

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE PROTOTIPOS DE MONITOREO REMOTO
BASADOS EN ESP32 Y PÁGINA WEB**

**IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA DE
MONITOREO AMBIENTAL DE TEMPERATURA, HUMEDAD,
PRESIÓN Y CALIDAD DEL AIRE BASADO EN ESP32 Y PÁGINA
WEB**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
SUPERIOR EN REDES Y TELECOMUNICACIONES**

JHON ALEXANDER TOSCANO AGUIRRE

DIRECTOR: ANDRÉS FERNANDO REYES CASTRO

DMQ, febrero 2024

CERTIFICACIONES

Yo, JHON ALEXANDER TOSCANO AGUIRRE declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no se ha presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas de este documento.

JHON ALEXANDER TOSCANO AGUIRRE

jhon.toscano@epn.edu.ec

jata_2022@hotmail.com

Certifico que este trabajo de integración curricular lo desarrolló JHON ALEXANDER TOSCANO AGUIRRE, bajo mi supervisión.

ANDRES FERNANDO REYES CASTRO
DIRECTOR

andres.reyes@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JHON ALEXANDER TOSCANO AGUIRRE

DEDICATORIA

Este proyecto está dedicado primeramente a Dios, el cual me ha demostrado que siempre ha estado conmigo a lo largo de este caminar al darme fuerzas para no desmayar.

A mi madre Anita y mis hermanos Stalin y Gaby que han sido siempre un pilar muy importante a lo largo de mi vida personal y profesional que con su amor incondicional supieron aconsejarme y guiarme para no rendirme, a seguir superar cada obstáculo que se me presente, siempre recordándome lo valiente y esforzado que soy y que no pare de soñar que el límite siempre va a hacer el cielo.

A mi grupo de amigos Denny, Ivette, Kevin, Arley quienes demuestran que tenemos una segunda familia con ellos, donde he aprendido mucho de ellos a lo largo de la carrera.

Finalmente dedico este proyecto a mi mejor amigo a mi compañero de vida Robert a la persona que incondicionalmente está ahí aconsejándome y guiándome.

Jhon Toscano

AGRADECIMIENTO

Agradezco a Dios por ser mi guía y mi camino, en todo este trayecto me has demostrado que todo sacrificio vale la pena.

Agradezco grandemente a mi madre Anita por cada consejo y cada enseñanza que me dio desde muy niño.

A mi hermana Gaby y a su esposo Geovanny por su motivación para no decaer a lo largo de esta carrera y por estar conmigo incondicionalmente.

A mi tutor Ing. Andrés Reyes por su tiempo y paciencia y la dedicación que tiene para guiarme en el desarrollo del proyecto brindándome todo su conocimiento y experiencia.

A mis amigos con los que compartí buenos y malos momentos, y siempre nos ayudábamos mutuamente para salir bien en todas las materias, agradezco su compañía su amistad y sus consejos.

Jhon Toscano

ÍNDICE DE CONTENIDOS

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
RESUMEN.....	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcance	2
1.4 Marco Teórico	2
Internet de las cosas (<i>Internet of Things - IoT</i>).....	2
Sensores	3
Microcontrolador ESP32.....	3
Arduino IDE	4
<i>Hosting</i>	4
MySQL	5
2 METODOLOGÍA.....	5
3 RESULTADOS	7
3.1 Establecimiento de los requerimientos necesarios para el prototipo.....	8
3.2 Definición de los componentes de hardware y software	8
Característica	11
3.3 DISEÑO DEL PROTOTIPO.....	13
Comunicación entre microcontrolador y el sensor.....	14
Circuito Electrónico.....	15
Creación del sitio web.....	15
Creación de la base de datos en MySQL.....	15

Conexión entre el servidor web y el microcontrolador	16
Creación de la página web	17
Justificación de rangos de operación y alertas del prototipo	18
Diagramas de flujo.....	20
3.4 Implementación del prototipo.....	23
Elaboración de la placa electrónica.....	23
Montaje de los componentes electrónicos	25
Implementación de la caja del prototipo.....	25
Programación del módulo ESP32	26
3.5 PRUEBAS DE FUNCIONAMIENTO	29
4 CONCLUSIONES	35
5 RECOMENDACIONES.....	36
6 Referencias	36
7 ANEXOS.....	41

RESUMEN

Este proyecto de integración define la implementación de un prototipo de un sistema de mediciones y monitoreo de varios factores ambientales como la temperatura, presión atmosférica, humedad y calidad del aire. En el primer capítulo se muestran los componentes de hardware y software que se utilizarán en la implementación, tales como el sensor BME680, el microcontrolador ESP32 WROOM-32, la base de datos MySQL y el servidor web Apache. También se presentan los objetivos generales y específicos para cumplir con el desarrollo del trabajo.

En el segundo capítulo se presenta la metodología a usarse para cumplir con cada objetivo específico, mostrando el proceso a seguir de acuerdo con el modelo de cascada.

El tercer capítulo corresponde a los resultados donde se establece los requerimientos necesarios para el proyecto, aquí se definen los componentes de hardware y software en base a un análisis de sus características físicas, técnicas y eléctricas, así como la disponibilidad en el mercado; facilitando la implementación de la tecnología *IoT* al usar un microcontrolador con un módulo *WiFi* integrado y un sensor de parámetros climáticos. Adicional se detalla el procedimiento de implementación del sistema de medición ambiental, así como las pruebas de funcionamiento realizadas.

Por último, las secciones 4 y 5 contienen las conclusiones y recomendaciones en función del proyecto, mismas que ayudaran en el desarrollo de futuros proyectos.

PALABRAS CLAVE: ESP32, sensor BME680, plataformas *IoT*, base de datos.

ABSTRACT

This project defines the implementation of a prototype about environmental measurements and monitor system such as temperature, air pressure, humidity and air quality. In the first chapter, it's displayed the hardware and software components which be used in the project, these are the BME680 sensor, the microcontroller ESP32 WROOM-32, MySQL database, and Apache web server. Also, it's defined the general and specific objectives to accomplish the development of the job.

In the second chapter it's talked about the methodology to be used to get solved each specific objective, showing the process to follow in order to waterfall model.

In the third chapter it's stablished the requirements to the project, here it's defined the hardware and software components in base to the analysis made of each physical, technical and electrical features, the availability of the elements in the market; giving an advantage to the implementation of IoT technology by using a microcontroller with WiFi module integrated and an environmental sensor. Also, it's defined the implementation procedure of the environment measurement system and the tests done with the prototype.

Finally, in the chapter 4 and 5, the conclusions and recommendations of the project are shown, which will help in the development of future similar projects.

KEYWORDS: *ESP32, sensor BME680, IoT, database, web server.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El proyecto planteó el prototipo de un sistema que recibe mediciones de presión (hPA), temperatura (°C), humedad % y calidad de aire (Ppm), mediante el uso del sensor BME680, de gran precisión y bajo consumo de energía. Los datos son enviados al módulo ESP32, obteniendo alertas de cambios repentinos o fuera de lo normal con respecto a los valores definidos en los rangos permitidos.

El prototipo ofrece un enfoque integral para el monitoreo ambiental, proporcionando información valiosa para una variedad de usuarios, contribuyendo a la creación de entornos más eficientes y saludables. El cerebro de este sistema es un módulo ESP32 WROOM-32 de 38 pines, que incluye una interfaz *WiFi*, lo que permite tener conectividad dentro de una red inalámbrica. Para programarlo se utiliza el software Arduino IDE, donde se define algunos aspectos como el nombre de red (SSID), la clave de autenticación para que el microcontrolador tenga conectividad, y el intervalo de tiempo para recibir una nueva medición del sensor BME680.

Al conectar el módulo ESP32 a la red, las mediciones son enviadas hacia el servidor web usando el método *HTTP POST*, y a una base datos MySQL, ambos alojados en *webhost*, este *hosting* ofrece una suscripción gratuita con un espacio en disco hasta de 300 (MB). El *front-end* está desarrollado en *PHP*, y consta de dos páginas, en la primera o página principal se presenta la información de las mediciones en tiempo real, obtenidas en un intervalo de 10 (segundos). La segunda página muestra dos tablas que contienen los datos históricos de una semana y mediciones que presentaron un valor fuera de los rangos definidos.

Cuando se desea acceder a la interfaz web donde se presentan las mediciones en tiempo real y los datos históricos, lo puede realizar sin necesidad de estar conectado en la misma red, debido a que el servicio de *hosting* permite acceder desde cualquier lugar, incluso desde un equipo móvil, sea Tablet o celular, ya que la interfaz permite visualizar los datos en pantallas grandes o pequeñas. Para esto, únicamente se debe utilizar un equipo que disponga un navegador web, e ingresar a la URL creada en *webhost* para poder visualizar toda la información brindada por BME680.

1.1 Objetivo general

Implementar un prototipo de un sistema de monitoreo remoto basado en ESP32 y página web.

1.2 Objetivos específicos

- Determinar las necesidades esenciales del prototipo.
- Describir los elementos fundamentales de hardware y software.
- Diseñar el prototipo de monitoreo remoto y sus respectivas alarmas.
- Materializar la estructura y disposición de los componentes en una maqueta.

1.3 Alcance

El prototipo desarrollado muestra las mediciones de temperatura, humedad, presión y calidad del aire recolectadas en el medio ambiente por un sensor, seleccionado tras un análisis realizado entre varios modelos de sensores, desplegadas mediante una página web en el navegador de un equipo móvil o de un computador; adicionalmente, la página web presenta una alerta si hay mediciones fuera de los límites establecidos. El componente de hardware consta de un circuito impreso en una placa electrónica, alimentada mediante una interfaz mini USB, colocada en una caja para la protección del prototipo.

1.4 Marco Teórico

Internet de las cosas (*Internet of Things - IoT*)

El Internet de las cosas se refiere al conjunto de dispositivos conectados a través de tecnología inalámbrica para comunicarse entre sí y que toda la información recolectada se almacene en la nube [1].

Este sistema funciona con la recopilación y el intercambio de datos en tiempo real, consta de tres componentes [2]:

- El primer componente son los dispositivos inteligentes que cumplen con el rol de recopilar información, la cual se comunica a través de Internet hacia la

aplicación, como ejemplo tenemos a los sensores, rastreadores, alarmas de seguridad etc.

- El segundo es la aplicación de *IoT*, que es el conjunto de software y servicios que integra los datos recopilados por todos los dispositivos inteligentes, la tecnología implementa *machine learning*, que se encarga de analizar los datos y de tomar decisiones hacia el dispositivo *IoT* y este responde de manera eficiente e inteligente.
- El tercero es el terminal *IoT* o todos los dispositivos que funcionen de manera inteligente, pueden ser administrados a través de una interfaz gráfica, como ejemplo podemos tener una aplicación celular o un sitio web que se vincula con los sensores o todo dispositivo que se encarga de recolectar datos.

Sensores

Los sensores son dispositivos que detectan y responden a cambios en el medio ambiente, estos se utilizan para medir diversos fenómenos físicos y químicos como temperatura, presión, luz, humedad, movimiento y concentración de sustancias químicas, etc [3].

Los sensores están fabricados a partir de una variedad de materiales, incluidos semiconductores, metales, polímeros y cerámicas. El tipo de material utilizado depende del tipo de fenómeno que mide. Por ejemplo, los sensores de temperatura suelen estar hechos de semiconductores y los sensores de presión de metal [4].

Microcontrolador ESP32

ESP32-WROOM-32 es una placa de desarrollo de microcontrolador de 32 (*hosting*) fabricada por *Espressif Systems* muy versátil, ya que se puede utilizar para una amplia gama de aplicaciones, desde proyectos de Internet de las cosas (*IoT*), hasta robótica.

La plataforma *ESP32* permite desarrollar aplicaciones en diferentes lenguajes de programación, bibliotecas y diversos recursos. La placa tiene 38 pines, un módulo *WiFi* y un módulo *Bluetooth* integrado. Soporta la tecnología *WiFi* 802.11 b/g/n/ac y puede actuar como punto de acceso y como estación inalámbrica. Con respecto al *Bluetooth*, trabaja con la versión 5.0 y soporta para *Bluetooth LE*. [5]

Los 38 pines de la placa se distribuyen de la siguiente manera: 20 pines GPIO (*General Purpose In/Out*), 10 pines analógicos, 4 pines de comunicación y 4 pines de potencia. A continuación, se detalla los pines:

- **Pines GPIO:** Los pines se utilizan para controlar dispositivos externos, como actuadores, sensores y *displays*. Estos se pueden configurar como entradas o salidas digitales.
- **Pines analógicos:** Estos pines se utilizan para leer los datos analógicos recolectados por los sensores, los pines analógicos tienen una resolución de 12 (bits).
- **Pines de comunicación:** Se utilizan para conectar el ESP32 con otros dispositivos entre algunos están TXD0 salida de datos, RXD0 entrada de datos, I2C0_SDA salida de datos y I2C0_SCL salidas de datos de reloj.
- **Pines de potencia:** Los cuales alimentan al microcontrolador el pin VDD tiene un voltaje de entrada de 3.3 (V) a 5.0 (V) y el pin EN, encienden o apagan el ESP32 [6].

Arduino IDE

Arduino IDE es un entorno de desarrollo integrado (*IDE*) de código abierto que permite escribir, compilar y cargar proyectos en un microcontrolador, está basado en el lenguaje de programación C++. Consta de un editor de texto, un compilador, un cargador de programas y un conjunto de herramientas de ayuda.

Los editores de texto permiten a los usuarios escribir código en un entorno cómodo. El compilador convierte el código a un formato que la placa Arduino pueda entender. El cargador de programas graba el código en la placa Arduino [7].

Hosting

El alojamiento web, también conocido como *hosting*, es un servicio que proporciona a los usuarios espacio de almacenamiento en un servidor para alojar sus sitios web. El servidor es una computadora que está conectada a Internet y tiene el software y hardware necesarios para almacenar archivos web [8].

El alojamiento compartido implica alojar sitios web de varios usuarios en el mismo servidor. Esto significa que los recursos del servidor se comparten entre los usuarios. El alojamiento compartido es el más barato y mejor para sitios web pequeños con poco tráfico. Tiene el alojamiento dedicado implica alojar un único sitio web en un servidor dedicado. Esto significa que los recursos del servidor están disponibles exclusivamente para este sitio web [9].

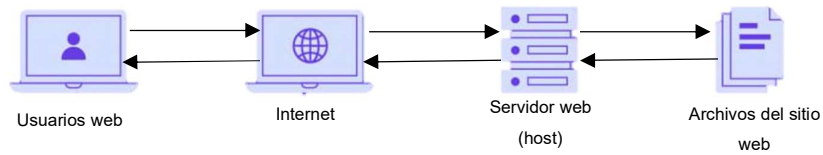


Figura 1.1 Proceso de Alojamiento Web

El alojamiento web es responsable de mantener el servidor en funcionamiento, protegerlo de ataques maliciosos y transferir su contenido (como texto, imágenes, etc.) desde el servidor a los navegadores de sus visitantes.

MySQL

Es un sistema de gestión de bases de datos relacionales de código abierto. Es uno de los sistemas de gestión de bases de datos más populares en todo el mundo y lo utilizan una amplia gama de aplicaciones y sitios web [10].

MySQL organiza la información en tablas con filas y columnas, siguiendo el modelo de base de datos relacional. Utiliza SQL (lenguaje de consulta estructurado) para consultar y manipular datos en la base de datos. Admite transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) para garantizar la integridad de los datos.

Es compatible con múltiples plataformas, incluidos sistemas operativos como Windows, Linux y macOS. Puede manejar datos y escalar para adaptarse a las crecientes necesidades de las aplicaciones [10].

2 METODOLOGÍA

El desarrollo de este proyecto está basado en un modelo de cascada (*Waterfall*), donde se definen en la Figura 2.1, empezando con los requerimientos, el diseño y la programación, la implementación del prototipo, las verificaciones y pruebas, y en caso de ser necesario el mantenimiento y las correcciones. Cada etapa está relacionada entre sí y debe llevarse a cabo en el orden mencionado.

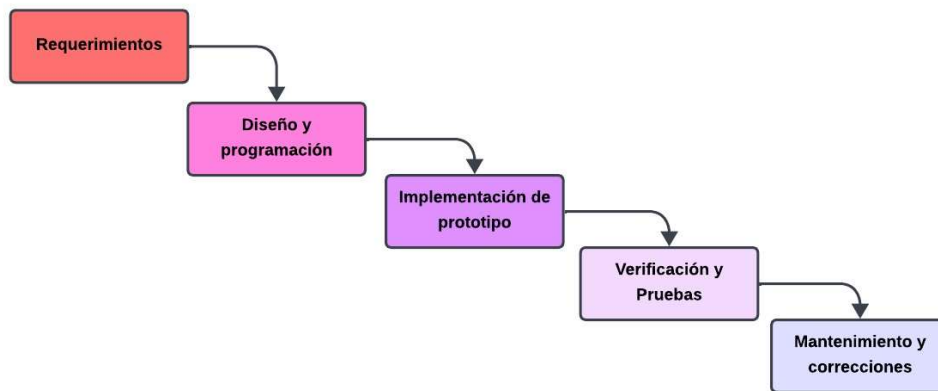


Figura 2.1 Estructura de la metodología del proyecto

Requerimientos: Etapa donde se analizan los requerimientos y los componentes de hardware y software que se utilizaron después de investigar las opciones existentes para el desarrollo del proyecto. En términos de hardware, se consideraron los componentes físicos del prototipo, según el tipo de sensor y tipo de microcontrolador. Además, para energizar el circuito se utilizará una toma eléctrica de 110 (V) AC a través de un adaptador a 5 (V) DC conectados al microcontrolador mediante un puerto micro USB.

Para el componente de software se consideró la facilidad en el desarrollo y la integración de los servicios en una sola herramienta, ya que se necesita un servidor web, y una base de datos que deben almacenarse en la nube para poder acceder a la información desde cualquier lugar del mundo, y puede almacenar las mediciones obtenidas mientras el sensor se encuentre activo.

Diseño y programación: Una vez definido e investigado los requerimientos, se plantea las herramientas para comenzar con el diseño del prototipo, el microcontrolador se encargará de obtener los datos que arroja el sensor, aquí se programará los intervalos entre cada medida, y la conexión entre el micro y la red *WiFi*.

En el servidor web, se programará la interfaz donde el usuario final podrá revisar la información y las alarmas, así como la conectividad y la obtención de la información con métodos POST, permitiendo almacenarla en la base de datos.

Implementación de prototipo: Para esta etapa se necesitó imprimir el circuito en una baquelita, para lo que se usó el *software* Proteus, considerando el tamaño de cada componente y la funcionalidad que el sensor debe tener en la placa.

En el momento de imprimir las pistas sobre la placa, se tomó en cuenta la distancia entre ellas para evitar tener un cortocircuito afectando el funcionamiento del prototipo, e incluso pudiendo dañar los componentes de este.

Para salvaguardarlo de cualquier daño que pueda tener por mala manipulación de quienes accederán, el circuito se alojará en una caja plástica, tendrá dos aberturas, una compuerta en la parte superior, que permitirá que el sensor sea visible y esté expuesto al medio ambiente para tomar muestras, y una abertura que servirá de acceso para el cable de alimentación del circuito.

Verificación y Pruebas: Una vez terminada la fase de implementación, las pruebas serán realizadas alimentando el circuito, asegurando que el mismo disponga de conectividad hacia la red inalámbrica, asegurando la recepción de los datos en la interfaz gráfica, donde se podrá seguir en vivo las mediciones de temperatura, humedad, presión, y calidad del aire. Se toma en cuenta también los parámetros considerados como alertas, que son mediciones fuera de los rangos mínimos y máximos establecidos.

Al realizar las pruebas se pueden modificar rangos solo para confirmar que la alerta se recibe y almacena correctamente en la base de datos.

Mantenimiento y correcciones: Para esta etapa únicamente serán consideradas las mejoras que se podrían dar a este proyecto a futuro, así como realizar ciertos ajustes y cambios en cables de conexión, y de hacer falta actualizar datos como *SSIDs* y claves de la red *WiFi*.

3 RESULTADOS

En esta sección del proyecto se presenta en detalle el establecimiento de los requerimientos necesarios para la implementación, los componentes como el microcontrolador ESP32 y el sensor que cumpla con lo estipulado en los objetivos y el alcance.

Se detalla el software utilizado para la programación del microcontrolador, así como el servidor web y la programación desarrollada en PHP para obtener la interfaz para el usuario final y la conectividad entre el componente de hardware y la base de datos. La información es visualizada en la página web de la plataforma usando un computador o un dispositivo móvil.

También se explica la implementación del prototipo, incluyendo la fabricación de la placa electrónica, montada en una estructura óptima para que, al realizar las pruebas, se pueda medir exactamente.

Para la definición de los parámetros que lanzaría una alerta, se realizó un breve estudio de las condiciones normales, mínimas y máximas de temperatura, humedad, presión atmosférica y calidad del aire de la ciudad de Quito.

3.1 Establecimiento de los requerimientos necesarios para el prototipo

Para el desarrollo del prototipo del sistema de monitoreo ambiental de temperatura, humedad, presión atmosférica y calidad de aire basado en el uso del microcontrolador ESP32 y mostrados en una página web, se definen los siguientes requerimientos para su correcto funcionamiento:

- La comunicación entre el microcontrolador y el servidor web y de base de datos será inalámbrica para evitar el uso de cables y tener acceso a Internet.
- El sistema debe tomar mediciones de humedad, la presión y la calidad del aire. Los datos se deben medir en intervalos regulares (configurables) y el rango de medición de cada sensor debe ser apropiado para el entorno en el que se encuentra.
- Una vez obtenidos los datos, serán enviados por medio del módulo *WiFi* de la placa hacia el servidor web, localizado en la nube, donde las mediciones serán presentadas al usuario final. Además, en la página web se mostrarán alertas en caso de tener mediciones por fuera de los rangos establecidos.
- El sistema debe almacenar la información obtenida en una base de datos en la nube, ya se conservarán disponibles y en línea para su análisis posterior, agregando la fecha y hora de medición.
- La página web debe mostrar tablas de los datos medidos en tiempo real, y tener una interfaz amigable y de fácil uso para el usuario final.
- El sistema debe soportar la conexión de múltiples usuarios a la página web, para este caso al ser un prototipo se considerará las condiciones de simultaneidad definidos por el proveedor del *hosting*.

3.2 Definición de los componentes de hardware y software

Selección del hardware

Este prototipo requiere el uso de un microcontrolador asociado a un sensor de medición de parámetros del medio ambiente, conectados mediante una red *WiFi*, para recolectar y mostrar la información en una página web, y almacenados en una base de datos para futuras revisiones.

Para elegir el sensor que se adapte más a este proyecto, se analizó varios aspectos, entre ellos el tener la mínima cantidad de elementos para disminuir el tamaño del componente y optimizar el uso de la placa.

Para la recolección de datos en el medio ambiente existen una variedad de sensores de temperatura, humedad, presión atmosférica y calidad del aire, sin embargo, como se mencionó anteriormente se ha realizado la comparación entre modelos que sean polifuncionales, es decir que brinden mediciones simultáneas de los parámetros deseados, para esto se ha escogido dos sensores BME680 Y BME280; cuyas características se presentan en Tabla 3.1.

El sensor BME680 es pequeño y de bajo consumo de energía, que puede medir diferentes parámetros ambientales, el dispositivo tiene variantes en sus rangos de medición como el de la temperatura de -40 a 85 grados Celsius un rango de humedad de 0 a 100%, un rango de presión de 300 a 1100 (hPa) y un rango de gas de 0 a 1000 (Ppm) [11].

Tabla 3.1 Comparación de los sensores de temperatura, humedad y presión atmosférica

Característica	BME680	BME280
Voltaje de alimentación	1.7 (V) a 3.6 (V)	1.7 (V) a 3.6 (V)
Pines de entrada y salida	10	7
Pines entrada analógicos	1	1
Puertos	I ² C, SPI	I ² C, SPI
Dimensión	20.3 x 2.6 x 1.4 (mm)	20.3 x 2.6 x 1.4 (mm)
Temperatura de operación	-40 (°C) a 85 (°C)	-40 (°C) a 85 (°C)
Porcentaje de humedad	0 % a 100 %	0 % a 100 %
Porcentaje de presión atmosférica	300 (hPa) a 1100 (hPa)	300 (hPa) a 1100 (hPa)
Consumo de corriente	0.5 (mA) a 1.2 (mA)	0.2 (mA) a 0.8 (mA)

El dispositivo tiene un consumo de corriente bajo, en modo de espera solo consume 0.002(mA) y en modo de medición de 0.2(mA).

La precisión del sensor es alta, para la temperatura cumple con ± 1 grado Celsius, en la humedad con $\pm 3\%$, la precisión del sensor de presión atmosférica es de ± 0.5 (hPa) y la

precisión del sensor de gas es de $\pm 20\%$. El sensor BME680 viene calibrado de fábrica, pero como mantenimiento para futuro se necesita recalibrar el sensor periódicamente para mantener la precisión al medir [11].

El BME280 es un sensor ambiental que mide únicamente temperatura y presión. Es un sensor pequeño y eficiente que se puede utilizar en diversas aplicaciones, como monitoreo del clima, navegación y medición de altitud [12].

El LM393 es un sensor de gas que mide la concentración de gas en el aire. Es simple y económico y se puede utilizar en una variedad de aplicaciones, como detección de fugas de gas, medición de la calidad del aire y monitoreo de la salud [13].

El BME680 es el más completo de los sensores. Proporciona mediciones de temperatura, presión atmosférica, humedad y calidad del aire. El BME280 es similar al BME680, pero no ofrece mediciones de la calidad del aire. El LM393 es un sensor de gas sencillo que puede medir la concentración de gas en el aire. La Tabla 3.2, representa las características de los sensores mencionados.

Tabla 3.2 Comparación de los sensores de calidad de aire de varios fabricantes.

Característica	LM393	MQ	CCS811	BME680
Voltaje de funcionamiento	5 (V)	5 (V)	2.4 (V)	1.7 (V) a 3.6 (V)
Corriente de funcionamiento	20 (mA)	10 (mA)	150 (μ A)	0.5 (mA) a 1.2 (mA)
Consumo de energía	100 (mW)	50 (mW)	3.6 (mW)	0.8 (mW) a 1.2 (mW)
Resistencia de carga	10 (k Ω)	10 (k Ω)	10 (k Ω)	10 (k Ω)
Detección de resistencia	Sí	Sí	Sí	Sí
Alcance de la concentración	100 (Ppm)	200 (Ppm)	1000 (Ppm)	1000 (Ppm)
Producción	Baja	Baja	Media	Media
Dimensiones	12.5 x 7.5 x 5.0 (mm)	5.0 x 3.0 x 2.0 (mm)	20.0 x 10.0 x 2.0 (mm)	20.3 x 2.6 x 1.4 (mm)
Pines analógicos	1	1	1	1

Por lo tanto, la mejor opción para la implementación es el sensor BME680, por incluir en un solo circuito las mediciones del medio ambiente requeridas en el prototipo, con un rango de medición más amplio que otros sensores, lo que lo hace adecuado para aplicaciones con mediciones extremas, con una amplia gama de aplicaciones, desde el monitoreo del clima hasta la calidad del aire interior y exterior.

El módulo *WiFi* está incorporado en el chip ESP32 WROOM, por lo tanto, no hace falta adicionar otro componente en el sistema, presenta velocidades de transmisión hasta de 150 (Mbps), lo cual facilita el uso y la implementación de equipos *IoT*.

Selección de Plataforma *Hosting* para visualización y almacenamiento de datos

El *hosting* donde se cargará el servidor web y la base de datos debe cumplir los requerimientos mínimos necesarios para mantener la información segura, sea siempre disponible y que sea de rápido acceso.

Para la elección se consideró características como la visualización de datos en tiempo real, el tipo de plan y las herramientas brindadas por el *hosting* para el desarrollo de la página web, además que todos los datos se vayan almacenando en la nube por un determinado tiempo.

Adicional a los requerimientos mínimos, en este proyecto se consideró la facilidad para poder realizar pruebas y obtener espacio de almacenamiento y conexiones limitadas sin costo, sin embargo, para un futuro proyecto o implementación de este se podrá elegir una plataforma con costo.

La muestra una comparación entre 4 plataformas de *hosting* ofrecidas en el internet, con las principales características para realizar el análisis y obtener la mejor solución para implementar el prototipo

Tabla 3.3 Comparación de Hosting y Página Web

Característica	000Webhost	Wix	Network Solutions	Bluehost
Tipos de alertas	Correo electrónico, SMS	Correo electrónico	Correo electrónico, SMS	Correo electrónico, SMS
Tipo de plan	Gratuito, básico, profesional	Gratuito, básico, ilimitado	Básico, estándar, avanzado	Básico, estándar, plus
Cantidad de dispositivos	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Duración en la nube	30 días	30 días	30 días	30 días
Base de datos	MySQL	MySQL	MySQL, PostgreSQL	MySQL, PostgreSQL
Página web	Sí	Sí	Sí	Sí

000Webhost.com es un servicio de *hosting* con planes gratuitos que ofrece alojamiento de sitios web básico. Este plan incluye 500 (MB) de espacio de almacenamiento, un motor de base de datos MySQL, un dominio gratuito y un ancho de banda limitado. El

plan básico tiene un costo mensual de 5 dólares e incluye (1GB) de espacio de almacenamiento, una base de datos MySQL, un dominio gratuito y ancho de banda ilimitado. El plan profesional cuesta 10 dólares por mes e incluye 5 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado, si se requiere también se puede contratar soporte técnico con un costo adicional [14].

Wix es un servicio de creación de sitios web que ofrece alojamiento web gratuito para sitios web creados con su plataforma, sin necesidad de realizar la programación en PHP. El plan gratuito incluye 500 (MB) de espacio de almacenamiento, un dominio personalizado y ancho de banda ilimitado. El plan básico cuesta 14 dólares por mes e incluye 3(GB) de espacio de almacenamiento, un dominio personalizado y ancho de banda ilimitado. El plan ilimitado cuesta 35 dólares por mes e incluye espacio de almacenamiento ilimitado, dominio personalizado y ancho de banda ilimitado. Algunas ventajas de este hosting es que se puede crear un sitio web con conocimientos básicos de programación, sin embargo, no permite tener mayor personalización al momento de diseñar la página web [15].

Network Solutions es un proveedor de servicios de Internet que ofrece una variedad de servicios, incluido alojamiento web. El plan básico de alojamiento web cuesta 6,99 dólares por mes e incluye 1 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado. El plan estándar cuesta 11,99 dólares por mes e incluye 5 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado. El plan avanzado cuesta 21,99 dólares por mes e incluye 10 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado [16].

Bluehost es uno de los proveedores de servicios de alojamiento web más populares del mundo. Ofrece varios planes de alojamiento web, desde básicos hasta de alto rendimiento. El plan básico cuesta 2,95 dólares por mes e incluye 50 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado. El plan estándar cuesta 5,95 dólares por mes e incluye 100 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado. El plan plus cuesta 11,95 dólares por mes e incluye 200 (GB) de espacio de almacenamiento, una base de datos MySQL, un dominio personalizado y ancho de banda ilimitado [17].

Para este caso se elige el *hosting* 000Webhost, ya que ofrece el mejor plan gratuito que los de otros proveedores, el panel de control es fácil de usar y navegar, así se puede ya

crear un sitio web desde cero y con conocimientos mínimos por su interfaz amigable. La creación y montaje en la plataforma escogida se explicará más adelante.

3.3 DISEÑO DEL PROTOTIPO

Esquema general del proyecto

La Figura 3.1 muestra el esquema del prototipo implementado por un sensor de parámetros ambientales, cuya información recolecta el microcontrolador ESP32 y envía a una base de datos en la nube, para que se presente en una página web y se almacena durante un tiempo, lo que permite acceder fácilmente a las mediciones para analizar los datos obtenidos.

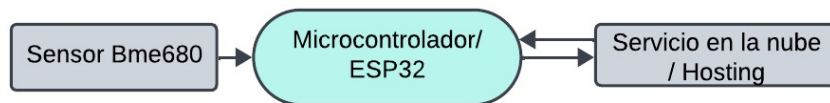


Figura 3.1. Esquema Centralizado

En la Figura 3.2. se detallan los componentes que forman parte del prototipo, así como la interacción entre ellos, desde que el sensor toma la medición, que llega al microcontrolador, en donde se transforma en una cadena de caracteres para ser enviada mediante una conexión *WiFi* a la base de datos, los cuales se almacenan en la tabla creada, para que el servidor web realice la consulta y muestre los datos en tiempo real en la página, marcando una alerta en caso de obtener una medición fuera de los parámetros establecidos.

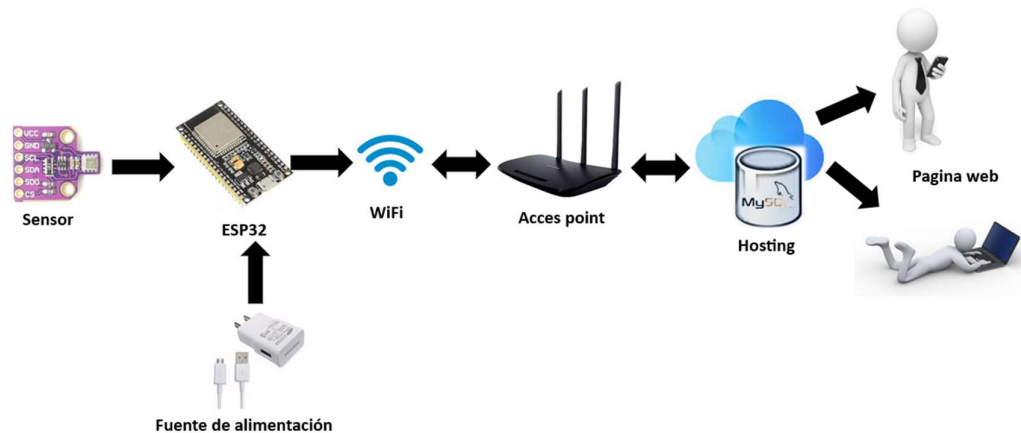


Figura 3.2. Esquema general del funcionamiento

Comunicación entre microcontrolador y el sensor

El microcontrolador ESP32 WROOM32 maneja 4 tipos de comunicación serial que permite comunicarse con otros dispositivos electrónicos; estos protocolos de comunicación son: I2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*), CAN (*controller area network*) y PWM (*Pulse Width Modulation*) [18].

En el caso del sensor de calidad ambiental BME680 maneja I2C y SPI para la comunicación. Para este proyecto se utiliza I2C, al ser un protocolo de comunicación serial más utiliza; sin embargo, no se descarta el uso de SPI para futuros casos, ya que la implementación no varía en mayor grado [6].

Es importante tener los diagramas de distribución de pines del microcontrolador y del sensor para estar seguros de las conexiones que se realizan al momento de diseñar el circuito para la comunicación I2C. En la Figura 3.3, se observa que los pines SCL y SDA del microcontrolador ESP32 WROOM32 corresponde al pin D22 (GPIO 22) y D21 (GPIO 21) respectivamente; mientras que en el sensor BME680 los pines vienen escritos, como se observa en la Figura 3.4

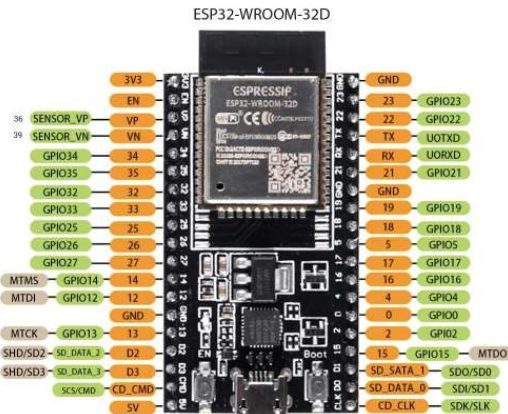


Figura 3.3. Esquema del microcontrolador

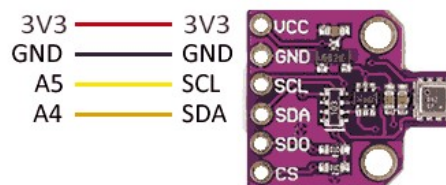


Figura 3.4. Esquema sensor BME680

Circuito Electrónico

Para empezar con el diseño, se realiza un diagrama con los componentes necesarios, en este caso el microcontrolador ESP32 y el sensor BME680 con sus respectivos diagramas de pines, para esto se usa el programa Proteus, el cual permite simular, diseñar y obtener el diagrama para la placa impresa del circuito *PCB (Printed Circuit Board)*. El diagrama del circuito prototipo se puede observar en la Figura 3.5.

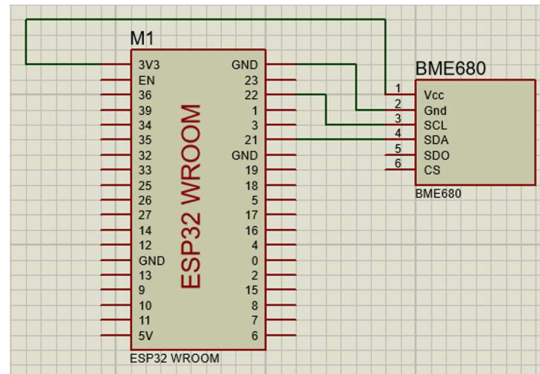


Figura 3.5. Prototipo creado en Proteus

Creación del sitio web

Para utilizar el hosting 000webhost, es necesario iniciar con la creación de la cuenta, seleccionando la opción gratuita, la cual permite tener un almacenamiento de 500MB, una base de datos MySQL, y adicional se genera automáticamente varios archivos útiles para la página web, como el *index.php*.

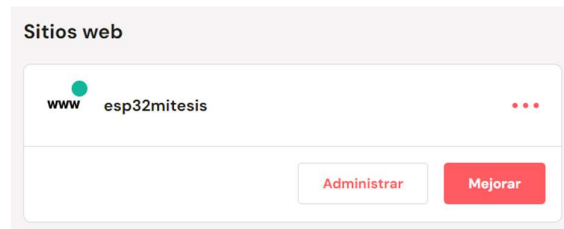


Figura 3.6 Hosting

Creación de la base de datos en MySQL

La creación de la base de datos dentro del sitio web creado se realiza usando la interfaz de administración *phpMyAdmin*, donde se genera una tabla con los campos id, temperatura, humedad, presión, calidad y fecha actual. Como se muestra en la Figura 3.6

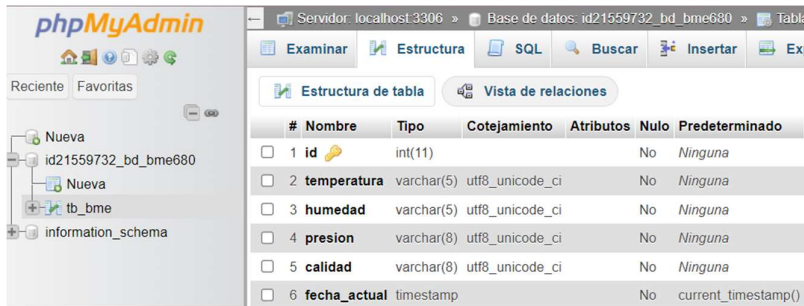


Figura 3.7 Base de datos Mysql

La base de datos almacenará la información obtenida mediante la conexión al microcontrolador ESP32, para lo que no hará falta que estén en la misma red local, ya que la ventaja del hosting es acceder a la información desde cualquier lugar.

Conexión entre el servidor web y el microcontrolador

Para obtener la información desde el microcontrolador y guardarla en la base de datos, se recurre al método http post, el cual permite enviar los datos de forma transparente para el usuario, como se muestra en la Figura 3.7, se tomará individualmente los datos de temperatura, humedad, presión atmosférica y calidad del aire, adicional se crea una variable donde se carga la fecha en la que se está obteniendo el registro desde del controlador.

```

if(isset($_POST['temperatura'])) {
    $temperatura = $_POST['temperatura'];
    echo " Temperatura : ".$temperatura;
}

if(isset($_POST['humedad'])) {
    $humedad = $_POST['humedad'];
    echo " humedad : ".$humedad;
}

if(isset($_POST['presion'])) {
    $presion = $_POST['presion'];
    echo " presion : ".$presion;
}

if(isset($_POST['calidad'])) {
    $calidad = $_POST['calidad'];
    echo " calidad : ".$calidad;
    date_default_timezone_set('america/bogota');
    $fecha_actual = date("Y-m-d H:i:s");
}

```

Figura 3.7 Método http Post

Una vez obtenida la información, se carga en la base de datos mediante la función *insert*, en el mismo orden en el que los componentes de la tabla fueron creados, la Figura 3.8. muestra el código utilizado.

```

$con consulta = "INSERT INTO tb_bme(temperatura, humedad, presion, calidad, fecha_actual) VALUES ('$temperatura', '$humedad', '$presion', '$calidad', '$fecha_actual')";
$resultado = mysqli_query($con, $consulta);

```

Figura 3.8 Función *insert*

Creación de la página web

La programación en *PHP* para generar la página web se realiza en el archivo *index.php*, donde se definen los siguientes parámetros:

- **Conexión a la base de datos:** la Figura 3.9, muestra la información necesaria para conectarse a la base, el nombre, el usuario, la clave y el nombre del servidor.

```
$user = "id21559732_jata";  
$pass = "2022J@ta";  
$server = "localhost";  
$db = "id21559732_bd_bme680";  
$con = mysqli_connect($server, $user, $pass, $db);
```

Figura 3.9 Conexión a MySQL

- **Rangos para alertas:** la Figura 3.10 muestra los rangos mínimos y máximos para las mediciones de temperatura, humedad, presión y calidad del aire, para cuando se obtenga un valor fuera de estos límites, se genere la alerta que se mostrará en la página.

```
$temnor = 20;  
$tem_min = 7; /*variables globales de temperatura*/  
$tem_max = 25;  
$hum_min = 30;  
$hum_max = 80;  
$pres_min = 700;  
$pres_max = 1025;  
$ca_min = 0;  
$ca_max = 50
```

Figura 3.10. Rangos establecidos

- **Parámetros de diseño de cada tabla:** los parámetros mostrados en la Figura 3.11, son los que definen el tamaño, el margen, el color de fondo, los bordes, y sombras de cada estructura.

```
table {  
width: 50%;  
margin: 20px auto;  
border-collapse: collapse;  
background-color: #fff;  
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
```

Figura 3.11. Diseño de tablas

- **Impresión de temperatura en la tabla:** La Figura 3.12, muestra el código utilizado para ilustrar en la página web la última medición obtenida y almacenada en la base de datos, junto a un mensaje de temperatura normal, si por el contrario

se obtiene un valor fuera de los rangos establecidos, el dato se imprimirá en rojo con un mensaje de alerta. Para observar todo el código fuente se encuentran en el anexo III.

```

<div>
  <table>
    <tr>
      <th>Temperatura</th>
    </tr>
  </table>
  <?php
  $sql = "SELECT temperatura FROM tb_bme ORDER by id DESC Limit 1";
  $result = mysqli_query($con, $sql);
  while ($mostrar = mysqli_fetch_array($result)) {
  >
  <?php
  if ($mostrar['temperatura'] < $tem_min || $mostrar['temperatura'] > $tem_max) {
    echo 'tr><td style="color: red;">¡Alerta! Temperatura fuera del rango Normal : ' .
      $mostrar['temperatura'] . ' °C</td></tr>';
  }
  } else {
    echo 'tr><td>Temperatura normal: ' . $mostrar['temperatura'] . ' °C</td></tr>';
  }
  >
  <?php
  }
  >
</table>
</div>

```

Figura 3.12 Impresión de temperatura

Los datos se mostrarán como en la Figura 3.13, los cuales pueden ser visualizados usando un computador o un dispositivo móvil, para mostrar las mediciones de humedad, presión y calidad del aire se utilizó el mismo código mostrado en la Figura 3.12.

Fecha de Medición	2024-01-17 18:34:30
Temperatura	Temperatura normal: 21.62°C
Humedad	Humedad normal: 52.11%
Presión	Presion normal: 727.18hPa
Calidad del Aire	Calidad de Aire normal: 31.83ppm

Figura 3.13. Página web.

Justificación de rangos de operación y alertas del prototipo

Los parámetros establecidos para este proyecto se tomaron según el estudio y mediciones reales en Quito, y son promedios mensualmente tomados en varios sectores de la ciudad, estas mediciones se realizaron todos los días en varias horas [19].

En la Tabla 3.4, se muestra los datos de las variaciones de temperatura de la ciudad durante el año 2023. Para el sensor se tomó en cuenta la temperatura mínima que fueron en julio y agosto y fue de 8 (°C) y como máxima de los meses de enero, febrero y diciembre y su pico fue de 22 (°C).

Tabla 3.4. Rango de Temperatura en la ciudad de Quito

Mes	Máxima (°C)	Mínima (°C)
Enero	22	9
Febrero	22	10
Marzo	21	10
Abril	21	10
Mayo	21	10
Junio	21	9
Julio	21	8
Agosto	21	8
Septiembre	21	9
Octubre	21	10
Noviembre	21	10
Diciembre	22	9

Los niveles de humedad en Quito como se muestra en la Tabla 3.5, son altos durante todo el año, siendo los meses más húmedos generalmente enero, febrero y marzo, mientras que los meses con menor humedad son generalmente julio y agosto. Para el sensor se eligió el rango de humedad del mes de junio que fue la más baja con un 40% y el mes más alto febrero con 85% [20].

Tabla 3.5. Intervalos de humedad en la ciudad de Quito

Mes	Humedad Relativa %
Enero	70 – 85
Febrero	75 – 85
Marzo	70 – 80
Abril	70 – 80
Mayo	70 – 80
Junio	40 – 50
Julio	65 – 75
Agosto	65 – 75
Septiembre	65 – 75
Octubre	70 – 80
Noviembre	70 – 80
Diciembre	75 – 85

La presión atmosférica en Quito es relativamente baja debido a la altitud, esta varía poco a lo largo del año, aunque puede ser más pronunciada en las zonas aledañas [21]. La unidad de medida es el hectopascal (hPa), y de acuerdo con las variaciones anuales de presión atmosférica de la ciudad, se muestra en la Tabla 3.6, se estableció un mínimo de 540 (hPa) y un máximo de 730 (hPa). [22].

Tabla 3.6. Variaciones Presión atmosférica en la Ciudad de Quito

Mes	Presión Atmosférica (hPa)
Enero	540 – 550
Febrero	540 – 550
Marzo	540 – 550
Abril	540 – 728
Mayo	540 – 550
Junio	540 – 550
Julio	540 – 550
Agosto	540 – 726
Septiembre	540 – 550
Octubre	540 – 550
Noviembre	550 – 730
Diciembre	540 – 730

Los rangos consultados son promedios generales y pueden variar significativamente de un día a otro debido a factores como el clima, la actividad humana o los incendios forestales como se muestra en la Tabla 3.7, las partes por millón (ppm) es una unidad que se utiliza para medir el volumen ocupado por una pequeña cantidad de un elemento. Se utiliza una ppm para medir la calidad del aire. Para el intervalo establecido en la programación se tomó como referencia los meses de abril y agosto donde muestran que la calidad mínima de aire es de 5 (ppm) y va aumentando su contaminación hasta 50 (ppm), donde la medición estaría dentro de los parámetros normales para la ciudad [23].

Tabla 3.7. Rango de la calidad de aire en la ciudad de Quito.

Mes	(Ppm)
Enero	25- 50
Febrero	25- 50
Marzo	25- 50
Abril	5- 50
Mayo	25- 50
Junio	25- 50
Julio	25- 50
Agosto	5- 50
Septiembre	25- 50
Octubre	25- 50
Noviembre	25- 50
Diciembre	25- 50

Diagramas de flujo

El diagrama de proceso de la Figura 3.14, es la base para la codificación de la placa ESP32 donde se la programó como una estación *WiFi*, para luego establecer la conexión inalámbrica a una red declarando su SSID y su clave, se procede a declarar las variables de lectura del sensor y se corrobora que el sensor esté en correcto funcionamiento.

Por último, se muestra la Figura 3.15, donde se configuró la conexión a la base de datos de *MySQL*, se realiza una consulta a los datos del sensor y se obtiene el registro más reciente, donde se ejecuta una serie de condicionales para mostrar los datos normales o si es el caso se mostrarán las alertas.

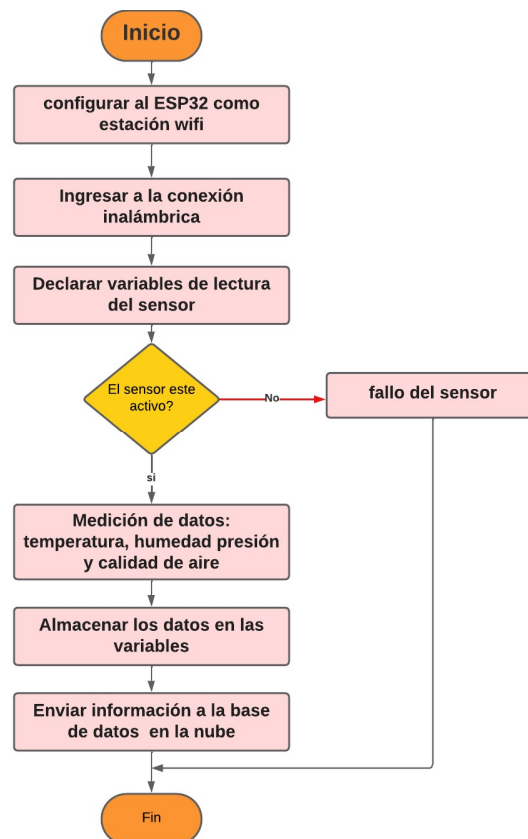


Figura 3.14. Diagrama de flujo de la placa ESP32.

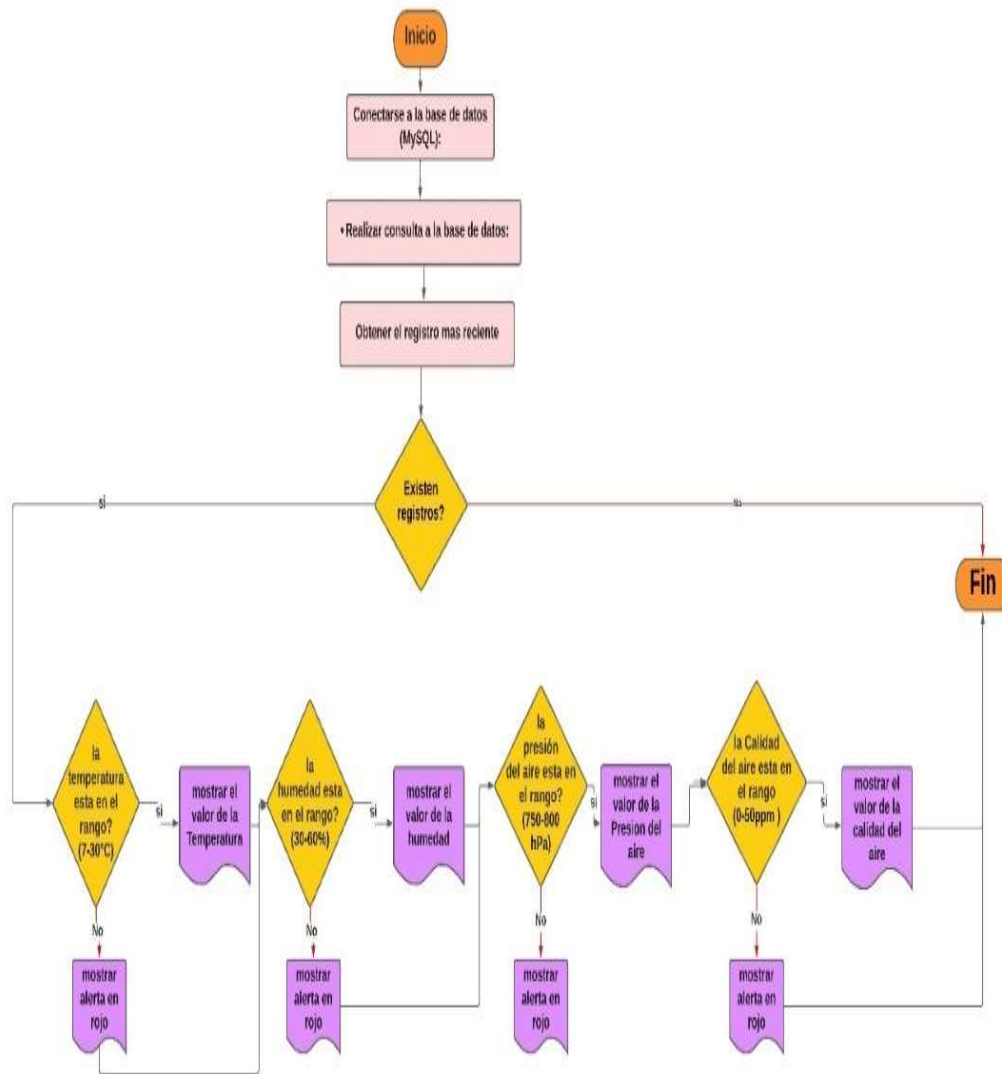


Figura 3.15. Diagrama de flujo de la página web.

En la Figura 3.16, se muestra la conexión a la base de datos *MySQL*, donde se hace la consulta de los valores de temperatura, humedad, presión y calidad de aire y respectivamente se guardarán las variables que tendrán los mismos nombres ya mencionados, para esto se tiene un proceso de registro, si el registro de las variables es exitoso se mostrará un mensaje de Registros Ok caso contrario se mostrará un mensaje con falla de Registros.

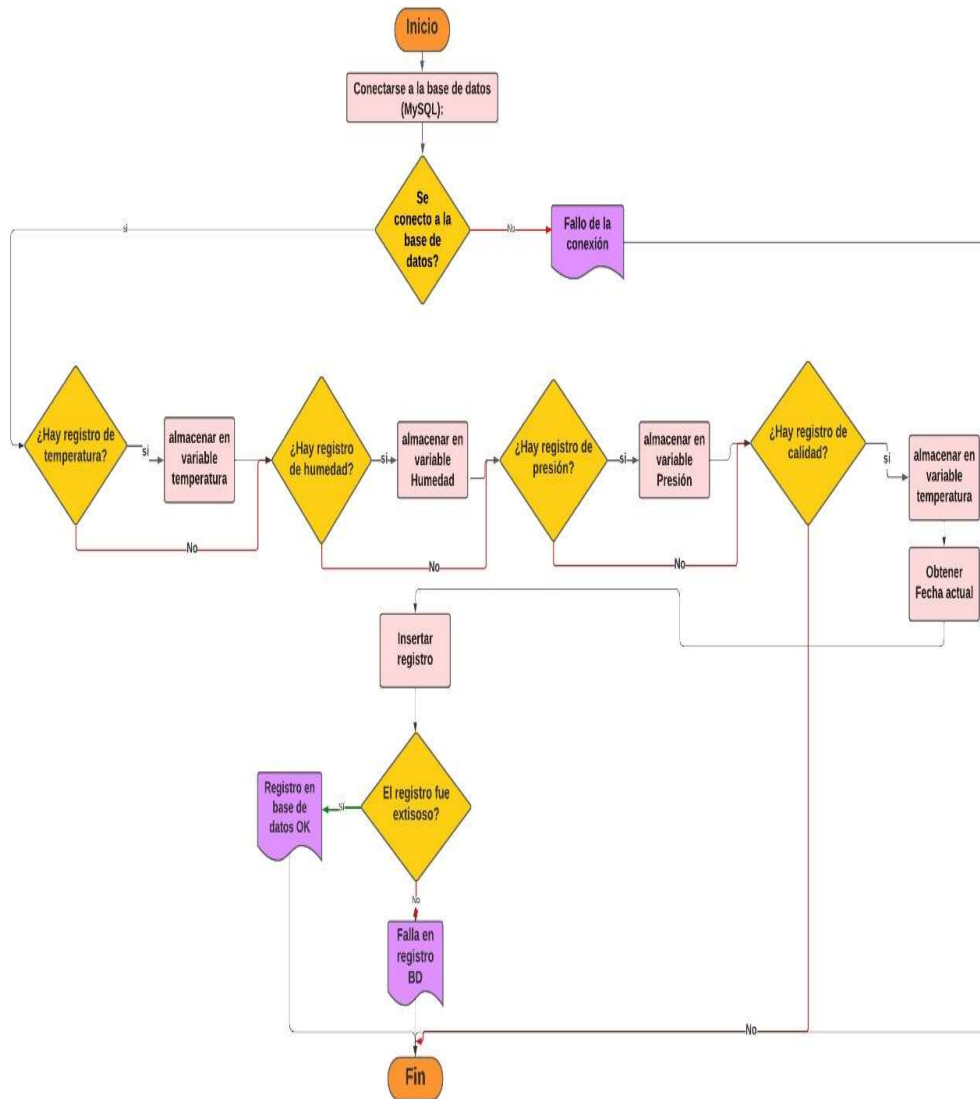


Figura 3.16 Diagrama de Flujo función Post

3.4 Implementación del prototipo

Elaboración de la placa electrónica

Para la elaboración del circuito impreso, se utilizó el programa Proteus, el cual brinda la facilidad de obtener la placa impresa que se muestra en la Figura 3.17, donde muestra las vías y la ubicación de cada componente, para este caso del sensor BME680 y del microcontrolador ESP32 WROOM

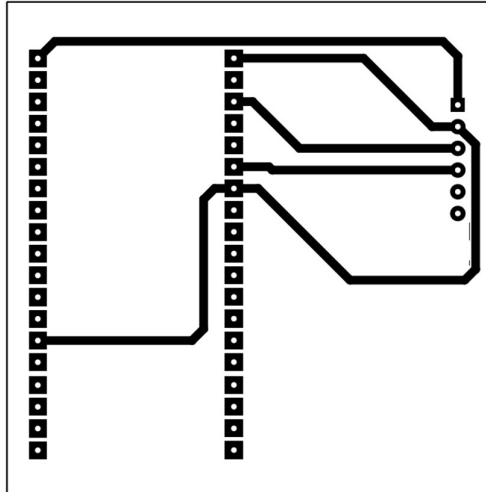


Figura 3.17. Placa impresa del circuito.

Los materiales que se utilizaron para realizar la placa PBC mostrada en la Figura 3.18, fueron: estaño, lija de agua, papel para estampado, estaño, ácido férrico una baquelita de 10 x 10 (cm) de tamaño un caudín y un multímetro. Se aplicó el método del planchado para la impresión sobre la placa.

Una vez que las vías están totalmente impresas, luego se enfría en agua y se introduce en el ácido férrico hasta que la baquelita solo tenga las vías de cobre, se sueldan todos los componentes en su sitio correspondiente, manteniendo el cuidado para no unir más de dos pines y tener un circuito en corto al conectar, lo que puede ocasionar lecturas falsas e incluso el daño de los elementos.

Al finalizar con el multímetro se valida la continuidad entre el circuito y entre pines para confirmar que no haya un corto.



Figura 3.18. Placa PBC final

Montaje de los componentes electrónicos

En la Figura 3.19 se observa el prototipo con todos sus componentes electrónicos ya soldados a la baquelita y fijos, la misma que será instalada más adelante en una estructura de protección.



Figura 3.19. Componentes soldados.

Implementación de la caja del prototipo

Es necesario disponer de una caja donde se almacenará la placa, lo cual evita que el mismo se encuentre expuesto y pueda sufrir daños por mala manipulación.

Esta caja dispondrá de dos orificios, el primero será para que el circuito pueda ser energizado mediante su conexión con un cable micro *USB*, y el segundo orificio para que el sensor BME quede expuesto al msmedio ambiente y permita obtener mediciones precisas durante su funcionamiento. Adicional el material de la caja no es aislante de ondas de radiofrecuencia para evitar problemas de conexión entre el microcontrolador y el *router* o el *Access point*.

Una vez que la parte electrónica está realizada se procede a fijarla en la caja de protección en la cual se realizó orificios como se muestra en la Figura 3.20, uno para la salida de cable de poder y otro para que el sensor quede descubierto al momento de recolectar la información.



Figura 3.20. Prototipo en la caja de protección.

Después de colocar todo se cierra la caja de protección y se obtiene ya el prototipo final como se muestra en la Figura 3.21.



Figura 3.21. Prototipo final.

Programación del módulo ESP32

Todas las librerías de la configuración del ESP32 se mostrarán en la Figura 3.22, cada una de ellas cumple una función específica y brinda las funciones necesarias para el

desarrollo del proyecto, la librería *Wire.h* permite la comunicación serial entre dispositivos a través de I2C, la librería *SPI.h* permite la comunicación sincrónica y la transferencia de datos entre el microcontrolador y los otros dispositivos, las librerías *Adafruit_Sensor.h* y *Adafruit_BME680.h* tiene las funciones para el uso del sensor BME680, las librerías *Wifi.h* y *WiFiClient.h* sirve para la conexión inalámbrica y los protocolos de comunicación TCP/IP, y la librería *HTTPClient.h* permite realizar solicitudes Get o post, las cuales serán usadas para enviar datos hacia el servidor web.

```
#include <Wire.h> //permite comunicarse con dispositivos por bus I2C
#include <SPI.h> //protocolo de comunicacion sincronica transferir datos de un microcontrolador a otros dispositivos
#include <Adafruit_Sensor.h> //librerias para el sensor
#include "Adafruit_BME680.h" //librerias para generar etiquetas
#include <WiFi.h> //funciona como AP o como una propia red wifi
#include <HTTPClient.h> //hacer solicitudes get o post enciar datos a un servidor web.
#include <WiFiClient.h> //establecer conexiones con protocolo TCP/Ip permite conecciones inalamblicas
//ingresar a la red para tener comunicacion inalambrica
const char* ssid ="SOLTEC"; //CAMPUS_EPN //identificador del router
const char* password ="JT1719080499"; //politecnica** //una contraseña
```

Figura 3.22. Librerías ESP32

Luego, se definen las variables en donde se almacenarán temporalmente los valores medidos desde el sensor. La Figura 3.23 muestra las variables de temperatura, humedad, presión y calidad, y están declaradas como *float*, ya que pueden tener números decimales.

```
Adafruit_BME680 bme; //Declaracion de un objeto llamado bme para obtener las lecturas del sensor
// Variables para lectura del BME680
float temperatura;
float humedad;
float presion;
float calidad;
```

Figura 3.23. Variables del sensor.

La Figura 3.24 muestra los parámetros de la función *setup*, donde se carga la configuración para el correcto funcionamiento de los componentes. La velocidad de transmisión de datos para la comunicación serial entre el computador desde donde se programará el microcontrolador y el circuito está definida en 115200 baudios. También se define el modo de operación que tendrá el módulo *WiFi* del ESP32, el cual en este caso será una estación, ya que el punto de acceso será el *router* del lugar donde se realizarán las pruebas. Se definen las variables *ssid* y *password* de la red inalámbrica. En este punto se inicia con el primer lazo para realizar la autenticación con el *router*.

```

void setup () {
  Serial.begin(115200); //velocidad de transmision del puerto serial en baudios
  WiFi.mode(WIFI_AP_STA);// Configurar el dispositivo como estación WiFi

  WiFi.begin(ssid, password);//inicia el proceso de conexion a la red WiFi
  while (WiFi.status() != WL_CONNECTED) { //Bucle hasta que se conecte a la red WiFi
    delay(1000); //evita que el bucle se ejecute demasiado rapido se introduce un retraso de 1 segundo p
    Serial.println("Conectando a red Wifi.."); //se muestra un mensaje indicando que esta en proceso de conexion
  }
}

```

Figura 3.24. Función *Setup*.

En la Figura 3.25, se observa la forma de obtener las mediciones desde el sensor, para la temperatura se define la obtención de 8 muestras, las cuales se promediarán y enviará al microcontrolador este resultado, para la humedad serán 2 muestras, para la presión 4 muestras. Para la calidad del aire por el contrario se define una operación distinta, donde primero se define un valor del filtro de 3, que significa que hará un promedio entre 3 muestras, y con la función *setGasHeater* se configura el calentador del gas en 320° y el ciclo de calentamiento de 150(ms), estos valores son configurados en base a lo estipulado en el *datasheet* del dispositivo, para la obtención de medidas precisas.

```

// configurar sobre-muestreo para la temperatura
bme.setTemperatureOversampling(BME680_OS_8X);//tomará 8 muestras de temperatura y promediará los resultados para proporcionar una lectura más precisa
bme.setHumidityOversampling(BME680_OS_2X);//se tomara 2 muestras
bme.setPressureOversampling(BME680_OS_4X);//4 muestras
bme.setIIRFilterSize(BME680_FILTER_SIZE_3);// indica que se está configurando el tamaño del filtro en 3. lo cual se hace un promedio entre los tres y da una lectura filtrada
bme.setGasHeater(320, 150); // configurar el calentador de gas donde los parametros son 320°C y 150 milisegundos de ciclo de calentamiento //diseño e implementación

```

Figura 3.25. Muestras tomadas por el sensor

Para continuar con el detalle del programa desarrollado en Arduino IDE, en la Figura 3.26, se muestra la función *EnvioDatos*, donde primero se realiza una validación si se encuentra conectado a la red *WiFi*, se arma una cadena de caracteres con la temperatura, la humedad, la presión y la calidad del aire, en la variable *datos_a_enviar*.

Con la función *http.begin* se llama al servidor web previamente configurado, se realiza la siguiente condicional donde en caso de que haya datos para enviar al servidor, y el servidor envíe una respuesta con código 200, que significa que la conexión se realizó satisfactoriamente, se envía la variable *cuerpo respuesta*.

```

// rutina de envio de datos por POST
void EnvioDatos(){
  if (WiFi.status() == WL_CONNECTED){
    HTTPClient http; // creo el objeto http
    String datos_a_enviar = "temperatura=" + String(temperatura) + "&humedad=" + String(humedad) + "&presion=" + String(presion) + "&calidad=" + String(calidad);

    http.begin("https://esp32mitesis.000webhostapp.com/EspPost.php");//servidor donde se va a reflejar la informacion enviada desde esp32
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");// se define texto plano.

    int codigo_respuesta = http.POST(datos_a_enviar); // variable tipo int para almacenar la respuesta
    //condicional para tomar captura de los datos a enviar al servidor web
    if (codigo_respuesta>0){
      Serial.println("Código HTTP: " + String(codigo_respuesta));
      if (codigo_respuesta == 200){ //condicional para tomar respuestas en caso de obtener un código 200 = exitoso.
        String cuerpo_respuesta = http.getString(); //almacenar el string obtenido en la variable cuerpo_respuesta
        Serial.println("El servidor respondió: "); //mostrar en consola
        Serial.println(cuerpo_respuesta); //mostrar en consola los valores a enviar
      }
    }
  }
}

```

Figura 3.26. Comunicación de módulo con el hosting

La función `LecturaBME680`, de la Figura 3.27, recolecta la información con los parámetros definidos en `setup`, aquí se abre la conexión con el sensor, y se emite la primera condición, donde si no existe respuesta se mostrará en la pantalla serial que el inicio de la lectura ha fallado.

En el segundo condicional, si la conexión termina anticipadamente, también se generará un mensaje de error, y se cerrará la conexión.

Una vez iniciada la conexión se empieza a guardar las mediciones de temperatura, presión, humedad y calidad del gas, en las variables antes declaradas.

```
void LecturaBME680(){
  // Empezar la lectura de datos del BME680
  unsigned long endTime = bme.beginReading();
  if (endTime == 0) {
    Serial.println(F("Failed to begin reading :("));
    return;
  }
  if (!bme.endReading()) {
    Serial.println(F("Failed to complete reading :("));
    return;
  } //condicional en caso de fallo en la lectura del sensor
  temperatura = bme.temperature; //variable temperatura para almacenar valor recibido por el sensor
  presion = bme.pressure / 100.0; //variable presion para almacenar valor recibido por el sensor
  humedad = bme.humidity; //variable humedad para almacenar valor recibido por el sensor
  calidad = bme.gas_resistance / 1000.0; //variable calidad para almacenar valor recibido por el sensor
}
```

Figura 3.27. Recolección de datos.

En la **Figura 3.28**, se muestra el final de la conexión http luego de haber enviado los datos, y el retraso de 10000(ms), lo que significa que se enviará una nueva medición al servidor web después de 10 (segundos). Se puede visualizar todo el código fuente en el Anexo III.

```
    http.end(); // liberar recursos
  } else {
    Serial.println("Error en la conexión WIFI");
  }
  delay(10000); //espera 10s
```

Figura 3.28. Tiempo de envío y liberación de recursos.

3.5 PRUEBAS DE FUNCIONAMIENTO

Una vez finalizado el montaje del sistema sobre la placa electrónica y en la caja de protección, se conecta el cable de alimentación para realizar la primera prueba, con la cual, mediante el uso de la consola de Arduino IDE, se valida que el microcontrolador se conecte a la red inalámbrica, adquiera una IP, como se muestra en la Figura 3.29, en este caso es la 192.168.100.91.

Una vez establecida la conexión *WiFi*, se obtiene en la consola el código HTTP 200, que indica que la conexión con el servidor web fue exitosa, se recibe el mensaje establecido en la página *EspPost.php* (Estación meteorológica) y se imprime los valores medidos por el sensor, primero temperatura, luego humedad, presión, y calidad del aire.

```

wlan0: <----->
Conectando a red Wifi..
Conectando a red Wifi..
Conectando a red Wifi..
IP Address del ESP32: 192.168.100.91

Código HTTP: 200
El servidor respondió:
Conexion con base de datos exitosa! Estación meteorológica Temperatura : 26.79 humedad : 55.97 presion : 726.98 calidad : 14.05 Registro en base de datos OK!

```

Figura 3.29 Consola de salida del Arduino IDE

Como siguiente prueba se observará si la base de datos está almacenando los registros en forma apropiada, y en el orden creado, donde adicional a los valores medidos se almacenará la fecha y hora a la cual se obtienen los datos.

La Figura 3.30, muestra la base de datos MySQL ya confirmación real y varias mediciones almacenadas. La base muestra los registros ordenados desde el más antiguo al más nuevo, sin embargo, se puede modificar para observar en distinto orden

Las mediciones son obtenidas desde el microcontrolador cada 12 (segundos), los cuales serán presentados más adelante en la página web.

				id	temperatura	humedad	presion	calidad	fecha_actual
<input type="checkbox"/>				1057	26.64	54.51	726.98	30.64	2024-02-12 10:27:34
<input type="checkbox"/>				1058	26.48	54.97	726.99	30.75	2024-02-12 10:27:46
<input type="checkbox"/>				1059	26.33	54.99	726.99	31.05	2024-02-12 10:27:57
<input type="checkbox"/>				1060	26.21	55.35	726.99	31.30	2024-02-12 10:28:09
<input type="checkbox"/>				1061	26.10	55.56	726.99	31.42	2024-02-12 10:28:20
<input type="checkbox"/>				1062	26.01	55.53	726.99	31.54	2024-02-12 10:28:31
<input type="checkbox"/>				1063	25.93	56.49	726.99	31.70	2024-02-12 10:28:42
<input type="checkbox"/>				1064	25.86	56.48	726.99	31.70	2024-02-12 10:28:54
<input type="checkbox"/>				1065	28.45	55.49	726.78	9.01	2024-02-13 10:55:22
<input type="checkbox"/>				1066	27.96	50.86	726.86	7.89	2024-02-13 10:55:33
<input type="checkbox"/>				1067	27.47	49.99	726.91	8.65	2024-02-13 10:55:45
<input type="checkbox"/>				1068	26.74	50.78	726.94	9.77	2024-02-13 10:55:56
<input type="checkbox"/>				1069	26.15	52.08	727.00	10.94	2024-02-13 10:56:07
<input type="checkbox"/>				1070	25.66	52.24	727.03	12.10	2024-02-13 10:56:19
<input type="checkbox"/>				1071	25.48	51.36	727.01	13.07	2024-02-13 10:56:30
<input type="checkbox"/>				1072	25.30	50.77	726.99	13.92	2024-02-13 10:56:42
<input type="checkbox"/>				1073	25.19	50.47	726.97	14.90	2024-02-13 10:56:53
<input type="checkbox"/>				1074	24.99	50.72	726.97	15.71	2024-02-13 10:57:04

Figura 3.30 Registros de la base de datos.

La información almacenada en la base de datos será mostrada en la página web mediante dos métodos, en la primera pantalla, se podrá observar la última medición tomada realizando una búsqueda en base a la fecha y hora de obtención del registro. Primero se muestra la fecha de medición, luego la temperatura, la humedad, la presión, y la calidad del aire.

Cada medición es mostrada en tablas individuales para evitar mezclar la información y tener una visualización más ordenada.

La Figura 3.31 muestra la interfaz inicial, el título, el autor, y todos los valores en tiempo real, los cuales se encuentran dentro del rango normal.

Para el otro método en la esquina inferior izquierda, se encuentra el botón Registros, al dar clic sobre él, se despliega la tabla con el historial de todos los registros incluyendo los datos de parámetros normales y los de alertas. Como se muestran en la Figura 3.32 y Figura 3.33.

Al final de la tabla se encuentra el botón Regresar, el cual permite regresar a la página principal donde se muestra los valores en tiempo real.



Figura 3.31. Interfaz principal.

Fecha de Medición	Temperatura	Humedad	Presión	Calidad del Aire
2023-12-01 16:46:05	Temperatura normal: 20.00°C	Humedad normal: 43.39%	Presion normal: 730.67hPa	Calidad de Aire normal: 45.24ppm
2023-11-27 19:00:54	Temperatura normal: 22.65°C	Humedad normal: 43.45%	Presion normal: 730.67hPa	Calidad de Aire normal: 41.45ppm
2023-11-27 19:01:07	Temperatura normal: 22.65°C	Humedad normal: 43.34%	Presion normal: 730.67hPa	Calidad de Aire normal: 42.84ppm
2023-11-27 19:01:40	Temperatura normal: 22.65°C	Humedad normal: 43.20%	Presion normal: 730.68hPa	Calidad de Aire normal: 42.76ppm
2023-11-27 19:01:56	Temperatura normal: 22.65°C	Humedad normal: 43.23%	Presion normal: 730.68hPa	Calidad de Aire normal: 42.28ppm
2023-11-27 19:02:45	Temperatura normal: 22.64°C	Humedad normal: 43.17%	Presion normal: 730.68hPa	Calidad de Aire normal: 40.97ppm
2023-11-27 19:02:57	Temperatura normal: 22.64°C	Humedad normal: 43.27%	Presion normal: 730.69hPa	Calidad de Aire normal: 42.89ppm
2023-11-27 19:03:10	Temperatura normal: 22.64°C	Humedad normal: 43.16%	Presion normal: 730.69hPa	Calidad de Aire normal: 44.47ppm
2023-11-27 19:03:23	Temperatura normal: 22.63°C	Humedad normal: 43.09%	Presion normal: 730.70hPa	Calidad de Aire normal: 45.24ppm
2023-11-27 19:03:39	Temperatura normal: 22.62°C	Humedad normal: 43.34%	Presion normal: 730.70hPa	Calidad de Aire normal: 45.09ppm
2023-11-27 19:09:49	Temperatura normal: 22.49°C	Humedad normal: 43.67%	Presion normal: 730.79hPa	Calidad de Aire normal: 43.02ppm
2023-11-27 19:10:02	Temperatura normal: 22.48°C	Humedad normal: 43.78%	Presion normal: 730.78hPa	Calidad de Aire normal: 42.37ppm

Figura 3.32. Datos históricos de parámetros normales.

2023-12-04 17:37:51	Temperatura normal: 24.52°C	Humedad normal: 48.14%	Presion normal: 728.72hPa	Calidad de Aire normal: 26.02ppm
2023-12-17 20:45:21	Temperatura normal: 22.00°C	¡Alerta Humedad fuera del rango Normal : 62.86%	Presion normal: 728.36hPa	Calidad de Aire normal: 5.53ppm
2023-12-17 20:45:33	Temperatura normal: 21.61°C	¡Alerta Humedad fuera del rango Normal : 62.04%	Presion normal: 728.38hPa	Calidad de Aire normal: 5.08ppm
2023-12-17 20:45:45	Temperatura normal: 21.32°C	¡Alerta Humedad fuera del rango Normal : 62.22%	Presion normal: 728.40hPa	Calidad de Aire normal: 5.67ppm
2023-12-17 20:45:58	Temperatura normal: 21.98°C	¡Alerta Humedad fuera del rango Normal : 62.56%	Presion normal: 728.36hPa	Calidad de Aire normal: 12.92ppm
2023-12-17 20:46:09	Temperatura normal: 21.60°C	¡Alerta Humedad fuera del rango Normal : 61.97%	Presion normal: 728.38hPa	Calidad de Aire normal: 11.21ppm
2023-12-17 20:46:21	Temperatura normal: 21.31°C	¡Alerta Humedad fuera del rango Normal : 62.96%	Presion normal: 728.39hPa	Calidad de Aire normal: 11.24ppm
2023-12-17 20:46:33	Temperatura normal: 21.09°C	¡Alerta Humedad fuera del rango Normal : 62.55%	Presion normal: 728.42hPa	Calidad de Aire normal: 11.70ppm
2023-12-17 20:46:45	Temperatura normal: 20.93°C	¡Alerta Humedad fuera del rango Normal : 62.29%	Presion normal: 728.43hPa	Calidad de Aire normal: 12.27ppm
2023-12-17 20:46:56	Temperatura normal: 20.82°C	¡Alerta Humedad fuera del rango Normal : 62.11%	Presion normal: 728.43hPa	Calidad de Aire normal: 12.94ppm
2024-01-10 18:03:53	Temperatura normal: 22.20°C	¡Alerta Humedad fuera del rango Normal : 63.66%	Presion normal: 726.82hPa	Calidad de Aire normal: 4.87ppm
2024-01-10 18:04:07	Temperatura normal: 21.82°C	¡Alerta Humedad fuera del rango Normal : 63.16%	Presion normal: 726.85hPa	Calidad de Aire normal: 4.50ppm
2024-01-10 18:04:34	Temperatura normal: 21.30°C	¡Alerta Humedad fuera del rango Normal : 63.27%	Presion normal: 726.87hPa	Calidad de Aire normal: 5.30ppm
2024-01-10 18:04:46	Temperatura normal: 21.12°C	¡Alerta Humedad fuera del rango Normal : 63.37%	Presion normal: 726.88hPa	Calidad de Aire normal: 5.89ppm

Figura 3.33. Datos mostrando alarma fuera de lo normal.

En la siguiente prueba realizada ingresamos desde un celular para monitorear las mediciones, se observó el correcto funcionamiento en la Figura 3.34.

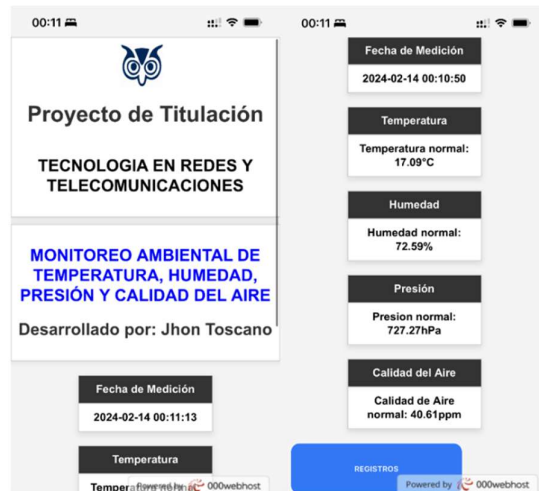


Figura 3.34. Mediciones vistas del celular.

Verificamos el funcionamiento del botón de la página principal para ingresar a los datos históricos y también el botón para luego regresar a la página inicial nuevamente como se ve en la Figura 3.35.



Figura 3.35. Funcionamiento de los botones.

Se realizaron pruebas en una pérgola el día 14 de febrero de 2024, donde se verificó las alertas mandadas por el sensor como se muestra en la Figura 3.36. Debido al ambiente cerrado de la pérgola, la temperatura se elevó al 30.57(°C) y salió de los parámetros establecidos al igual de la humedad esta fuera del rango con un 39,38%, lo que quiere decir que el lugar se encuentra seco.



Figura 3.36. Alertas de pruebas.

Una vez culminada las pruebas de funcionamiento, el prototipo se conectará en un ambiente exterior, que disponga de cobertura *WiFi* para que la transmisión de los datos a la página web sea exitosa.

4 CONCLUSIONES

- Se implementó un prototipo de un sistema de monitoreo ambiental remoto basado en el microcontrolador ESP32 y por medio de una página web se muestran los datos arrojados por el sensor.
- Se determinaron las necesidades esenciales del prototipo, así como los elementos fundamentales de hardware y software para el correcto funcionamiento, usando el sensor BME680, que es un componente multifuncional y arroja mediciones de temperatura, humedad, presión y calidad de aire.
- El prototipo se instaló en una caja que permite asegurar al circuito y a sus componentes evitando que exista daños ya sea por la manipulación o por factores ambientales, brindando la comodidad de tener el cable de energización a la mano, y el sensor expuesto para tomar mediciones reales.
- El microcontrolador ESP32 es un circuito versátil que permite trabajar con proyectos de Internet de las cosas, ya que incluye características como un módulo de conexión *WiFi*.
- La conexión HTTP post permite realizar la interconexión entre el servidor web y el microcontrolador ESP32, ofreciendo la ventaja del uso de tecnologías *IoT* para poder almacenar la información obtenida de forma analógica en cualquier servidor desde donde se puede realizar un tratamiento de los datos.
- La programación PHP facilitó el diseño de la página web de forma simple, y con herramientas básicas y necesarias para obtener una interfaz de fácil uso y amigable para el usuario final, quien podrá revisar los parámetros usando un dispositivo móvil o un computador.
- La visualización de alarmas en la página web se realizó sobre la misma interfaz principal, con el fin de no tener demasiadas páginas y evitar que el usuario final pueda cometer equivocaciones al momento de leer e interpretar las mediciones.
- Los datos históricos de las mediciones de los parámetros en el rango normal y de las alertas, han sido mostrados en una tabla accesible para el usuario mediante el ingreso a la página web, que junto con la fecha de medición se podrá analizar las variaciones de los datos para posteriores análisis estadísticos.

5 RECOMENDACIONES

- Al momento de realizar el circuito PCB en Proteus, las pistas obtenidas por defecto fueron demasiado finas, lo cual es recomendable cambiarlo y ensancharlas para que, al momento de quemarlo en la baquelita, no existan secciones donde se puede perder conectividad.
- Una vez instalado el prototipo en el lugar donde se obtendrá las mediciones, es necesario validar la conectividad al mismo, usando la herramienta ping, mediante un computador conectado en la misma red, con el fin de validar que no exista caídas debido a la ubicación y a la distancia entre la caja y el *router* o el *access point*.
- Es necesario realizar las pruebas y cambios necesarios en la programación del microcontrolador, para evitar así la manipulación del ESP32 luego de haberlo instalado para su funcionamiento.
- Se recomienda que las claves de ingreso a la plataforma de hosting, así como para la base de datos sean distintas, y sean cambiadas periódicamente, con el fin de evitar problemas de seguridad y pérdida de información.

6 REFERENCIAS

- [1] T. I. d. reservados., «¿Qué es la Internet de las cosas? Definición y explicación,» Kaspersky, 26 05 2023. [En línea]. Available: <https://latam.kaspersky.com/resource-center/definitions/what-is-iot>. [Último acceso: 15 Enero 2024].
- [2] T. I. d. reservados., «¿Qué es IoT (Internet de las cosas)?,» Amazon, 2023. [En línea]. Available: <https://aws.amazon.com/es/what-is/iot/#:~:text=Una%20aplicaci%C3%B3n%20de%20IoT%20es,datos%20y%20tomar%20decisiones%20informadas>. [Último acceso: 15 Enero 2024].

- [3] G. M. Smith, «¿Qué es un sensor y qué hace?,» 14 Marzo 2023. [En línea]. Available: <https://dewesoft.com/es/blog/que-es-un-sensor>. [Último acceso: 19 Febrero 2024].
- [4] SDI, «¿Qué es un sensor?», [En línea]. Available: <https://sdindustrial.com.mx/blog/sensores/>. [Último acceso: 19 Febrero 2024].
- [5] P. Marcken, «Introducción al funcionamiento de PWM en ESP32,» [En línea]. Available: https://polaridad.es/introduccion-al-funcionamiento-de-pwm-en-esp32/#google_vignette. [Último acceso: 17 Enero 2024].
- [6] «Sensor BME680 para Gas, Temp., Hum. y Presión Atmosferica,» [En línea]. Available: <http://www.jopapa.me/GasBME680.html>. [Último acceso: 18 Enero 2024].
- [7] A. Arduino, «IDE Arduino,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>. [Último acceso: 5 Enero 2024].
- [8] D. reservados, «Qué es Hosting,» [En línea]. Available: <https://www.arimetrics.com/glosario-digital/hosting>. [Último acceso: 15 diciembre 2023].
- [9] D. reservados, «Características De Hosting Gratis,» 15 Febrero 2022. [En línea]. Available: <https://es.000webhost.com/caracteristicas>. [Último acceso: 12 Enero 2024].
- [10] D. reservados, «¿Por qué MySQL?,» [En línea]. Available: <https://www.mysql.com/why-mysql/>. [Último acceso: 15 Enero 2024].
- [11] I+D, «Sensor de humedad, temperatura, presión y gas. BME680,» [En línea]. Available: <https://www.didacticaselectronicas.com/index.php/sensores/temperatura/cjmcu-680-detail>. [Último acceso: 18 Enero 2024].
- [12] J. M. Laorden, «BME280. Sensor,» [En línea]. Available: <https://www.tecnoteca.es/producto/schmod151-sensor-de-presion-atmosferica-bme280/>. [Último acceso: 19 Enero 2024].

- [13] T. I. d. reservados, «Sensor de llama 3 pines LM393,» [En línea]. Available: <https://triacs.cl/sensores/281-sensor-de-llama-3-pines-lm393-.html#:~:text=Sensor%20de%20detecci%C3%B3n%20de%20llama,760%20nm%20a%201100%20nm..> [Último acceso: 18 Enero 2024].
- [14] K. SALOME, «¿Cómo funciona 000webhost,» 25 Febrero 2022. [En línea]. Available: <https://hostingsgratis.com/000webhost-wordpress/>. [Último acceso: 15 Enero 2024].
- [15] D. Reservador, «Líderes en creación de Páginas Web,» [En línea]. Available: <https://es.wix.com/about/us#:~:text=La%20plataforma%20para%20crear%20sitios,y%20expandir%20su%20presencia%20online.&text=Una%20plataforma%20integral%20para%20crear,clientes%20de%20todas%20las%20escalas..> [Último acceso: 18 Enero 2024].
- [16] O. Taylor, «¿Por qué usar Network Solutions?,» [En línea]. Available: <https://www.websiteplanet.com/es/website-builders/network-solutions-website-builder/#overview>. [Último acceso: 19 Enero 2024].
- [17] I. d. Souza, «Bluehost: conoce esta solución de hospedaje y decide si es la ideal para tu página web,» 3 Marzo 2020. [En línea]. Available: <https://rockcontent.com/es/blog/bluehost/#:~:text=Bluehost%20es%20un%20servicio%20de,a%20buena%20relaci%C3%B3n%20costo%2Dbeneficio..> [Último acceso: 19 Enero 2024].
- [18] J. Beningo, «Cómo seleccionar y usar el módulo ESP32 con Wi-Fi/Bluetooth adecuado para una aplicación de IoT industrial,» 09 2018. [En línea].
- [19] H. Gringo, «Clima En Quito Ecuador Por Mes,» [En línea]. Available: <https://es.happygringo.com/blog/quito-weather/#:~:text=Temperatura%20promedio%20en%20Quito,una%20ciudad%20de%20clima%20extremo..> [Último acceso: 15 Enero 2023].
- [20] J. Torres, «Estudio bioclimático,» Abril 2021. [En línea]. Available: https://www7.quito.gob.ec/mdmq_ordenanzas/Administraci%C3%B3n%202019-2023/Comisiones%20del%20Concejo%20Metropolitano/Areas%20Hist%C3%B3ricas%20y%20Patrimonio/2021/2021-10-

25/Documentos%20para%20tratamiento/4.%20%20018-SCAHP-2021/estudio_bioclim+%C3%BCtico. [Último acceso: 18 Enero 2024].

- [21] M. H. Proaño, «CIRCULACIONES ATMOSFÉRICAS EN EL ECUADOR Y SU INFLUENCIA SOBRE LA CIUDAD DE QUITO,» 16 Enero 2016. [En línea]. Available: <http://portal.amelica.org/ameli/journal/367/3671557002/html/>. [Último acceso: 18 Enero 2024].
- [22] M. Navarra, «Presión atmosférica,» [En línea]. Available: http://meteo.navarra.es/definiciones/presion_atmosferica.cfm#:~:text=La%20unidad%20de%20medida%20es,atm%C3%B3sfera%20en%20un%20momento%20determinado. [Último acceso: 18 Enero 2024].
- [23] «Calidad del aire en Quito,» [En línea]. Available: <https://www.iqair.com/es/ecuador/pichincha/quito.> [Último acceso: 18 Enero 2024].
- [24] L. LLamas, «Sensor de calidad ambiental con Arduino y BME680,» [En línea]. Available: <https://www.luisllamas.es/sensor-de-calidad-ambiental-con-arduino-y-bme680/>. [Último acceso: 2024 Enero 15].
- [25] S. Carranza, «CONOCIENDO AL ESP32,» 3 11 2021. [En línea]. Available: <https://todomaker.com/blog/conociendo-al-esp32/#:~:text=ESP32%20es%20la%20denominaci%C3%B3n%20de,de%20la%20la%20familia%20ESP8266.> [Último acceso: 17 Enero 2024].
- [26] G. B., «¿Qué es un hosting y cómo funciona?,» 25 Enero 2024. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-un-hosting.> [Último acceso: 5 Febrero 2024].
- [27] Bosch, «Gas sensor BME680,» [En línea]. Available: <https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/>. [Último acceso: 17 Enero 2024].
- [28] V. Mohanan, «¿Qué es el bus CAN y cómo usar la interfaz CAN con ESP32 y Arduino?,» [En línea]. Available: <https://www.circuitstate.com/tutorials/what-is-can-bus-how-to-use-can-interface-with-esp32-and-arduino/>. [Último acceso: 17 Enero 2024].

- [29] E. Vicent, «Sensor BME680; medir la calidad del aire.,» 30 Marzo 2021. [En línea]. Available: <https://softwaremakeshardware.wordpress.com/2021/03/30/sensor-bme680-medir-la-calidad-del-aire/>. [Último acceso: 17 Enero 2024].
- [30] AQI, «FAQs de Quito Índice de calidad del aire,» [En línea]. Available: <https://www.aqi.in/es/dashboard/ecuador/pichincha/quito>. [Último acceso: 18 Enero 2024].

7 ANEXOS

La lista de los Anexos se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Conjunto de datos extenso