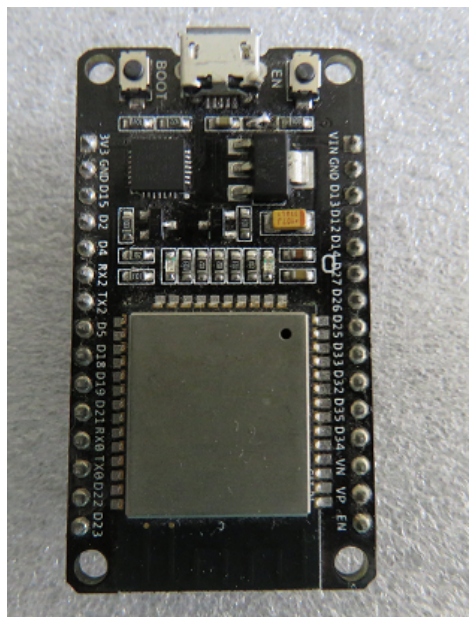


Figura 2: Descripción de pines (pinout) de ESP32 DOIT DEV KIT V1

**3.1.2 Descripción de los pines ESP32.** A continuación se describen los periféricos del ESP32 utilizados para este proyecto. El ESP32 se alimenta por medio de una tensión de 5 [V] presente en un conector micro USB que puede venir de un puerto USB de computador o de un adaptador de tensión de hasta 12 [V]. Este mismo conector permite programar el ESP32 desde un computador (plataforma Arduino). una descripción detallada se encuentra en su hoja de datos. [5]

- ❖ **Pines GPIO (General Purpose Input/Output):** como sus siglas lo dicen, son puertos de propósito general ya sea de entrada o de salida independientemente si son analógicas o digitales.
- ❖ **Pines ADC o TOUT (Analog to digital converter):** se utilizan para recibir señales analógicas de diferentes dispositivos o sensores, pero hay que tener en cuenta el rango de tensión de entrada que manejan para no perjudicar el puerto de entrada, que en este caso la tensión máxima que pueden recibir dichos pines es de 3.3 voltios, a su vez cuentan con una resolución de 12 bits. Se utilizaron 3 puertos donde se reciben las señales de los sensores seleccionados.
- ❖ **Pines SPI (Serial Peripheral Interface):** se utilizan para las comunicaciones entre circuitos integrados siguiendo un protocolo de comunicación síncrono, como por ejemplo con pantallas OLED, microSD, Etc. Entre los pines SPI tenemos SPICLK, SPIQ (MISO), SPID (MOSI) y SPICS, Estos pines que serán utilizados para la comunicación con el módulo microSD.
- ❖ **Pines UART (Universal asynchronous Receiver / Transmitter):** La ESP32 cuenta con 3 puertos (TX y RX) de comunicación UART, de los cuales están solo 2 disponibles para propósito general. En este proyecto se utilizó el serial 2 (Rx2 o GPIO16, Tx2 o GPIO17) para la programación de la pantalla nextion y el serial 1 (Rx0 o GPIO3, Tx0 o GPIO1) para la programación de la ESP32, se utilizan para la recepción (RX) y transmisión (TX) de datos de la nodemcu. Los datos enviados a través de los pines UART permiten configurar tanto la velocidad de transmisión de los datos como su formato.

## 3.2. SENSORES Y ADECUACIÓN

Existen en el mercado diferentes sensores que sirven para la medición tanto para la concentración de monóxido de carbono como para la de ozono, como lo son el ZE07-CO, el MICS-6814, el MQ9 o el MQ7 para el monóxido de carbono, y el MQ131 para el ozono, siendo estos dos últimos por disponibilidad y costo se seleccionaron, se encuentran en el mercado local, esto quiere decir que los sensores seleccionados son los MQ7 y MQ131 para el prototipo IoT.

**3.2.1 Los sensores** de la serie MQ son analógicos y de funcionamiento electroquímicos resistivos. Los sensores varían su resistencia ( $R_s$ ) cuando se exponen a un determinado gas, estos dispositivos poseen un calentador encargado de aumentar la temperatura interna para permitir que el sensor pueda reaccionar correctamente ante la presencia de los gases provocando un cambio en el valor de la resistencia. Ambos sensores se deben precalentar con una tensión de 5 voltios durante 48 horas, pasado este tiempo, el sensor se puede utilizar para determinar la concentración del gas, para ello se debe leer su resistencia, en este proyecto se utilizó un divisor de tensión para realizar esta operación como se observa en la figura 3, como siguiente medida se procedió a ubicar filtros *antialiasing* en los pines donde se van a tomar las muestras analógicas de los sensores en este caso para las señales SMQ131, SMQ71 y SMQ72. La potencia consumida varía según el modelo, el sensor se comporta como una resistencia ( $R_s$ ) y necesita una resistencia de carga ( $R_L$ ) para cerrar el circuito y con este hacer un divisor de tensión de forma tal que en el peor de los casos las lecturas no superasen los 3,3 voltios en el microcontrolador que en este caso es la ESP32.

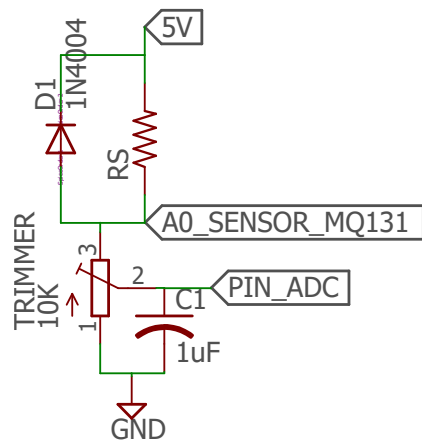


Figura 3: Circuito para la lectura de  $R_s$  para el sensor de ozono MQ131

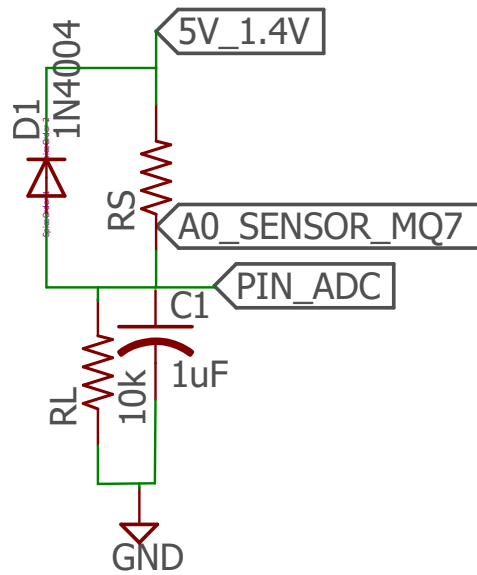


Figura 4: Circuito para la lectura de  $R_s$  para el sensor de monóxido de carbono MQ7



Figura 5: Sensor de monóxido de carbono MQ7

Fuente: <https://www.ardobot.com/sensor-mq7-monoxido-de-carbono.html/>

Para el sensor de monóxido de carbono MQ7 <sup>1</sup> 1 ver figura 5, el fabricante recomienda hacer un precalentamiento de 48 horas a 5 voltios (tabla 3) antes de colocar en funcionamiento el módulo, esto se hace con el fin de que las lecturas del sensor sean más precisas, luego de la etapa de precalentamiento el sensor estará listo para su uso. Para el funcionamiento correcto de este dispositivo se alternarán cambios de tensión en el pin de alimentación entre 5 voltios por 60 segundos que dicho de otra forma es una etapa de calentamiento, y 1.4 voltios por 90 segundos para hacer lecturas del sensor como se puede evidenciar en la figura 6, como siguiente medida se proceden a utilizar filtros *antialiasing* <sup>2</sup> en los pines donde se van a tomar las muestras analógicas de los sensores en este caso para las señales SMQ131, SMQ71, SMQ72.

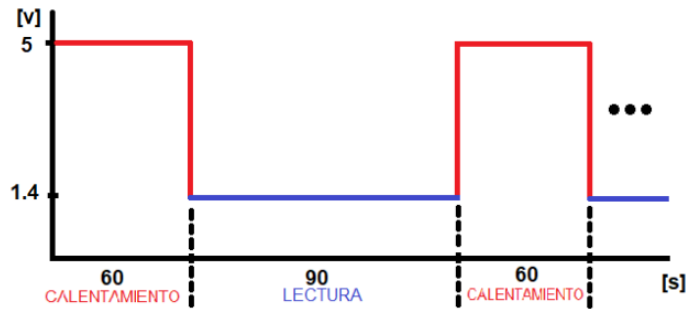


Figura 6: Ciclos sensor de monóxido de carbono MQ7

Símbolo	Parámetro	Condiciones técnicas	Observación
$V_C$	Tensión de alimentación	$5V \pm 0.1$	Ac o Dc
$V_H(H)$	Tensión de calentamiento (Alto)	$5V \pm 0.1$	Ac o Dc
$V_H(L)$	Tensión de calentamiento (Bajo)	$1.4V \pm 0.1$	Ac o Dc
$R_L$	Resistencia de carga	Ajustable	
$R_H$	Resistencia de calentamiento	$33 \Omega \pm 5\%$	Temperatura ambiente
$T_H(H)$	Tiempo de calentamiento (Alto)	$60 \pm 1$ segundos	
$T_H(L)$	Tiempo de calentamiento (Bajo)	$90 \pm 1$ segundos	
PH	Heating consumption	About 350mW	

Tabla 3: Tabla especificaciones sensor MQ7

<sup>1</sup>Sensor de monóxido de carbono.

<sup>2</sup>Es un filtro paso bajo analógico, limpia la señal antes de hacer una conversión analógica-digital



Figura 7: Sensor de Ozono MQ131

*Fuente:*

<https://didacticaselectronicas.com/index.php/sensores/sensor-de-ozono-mq131-gas-sen-ozono-2-detail/>

Para el sensor MQ131<sup>1</sup> que se puede observar en la figura 7, se realizó también una etapa de precalentamiento de 48 horas a una tensión de 5 voltios y después de este tiempo se tomaron lecturas del sensor, con la única diferencia que la tensión de alimentación se mantiene a 5 voltios.

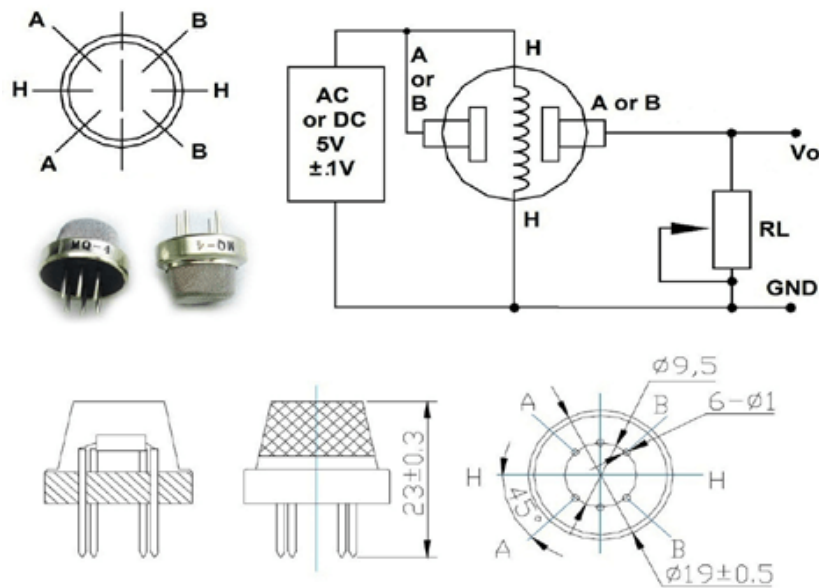


Figura 8: Conexión eléctrica interna sensores serie MQ

*Fuente:*

[https://cdn.shopify.com/s/files/1/1723/8439/products/TECH1576\\_b\\_large.png?v=1492143532](https://cdn.shopify.com/s/files/1/1723/8439/products/TECH1576_b_large.png?v=1492143532)

<sup>1</sup>Sensor de ozono.

### 3.3. DISEÑO DEL ESQUEMA Y CIRCUITO IMPRESO

EL diseño del circuito impreso está sujeto 4 parametros importantes que son la conexión de los sensores, módulo microSD, Pantalla Nextion y su respectiva alimentación como se muestra en el siguiente diagrama de potencia de la figura 9.

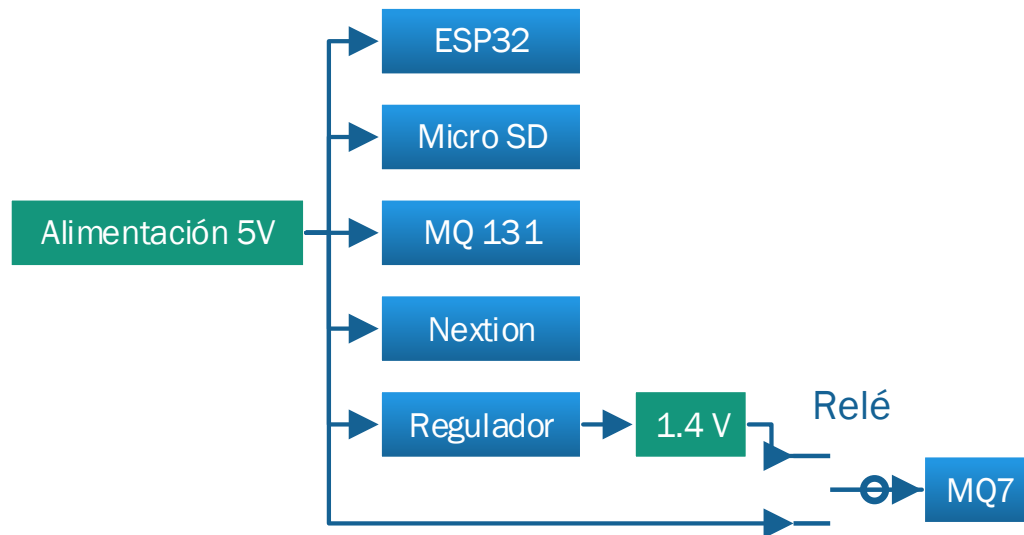


Figura 9: Diagrama de potencia del circuito

Para la alimentación del prototipo se decidió que la fuente del dispositivo fuera de 5 voltios, esto con el fin de utilizar la salida USB de un computador o un cargador con salida “micro USB”, típico de celular y con capacidad de corriente de mínimo 1 Amperio. Para cumplir con las especificaciones del sensor MQ7, se implementó una fuente de tensión estabilizada en 1,4 V por medio del LM 317 [10], ver figura 10.

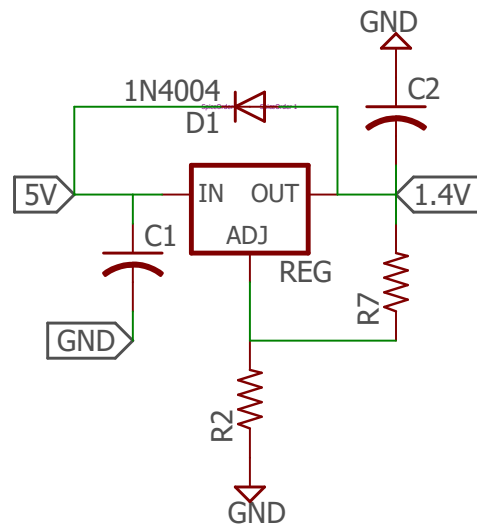


Figura 10: Esquema circuito de regulación de 1.4 V

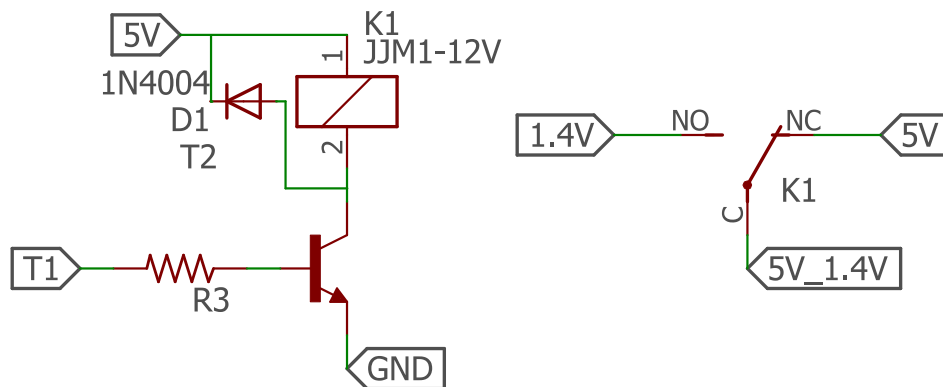


Figura 11: Esquema circuito de conmutación de la tensión para sensor MQ7

La hoja de datos del sensor MQ7 especifica que el módulo, previamente energizado (48 Horas), se debe operar de la siguiente manera: Una etapa de precalentamiento a 5 V durante 60 segundos y seguidamente una conexión a 1,4 V durante 90 segundos para realizar la lectura analógica. Como estos cambios de tensión son relativamente lentos, se decidió utilizar un relé de 5 voltios, que por medio de una salida digital del ESP32, permite seleccionar la tensión de alimentación del sensor MQ7 entre 5 y 1.4 voltios. Para favorecer el manejo de corriente y tensión del relevo, se utilizó un transistor bipolar BJT<sup>1</sup>, 2N2222. Ver figura 11. El diodo en antiparalelo con la bobina del relé protege el circuito contra sobretensiones.

<sup>1</sup>Bipolar Junction Transistor

- ❖ **Conexión módulo microSD:** la comunicación del módulo se hace a través de comunicación serial y los pines utilizados para dicha comunicación se muestran en la tabla 4.

Pines Tarjeta MicroSD	ESP32
SCK	D18 o GPIO18
CS	D5 o GPIO5
MISO	D19 o GPIO19
MOSI	D23 o GPIO23
VSS	GND
VDD	3V

Tabla 4: Distribución de pines módulo microSD

- ❖ **Conexión pantalla Nextion:** este dispositivo consta de una pantalla con sensor capacitivo, a su vez posee un microcontrolador que carga la visualización de la pantalla y realiza la comunicación con la tarjeta programable, cuenta con un módulo microSD que se utiliza para programar la visualización de los archivos generados por el software de dicha pantalla.

EL tipo de comunicación que maneja este dispositivo es UART, para ello se van a utilizar los puertos seriales TX2 y RX2 de la ESP32 como se muestra en la tabla 5, esto se hace con el fin de utilizar a la vez el puerto serial de comunicación del computador para efectos de comodidad a la hora de programar el dispositivo con todo el montaje y en caso de que haya que hacer alguna depuración en el código final del dispositivo.

Pines Nextion	ESP32
TX	RX2 o GPIO 16
RX	TX2 o GPIO 17
GND	GND
5V	5V

Tabla 5: Distribución de pines pantalla nextion

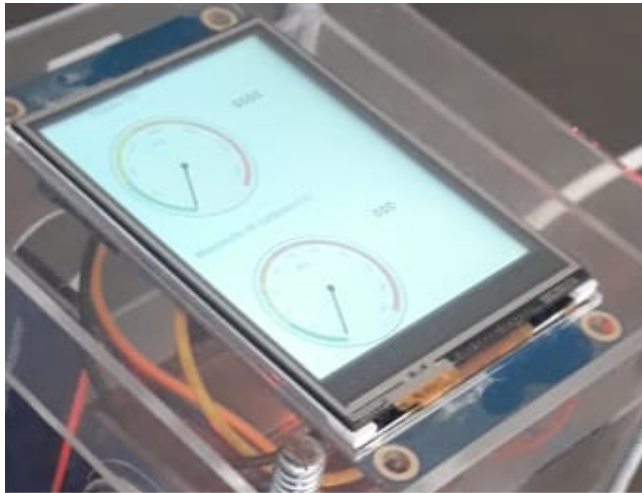


Figura 12: Pantalla Nextion

- ❖ **Selección de pines de lectura analógica y demás conexiones con la ESP32:** en la figura 13, se muestran todos los pines utilizados en el ESP32.

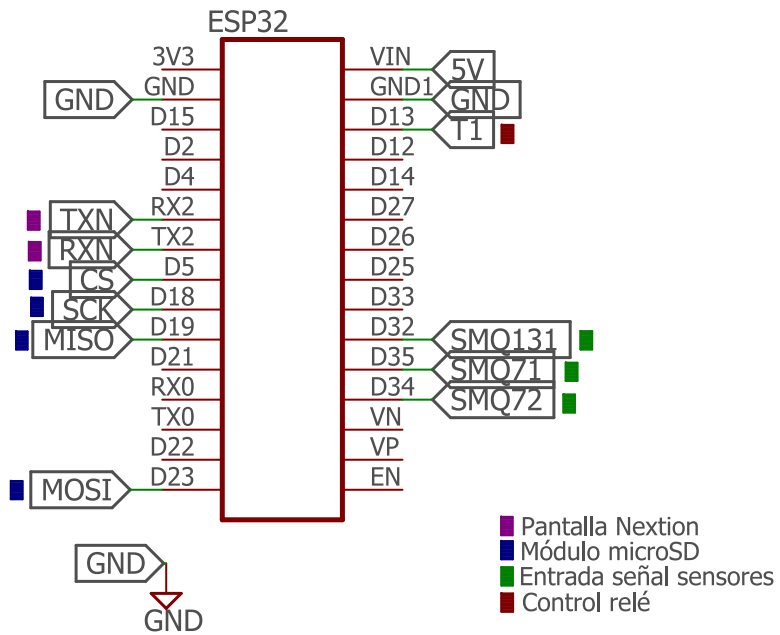


Figura 13: Distribución de pines ESP32 para el prototipo

- ❖ **Diseño circuito impreso:** para reducir el tamaño de la placa se decidió utilizar elementos superficiales para algunas resistencias y capacitores de tamaño 0805 [7] y a su vez se utilizaron dos capas en el diseño de la PCB<sup>1</sup>. El resultado fue una placa de 63 x 65 mm como se puede observar en la figura 14.

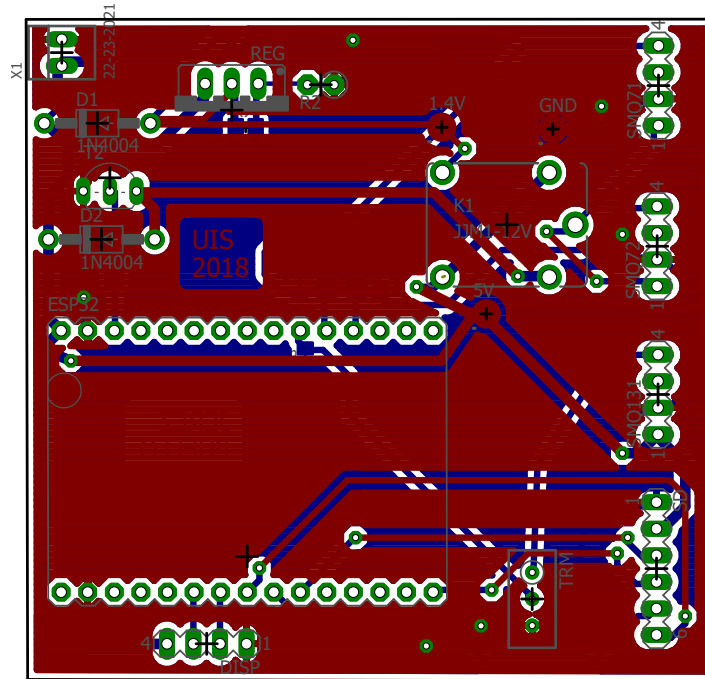


Figura 14: Diseño circuito impreso

---

<sup>1</sup>“Placa de circuito impreso” (del inglés: Printed Circuit Board, PCB).

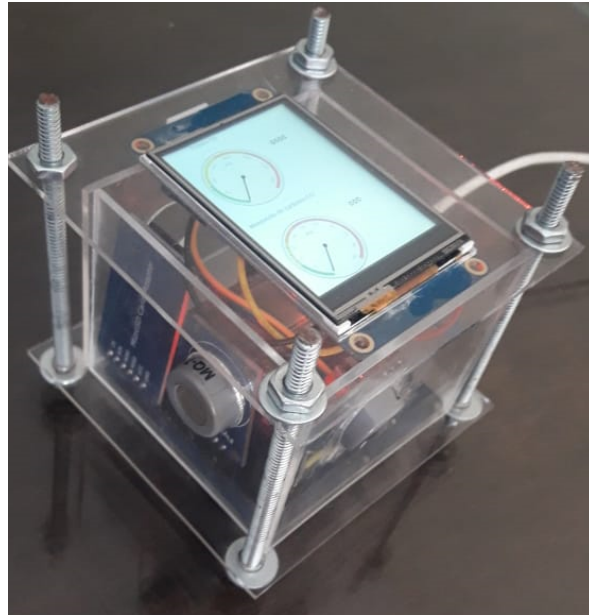


Figura 15: prototipo de medición de calidad del aire

Especificaciones	
Tensión de alimentación	5 V
Peso	200 [g]
Corriente de consumo	1,2 A. Aproximadamente
Dimensiones	9 x 9 x 10 cm

Tabla 6: Especificaciones del prototipo

**3.3.1 Ajuste de los sensores** Para realizar la medición de la calidad del aire el primer paso que se dio fue el de obtener las ecuaciones de los sensores MQ7 y MQ131 utilizando la información gráfica suministrada por el fabricante (Hoja de datos), este procedimiento utilizó la herramienta webplotDigitilizer que permite insertar una imagen de la gráfica del sensor en escala LOG-LOG (logarítmica), seleccionar los puntos de la línea de interés y obtener por medio de regresión una ecuación que describe esa curva.

**3.3.1.1 Sensor de monóxido de carbono MQ7** Para obtener la ecuación de este sensor se tomó la gráfica de la figura 16, que relaciona sus cambios de resistencia ( $R_S/R_0$ ) versus concentración en PPM de CO (partes por millón), se tomaron los puntos de la curva para el gas de monóxido de carbono mediante una herramienta en línea

gratuita bastante útil WebPlotDigitizer, esta herramienta permite tomar los puntos de la curva de interés bastante precisos. Se ingresó esta imagen en *webplotDigitizer* y siguiendo el procedimiento [1] se exporta esta información a Excel y allí se realiza una regresión (lineal, potencial, polinomial), para este sensor la mejor regresión fue potencial y se obtuvo la ecuación 3.1(coeficiente de correlación R aproximadamente igual a 1).

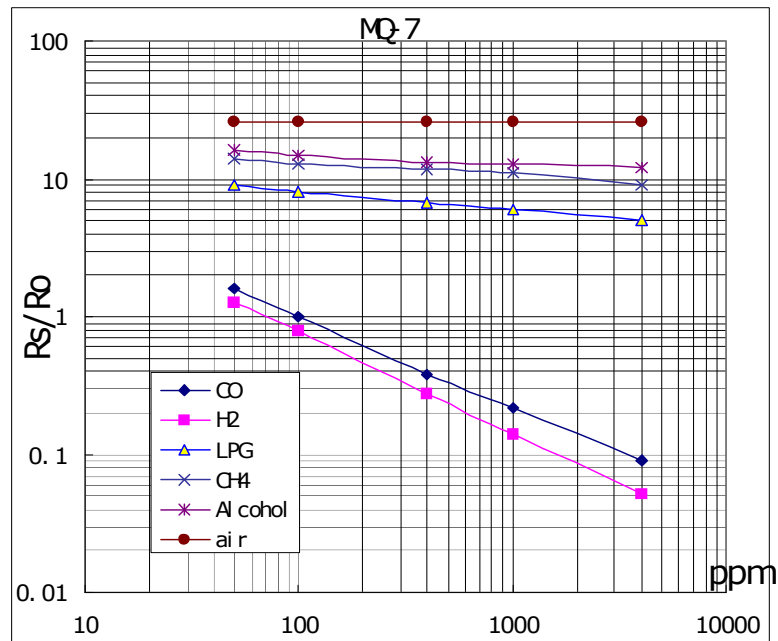


Figura 16: Curvas de gases para el MQ7

Fuente: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>

$$PPM = 97,872(Rs/Ro)^{-1,514}. \quad (3.1)$$

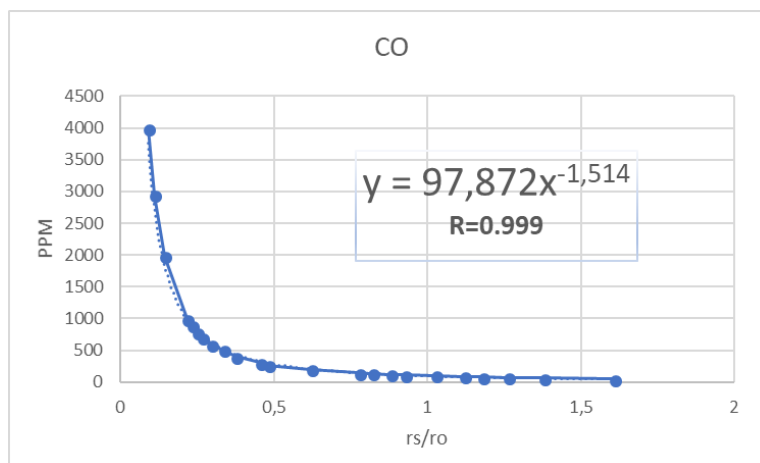


Figura 17: Curva CO, ppm vs (Rs/Ro) en Excel

Las gráficas mencionadas anteriormente, representan la relación de resistencias ( $R_s/R_o$ ) y su respectiva equivalencia en PPM, donde  $R_s$  es la resistencia que se mide a cualquier concentración y  $R_o$  es la resistencia equivalente a una concentración específica, para el sensor de monóxido de carbono es a 100 ppm.

**3.3.1.2 Sensor de ozono MQ131** Siguiendo un procedimiento similar, a partir de la gráfica del sensor, se obtuvo la ecuación 3.2, y funcionó mejor la regresión polinomial (coeficiente de correlación  $R$  aproximadamente igual a 1), la hoja de datos del sensor MQ131 suministra una curva que representa la tensión en el puerto analógico versus la concentración en partículas por millón, dando la posibilidad de hacer mediciones sin calcular la resistencia  $R_o$  del sistema para el MQ131 como se observa en la figura 19, se decidió utilizar la ecuación que relaciona PPM vs tensión analógica.

$$PPM = 165,51 * v^4 - 2140,9 * v^3 + 10303 * v^2 - 21984 * v + 17757 \quad (3.2)$$

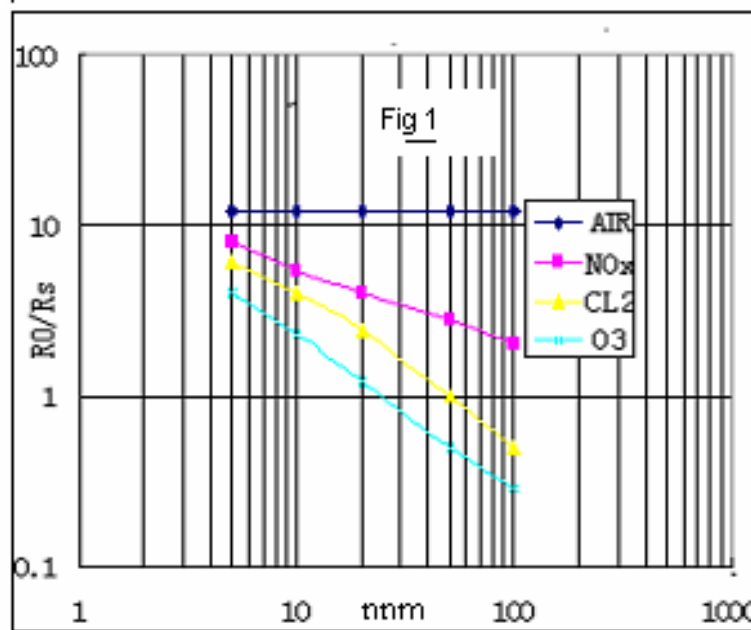


Figura 18: Curvas de gases para el MQ131

Fuente: <http://www.sensorsportal.com/DOWNLOADS/MQ131.pdf>

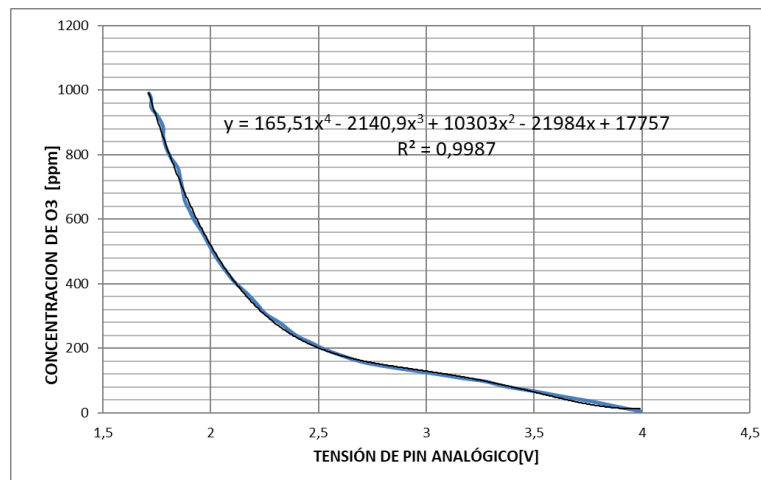


Figura 19: Curvas de tensión de salida analógica vs concentración O<sub>3</sub> para sensor MQ131

Fuente: Hoja de datos MQ131

## Capítulo 4

# DESCRIPCIÓN DE SOFTWARE

### 4.1. SERVIDOR LOCAL: XAMPP

Para habilitar la presentación de los datos de los sensores por medio del ESP32, en una página web, se analizaron múltiples alternativas como lo son Blynk, Cayenne, entre otras y en donde se escogió el programa XAMPP [9], que simula un servidor en el computador, es de código libre, una herramienta de desarrollo que consiste principalmente en el sistema de gestión de bases de datos MySQL <sup>1</sup>, el servidor web Apache<sup>2</sup> y los intérpretes para lenguajes de *script* PHP<sup>3</sup> y Perl. La forma en como se va a establecer la comunicación con el XAMPP se puede observar en la imagen.

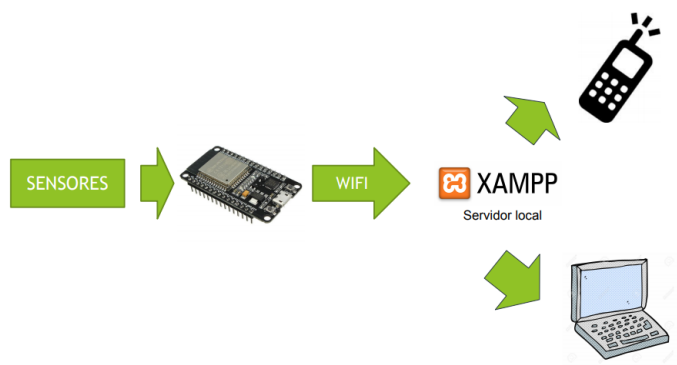


Figura 19: Diagrama de comunicación IoT

El programa actúa como un servidor web libre, fácil de usar e instalar y capaz de

---

<sup>1</sup>Sistema de gestión de base de datos relacional (RDBMS) de código abierto.

<sup>2</sup>Servidor web HTTP de código abierto para la creación de páginas y servicios web.

<sup>3</sup>Hypertext Preprocessor, adecuado para el desarrollo web y que puede ser incrustado en HTML.

interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows el cual fue el sistema operativo que se utilizó para el proyecto, además también funciona en sistemas operativos como GNU/Linux, Solaris y Mac OS X.

En su instalación crea una carpeta por defecto en el disco local del computador en la dirección C:/xampp/htdocs en donde se almacenará toda la información en lenguaje HTML<sup>1</sup>, php y JavaScript<sup>2</sup>, estos archivos contienen la programación de gráficas y la comunicación entre el módulo wifi y el servidor.

**4.1.1 Montaje de servidor y base de datos con XAMPP** A continuación se ilustra la interfaz de XAMPP que permite generar un servidor web y una base de datos de manera local. La interfaz es muy sencilla de utilizar, solo con hacer un par de clics en el botón *start* en la columna *actions* de los servicios *apache* y *MySQL*<sup>3</sup> respectivamente, para verificar que se hayan activado el nombre de los recursos cambiara a color verde.

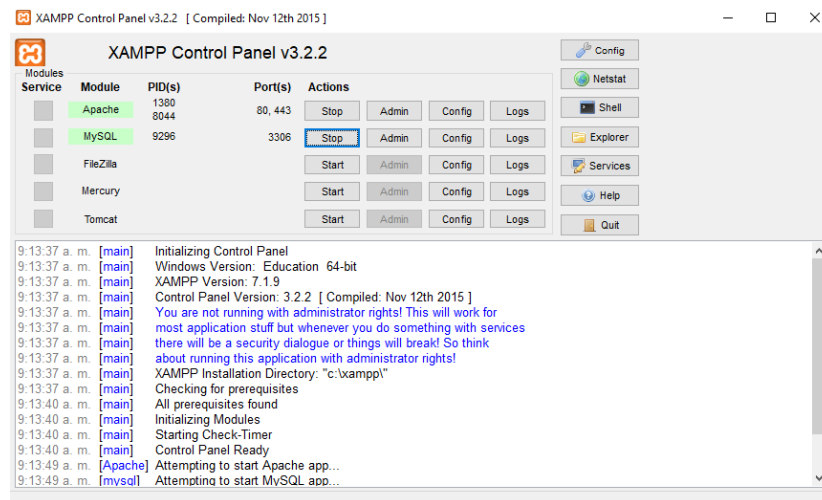


Figura 19: Interfaz de XAMPP

Cuando se accede en el navegador de preferencia a la dirección *localhost* esta página muestra las carpetas contenidas en el disco local en la dirección donde se almacena la

<sup>1</sup>Hypertext Markup Language, lenguaje que se utiliza para el desarrollo de páginas de Internet

<sup>2</sup>JavaScript Object Notation, es un formato ligero de intercambio de datos

<sup>3</sup>MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

información concerniente al servidor local en archivos de tipo HTML, PHP, JavaScript e imágenes entre otro tipo de archivos. Si se quiere acceder a este servidor desde otro dispositivo vinculado a esa red local, solo basta con acceder a la dirección *ip*<sup>1</sup> del equipo donde está ejecutándose el XAMPP desde cualquier navegador.

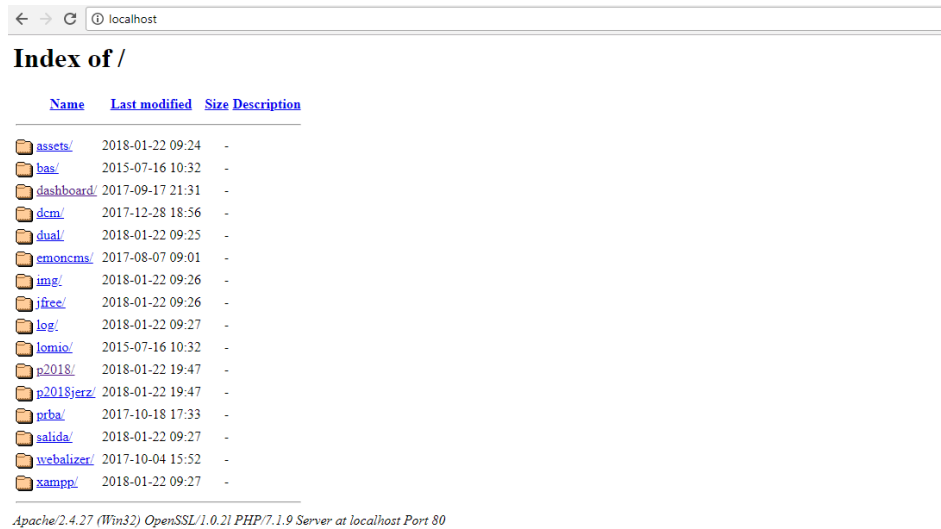


Figura 20: Página principal del servidor local

Al dar clic en la carpeta llamada *dashboard* se accede a la página que trae por defecto XAMPP.

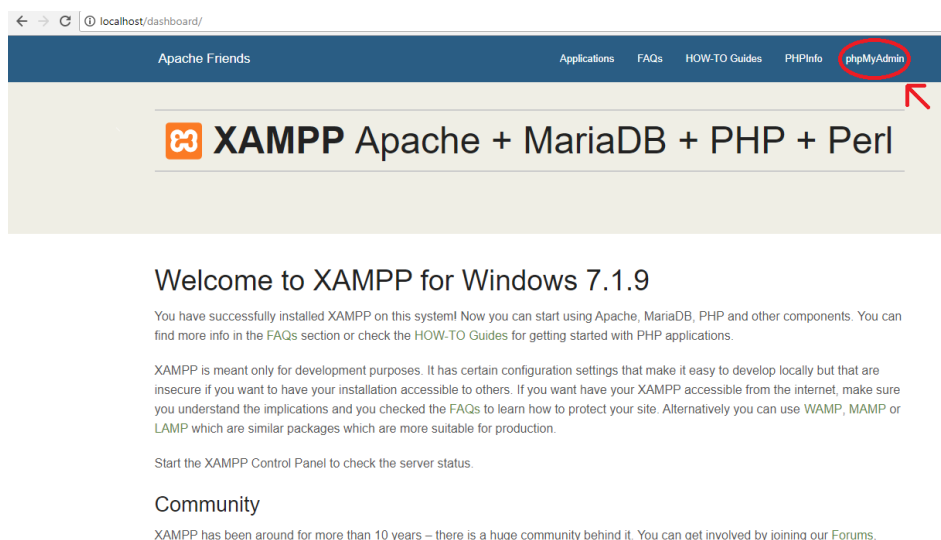


Figura 21: Acceso a base de datos

<sup>1</sup>Protocolo de internet (Internet Protocol),

Para acceder a la base de datos solo basta con dar clic en *phpMyAdmin*, ver figura 21, en dicho sitio es donde se podrá configurar la base de datos, como el tipo de variable (*float*, *String*, *varchar*), etc., el número de dígitos y decimales con los cuales va a contar cada variable entre otras opciones, esto se hace con el fin de almacenar toda la información proveniente de los 3 sensores (2 de monóxido de carbono y uno de ozono).

The screenshot shows the phpMyAdmin interface for a database named 'esp32'. The 'datos' table is selected, and the following data is displayed:

id	fecha	SMC1	O3	SMC2
1533	2018-06-06 10:28:11	0.00	165.00	0.00
1532	2018-06-06 10:28:09	0.00	399.00	0.00
1531	2018-06-06 10:28:06	0.00	990.00	0.00
1530	2018-06-06 10:28:04	0.00	228.00	0.00
1529	2018-06-06 10:28:01	0.00	873.00	0.00
1528	2018-06-06 10:27:59	0.00	343.00	0.00
1527	2018-06-06 10:27:57	0.00	45.00	0.00
1526	2018-06-06 10:27:55	0.00	759.00	0.00
1525	2018-06-06 10:27:53	0.00	65.00	0.00
1524	2018-06-06 10:27:51	0.00	407.00	0.00

Figura 22: Base de datos

## 4.2. COMUNICACIÓN ENTRE EL SERVIDOR LOCAL Y EL MÓDULO WIFI ESP32

Una vez realizado el montaje del servidor y la base de datos que almacena información proveniente de los sensores de monóxido de carbono y ozono que son enviados mediante el módulo wifi ESP32, se estableció un protocolo de envío de información de la tarjeta programable hacia el servidor, para ello se utilizó el lenguaje PHP que permitió crear una conexión con la base de datos y se asignaron los valores entrantes a la tabla.

### 4.2.1 Solicitud GET HTTP

Existen varios tipos de solicitud, *GET*, *POST*, *HEAD*, *PUT*, *DELTE*, *CONNECT*, *OPTIONS*, *TRACE*, *PACTH*, pero para establecer la comunicación entre el módulo ESP32 y el servidor se va a utilizar el método *GET* ver figura 23, que generalmente es utilizado para enviar información, también conocido como *Request-method*, consiste en una línea por donde se envía los datos, usando la URL ver figura 24.

Este método permite el acceso mediante una dirección URL, que sirve para enviar o actualizar información al servidor; es muy utilizada cuando no se necesita seguridad o protección fuerte de la información, ya que al ser un enlace al que se va a acceder se corre el riesgo de que esa información quede almacenada en el historial del navegador o registrada por un motor de búsqueda, pero por naturaleza del proyecto estas direcciones van a ser manejadas por el dispositivo IoT donde no se corre riesgo que se fugue la información del enlace.

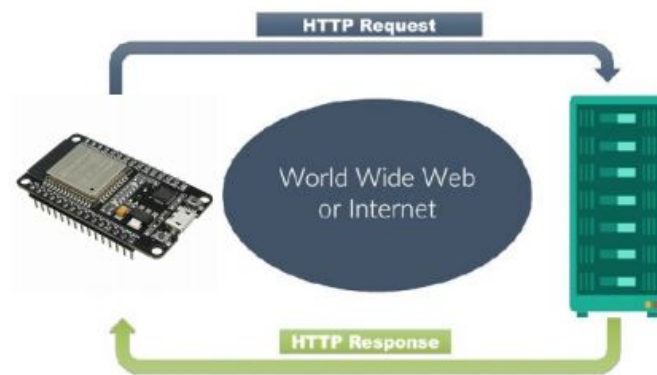


Figura 23: Esquema solicitud cliente-servidor

Fuente: <https://www.webnots.com/what-is-http/>

La estructura de una solicitud tipo GET es muy sencilla como se muestra en la figura 24, consta de nombrar el dominio (si se está utilizando desde internet) o la dirección IP del servidor local que se esté utilizando, seguido de las carpetas donde se ubica el archivo que contiene el lenguaje PHP junto con las variables y sus respectivos valores, como se puede evidenciar no hay límite en la cantidad de variables que se puedan enviar

`carpeta/archivo.php?variable1=valor1&variable2=valor2&variable3=valor3`

`localhost/entrada_datos1.php?SMC1=valor1&O3=valor2&SMC2=valor3`

Figura 24: Esquema solicitud tipo GET

**4.2.2 Ingreso de datos al servidor** Mediante la siguiente dirección URL de lenguaje PHP que se muestra en la siguiente ecuación 4.1, serán ingresados los datos de monóxido de carbono y ozono a la base de datos, para almacenarlos y visualizarlos.

$$localhost/entrada\_datos1.php?SMC1 = 100\&O3 = 2\&SMC2 = 12 \quad (4.1)$$

La imagen 25 muestra el código, el cual establece contacto con el servidor local, donde como primera medida se hace una conexión con la base de datos, ingresando su nombre y contraseña de esta, seguido de la tabla en la cual van a ser insertados la información, se crean las variables que van a componer la estructura tipo GET que está explicado en la figura 24, luego mediante un comando (línea 12 figura 25) de la base de datos SQL se ingresa la información, y como paso final se cierra la conexión.

```

1 <?php
2 $db_host = "localhost";
3 $db_user = "root";
4 $db_password = "";
5 $db_name = "esp32";
6 $db_table_name = "datos";
7 $db_connection = mysqli_connect($db_host, $db_user, $db_password);
8 mysqli_select_db($db_connection,$db_name) or die;
9
10 $dato = $_GET['SMC1'];
11 $dato2= $_GET['O3'];
12 $dato3= $_GET['SMC2'];
13 $insertar="INSERT INTO datos (id,fecha,SMC1,O3,SMC2) VALUES (NULL,CURRENT_TIMESTAMP ,'$dato', '$dato2' ,'$dato3')
14 ";
15 $retry_value = mysqli_query($db_connection,$insertar);
16 mysqli_close($db_connection);
17 ?>
18
19 |

```

Figura 25: formulario de datos

**4.2.3 Programación módulo ESP32** Para la programación del módulo ESP32 se tiene que tener en cuenta la conexión a una red Wifi, la lectura de los sensores para el desarrollo de este proyecto son analógicos, el muestreo de los datos obtenidos del pin analógico, la comunicación con la base de datos del servidor local, soporte de almacenamiento en una memoria micro SD y visualización de los datos obtenidos en un display siguiendo el diagrama de flujo que se muestra en la figura 26; para ello la ESP32 en su plataforma de Arduino cuenta con librerías con sus respectivas plantillas, creadas especialmente para esta tarjeta y que se mostrarán a continuación:

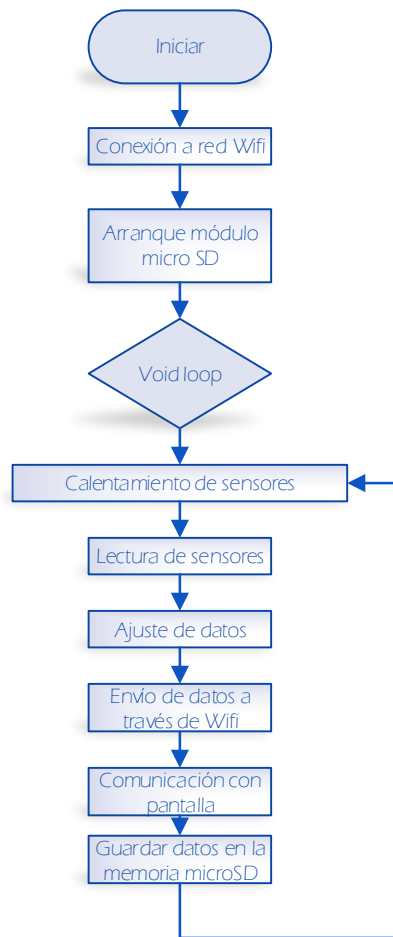


Figura 26: Diagrama de flujo proceso ESP32

- ❖ **Lectura analógica de los sensores:** La ESP32 cuenta con 18 canales adc de 12 bits de resolución y pueden muestrear tensiones comprendidas entre 0 y 3,3 voltios a una frecuencia de más de 1khz, es decir que 3.3 Voltios es igual a 4095, utilizando comandos propios de la plataforma Arduino, se toman lecturas del pin ADC, pero el dato que se obtiene es número cuantizado que adopta valores entre 0 y 4095 por la resolución de 12 bits, para ello se convierte el valor a uno de tensión dado a sabiendas que el máximo valor de tensión que puede registrar el ADC es de 3.3 voltios como se muestra en la figura 27. Luego de adquirir el valor se convierte en formato tipo *String* con el fin de hacer el envío por wifi,

para garantizar una lectura confiable de los sensores se realizó por *software* un promediado de la tensión.

```
int val = (analogRead(analogPin));
float val2 = (3.3/4095)*val;
String val3 = val2;
```

Figura 27: Código lectura analógica de los sensores

- ❖ **Muestreo de los datos obtenidos del pin analógico:** Para la lectura de los sensores es necesario tomar 10 muestras en un segundo y seguidamente obtener su promedio, para hacer la toma del dato lo más exacto posible, el código a utilizar se muestra en la figura 28.

```
void muestreo(){
  v0=0;
  for(byte i =0; i<10; i++){
    float A=(3.3/4095)*analogRead(32);
    v0+=A;
    delay(100);
  }
  v0/=10;
}
```

Figura 28: Código de muestreo de la señal analógica

- ❖ **Conexión Wifi:** Dentro de las librerías suministradas para la ESP32, se cuenta con una librería que establece la conexión Wifi con una red local, solo es necesario agregar al código el nombre de la red y la contraseña, se puede decidir en qué momento se quiere tener encendida la comunicación para disminuir la exigencia del procesador, como se muestra en la figura 29.

```
state=0;
wifiMulti.addAP("Redwifi", "contraseña");
if((wifiMulti.run() == WL_CONNECTED)) {
  Serial.print("Connected...\n");
}
else{
  Serial.print("Error al conectar a la red...\n");
}
if (state==1){
  WiFi.disconnect(true);
}
```

Figura 29: Ejemplo código de conexión y desconexión a una red Wifi

- ❖ **Comunicación con la base de datos del servidor local:** Para la comunicación de la ESP32 con el servidor local se utilizó una librería del módulo llamada *HTTPclient*. Esta librería tiene líneas de código importantes para hacer una solicitud tipo GET, como ya la plantilla está diseñada, solo basta con modificar la línea donde se ingresa la dirección URL a la cual se va a comunicar la tarjeta como se muestra en la figura 30. Los datos obtenidos por el pin analógico se tienen que convertir a tipo *string*<sup>1</sup>.

```
HTTPClient http;

http.begin("http://192.168.1.6/entrada_datos1.php?variable="+valor);
int httpCode = http.GET();

if(httpCode > 0) {
  if(httpCode == HTTP_CODE_OK) {
    payload = http.getString();
  }
} else {
  USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
    http.errorToString(httpCode).c_str());
}
```

Figura 30: Plantilla HTTPclient para solicitud tipo GET

- ❖ **Almacenamiento de datos:** Se seleccionó como recurso de almacenamiento una memoria microSD, donde por medio de librerías que aportaron los desarrolladores de la ESP32 para Arduino, se guardarán los datos en un archivo de formato csv<sup>2</sup>, esto se hace con el fin de tener un respaldo de información de los datos obtenidos por la tarjeta y también para pruebas de campo donde no se tenga acceso a una red local. Dentro de ese archivo la ESP32 guarda los datos de cada sensor llenando una tabla delimitado por comas, donde el límite de datos que se pueden llenar está dado por la capacidad de la memoria microSD.

---

<sup>1</sup>Tipo de variable para representar texto

<sup>2</sup>Los archivos CSV (del inglés comma-separated values) formato abierto sencillo para representar datos en forma de tabla.

```

void microsd(){
  if(!SD.begin()){
    Serial.println("Card Mount Failed");
    return;
  }
  uint8_t cardType = SD.cardType();
  if(cardType == CARD_NONE){
    Serial.println("No SD card attached");
    return;
  }
  Serial.print("SD Card Type: ");
  if(cardType == CARD_MMC){
    Serial.println("MMC");
  } else if(cardType == CARD_SD){
    Serial.println("SDSC");
  } else if(cardType == CARD_SDHC){
    Serial.println("SDHC");
  } else {
    Serial.println("UNKNOWN");
  }
  uint64_t cardSize = SD.cardSize() / (1024 * 1024);
  Serial.printf("SD Card Size: %lluMB\n", cardSize);
  appendFile(SD, "/hello.csv", "dato1=valor\n");
}

```

Figura 31: Plantilla almacenamiento memoria microSD

❖ **Visualización pantallas Nextion [4]:** Para la utilización de este dispositivo se tiene que tener en cuenta lo siguiente:

- **Programación pantalla:** Se realizó a través de un software suministrado por la misma compañía creadora de la pantalla NEXTION, en donde se insertan los objetos que se requieran visualizar en la pantalla. Como se puede evidenciar en la figura 32 se presenta a manera de ejemplo el diseño para explicar el funcionamiento del dispositivo, que consta de 3 botones y un indicador de números, el software agrega etiquetas con fondo amarillo a cada objeto utilizado en la visualización.

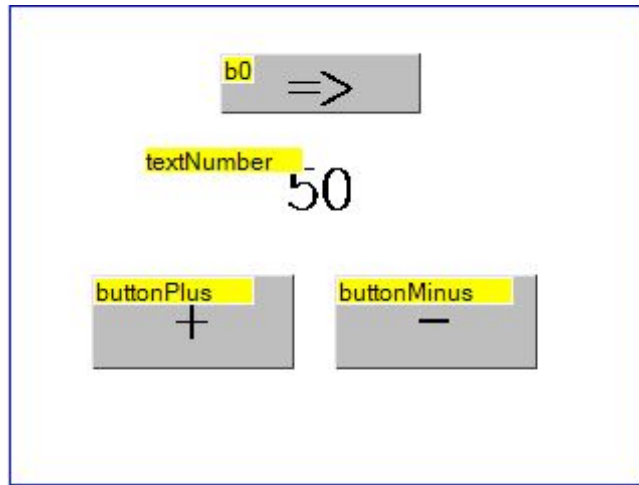


Figura 32: Diseño de la visualización en NEXTION Editor

Luego de haber definido el entorno gráfico de la pantalla el software crea un archivo de formato *.TFT*, este archivo se copia a una memoria micro SD y se conecta a la pantalla donde actualiza los archivos y verifica que los datos fueron cargados correctamente como se muestra en la figura 33.

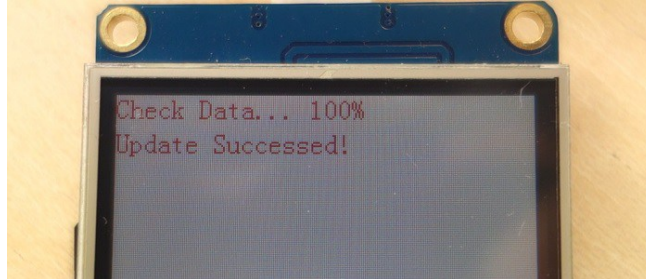


Figura 33: Verificación de los datos cargados a la pantalla Nextion

- **Programación en la ESP32:** la comunicación entre la pantalla y la ESP32 se realiza por comunicación UART, donde el proveedor NEXTION, aporta la librería para su trabajo en Arduino, en el código se especifican las etiquetas que se desean modificar, como lo son el texto y las que se desean escuchar (Acciones hechas por los botones desde la pantalla), que en este caso es cuando se accione un botón desde la pantalla, como se evidencia en la figura 34.

```

#include "Nextion.h"
#include "HardwareSerial.h"
HardwareSerial Serial2(2);

NexText textNumber = NexText(0, 2, "textNumber");
NexButton buttonPlus = NexButton(0, 3, "buttonPlus");
NexButton buttonMinus = NexButton(0, 4, "buttonMinus");

int number = 50;
char buffer[10] = {0};

NexTouch *nex_listen_list[] =
{
    &textNumber,
    &buttonPlus,
    &buttonMinus,
    NULL
};

void buttonPlusPushCallback(void *ptr)
{
    if (number < 100) number += 1;
    memset(buffer, 0, sizeof(buffer)); // clear buffer
    itoa(number, buffer, 10);
    textNumber.setText(buffer);
}

void buttonMinusPushCallback(void *ptr)
{
    if (number > 0) number -= 1;
    memset(buffer, 0, sizeof(buffer)); // clear buffer
    itoa(number, buffer, 10);
    textNumber.setText(buffer);
}

void setup(void)
{
    nexInit();
    Serial.begin(115200);
    Serial.println("Start");
    buttonPlus.attachPush(buttonPlusPushCallback);
    buttonMinus.attachPush(buttonMinusPushCallback);
    dbSerialPrintln("Setup done");
}

void loop(void)
{
    nexLoop(nex_listen_list);
}

```

Figura 34: Plantilla de visualización NEXTION

### 4.3. VISUALIZACIÓN *WEB*

Para la visualización existen varias opciones *Blynk*, *thinger io*, *Emoncms*, *Cayene* donde unas se encuentran limitadas en número de dispositivos conectados para enviar información, velocidad de trasmisión de datos y capacidad de almacenamiento o acceder a ilustraciones más llamativas entre otras, o en otros casos pagar para acceder a mejoras.

Las lecturas de los sensores utilizarán programación en JavaScript que permite ilustrar al usuario para su fácil entendimiento e interpretación de datos, proveniente de

unas plantillas de gráficas de uso libre, que están previamente adecuadas para lo que se quiere desde *highcharts* [6].



Figura 35: Página principal web

En esta página creada a partir de *Wordpress* [11], en la barra principal dándole clic a los enlaces gráficas (Figura 36) PPMs o Gráficas podemos acceder a la visualización de los datos de los sensores ambos en tiempo real y en diferentes tipos de presentación.

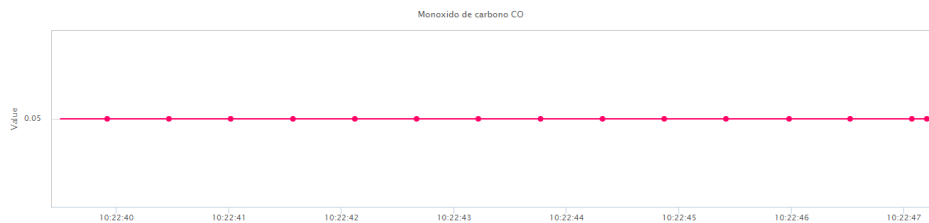


Figura 36: Enlace Gráficas

Como se presenta en la figura 37 se visualiza la cantidad en ppm de concentración de monóxido de carbono y de ozono en ppb forma de indicador pensados de fácil entendimiento por colores [2] siendo el verde el menos perjudicial para la salud y el rojo por el contrario el más perjudicial para el cuerpo humano.

El Gráfico indicador esta por colores siendo el verde bueno la calidad de aire es considerada como satisfactoria y la afectación en la contaminación del aire es pequeña y no evidencia ningún efecto en la salud humana. El amarillo, Moderado, La calidad de aire es aceptable y no tiene ningún efecto sobre la población en general, Naranja, Regular, Aumento de molestias en personas con padecimientos respiratorios y cardiovasculares;

aparición de ligeras molestias en la población en general. Rojo, Malo, Agravamiento significativo de la salud en personas con enfermedades cardíacas o respiratorias. Afectación de la población sana. para el monóxido de carbono en PPM (partes por millón) y el del ozono en PPB (partes por billón), esta información también se encuentra en la página desarrollada para el proyecto, de tal manera que el usuario pueda entender la clasificación de los colores.

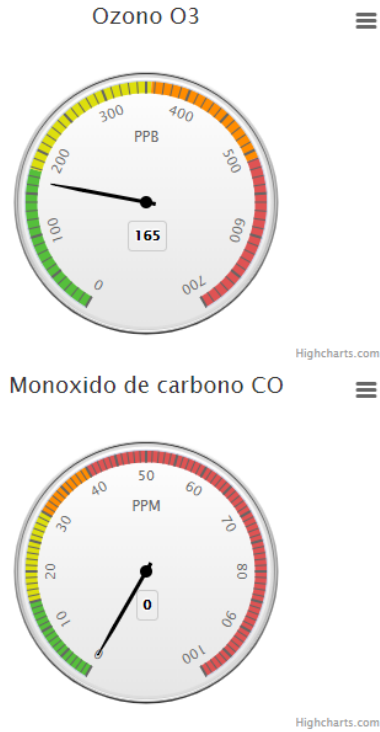


Figura 37: Enlace PPMs

## Capítulo 5

# PRUEBAS DE LABORATORIO Y CAMPO

Se realizaron las siguientes pruebas para garantizar que la lectura de los sensores que se va a mostrar en el servidor local sea acorde a la información suministrada por la hoja de datos de los sensores.

### 5.1. CARACTERIZACIÓN PUERTOS ADC

En la prueba de los puertos analógicos a digital, se realizó un barrido de voltaje desde 0 a 3.3 voltios con una fuente variable de tensión, el porcentaje de error en dichos pines supera el 49% para valores cercanos a cero y se va reduciendo a medida que se aumenta la tensión como se puede observar en la figura 37 y en la tabla 6.

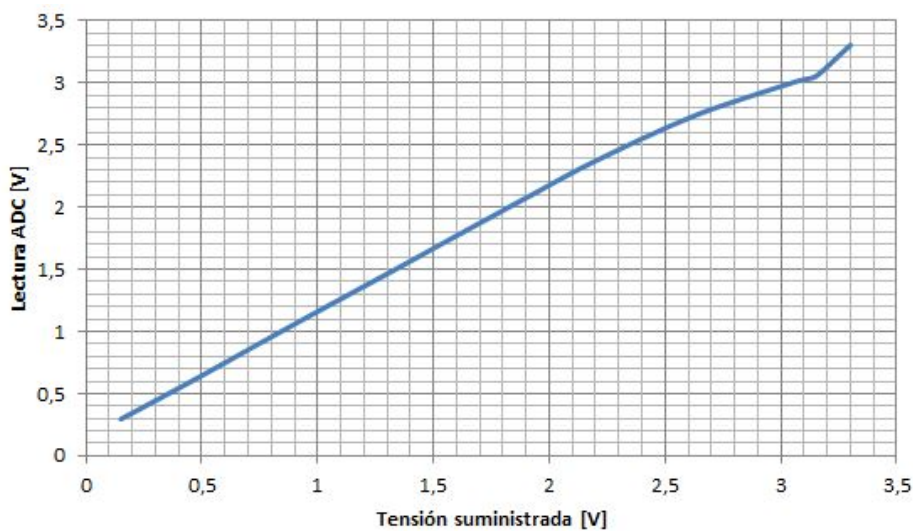


Figura 37: lectura ADC vs tensión suministrada

Tensión suministrada [V]	Vadc [v]	Error absoluto [%]
0.295	0.150	49.15
0.625	0.480	23.2
0.936	0.780	16.67
1.191	1.030	13.51
1.535	1.370	10.74
1.781	1.610	9.6
2.095	1.920	8.35
2.372	2.200	7.25
2.738	2.630	3.94
3.013	3.070	1.89
3.052	3.150	3.21
3.304	3.300	0.12

Tabla 6: Porcentaje de error puerto ADC

Para corregir la lectura analógica se ajustó los datos obtenidos, dando como resultado la ecuación 5.1, y la tabla 7 muestra el porcentaje de error con la tensión corregida. Como se puede ver, el porcentaje de error disminuye, pero el error sigue siendo elevado a valores pequeños de tensión.

$$V_o = 0,9343 * V_{adc} - 0,2227 \quad (5.1)$$

Tensión suministrada [v]	Tensión corregida Vo [V]	Error [%]
0.295	0.364	23.44
0.625	0.675	8.05
0.936	0.958	2.37
1.191	1.193	0.25
1.535	1.514	1.32
1.781	1.74	2.24
2.095	2.033	2.94
2.372	2.297	3.14
2.738	2.702	1.28
3.013	3.117	3.47
3.052	3.193	4.62
3.304	3.334	0.92

Tabla 7: Porcentaje de error con ajuste lineal

Se procedió a realizar otra regresión ajustando la curva a una función polinómica de orden 4, donde se obtuvo la ecuación 5.2 y su respectivo porcentaje de error en la tabla 8, como se puede observar el error se redujo sustancialmente en valores inferiores

a 0.6 voltios, y con una tasa de error inferior al 3.65 % en todo el rango de tensiones permitidos por el puerto ADC. El ajuste de la ecuación 5.2, se aplicó en los puertos q van a ser utilizados para la lectura de los sensores, obteniendo respuestas de error similares a las de la tabla 8.

$$V_o = 0,0268 * V_{adc}^4 - 0,1996 * V_{adc}^3 + 0,4322 * V_{adc}^2 + 0,7102 * V_{adc} + 0,1902 \quad (5.2)$$

Tensión suministrada [v]	Tensión corregida Vo [V]	Error [%]
0.295	0.305	3.65
0.625	0.610	2.39
0.936	0.922	1.46
1.191	1.192	0.10
1.535	1.555	1.33
1.781	1.801	1.12
2.095	2.098	0.16
2.372	2.346	1.05
2.738	2.698	1.43
3.013	3.049	1.20
3.052	3.115	2.08
3.304	3.245	1.76

Tabla 8: Porcentaje de error con ajuste de orden 4

## 5.2. PRUEBA PARA AJUSTE DE LOS SENSORES

Para obtener de forma completa la ecuación que relaciona la tensión leída por los puertos ADC con la concentración en PPM se realizaron pruebas en ambientes “controlados” y se almacenaron las lecturas de los sensores junto con las obtenidas por equipos comerciales que se consideraron como patrones. En las pruebas realizadas para los sensores de monóxido de carbono se planteó un proceso de combustión controlado en un recipiente donde se pueda contener la concentración de CO, junto con una medición patrón, que en este caso es un medidor de monóxido de carbono fluke CO-220 adquirido por la Universidad Industrial de Santander y el (AS8700A) adquirido por los autores. Como se puede evidenciar en la figura 38 se realizó el proceso de ajuste de los sensores y pruebas. Después de realizado este ajuste del sensor, con los dos sensores MQ7 que se implementaron, se reguló la concentración de monóxido de carbono a la recomendada por la hoja de datos que son 100 ppm, y se determinó la resistencia Ro, 14176 y 8180

ohmios respectivamente para los sensores de monóxido de carbono.

Dentro de las pruebas desarrolladas para el sensor de monóxido de carbono, hechas en un recipiente como se puede evidenciar en la figura 38 y dado que se hicieron repetidas pruebas para calcular la resistencia  $R_0$ , el recipiente y los sensores quedan con concentraciones remanentes, lo cual afecta la medición y a su vez el ajuste de los sensores. Para futuras pruebas es necesario después de una prueba hacer limpieza o ventilar los sensores para que no se acumulen residuos de pruebas pasadas.



Figura 38: Ensayo calibración del sensor MQ7

MQ7-1 [PPM]	MQ7-2 [PPM]	Fluke [PPM]	AS8700A [PPM]	Error MQ7-1 [%]	Error MQ7-2 [%]
715	392	845	837	16	54
608	312	705	698	14	56
568	306	670	657	16	54
390	213	445	445	12	52
249	136	215	215	16	37
226	126	175	172	30	28
192	109	145	145	32	25
185	103	134	134	38	23
102	57	110	108	7	48
44	29	69	72	35	58
33	24	53	55	36	55
12	9	39	44	61	77
10	7	36	37	70	81

Tabla 9: Porcentaje de error absoluto respecto al sensor fluke CO-220

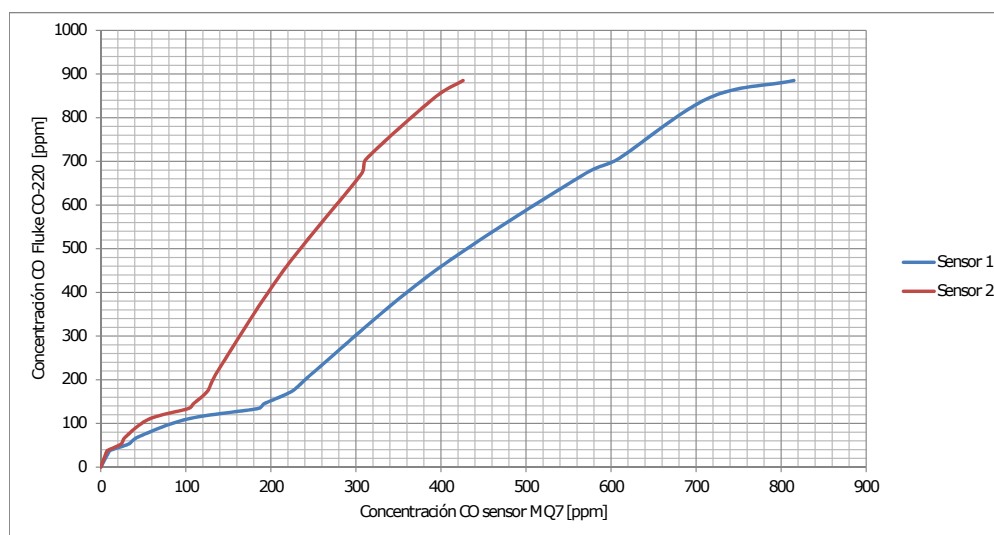


Figura 39: gráficas de respuesta de sensores MQ7

Como se puede evidenciar en la tabla 9 el porcentaje de error para el sensor 1 es menor en comparación al sensor 2, aun así el error es elevado para un dispositivo de medición, con ayuda del equipo patrón se hizo un ajuste para los dos sensores para disminuir el error, se ajusta haciendo regresión y/o cambiando R0 y haciendo regresión polinómica de orden 3, ecuaciones 5.3 y 5.4, obteniendo porcentaje de errores menores,

como se evidencia en la tabla 10.

$$PPM1 = -0,000003 * ppms1^3 - 0,0032 * ppms1^2 + 0,111 * ppms1 + 34,85 \quad (5.3)$$

$$PPM2 = -0,00002 * ppms2^3 - 0,0124 * ppms2^2 + 0,0082 * ppms1 + 35,24 \quad (5.4)$$

MQ7-1 [PPM]	MQ7-2 [PPM]	Fluke [PPM]	AS8700A [PPM]	Error MQ7-1 [%]	Error MQ7-2 [%]
649	725	865	960	25	16
653	655	640	645	2	2
470	481	500	503	6	3,8
474	460	425	428	12	8
425	406	405	407	5	0,24
363	342	355	358	2	3,6
343	325	340	344	0,8	4,4
280	260	280	281	0	7
232	252	265	278	12	5
195	183	198	200	1,5	8
133	133	139	137	4	4,3
67	73	75	79	17	3
56	66	68	73	18	3
46	52	60	64	23	13
41	47	49	53	16	4
37	37	38	43	2,7	3

Tabla 10: Porcentaje de error absoluto de los sensores MQ7 corregido respecto al sensor fluke CO-220

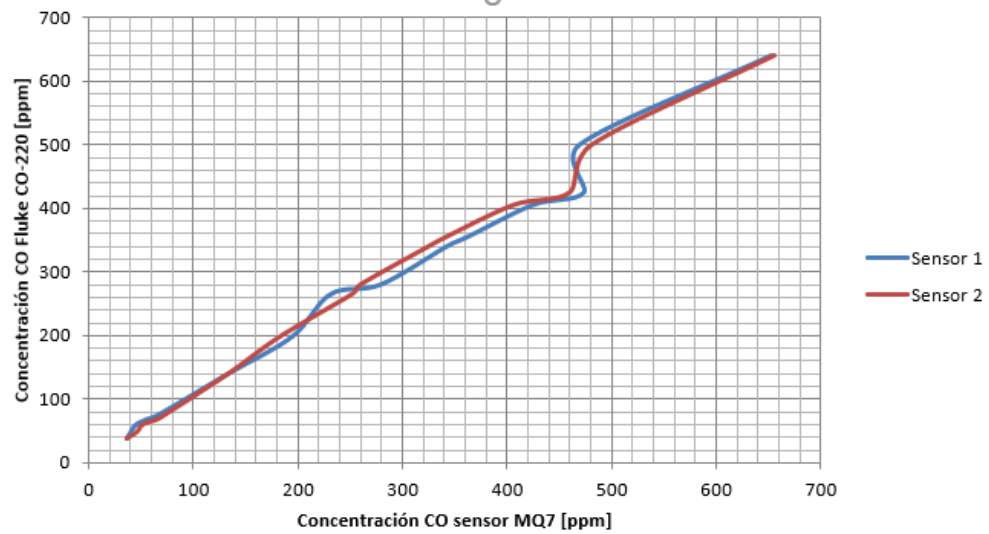


Figura 40: gráficas de respuesta de sensores MQ7 con corrección polinómica

Como se pudo observar en las anteriores tablas y gráficas el error de medición de monóxido de carbono en PPM respecto al sensor Fluke CO-220 disminuyó considerablemente obteniendo un porcentaje de error mínimo del 3% y máximo de un 16%.



Figura 41: Medidor de monóxido de carbono AS8700A

Característica	AS8700A
Tipo de sensor	Electroquímico estabilizado para monóxido de carbono
Unidades de medición	CO: ppm, Temperatura: °C / °F
Rango de medición	CO: 0 - 1000 PPM, Temperatura: 0 - 50 °C/ 32 - 122 °F
Resolución	CO:1 PPM, Temperatura: 0.1 °C / 0.1 °F
Precisión	CO: ±5 % o ±10 PPM, temperatura: ±0.1 °C
Tiempo de respuesta	<=60s

Tabla 11: Ficha técnica Medidor de monóxido de carbono AS8700A



Figura 42: Medidor de monóxido de carbono Fluke CO-220

Característica	AS8700A
Tipo de sensor	Electroquímico estabilizado para monóxido de carbono
Unidades de medición	CO: ppm
Rango de medición	CO: 0 - 1000 PPM
Resolución	CO:1 PPM
Precisión	CO: ±5 % o ±2 PPM
Tiempo de respuesta	<20s

Tabla 12: Ficha técnica Medidor de monóxido de carbono fluke CO-220



Figura 43: Medidor de ozono O3 42M

*Fuente: Cotización de Electrónicas DC. Ltda*

Característica	O3 42M
Tipo de sensor	Absorción Ultravioleta con control por microprocesador.
Unidades de medición	O3: PPM
Resolución	0-0,1/0.2/0.5/1/2/5/10 PPM, modificables.
Ruido	0.4 ppb
Tiempo de respuesta	automático y programable (mínimo 20 segundos)

Tabla 13: Ficha técnica Medidor de ozono O3 42M, propiedad de la CDMB

Para probar el sensor de ozono se contó con la cooperación por parte de la Corporación Autónoma Regional Para la Defensa de la Meseta de Bucaramanga (CDMB) que tiene dentro de sus equipos activos un sensor de ozono como el que se puede evidenciar en la figura 43, donde se tomarán lecturas de ambiente para calcular el porcentaje de error con respecto al suministrado por la hoja de datos del sensor MQ131 como se evidencia en la imagen 44, donde al obtener los resultados se observó que los valores de concentración de Ozono en el medio ambiente son inferiores a 1 PPM, por lo tanto los valores obtenidos no se podían comparar con los del sensor MQ131, ya que su rango de valores es hasta de 10 hasta 100 PPM.



Figura 44: Prueba del sensor MQ131 con generador de ozono

Por tal motivo, se diseñó un generador de ozono, y dentro de las pruebas desarrolladas con el sensor MQ131 Se observó que cuando se activa el generador, la concentración en PPM de O<sub>3</sub> aumenta, ...como en el ambiente, de manera general, la concentración de O<sub>3</sub> no es muy alta (normal), se decidió construir un generador de ozono y realizar las mediciones en un ambiente controlado. se observó que a mayor concentración de O<sub>3</sub>, el sensor, controlado por el ESP32 respondió como se esperaba. ver figura 46, esta respuesta era esperada, ya que su hoja de datos indica este comportamiento.



Figura 45: Prueba del sensor MQ131 con generador de ozono

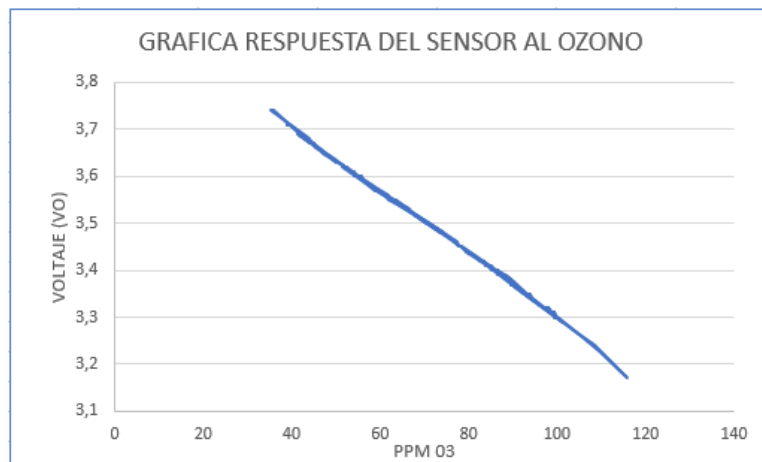


Figura 46: Respuesta del sensor MQ131

## *Capítulo 6*

# *CONCLUSIONES*

El microcontrolador elegido para el desarrollo de este proyecto fue una buena opción, ya que gracias a su precio y a su capacidad de computo, se logró una buena comunicación con los sensores parte fundamental para el desarrollo de este proyecto y wifi. A su vez es muy útil que su programación sea a través de la plataforma de Arduino.

Para los dos modos de visualización, indicador redondo (*gauge*) y PPM vs tiempo que se implementaron, se encontró que el tiempo de muestreo mínimo que se puede programar en el código de visualización web fue de 500 ms (si no se superponen o se pierden los datos), y como las variables que se están midiendo desde el punto de vista eléctrico son relativamente lentas, se tomó la decisión de programar al ESP32 (datos promediados) para enviar datos al servidor cada segundo y garantizar un buen funcionamiento del sistema.

Aunque existen grandes variedades de plataformas IoT, el diseño de esta, como en nuestro caso, ofrece la posibilidad de personalizar desde el entorno hasta el método de envío de datos, tiempo de muestreo, número de variables a visualizar, capacidad de almacenamiento y cantidad de dispositivos conectados al servidor.

Para el ajuste de los sensores de la serie MQ es indispensable un equipo patrón, de esta forma estos dispositivos que son de bajo costo puedan tomar lecturas con un porcentaje de error inferior al 10 %, esto con el fin de que los datos sean lo más confiables posible.

## *Capítulo 7*

# ***RECOMENDACIONES PARA TRABAJOS FUTUROS***

Es necesario continuar con el desarrollo de prototipos de medición de la calidad del aire ya que nos indican que tan perjudicial es el aire que respiramos para nuestra salud.

Las variables que se manejan en este proyecto, monóxido de carbono y ozono son algunas de las que se requieren para determinar la calidad del aire, pero no son suficientes para definir la calidad de aire, en este proyecto se pudo mostrar que con unos sensores de bajo costo se pueden hacer mediciones de monóxido de carbono y ozono. Se recomienda expandir el prototipo a más variables como lo son la temperatura, humedad, dirección del viento, material particulado, dióxido de carbono, entre otras.

La protección contra sobre tensiones, cuando se operan bobinas de relés, ya que puede dañar componentes electrónicos como la ESP32, por esta razón se deben usar diodos antiparalelos con las bobinas (diodo de libre paso).

Se recomienda adicionar una etapa que le permita al equipo hacer uso de baterías y de su respectiva carga, donde ya hay ESP32 que cuentan con el módulo de carga o el diseño de dicha etapa, esto con el fin de hacerlo portátil.

El prototipo diseñado depende de la cobertura de la señal Wifi, si se desea aumentar el alcance se puede agregar un módulo GSM que permita el uso de la navegación web para poder enviar la información.

Es fundamental implementar un sistema de seguridad tanto en la transmisión de los datos al servidor, como en la página *web* donde se están visualizando los datos.

## *Bibliografía*

- [1] ]. Webplotdigitizer. <https://apps.automeris.io/wpd/>. Herramienta para la extracción de puntos de una curva en una gráfica con alta precisión.
- [2] CDMB. Informe de red de monitoreo decalidad del aire del Área metropolitana de bucaramanga. <http://www.cdm-b.gov.co/web/ciudadano/centro-de-descargas/193-informe-enero-marzo-2010-1/file>, keywords = rangos.indices ,, Mar. 2010. informacion sobre la calidad del aire-rangos.
- [3] CINJORDIZ, C. Nodemcu esp32. <https://www.infootec.net/nodemcu-esp32/>. Formación en electricidad y electrónica, aficionado a la robótica y seguidor en todos los temas relacionados con la programación y nuevas tecnologías. CesarCinjordiz@gmail.com.
- [4] EDITOR, N. Nextion editor. <https://nextion.itead.cc/resources/download/nextion-editor/>. Software de desarrollo para las pantallas nextion.
- [5] ESPRESSIF. datasheet. [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Hoja de datos del módulo ESP32.
- [6] HIGHCHARTS. <https://www.highcharts.com/>. HighCharts es una librería escrita en JavaScript que permite la creación de gráficas. La librería ofrece un método fácil e interactivo para insertar graficas en su sitio web o aplicación web.
- [7] MURATA]. información. <http://uk.farnell.com/murata/grm21br61c106ke151/cap-mlcc-x5r-10uf-16v-0805/dp/1762635?MER=sy-me-pd-mi-alte>. capacitor superficial 0805.

- [8] OMS. Organizaci3n mundial de la salud, calidad del aire ambiente (exterior) y salud. [http://www.who.int/es/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](http://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health), Feb. 2018.
- [9] RUEDA, B., AND RIDRUEJO, D. L. Xampp entorno de desarrollo con php. <https://www.apachefriends.org/es/index.html>, 2008. Apache Friends es un proyecto para promocionar el servidor de web Apache y es el hogar del proyecto XAMPP. XAMPP es una distribuci3n de Apache f3cil de instalar que contiene MariaDB, PHP y Perl.
- [10] SEMICONDUCTOR], O. O. datasheet. <https://pdf1.alldatasheet.com/datasheet-pdf/view/11662/ONSEMI/LM317.html>. Hoja de datos del LM317.
- [11] WORDPRESS. <https://es.wordpress.com/>. WordPress es un sistema de gesti3n de contenidos (CMS) que permite crear y mantener un blog u otro tipo de web.