

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

TEMA:

**DISEÑO DE SISTEMA DE MONITOREO DE CALIDAD DE AIRE EN ENTORNOS
INDUSTRIALES UTILIZANDO SENSORES DE BAJO COSTO CON COMUNICACIÓN
ESP-MESH**

PREVIO LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

PRESENTADO POR:

GUSTAVO FERNANDO RUIZ BARROS

GUAYAQUIL - ECUADOR

AÑO: 2023

DEDICATORIA

Quiero dedicar este proyecto de grado a mi familia, en especial a mi madre la Econ. Lidia Guerrero que fue mi motor y mi inspiración durante todo este proceso.

A mi hermana que con su amor y su ayuda me ha motivado a seguir adelante.

A mis padres que, aunque estén ausentes físicamente los llevo siempre en el corazón.

A mi hijo y a mi esposa, quienes son mi principal motivo para seguir formándome y poder lograr mis metas.

CON AMOR GUSTAVO RUIZ

AGRADECIMIENTOS

Un agradecimiento especial a los profesores que me impartieron clases durante mi carrera universitaria, que con su ayuda pude lograr formarme y obtener conocimientos en las diferentes áreas en mi vida estudiantil.

Agradecer a las autoridades de la Universidad por permitir tener una excelente educación acompañada con la integración de diferentes áreas de deportes.

Un agradecimiento en especial al PhD. Efrén Herrera Muentes y al M.Sc. Jonathan Avilés, que con su ayuda y su colaboración se pudo concluir este proyecto.

A mis compañeros y amigos que me dio la Universidad también les quiero agradecer por su apoyo y su amistad.

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL).



Gustavo Fernando Ruiz Barros

EVALUADORES

EFRÉN HERRERA MUENTES
PROFESOR DE LA MATERIA

JONATHAN AVILÉS CEDEÑO
PROFESOR TUTOR

RESUMEN

El tema de la salud ocupacional está tomando cada vez más relevancia para los trabajadores, quienes se preocupan por las condiciones laborales en las que operan y cuidan su alimentación y procuran asistir a centros de entrenamiento y gimnasios.

Algo importante a mencionar es que, durante la jornada laboral, existen tareas que exponen a los trabajadores a diferentes tipos de agentes, tanto químicos como físicos, que pueden afectar la salud de las personas, lo cual es más preocupante en áreas como la de pintado, torno y soldado, en las cuales puede haber una gran cantidad de partículas suspendidas en el ambiente.

Las partículas suspendidas en el ambiente están clasificadas de acuerdo con La Agencia de Protección Ambiental de los Estados Unidos (EPA) que regula las partículas inhalables. Estas pueden ser PM10, las cuales son partículas conocidas también como pesadas y son típicamente retenidas por la mucosa. Otra categoría son las PM2.5, conocidas como partículas finas y que son retenidas por los bronquiolos. Finalmente tenemos las PM1.0 o partículas ultrafinas que no pueden ser retenidas y entran directamente al torrente sanguíneo.

La acumulación prolongada de estos tipos de partículas termina siendo perjudicial por los problemas cardíacos y respiratorios que provocan a las personas. Es por ello que se propuso el diseño de un sistema que pueda captar estas concentraciones de partículas nocivas y que, además pueda abarcar grandes extensiones de áreas con protocolo WiFi Mesh. Esto se lo hace mediante Esp-32 y un sensor de calidad de aire para recolectar y enviar los datos de dispositivos colocados en el taller y en la oficina permitiendo que se vean reflejados en un panel (dashboard).

El sistema implementado nos permitió conocer la cantidad de partículas existentes en distintas áreas de manera precisa, lo que posteriormente permitirá contrastar las concentraciones en áreas como el taller versus las oficinas. Esta información recolectada permite conocer las condiciones de operación durante la jornada laboral y así proceder a ver alternativas para reducir dichos niveles de contaminación.

Palabras Claves: IoT, WiFi Mesh

ABSTRACT

Occupational health is an issue becoming increasingly relevant for employees, who are concerned about the working conditions in which they operate and take care of their diet and seek to attend training centers and gyms.

It is important to mention that during a labor day, there are tasks that expose workers to different types of agents, both chemical and physical, that can affect people's health, which is more worrying in areas such as painting, lathe, and welding where there may be a large number of particles suspended in the environment.

Particles suspended in the environment are classified according to the U.S. Environmental Protection Agency (EPA) which regulates inhalable particulate matter. These may be PM10, which are also known as heavy particles and are typically retained by the mucosa. Another category is PM2.5, known as fine particles and that are retained by the bronchioles. Finally, we have PM1.0 or ultrafine particles that cannot be retained and enter directly into the bloodstream.

The prolonged accumulation of these types of particles becomes harmful due to the heart and respiratory problems caused to people. That is why the design of a system that can capture these concentrations and that can also cover large area extensions with WiFi Mesh protocol was proposed. This is done using Esp-32 and an air quality sensor to collect and send data from devices placed in the workshop and in offices and then they can be reflected on a panel.

The implemented system allowed us to know the number of particles existing in different areas in a precise manner, which will later allow us to compare the concentrations in areas such as the workshop versus offices. This information allows us to know the operating conditions during a working day and thus proceed to see alternatives to reduce said levels of contamination.

Key words: IoT, WiFi-Mesh

INDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
INDICE GENERAL.....	III
ÍNDICE DE IMÁGENES.....	V
ÍNDICE DE TABLAS.....	VI
ABREVIATURAS.....	VII
1. INTRODUCCIÓN.....	1
1.1 <i>Justificación del problema</i>	1
1.2 <i>Descripción del problema</i>	2
1.3 <i>Objetivos</i>	3
1.3.1 <i>Objetivo general</i>	3
1.3.2 <i>Objetivos específicos</i>	3
1.4 <i>Marco teórico</i>	3
1.4.1 <i>Partículas Nocivas</i>	3
1.4.2 <i>IoT</i>	5
1.4.2.1 <i>Capas de IoT</i>	6
1.4.3 <i>Protocolo Wifi-Mesh</i>	6
1.4.4 <i>Base de Datos</i>	7
1.4.5 <i>Sensor de Calidad de Aire</i>	8
1.4.6 <i>Inteligencia de Negocios (Business Intelligence)</i>	8
1.4.7 <i>Protocolos de Comunicación</i>	8
CAPÍTULO 2.....	11
2. METODOLOGÍA.....	11
2.1 <i>Concepción del dispositivo de monitoreo</i>	11
2.2 <i>Diseño del proyecto</i>	12
2.3 <i>Selección de equipos</i>	14
2.3.1 <i>Esp-32</i>	15
2.3.2 <i>Sensor MQ-135</i>	15
2.3.3 <i>Fuente DC</i>	17
2.4 <i>Selección de software y protocolo de comunicación</i>	18
2.4.1 <i>Protocolos de comunicación</i>	18
2.4.2 <i>Lenguajes de programación</i>	19

2.5 Descripción de la programación utilizada	20
2.6 Estructura del REST API	22
2.7 Diseño Electrónico.....	23
2.7.1 Diseño esquemático	23
2.8 Montaje del REST API	26
2.9 Puesta en marcha esp-32	31
CAPITULO 3	34
3. Análisis y Resultados.....	34
3.1 Interfaz.....	34
3.2 Tabla de costos.....	38
3.3 Diseño del Prototipo.....	40
CAPITULO 4	41
4. Conclusiones y Recomendaciones.....	41
4.1 Conclusiones.....	41
4.2 Recomendaciones.....	42
Referencias	43
ANEXOS	44
Anexo 1: Montaje de los equipos.....	44
Anexo 2: Códigos de programación Arduino	47

ÍNDICE DE IMÁGENES

Figura 1. 1 Rangos de calidad de aire (Universidad de los Andes de Colombia, 2008)	4
Figura 1. 2 Principales características de las partículas ultrafinas, finas y gruesas (Papayanopoulos, 2011)	5
Figura 1. 3 Capas de la Arquitectura IoT (Somayya Madakam, 2015)	6
Figura 1. 4 Vista lógica de una red MadMesh (Brik, Rayanchu, Saha, Sen, & Shrivastava, 2008)	7
Figura 1. 5 Capas del Protocolo TCP/IP (Quesada, 2016)	9
Figura 1. 6 Esquema del Protocolo HTTP (Mozilla Corporation, 2022)	9
Figura 1. 7 Representación Wifi-Mesh (Fernández, 2020)	10
Figura 2. 1 Diagrama de bloques del proceso de adquisición de datos de la cantidad de partículas nocivas (en PPM)	11
Figura 2. 2 Representación del prototipo del sistema medidor de aire	12
Figura 2. 3 Esquema de Comunicación del prototipo	13
Figura 2. 4 Esp-32	15
Figura 2. 5 sensor MQ-135	16
Figura 2. 6 Cargador de pared	17
Figura 2. 7 Diagrama de funcionamiento de componentes	20
Figura 2. 8 Diagrama de funcionamiento de flujo	21
Figura 2. 9 Estructura del REST API de adquisición de datos	22
Figura 2. 10 Diseño esquemático esp-32 con MQ-135	23
Figura 2. 11 Diseño PCB de la placa esp-32 con MQ-135	25
Figura 2. 12 Programación del REST API	26
Figura 2. 13 Código de arranque del servidor	26
Figura 2. 14 Programación de las rutas	27
Figura 2. 15 Programación de las consultas a la base de datos	28
Figura 2. 16 Pantalla del gestor de la base de datos esp32database	30
Figura 2. 17 Pantalla del requerimiento http listando los dispositivos	30
Figura 2. 18 Prueba de la ruta post /devices	31
Figura 2. 19 Prueba de la ruta /measure	31
Figura 2. 20 Recepción de la información del nodo 2 al nodo 1	32
Figura 2. 21 Recepción de la información del nodo 1 al nodo 2	33
Figura 3.1 Interfaz de las partículas presentes en el medio en el área de taller	34
Figura 3. 2 Interfaz de las partículas presentes en el medio en el área de Oficina	35
Figura 3. 3 Dashboard del área de pintado	35
Figura 3. 4 Dashboard del área de soldado	36
Figura 3. 5 Dashboard del área de Torno	36
Figura 3. 6 Dashboard del área de oficina planta alta	37
Figura 3. 7 Prototipo físico	40

ÍNDICE DE TABLAS

Tabla 2. 1 Equipos a utilizar en la implementación	14
Tabla 2. 2 Características del microcontrolador sugerido	15
Tabla 2. 3 Características del sensor MQ-135	16
Tabla 2. 4 Características del cargador de pared	17
Tabla 2. 5 Listado de las rutas http implementadas	28
Tabla 2. 6 Listado de las funciones utilizadas para interactuar con la base de datos	28
Tabla 3. 1 Costo de los equipos	39
Tabla 3. 2 Costo de los Servicios	39

ABREVIATURAS

WiFi	Wireless Fidelity.
API	Interfaz de programación de aplicaciones.
REST	Transferencia de estado proporcional.
HTTP	Protocolo de transferencia de hipertexto.
PM	Materia particulada
IOT	Internet de las cosas
SQL	Lenguaje de consulta estructurada
SO	Sistema Operativo

CAPÍTULO I

1. INTRODUCCIÓN

Actualmente, la contaminación es un problema que no solo afecta al medio ambiente, sino que también es un problema que afecta a la salud de las personas debido a los diferentes medios contaminantes presentes, los diferentes vehículos circulando por las calles de la ciudad, la basura generada por las personas las cuales contienen productos de lenta degradación y que generan partículas nocivas presentes en el aire y que según su concentración pueden representar un riesgo para la salud.

Los entornos industriales son otro tipo de ambientes que pueden presentar una concentración elevada de partículas nocivas, sin embargo, esto depende del área en donde la persona trabaja. En áreas donde el trabajador realiza sus tareas de pintado de piezas de diferente composición, el individuo puede absorber una gran cantidad de partículas nocivas por medio de inhalación, contacto por heridas o piel. Otra área que también presenta partículas nocivas es donde los trabajadores sueldan y la exposición por inhalación resulta muy perjudicial.

El presente proyecto consiste en desarrollar e implementar un dispositivo que pueda medir la cantidad de partículas nocivas en el ambiente con el fin de generar un reporte que pueda ser clave para evidenciar y mejorar las condiciones laborales en sectores donde se presentan riesgos nocivos por partículas pesadas para ayudar a la salud del personal.

1.1 Justificación del problema

En la actualidad las grandes y medianas empresas de productos y servicios cuentan con diferentes áreas que representan un peligro para la salud de las personas. Entre dichas áreas podemos considerar al área de pintura, por los químicos presentes en las pinturas que desprenden componentes orgánicos volátiles (COV) los cuales representan riesgos a la inhalación prolongada, teniendo consecuencias para el organismo humano como: irritación ocular, somnolencia, vértigo, malestar, dolor de cabeza y vómitos. A nivel de absorción por la piel puede ocasionar dermatitis y úlceras en la piel. También puede causar

por inhalación asma bronquial y otras alteraciones respiratorias. Por último, hay que destacar problemas relacionados con alteraciones nerviosas, problemas renales y hepáticos o neoplasias.

Otra área de gran importancia es el área de soldadura la cual representa un peligro por las partículas metálicas que se desprenden por la soldadura que también generan un peligro a la salud, debido a la inhalación de componentes pesados presentes en el aire.

Por esta razón se considera de alta importancia el poder supervisar y monitorear la calidad de aire utilizando sensores de bajo costo y así poder enviar la información a un centro de procesamiento a través de tecnología inalámbrica como es el WiFi Mesh.

El cliente que presenta el problema menciona que, el área de taller cuenta con espacios poco abiertos tales como el área de pintado, de torno y la de ensamble eléctrico y el interés del cliente es conocer el grado de la calidad del aire ya que está preocupado por la rotación de personal, dado que se presentan solicitudes de permisos médicos por parte del personal del área de taller.

1.2 Descripción del problema

El presente proyecto proporciona al cliente el diseño de un sistema automatizado que se encarga de monitorizar la calidad de aire de diferentes áreas. Para lograr la conexión necesaria se usarán dispositivos Esp-32 los cuales tienen integrado un módulo para hacer uso del protocolo WiFi Mesh, lo que permite que cada dispositivo actúe como punto de acceso para poder generar una red de dispositivos de recolección de datos. Estos datos van a ser almacenados en una base de datos SQL para su posterior tratamiento, con la finalidad de generar estadísticas sobre las partículas nocivas presentes, lo que permitirá gestionar medidas para mitigar el daño en la salud del personal a corto y largo plazo.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un sistema que mida la calidad de aire en entornos de oficina y de taller de ensamble y mantenimiento de equipos de bombeo para su respectivo monitoreo usando ESP-32 y tecnología WiFi Mesh con el fin de poder favorecer el confort y salud de los empleados.

1.3.2 Objetivos específicos

- Diseñar la arquitectura de red de comunicación de los sensores usando protocolo WiFi Mesh.
- Manejar y visualizar las estadísticas de la calidad del aire usando Power BI.
- Configurar los sensores de calidad de aire para la adquisición apropiada de datos de las partículas y la activación de alarmas en caso de encontrarse condiciones peligrosas para los operarios.
- Alojarse la información del sensor de las partículas en partes por millón (ppm) utilizando una base de datos relacional SQL.

1.4 Marco teórico

1.4.1 Partículas Nocivas

Todo proceso mecánico o industrial genera residuos que en algunos casos se pueden aprovechar nuevamente, y en otros casos no, debido a que dichos residuos son extremadamente pequeños, como lo son las partículas sólidas que cuentan con un tamaño en el orden de las micras, siendo clasificadas dependiendo del rango de tamaño exacto.

AIR QUALITY INDEX (AQI)	NIVELES DE INTERES EN SALUD	COLORES
cuando el AQI esta en...	... las condiciones de aire son...	simbolizado con color:
0 a 50	buenas	verde
51 a 100	moderadas	amarillo
101 a 150	dañinas grupos sensibles	naranja
151 a 200	dañinas	rojo
201 a 300	muy dañinas	morado
301 a 500	peligrosas	vinotinto

Figura 1. 1 Rangos de calidad de aire (Universidad de los Andes de Colombia, 2008)

Como se puede observar en la figura 1.1 la cual muestra la cantidad de partículas que deben estar presentes en el ambiente, la figura considera un rango bueno entre 0 a 50, y empiezan a ser dañinas cuando sobrepasan el rango de los 150.

1.4.1.1 Partículas PM10

Las PM10 son partículas sólidas, líquidas o de polvo, cenizas, hollín, partículas metálicas, cemento o polen (Ministerio para la transición ecológica y el reto demográfico, 2007) las cuales se encuentran dispersas en la atmósfera cuyo diámetro varía de 2.5 a 10 μm . Entre ellas podemos encontrar los silicatos y los aluminatos metálicos, los cuales son partículas pesadas. En cuanto a la parte orgánica estas cuentan con partículas de carbono en forma de hollín.

La exposición a este tipo de partículas puede afectar al sistema respiratorio de la persona, pero no llegan a atravesar los alvéolos pulmonares quedándose filtradas por la mucosa nasal.

1.4.1.2 Partículas PM2.5

Las partículas PM2.5 (OEHHA, 2022) son partículas que tienen un diámetro de 2.5 micrómetros, menor al grosor de un cabello. Su

exposición prolongada puede pasar a través de la mucosa nasal y desplazarse dentro de los pulmones; la exposición a este tipo de partículas causa efectos serios como lo son patologías cardiacas y pulmonares.

1.4.1.3 Partículas PM1.0

Denominadas partículas ultrafinas (Camfil, 2019) y son consideradas partículas ultra perjudiciales, siendo la materia particulada (PM) más pequeña y dañina, debido a su capacidad de entrar con facilidad a los alvéolos pulmonares, los que también pueden filtrarse al torrente sanguíneo. Su exposición puede provocar efectos de alto índice de mortalidad como anginas de pecho, ataques cardiacos, generación de cáncer al pulmón y demencia.

	Tipo de partículas		
	Ultrafinas (PM _{0.1})	Finas (≤ PM _{2.5})	Gruesas (PM _{2.5} -PM ₁₀)
Proceso de formación	<ul style="list-style-type: none"> • Nucleación de gases atmosféricos, incluidos H₂SO₄, NH₃ y algunos compuestos orgánicos • Condensación de gases 	<ul style="list-style-type: none"> • Condensación de gases • Coagulación de partículas pequeñas • Reacción de gases en o sobre las partículas • Evaporación de neblina y gotas de agua en las que los gases se han disuelto y reaccionado 	<ul style="list-style-type: none"> • Procesos mecánicos (prensado, molienda, abrasión, rompimiento de sólidos/gotas) • Evaporación de aerosoles • Suspensión de polvos • Reacción de gases en o sobre partículas

Figura 1. 2 Principales características de las partículas ultrafinas, finas y gruesas (Papayanopoulos, 2011)

1.4.2 IoT

El internet de las cosas o también conocido como IoT (Somayya Madakam, 2015) es la composición de 2 términos; el primero es internet el cual engloba la conexión mediante un sistema global y la segunda palabra es “cosas”, los cuales combinados indican la conexión global de artefactos y dispositivos que faciliten las tareas diarias y mejoren el confort según el área donde se aplica dicho paradigma.

Las arquitecturas en IoT no tienen definición o topología única, sino que dependen del criterio o idea a trabajar con la cantidad de sensores o dispositivos a implementar.

1.4.2.1 Capas de IoT

Basado en las recomendaciones de la unión internacional de telecomunicaciones (ITU), la arquitectura de las capas de IoT consiste en:

- La capa de sensado.
- La capa de acceso.
- La capa de red.
- La capa de aplicación.



Figura 1. 3 Capas de la Arquitectura IoT (Somayya Madakam, 2015)

1.4.3 Protocolo Wifi-Mesh

Es una tecnología de redes inalámbricas (Brik, Rayanchu, Saha, Sen, & Shrivastava, 2008) la cual une las infraestructuras ad-hoc y LAN; estas redes se auto organizan y se auto configuran dinámicamente. Las configuraciones de las redes Mesh poseen características específicas, por ende, la infraestructura consiste en dispositivos que se enlazan a internet y cuyo tráfico es controlado por routers Mesh. La otra característica son los clientes Mesh que proporcionan conexión punto a punto entre los dispositivos, además de realizar funciones básicas de red, como

encaminamiento o configuración, siendo otra alternativa de conexión, que los clientes también pueden proporcionar enlace a internet.

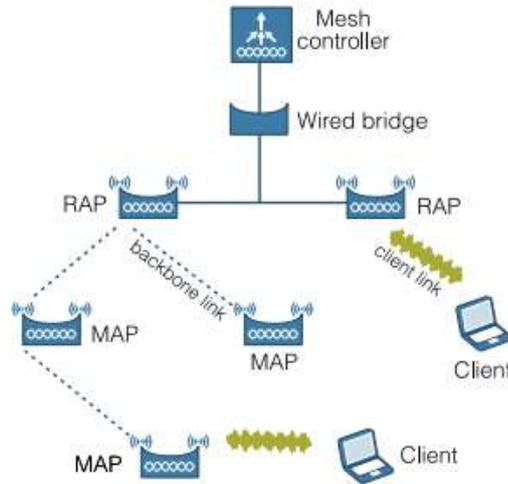


Figura 1. 4 Vista lógica de una red MadMesh (Brik, Rayanchu, Saha, Sen, & Shrivastava, 2008)

1.4.4 Base de Datos

Una base de datos es un conjunto de información almacenada y con la capacidad de ser consultada sistemáticamente. Los modelos presentes en las bases de datos son: jerárquicas, las cuales son organizadas en forma de árbol invertido, transaccionales, cuyo propósito es enviar y recibir datos a grandes velocidades, las relacionales, orientadas a objetos, grafos entre otros tipos de bases que pueden variar según el propósito a emplear.

1.4.4.1 Lenguaje SQL

Significa Structured Query Language (lenguaje de consulta estructurada) y es un lenguaje de programación estándar diseñado para administrar bases de datos relacionales. Esto significa establecer relaciones entre distintas tablas o entidades dentro de una base de datos. El lenguaje permite crear, insertar, modificar, eliminar, consultar datos y tablas.

1.4.5 Sensor de Calidad de Aire

Son dispositivos para detectar contaminantes en el aire, tales como partículas y gases nocivos que pueden afectar la salud humana. Estos dispositivos son aplicaciones para el monitoreo de calidad de aire, detección de gases, entre otros.

Su proceso de detección está basado en láser para estimar el número y tamaño de las partículas que pasan a través de ranuras. Estos sensores usan algoritmos que convierten la data bruta entrante en mediciones útiles.

1.4.6 Inteligencia de Negocios (Business Intelligence)

Su denominación está basada en un conjunto de estrategias, aplicaciones, datos, productos o tecnologías cuyo enfoque está fundamentado en la administración y creación de conocimiento por medio de análisis de datos existentes en una organización o empresa.

1.4.7 Protocolos de Comunicación

1.4.7.1 TCP/IP

Protocolo de Control de Transmisión que permite establecer una conexión segura y fiable en el intercambio de datos entre dos dispositivos dentro de una misma red. Trabaja a través de un sistema de capas ordenadas de manera jerárquica desde la superior (en la cual se envía el paquete de datos) y la inferior (donde se solicita el servicio de transmisión de datos) (Figura 1.4).

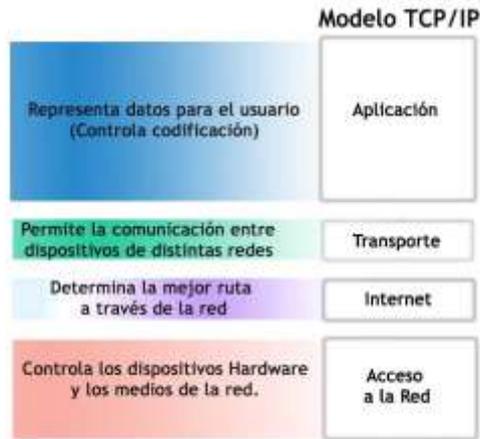


Figura 1. 5 Capas del Protocolo TCP/IP (Quesada, 2016)

1.4.7.2 HTTP

Significa (Hyper Text Transfer Protocol) es un protocolo el cual nos permite realizar petición de datos y recursos que pueden ser, imágenes, vistas HTML, datos, entre otros. La arquitectura más básica donde está presente el protocolo HTTP es cliente-servidor donde las peticiones pueden ser realizadas desde cualquier dispositivo y el envío de información es mediante estructura de texto el cual se basa en directorios donde uno hace el requerimiento.

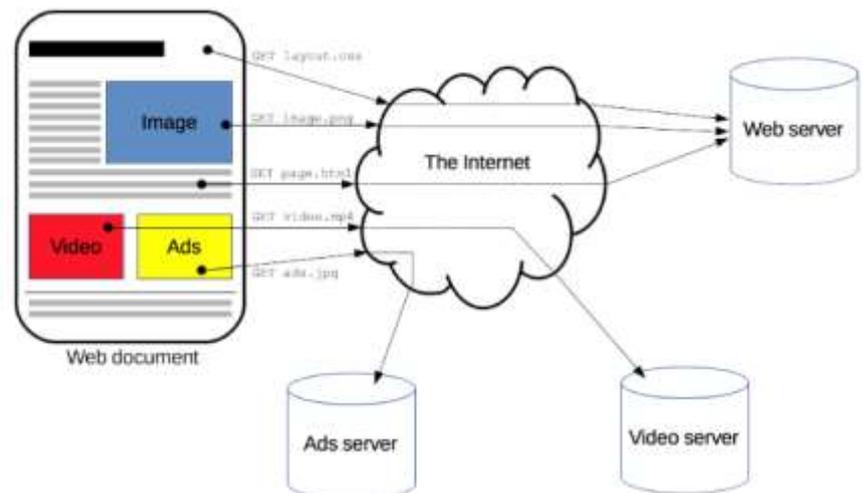


Figura 1. 6 Esquema del Protocolo HTTP (Mozilla Corporation, 2022)

1.4.7.3 Wifi-Mesh

Es una red WiFi conectada en malla y compuesta por varios dispositivos. La característica de esta red es que dichos dispositivos usan el mismo SSID o nombre de red con la peculiaridad de que pueden usar también la misma contraseña.

Comienza con un dispositivo base y luego se compone de nodos llamados satélites o puntos de acceso haciendo función de repetidores.



Figura 1. 7 Representación Wifi-Mesh (Fernández, 2020)

1.4.8 Lenguajes de Programación

1.4.8.1 C++

Es un lenguaje cuyo diseño se enfocó en extender el lenguaje C permitiendo que soporte el paradigma orientado a objetos, y la programación estructurada haciendo un lenguaje híbrido y multipropósito con un tiempo de ejecución relativamente rápido.

El lenguaje fue propuesto por Rick Mascitti en el año 1983 y actualmente es uno de los lenguajes más utilizados por los programadores.

1.4.8.2 Node.js

Es un entorno de tiempo de ejecución multiplataforma con código abierto, para la parte de servidor la cual está basado en el lenguaje de programación JavaScript que es un lenguaje con capacidad de tener funcionalidades asíncronas además de contar con un V8 desarrollado por Google el cual le permite correr desde un computador.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se abordará una propuesta para la medición de la calidad de aire, en el que se especificará su constitución en cuanto a hardware y software necesario para su implementación.

2.1 Concepción del dispositivo de monitoreo

El hardware de monitoreo está basado en un dispositivo Esp-32; este dispositivo se conecta a un sensor AQ (air quality) para leer la información del ambiente y transmitirla mediante protocolo WiFi a un servidor web. El servidor web almacenará los datos en la base de datos y se utilizarán para crear un panel (dashboard) con la información recolectada.

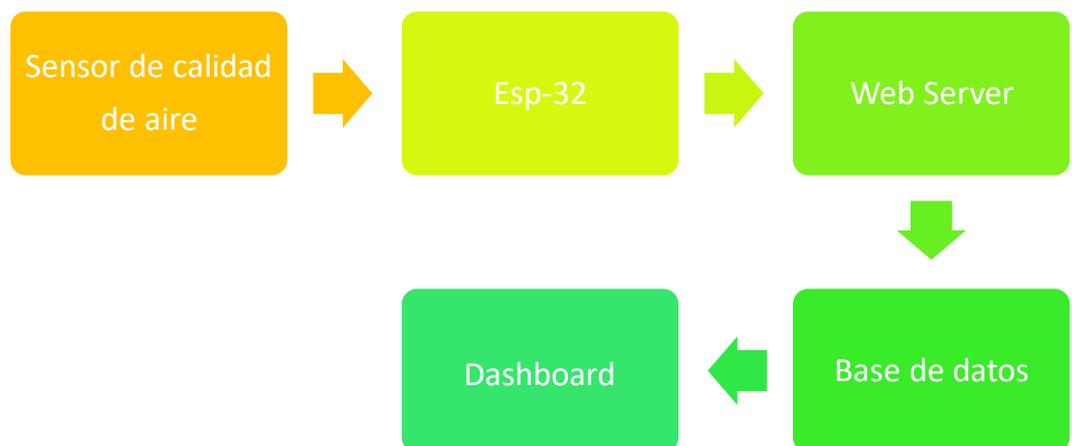


Figura 2. 1 Diagrama de bloques del proceso de adquisición de datos de la cantidad de partículas nocivas (en PPM).

2.2 Diseño del proyecto



Figura 2. 2 Representación del prototipo del sistema medidor de aire

Como se puede observar en la figura 2.2, el proyecto consistirá en un módulo Esp-32 y un sensor de calidad de aire, encapsulados en una caja que tiene como fin el proteger los dispositivos.

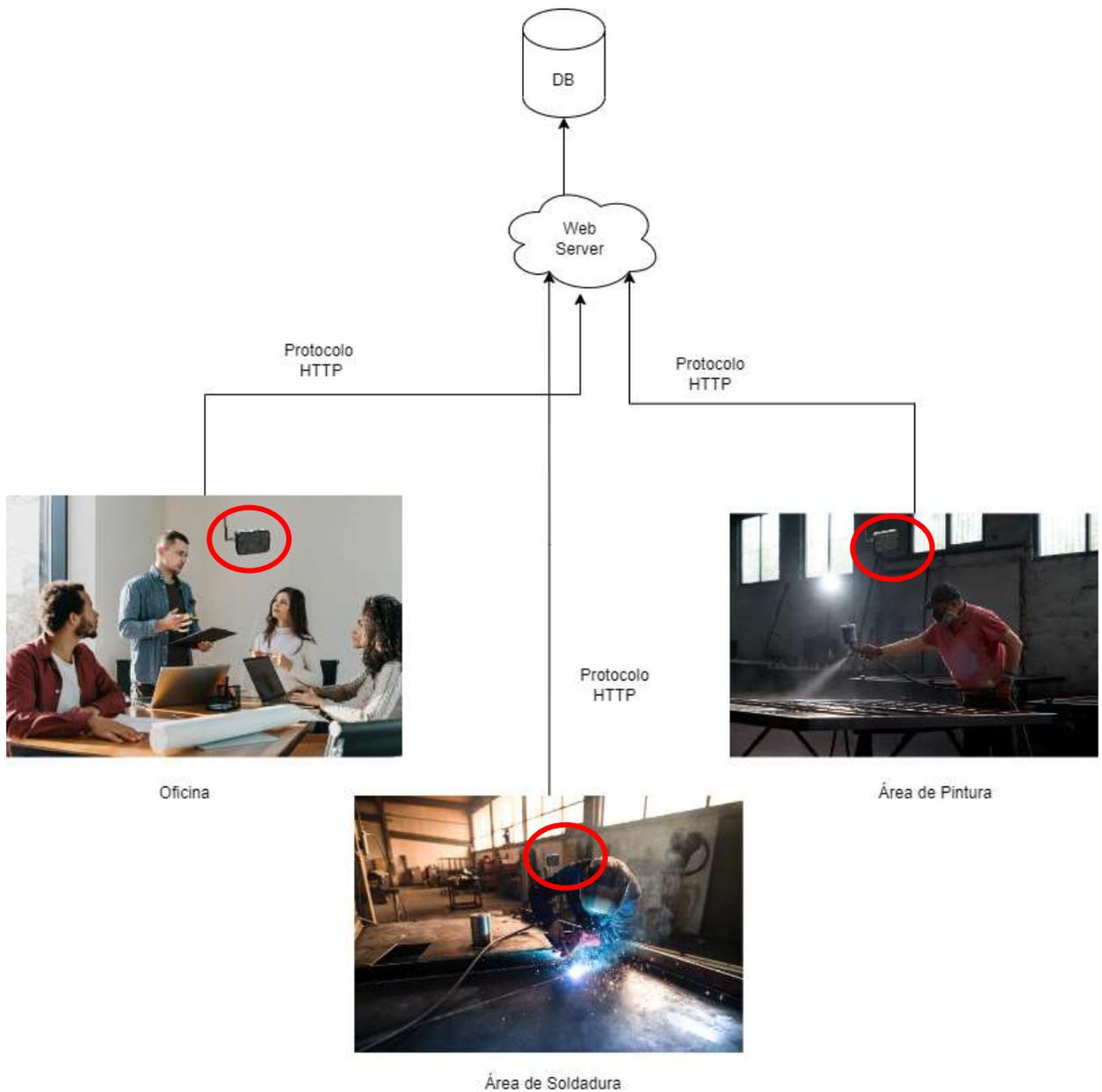


Figura 2. 3 Esquema de Comunicación del prototipo

El prototipo se va a ubicar en una posición estratégica (ver figura 2.3), usando conexión WiFi Mesh se podrá conectar al punto de acceso a internet más cercano (otro módulo Esp-32 cercano que actúe como AP). La comunicación establecida entre el prototipo y la base de datos va a ser mediante protocolo HTTP y este se va a conectar a un servidor web que tiene como objetivo procesar las peticiones que le lleguen e ingresarlas a la base de datos para que puedan ser consultadas posteriormente mediante programas de BI(business intelligence).

2.3 Selección de equipos

A continuación, se determinan los equipos, así como el software que será utilizado en la implementación del medidor de calidad de aire. (Tabla 2.1).

Tabla 2. 1 Equipos a utilizar en la implementación

Equipo	Ilustración
ESP-32	 A photograph of an ESP-32 microcontroller board, showing its various pins and components.
Sensor MQ-135	 A photograph of an MQ-135 gas sensor module, which includes a sensor element and a small PCB with pins.
Fuente DC	 A photograph showing a white DC power source (likely a USB power bank or adapter) and two white cables, one with a USB-A connector and the other with a USB-C connector.
Computador	 A photograph of a desktop computer system, including a monitor, keyboard, mouse, and tower PC case.
Router wifi	 A photograph of a white wireless router with multiple ports and antennas.

2.3.1 Esp-32

El módulo Esp-32 fue seleccionado en lugar de otros debido a su eficiente, veloz y confiable capacidad de conexión a redes inalámbricas. Además, su uso del protocolo HTTP y WiFi Mesh lo convierte en una opción perfecta para la conexión de nodos de información.



Figura 2. 4 Esp-32

Tabla 2. 2 Características del microcontrolador sugerido

Características Esp-32	
WiFi	WiFi integrado: Access point & Station
Bluetooth	Bluetooth 4.2 2.4 GHz; BT 2.0 y 4.0 BLE
Pines	36 GPIO pins. 16 x Analog-to-Digital Converter (ADC) de 12 bits de resolución y se pueden programar con límite de entrada a 1V, 2 V y 4V
RAM	520 kb
Procesador	Procesador dual core Xtensa® LX6 de 32 bits
Frecuencia de reloj	160-240 MHz
Voltaje	3V

2.3.2 Sensor MQ-135

El MQ-135 es un sensor de control de calidad de aire que se utiliza para detectar la contaminación en el área donde se encuentra ubicado. Este dispositivo permite detectar contaminación en casas, oficinas e industrias donde se manejan compuestos nocivos para la salud. El sensor detecta la concentración de gas en diferentes porcentajes, además la señal de salida que brinda el sensor puede ser analógica o digital.



Figura 2. 5 sensor MQ-135

Tabla 2. 3 Características del sensor MQ-135

Características MQ-135	
Corriente de operación	150 mA
Voltaje	5 VDC
Potencia de consumo	800mW
Tiempo de precalentamiento	20 segundos
Rango de detección (PPM)	10-1000ppm
Concentración detectable	Humo, benceno, amoniac, sulf

2.3.3 Fuente DC

Para suplir la demanda de energía de los dispositivos instalados se usará una fuente de corriente continua, similar a las usadas en los celulares, lo cual será útil debido a que el puerto de entrada del Esp-32 es Micro USB.



Figura 2. 6 Cargador de pared

Tabla 2. 4
del cargador de

Características Cargador	
Voltaje de salida	5V y 9V
Corriente nominal	2A y 1.67A

Características
pared

2.4 Selección de software y protocolo de comunicación

2.4.1 Protocolos de comunicación

2.4.1.1 TCP/IP

La elección de este protocolo de comunicación se debe a que, al ser el más usado en el mundo, facilita la configuración de los equipos; además, bajo este protocolo se maneja también el estándar WiFi, y para el Esp-32 es esencial debido a su chip WiFi que se basa en el estándar IEEE 802.11.ac. En cuanto al punto de vista de la empresa, el protocolo es el más usado para la asignación de dirección a los dispositivos bajo la opción de IP fija.

2.4.1.2 Protocolo HTTP

El protocolo HTTP permite una conexión directa a una dirección IP bajo una ruta, siendo esta opción la ideal debido a que el montaje del servidor estará alojado en una página que provee soluciones en la nube y la mejor forma de conectarse a la máquina virtual es mediante protocolo HTTP, el cual permite escuchar peticiones entrantes y salientes para su posterior procesamiento.

2.4.1.3 Protocolo WiFi Mesh

Este protocolo es esencial para la implementación ya que no se requiere instalar una cantidad grande de routers para que se pueda conectar cada dispositivo Esp-32 que ingrese en la red. Este protocolo permite que el Esp-32 actúe como punto de acceso para otros Esp-32, evitándonos la necesidad de agregar puntos de acceso especializados y colocar dispositivos para enviar información.

2.4.2 Lenguajes de programación

2.4.2.1 C++

Es el referente para la programación de dispositivos como los microcontroladores, ya que es el lenguaje de programación que permite administrar de mejor manera los recursos, gracias a que se puede manipular a bajo nivel y eso facilita explotar las características del Esp-32 al máximo.

2.4.2.2 Node.JS

Node.js es un entorno que proporciona un conjunto de herramientas que permiten la ejecución de códigos de JavaScript sin necesidad de un navegador web. Su principal característica es la capacidad de instalarse y ejecutarse en múltiples sistemas operativos. Las herramientas que brinda Node.js permiten acceder al sistema operativo, incluyendo el sistema de archivos y la ejecución de comandos del sistema, lo que facilita el montaje de servidores físicos o virtuales. De este modo, es posible interactuar con otros servidores o dispositivos, como celulares, computadoras o dispositivos IoT, y así intercambiar datos. Gracias a estas funcionalidades, se puede desarrollar el back-end de la web (es decir, el desarrollo del lado del servidor) y programar el proyecto adecuadamente.

2.5 Descripción de la programación utilizada

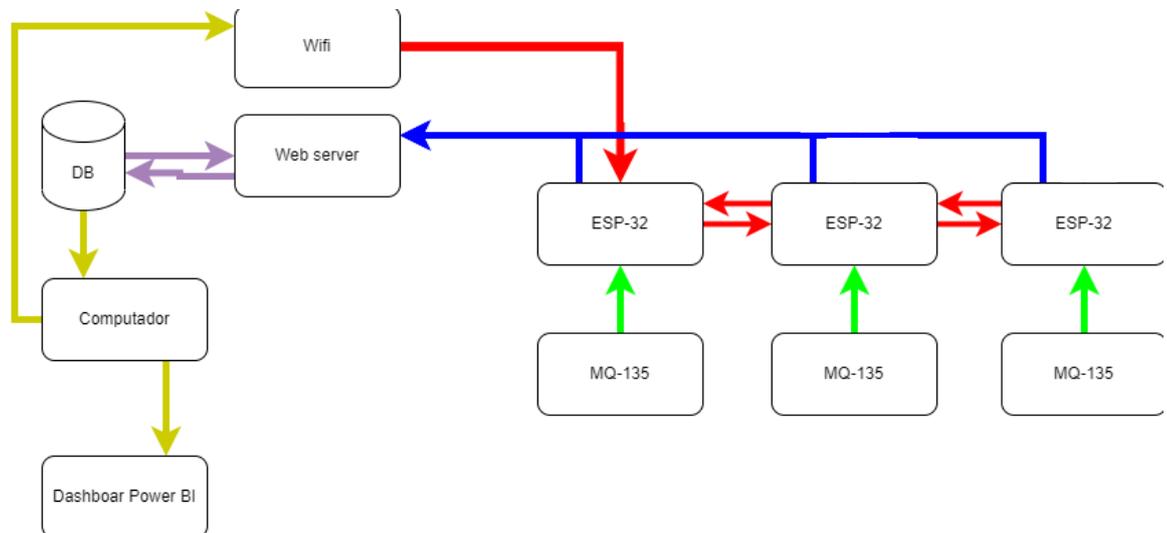


Figura 2. 7 Diagrama de funcionamiento de componentes

Como se puede observar el diagrama (Figura 2.7), el Esp-32 principal se conecta a la red WiFi y se configura como punto de acceso con el fin de extender una red de malla que no solo proporcione salida a internet, sino que además actúe como si fuese repetidor de señal WiFi para no tener que implementar estructura de red extra que tenga salida a internet.

Cada dispositivo Esp-32 contará con un sensor de calidad de aire MQ-135, el cual recolecta datos sobre los parámetros del aire en las áreas designadas y esa información es enviada mediante protocolo HTTP a un servidor web para su posterior almacenamiento. Una representación del flujo del programa es la que tenemos en la figura 2.8.



Figura 2. 8 Diagrama de funcionamiento de flujo

2.6 Estructura del REST API

El manejo adecuado de un proyecto es indispensable para su escalabilidad, por lo que la estructura del proyecto debe ser lo más clara posible para su fácil comprensión.

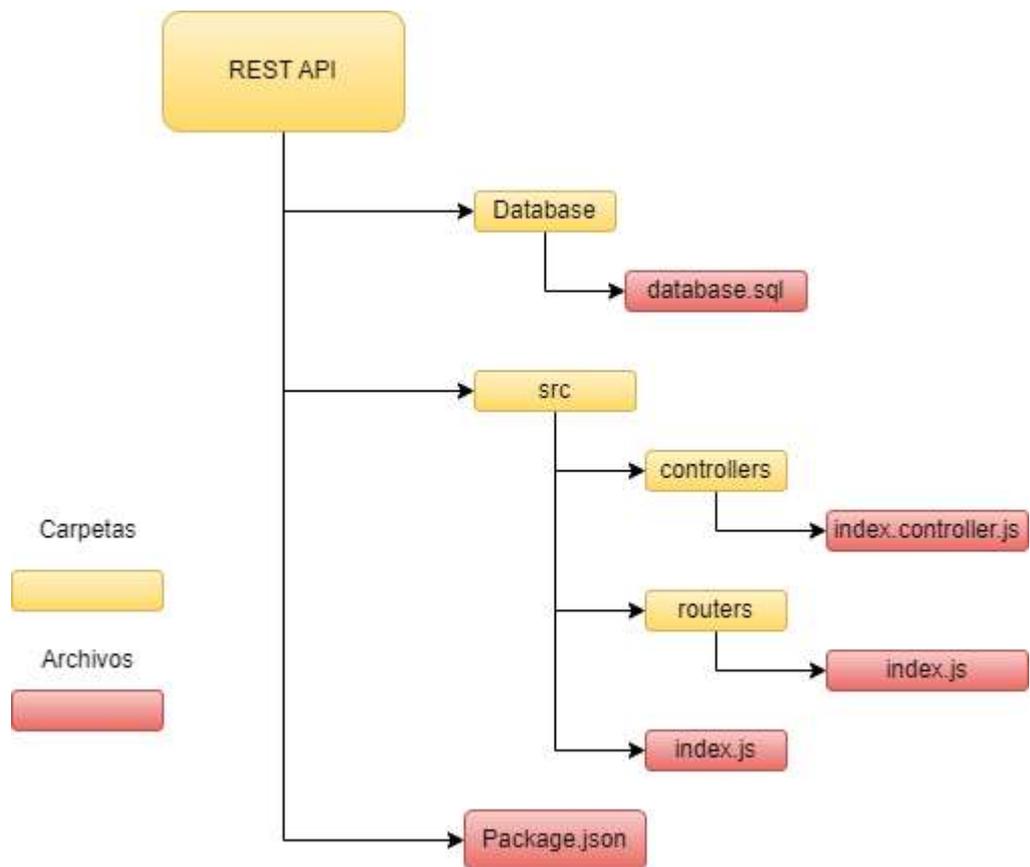


Figura 2. 9 Estructura del REST API de adquisición de datos.

Con respecto al proyecto se cuenta con el archivo package.json que es el encargado de guardar la información referente al proyecto, la carpeta src es la que contiene todo lo referente al manejo de las rutas HTTP y los comandos para insertar los datos a la base de información, al momento de realizar un requerimiento http. La carpeta routers es la encargada de almacenar las rutas para cuando los dispositivos Esp-32 realicen peticiones http, tengan rutas para consultar y enviar información.

2.7 Diseño Electrónico

2.7.1 Diseño esquemático

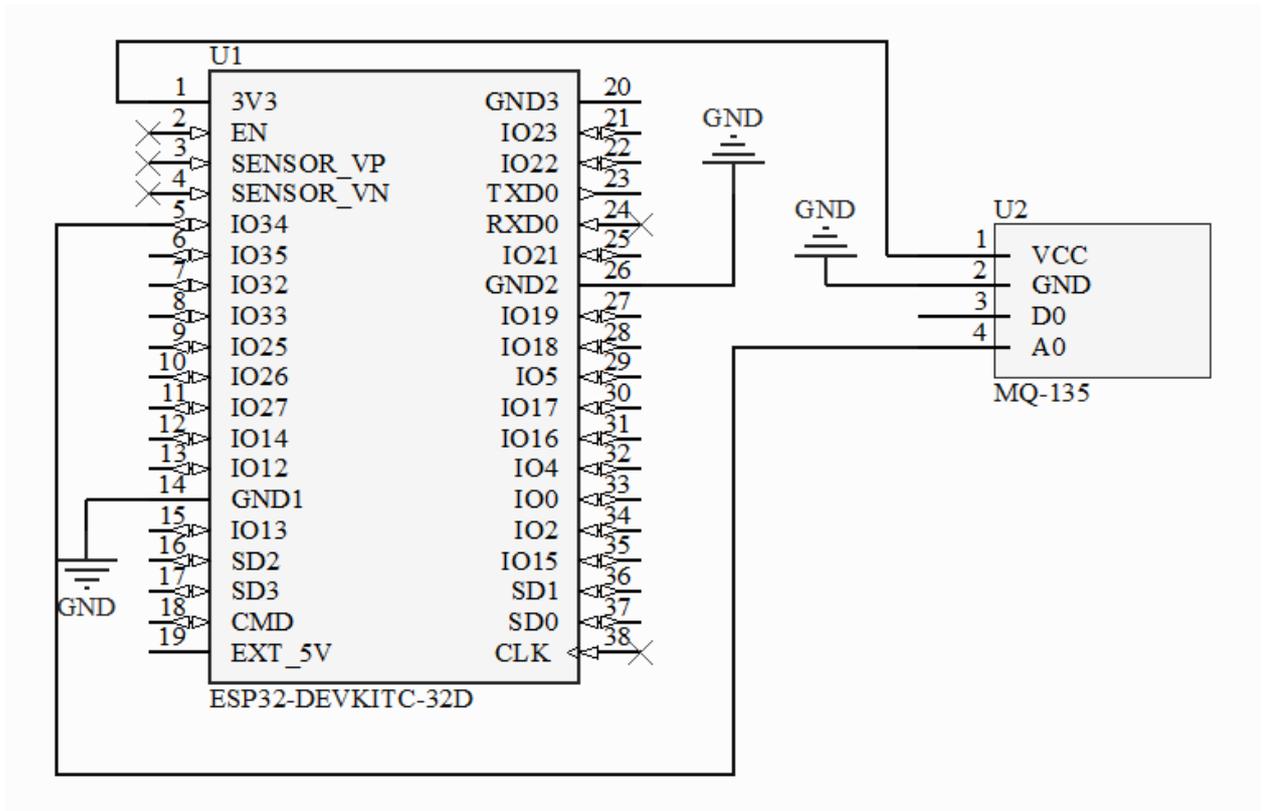


Figura 2. 10 Diseño esquemático esp-32 con MQ-135

En cuanto a la configuración del diseño esquemático en el proyecto, se ha establecido la conexión del sensor MQ-135 a través de la utilización de los pines de alimentación del ESP-32. El sensor MQ-135, es un dispositivo electroquímico que permite la medición de la calidad del aire en tiempo real, lo que permite obtener información valiosa sobre la composición química del ambiente en el que se encuentre.

En este caso, se ha configurado la lectura de los datos a través del pin IO34, que permite la lectura de información en el ESP-32. Una vez que se han obtenido los datos, se procede a su envío al servidor correspondiente mediante la conexión establecida en la red de dispositivos IoT.

Es importante destacar que el diseño esquemático es fundamental en el desarrollo de proyectos IoT, ya que permite establecer la conexión entre los diferentes dispositivos y sensores, lo que permite la recolección de información valiosa para el análisis y la toma de decisiones.

En conclusión, la configuración del diseño esquemático es una parte fundamental del proyecto IoT, ya que permite establecer la conexión entre los diferentes dispositivos y sensores, lo que permite la recolección de información valiosa para el análisis y la toma de decisiones. Es importante contar con un diseño adecuado para garantizar la seguridad de la información y la correcta transmisión de datos en la red, lo que es esencial en proyectos de este tipo.

2.7.2 Diseño PCB

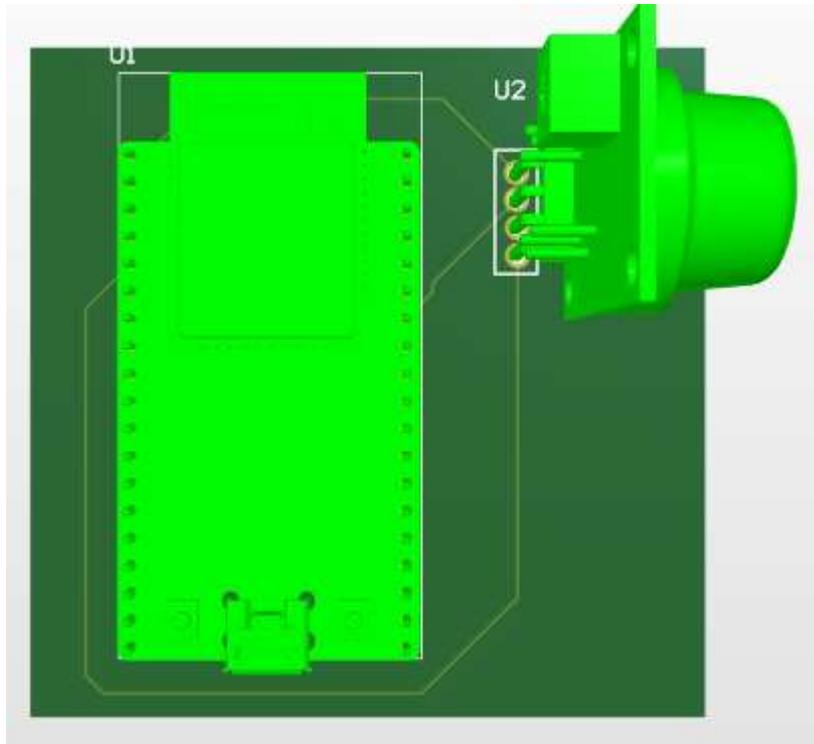
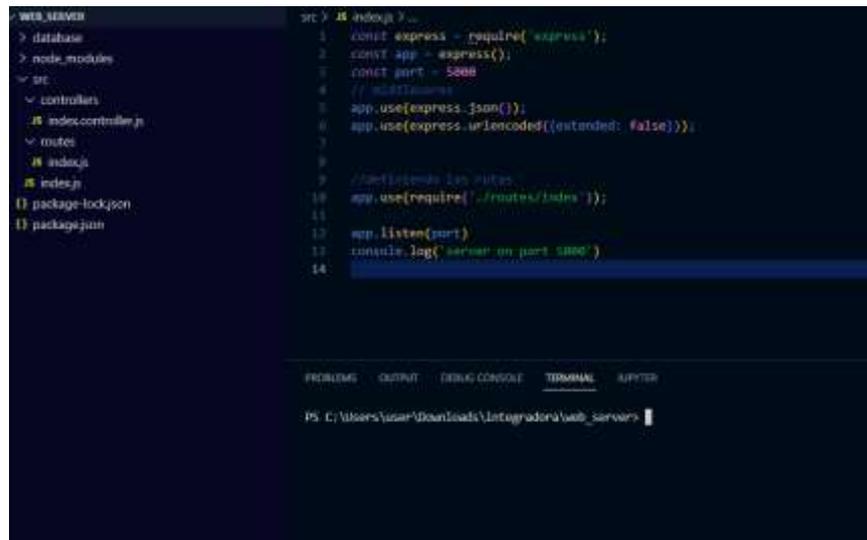


Figura 2. 11 Diseño PCB de la placa esp-32 con MQ-135

La placa está compuesta por dos componentes principales: el Esp-32 y el sensor MQ-135. Estos dos elementos son esenciales para la funcionalidad del dispositivo, ya que el Esp-32 es el encargado de recibir y enviar datos al servidor y el sensor MQ-135 es el responsable de recopilar información sobre la calidad del aire en el ambiente en el que se encuentra el dispositivo.

En cuanto a las dimensiones de la placa, se han elegido cuidadosamente para lograr un equilibrio entre tamaño y funcionalidad. La placa mide 6.22 cm tanto de largo como de ancho, lo que resulta en una PCB cuadrada. Este tamaño compacto permite que la placa sea fácil de montar en diferentes lugares, lo que aumenta su versatilidad en términos de aplicación. Además, el tamaño reducido también hace que el dispositivo sea portátil y fácil de transportar, lo que puede ser una ventaja para aplicaciones en el campo de la monitorización ambiental.

2.8 Montaje del REST API



```
src > index.js
1 const express = require('express');
2 const app = express();
3 const port = 5000
4 // middlewares
5 app.use(express.json());
6 app.use(express.urlencoded({extended: false}));
7
8
9 //definiendo las rutas
10 app.use(require('./routes/index'));
11
12 app.listen(port)
13 console.log('server on port 5000')
14
```

Figura 2. 12 Programación del REST API

En la figura 2.12 se observa la programación del servidor bajo el lenguaje de nodejs, con la configuración del esquema que se presentó en la figura 2.9.

```
1. const express = require('express');
2. const app = express();
3. const port = 5000
4. // middlewares
5. app.use(express.json());
6. app.use(express.urlencoded({extended: false}));
7.
8.
9. //definiendo las rutas
10. app.use(require('./routes/index'));
11.
12. app.listen(port)
13. console.log('server on port 5000')
14.
```

Figura 2. 13 Código de arranque del servidor

Las 2 primeras líneas importan el framework express para poder configurar las rutas HTTP; la línea 3 presente en el código fuente contiene el puerto en donde va a recibir las peticiones de los dispositivos. Los middlewares para este proyecto necesitan tener el formato json y que el encriptamiento de las url en la línea 6 del código tenga configurado que solo acepte texto.

Una vez definido el archivo de rutas se inicia la aplicación para que escuche en el puerto 5000.

```
1. const { Router } = require('express');
2. const router = Router();
3. //funciones a usar
4. const {getdevices, uploadMeasures, createDevices} = require('../controllers/index.controller')
5.
6. module.exports = router;
7. //rutas implementadas
8. router.get('/devices', getdevices)
9. router.post('/devices/:name.:location', createDevices)
10. router.post('/measure/:name.:measure.:time', uploadMeasures)
```

Figura 2. 14 Programación de las rutas

El archivo de rutas es un archivo importante dentro de un proyecto que utiliza el patrón de arquitectura MVC (Modelo-Vista-Controlador), ya que es el encargado de definir las diferentes rutas que el servidor puede manejar. En este caso, se especifica que hay una ruta get y dos rutas post.

La ruta get permite acceder a un recurso y obtener información de él, mientras que las rutas post permiten enviar información al servidor para procesarla. Cada una de estas rutas es manejada por una función de control que es la encargada de interactuar con la base de datos y devolver los datos correspondientes al cliente. Estas funciones de control pueden realizar operaciones como agregar, modificar o eliminar información de la base de datos según lo requiera la lógica del negocio. Por lo tanto, el archivo de rutas es una parte esencial de un proyecto que involucra una base de datos, ya que permite al cliente interactuar con la información almacenada en ella.

Tabla 2. 5 Listado de las rutas http implementadas

Ruta	Tipo de requerimiento	Descripción
/devices	GET	Ruta usada para obtener los dispositivos registrados en la base de datos
/devices/:name.:location	POST	Ruta usada para enviar el registro de un nuevo dispositivo a la base de datos.
/measure/:name.:measure.:time	POST	Ruta usada para enviar registro a la tabla measures donde almacena el nombre del dispositivo la medida tomada y el registro del tiempo.

```

1.
2. const { Pool } = require('pg');
3. const pool = new Pool({
4.   host: 'localhost',
5.   user: 'gruis',
6.   password: 'gruis1234',
7.   database: 'esp32database',
8.   port: '5432'
9. });
10. const getDevices = async (req, res) => {
11.   const response = await pool.query("SELECT * FROM devices");
12.   res.status(200).json(response.rows);
13. }
14. const createDevices = async (req, res) => {
15.   const name = req.params.name;
16.   const location = req.params.location;
17.   const response = await pool.query("INSERT INTO devices (name, location) VALUES ($1, $2)", [name, location]);
18.   res.json({
19.     message: 'Measure Added Successfully',
20.     body: {
21.       user: {name, location}
22.     }
23.   });
24. }
25. const uploadMeasures = async (req, res) => {
26.   const name = req.params.name;
27.   const measure = req.params.measure;
28.   const time = req.params.time;
29.   const response = await pool.query("INSERT INTO measures (name, measure, time_measure) VALUES ($1, $2, $3)", [name, measure, time]);
30.   res.json({
31.     message: 'Measure Added Successfully',
32.     body: {
33.       user: {name, measure, time}
34.     }
35.   });
36. }
37. module.exports = {
38.   getDevices,
39.   uploadMeasures,
40.   createDevices
41. }

```

Figura 2. 15 Programación de las consultas a la base de datos

Esta sección contiene las funciones que ejecutan las rutas al momento de recibir una petición HTTP, para la cual mediante la siguiente tabla se ilustrará el funcionamiento.

Tabla 2. 6 Listado de las funciones utilizadas para interactuar con la base de datos

Función	Descripción
getdevices	La función tiene como objetivo consultar la

	<p>tabla "devices" con el fin de mostrar los dispositivos registrados y su respectiva ubicación donde se están realizando las mediciones. Esta consulta es esencial para el correcto monitoreo de los dispositivos, ya que permite visualizar la información recolectada en función de la ubicación geográfica, lo que facilita la toma de decisiones y la identificación de posibles problemas en un área específica</p>
<p>createDevices</p>	<p>La función se encarga de realizar una consulta en la tabla "devices" con el fin de insertar nuevos dispositivos en la base de datos. Para llevar a cabo esta tarea, se requieren dos parámetros esenciales: el nombre y la ubicación del dispositivo. De esta manera, se asegura que los nuevos dispositivos registrados en el sistema cuenten con la información necesaria para su correcta identificación y localización en la red. Gracias a esta función, se pueden agregar nuevos dispositivos al sistema de manera sencilla y eficiente.</p>
<p>uploadMeasures</p>	<p>Se trata de una función que se encarga de insertar mediciones realizadas por dispositivos en la base de datos "measures". Es importante destacar que esta función es clave en el proceso de recolección de datos del sistema de monitoreo, ya que permite almacenar la información recopilada por los dispositivos en un lugar seguro y accesible para su posterior análisis y visualización. La función debe recibir los datos de la medición, que normalmente incluyen la fecha y hora de la toma, el valor de la medición y el identificador del dispositivo que realizó la medición. De esta forma, se asegura que la información esté bien estructurada y se pueda buscar fácilmente en la base de datos en el futuro.</p>

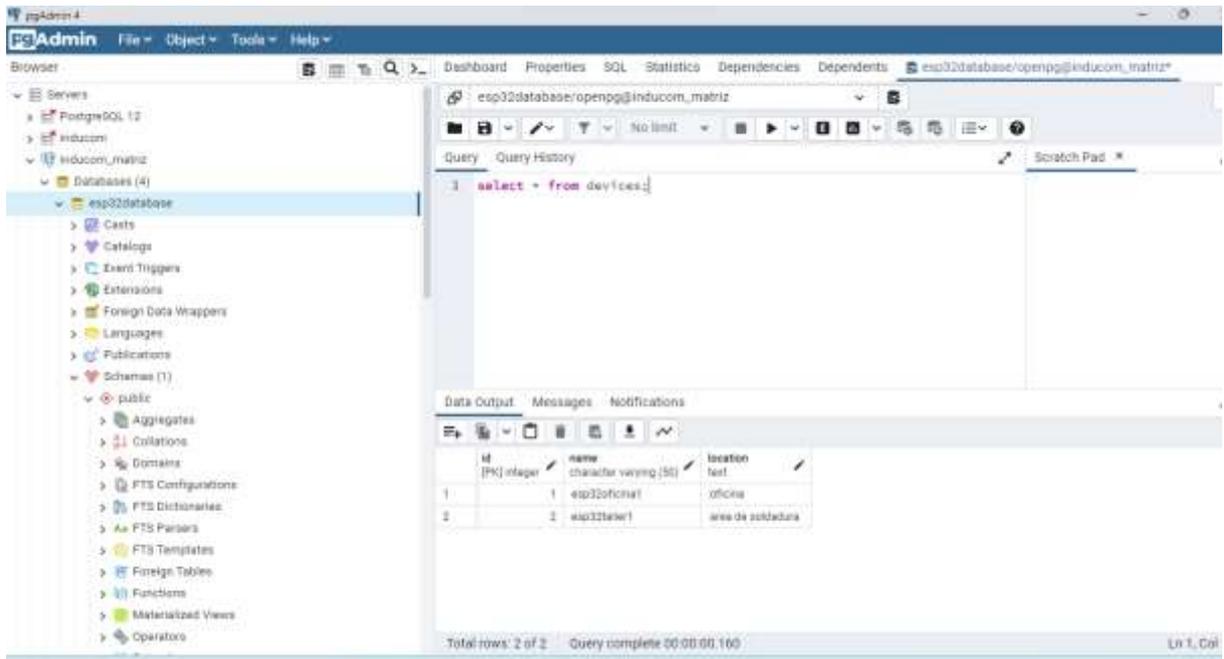


Figura 2. 16 Pantalla del gestor de la base de datos esp32database

A continuación se presenta una captura de pantalla que muestra los dispositivos registrados en la tabla "devices" de la base de datos después de realizar una consulta. En la imagen se pueden visualizar los nombres de los dispositivos y su respectiva ubicación.

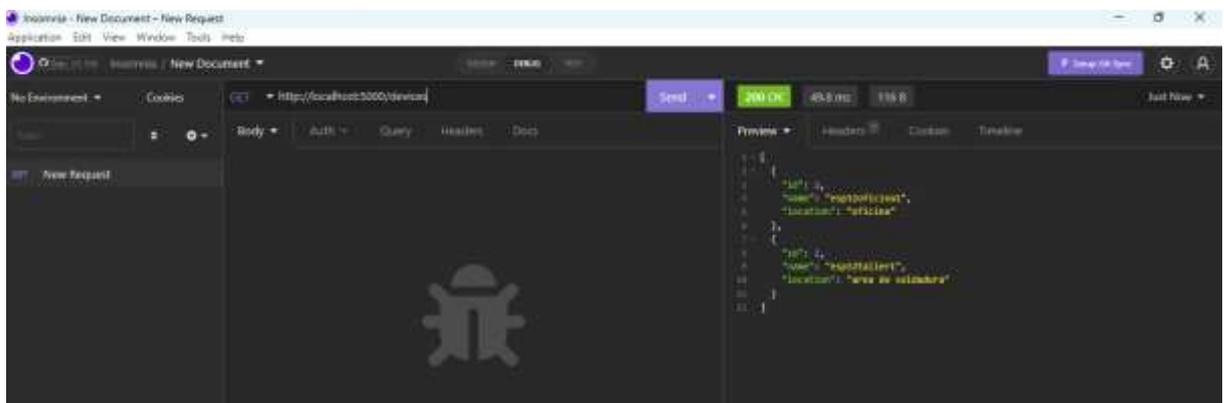


Figura 2. 17 Pantalla del requerimiento http listando los dispositivos

Usando el software de Insomnia se hace las pruebas a las rutas de manera local, en este caso se está probando la ruta /devices.

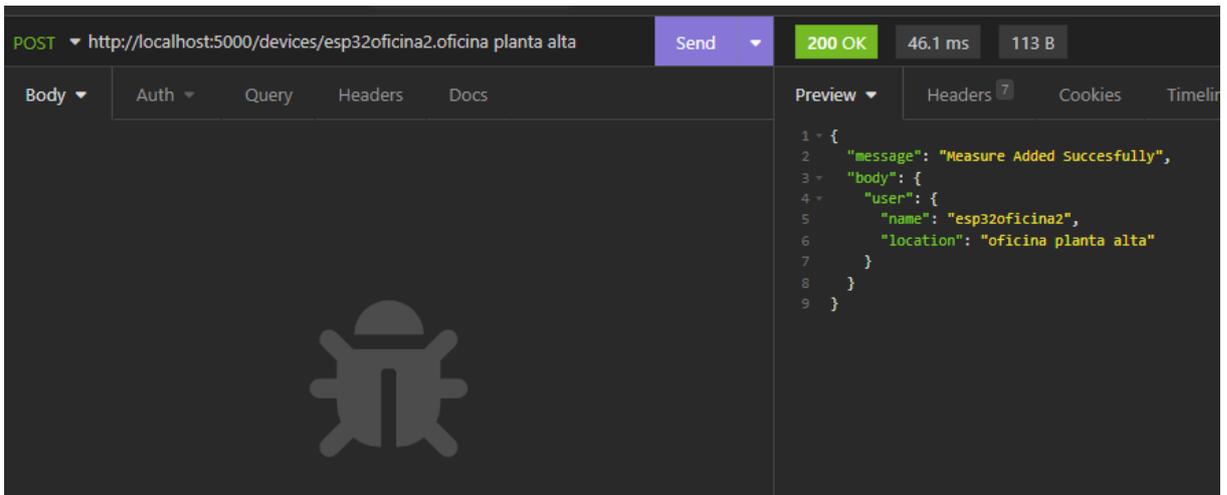


Figura 2. 18 Prueba de la ruta post /devices

En la figura 2.18 se muestra el resultado de la prueba ejecutada a la ruta post /devices con los parámetros nombre y localización del dispositivo.

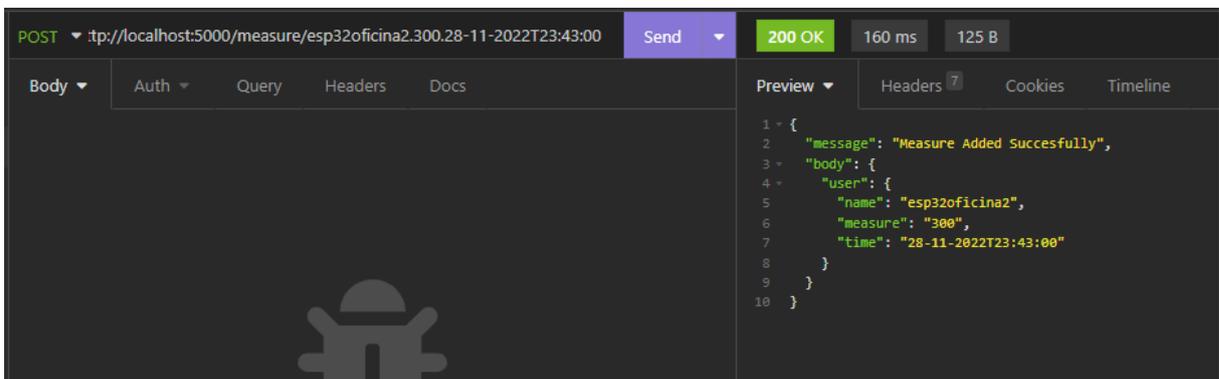


Figura 2. 19 Prueba de la ruta /measure

Finalmente, se procede a realizar la última prueba de la ruta /measure, la cual consiste en enviar los parámetros de nombre del dispositivo, medida en partes por millón (PPM) y tiempo de la medida.

2.9 Puesta en marcha esp-32

Mediante el uso de librerías adaptadas a la Esp-32, se conectó la red Mesh que permite la intercomunicación de los equipos y el envío al Api Rest para la gestión a la base de datos.

Message (Enter to send message to 'NodeMCU-32S' on 'COM3')

```
Node: 2
Valor: 698.00
Received from 365689597 msg={"nodo":2,"value":816}
Node: 2
Valor: 816.00
Received from 365689597 msg={"nodo":2,"value":711}
Node: 2
Valor: 711.00
Received from 365689597 msg={"nodo":2,"value":654}
Node: 2
Valor: 654.00
Received from 365689597 msg={"nodo":2,"value":925}
Node: 2
Valor: 925.00
Received from 365689597 msg={"nodo":2,"value":893}
Node: 2
Valor: 893.00
Received from 365689597 msg={"nodo":2,"value":799}
Node: 2
Valor: 799.00
Received from 365689597 msg={"nodo":2,"value":723}
Node: 2
Valor: 723.00
Received from 365689597 msg={"nodo":2,"value":783}
Node: 2
Valor: 783.00
```

Figura 2. 20 Recepción de la información del nodo 2 al nodo 1

En la imagen se observa cómo se recibe la información mediante impresión serial en el nodo 1 desde el nodo 2.

```
Received from 2662114169 msg={"nodo":1,"value":576}
Node: 1
Valor: 576.00
Received from 2662114169 msg={"nodo":1,"value":496}
Node: 1
Valor: 496.00
Received from 2662114169 msg={"nodo":1,"value":528}
Node: 1
Valor: 528.00
Received from 2662114169 msg={"nodo":1,"value":563}
Node: 1
Valor: 563.00
Received from 2662114169 msg={"nodo":1,"value":554}
Node: 1
Valor: 554.00
Received from 2662114169 msg={"nodo":1,"value":557}
Node: 1
Valor: 557.00
Adjusted time 831974385. Offset = 2184
Adjusted time 831980735. Offset = 1619
Received from 2662114169 msg={"nodo":1,"value":485}
Node: 1
Valor: 485.00
Received from 2662114169 msg={"nodo":1,"value":544}
Node: 1
Valor: 544.00
```

Figura 2. 21 Recepción de la información del nodo 1 al nodo 2

El diagrama de la figura 2.21 muestra la recepción de los datos enviados al nodo 2, en el cual se puede observar la información recibida y su correspondiente valor.

CAPITULO 3

3. Análisis y Resultados

Una vez realizada la selección de equipos y software a utilizar, se procederá a realizar su implementación de estos con el propósito de cumplir los objetivos planteados en este proyecto.

3.1 Interfaz

Un elemento clave del proyecto es la interfaz que permite llevar la información adquirida a información gráfica, la cual facilita una toma de decisiones.



Figura 3.1 Interfaz de las partículas presentes en el medio en el área de taller

La primera interfaz está basada en los datos que se recolectaron en un lapso de 15 días los cuales arrojaban gran cantidad de PPM. Los componentes de esta interfaz son un filtro por fecha la que se encarga de filtrar la información según avance el tiempo y un filtro por área interna. Se

tienen 3 tarjetas indicadoras las cuales manejan el promedio, mediana y desviación estándar del intervalo de tiempo filtrado y por último un gráfico de tendencias que refleja la recolección de datos a lo largo del tiempo.

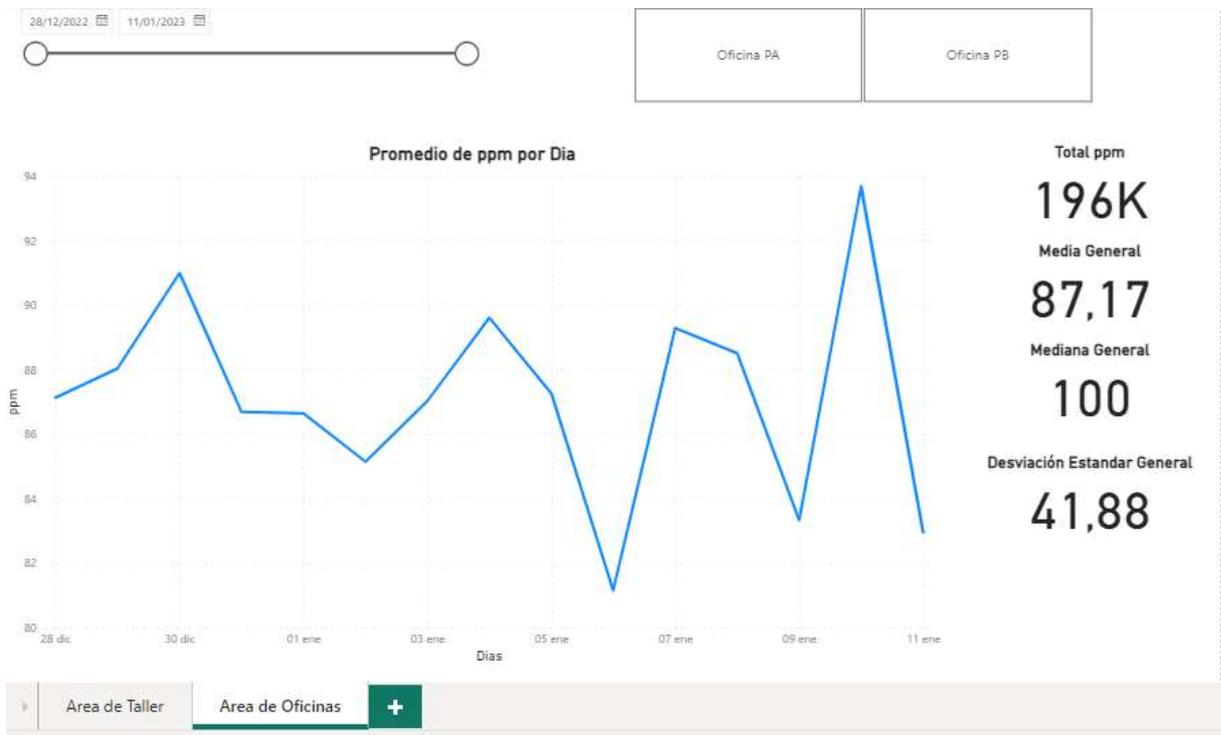


Figura 3. 2 Interfaz de las partículas presentes en el medio en el área de Oficina

Similar a la ventana anterior, la segunda interfaz cuenta con filtros por área y fecha, indicadores estadísticos y gráfico de tendencias.



Figura 3. 3 Dashboard del área de pintado

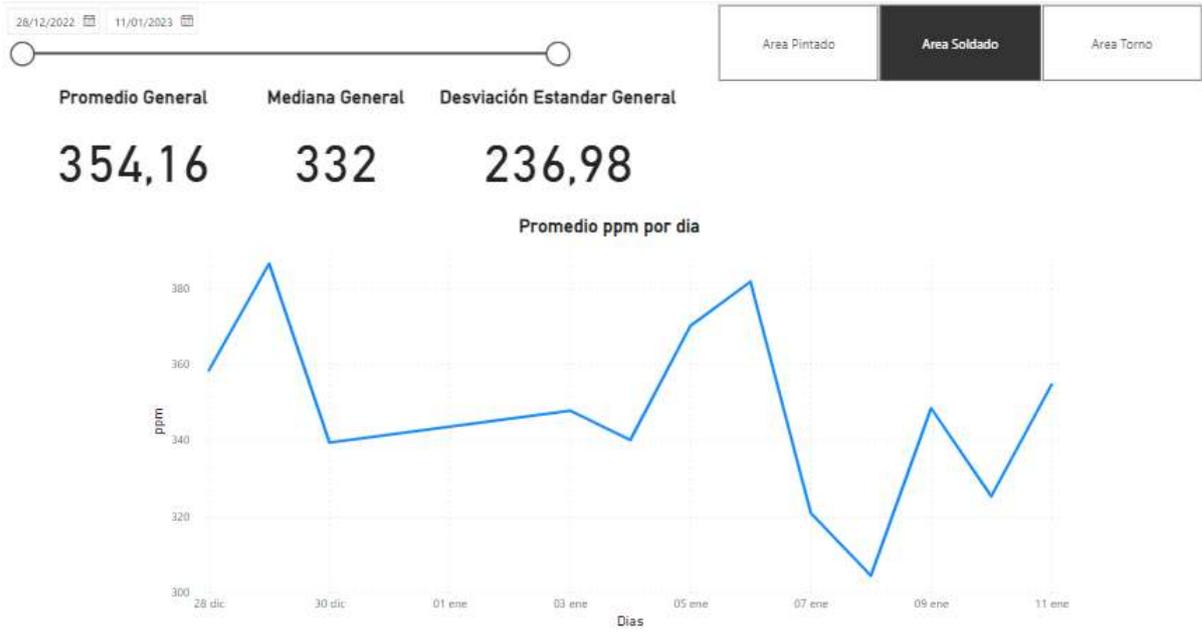


Figura 3. 4 Dashboard del área de soldado



Figura 3. 5 Dashboard del área de Torno

Como se muestra en las figuras 3.3, 3.4, 3.5 se tiene los datos estadísticos de cada área que compone la parte de taller, las que presentan una concentración de partículas que oscilan entre los 300 ppm en promedio, además de apreciarse días en donde el personal de un área esta más activa que en otras, lo que se debe al tipo de trabajos que se realizan en las diferentes áreas, más que todo cuando hay épocas de ensamblaje de equipos y reparación de los que se han vendido anteriormente.



Figura 3. 6 Dashboard del área de oficina planta alta.

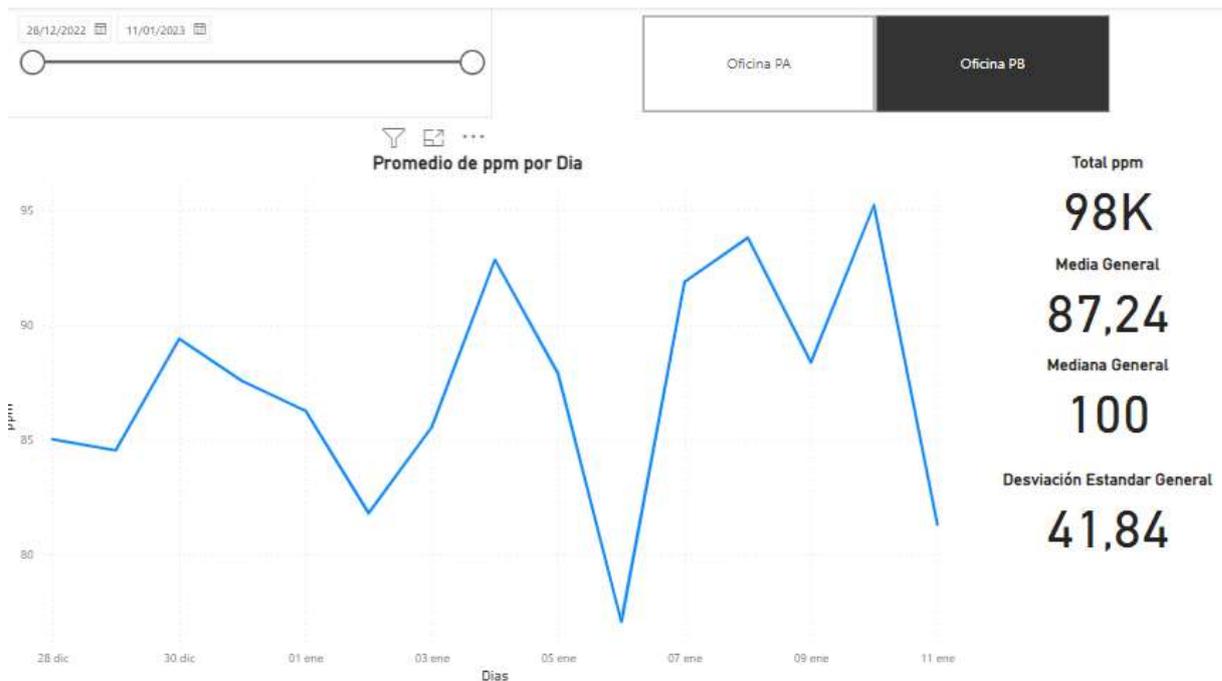


Figura 3. 7 Dashboard del área de oficina planta baja

En relación con las figuras 3.6 y 3.7 las concentraciones de ppm de las oficinas son menores en comparación con las de taller. En la planta baja de la oficina presenta mayores picos de concentraciones en promedio, debido a que el personal se encuentra más tiempo en ese sector. La planta alta presenta esos picos cuando el personal se moviliza para capacitaciones o reuniones, ya que la planta cuenta con salas para estas actividades.

3.2 Tabla de costos

La tabla de costos se estableció bajo los modelos especificados en el capítulo anterior, por ende, el costo se va a dividir en dos partes, en costos de implementación y costos por servicio para entender mejor el panorama del proyecto.

Tabla 3. 1 Costo de los equipos

Equipo	Modelo	Tensión	Cantidad	Costo (USD)	Total (USD)
Esp32	Node-MCU	3.3V/5V	3	12	36
Sensor calidad de aire	MQ135	3.3V/5V	3	4.50	13.50
Placa PCB			3	2.50	7.50
Cargador USB	Samsung	5V	3	8	24
Carcasa			3	15	45
Total					\$112.50

Tabla 3. 2 Costo de los Servicios

Servicio	Tipo	Cantidad	Costo(USD)	Total(USD)
Servidor web	mensual	12	5	60
Power Bi	mensual	0	0	0
Total				\$60

El software a utilizar incluye un servidor en la nube levantado en Digital Ocean, con un costo alrededor de 5 dólares mensuales y una cuenta gratuita de Power Bi para realizar las operaciones de conexión a la base de datos, por ende, el costo total del proyecto es de 172.50 dólares americanos, representando un beneficio económico a la empresa mediante la mejora del confort del empleado.

Con respecto al costo por dispositivo planteado en el proyecto es de 42 dólares americanos. En comparación con otras alternativas en el mercado, como Nextgen y Fractall, nuestro proyecto ofrece una solución más económica para la medición de la calidad del aire. Nextgen ofrece dispositivos de medición que parten desde los 80 dólares americanos en adelante, mientras que Fractall ofrece una amplia variedad de dispositivos, siendo el más parecido a nuestro proyecto AIRQ, el cual tiene un costo de 120 dólares más el servicio que ofrecen en su plataforma, lo que genera un mayor costo de implementación y operación. En cambio, nuestro proyecto es más accesible tanto en términos de

costo inicial como de operación a largo plazo, lo que lo hace más atractivo para los usuarios.

3.3 Diseño del Prototipo



Figura 3. 7 Prototipo físico

El prototipo físico está protegido por una carcasa de madera cerrada en el interior para evitar la entrada de polvo, en la salida se encuentra el módulo para medir la calidad de aire en los diferentes ambientes. Internamente cuenta con una batería y ranuras para la salida de calor, además tiene una ranura para cargar los dispositivos por USB.

CAPITULO 4

4. Conclusiones y Recomendaciones

4.1 Conclusiones

- La tecnología implementada da la posibilidad de llevar un control de la liberación de partículas pesadas en las áreas de taller y oficina mediante una interfaz la cual tiene una facilidad para ser actualizada de manera constante.
- El sistema propuesto tiene como objetivo mostrar la realidad de los ambientes laborales en los que los empleados se encuentran expuestos a diferentes niveles de partículas presentes en el aire. Se trata de un proyecto que se enfoca en los trabajadores de talleres, donde los niveles de partículas pueden ser elevados, en comparación con aquellos que trabajan en oficinas, donde los niveles son generalmente más saludables. De esta manera, el sistema permitirá a los empleados conocer el estado de la calidad del aire en su entorno laboral, lo que les permitirá tomar medidas preventivas para cuidar su salud y bienestar.
- Desarrollar el prototipo representa más de un 50% de ahorro frente a alternativas del mercado ecuatoriano y colombiano, permitiendo que se pueda desarrollar de manera sólida una implementación a mayor escala dentro de la empresa.
- La adquisición de datos que realiza el prototipo permite obtener y presentar los datos en tiempo real, facilitando a las áreas gerenciales datos que les permitan realizar toma de decisiones con respecto a la salud y seguridad de los empleados y las áreas en donde estos laboran.

4.2 Recomendaciones

- El presente proyecto al ser desarrollado con código abierto y dispositivos flexibles permite actualizaciones y mejoras en el futuro en función del cliente, como el poder agregar más sensores y más dispositivos a la red.
- Un buen control sobre los dispositivos permite preservar y prolongar la vida útil de los equipos a utilizar, para lo que se recomienda estrategias como mantenimientos preventivos y adquisición anticipada de existencias para piezas de recambio.
- Se recomienda considerar la escalabilidad del proyecto mediante la utilización de las tecnologías implementadas tanto en el hardware como en el software. Las tecnologías empleadas en la parte web son altamente eficientes y están actualizadas debido al gran respaldo de una comunidad que se siente identificada con dichas tecnologías. Esta decisión permitirá una mayor flexibilidad y adaptabilidad en el futuro, lo que puede resultar beneficioso para el proyecto a largo plazo.

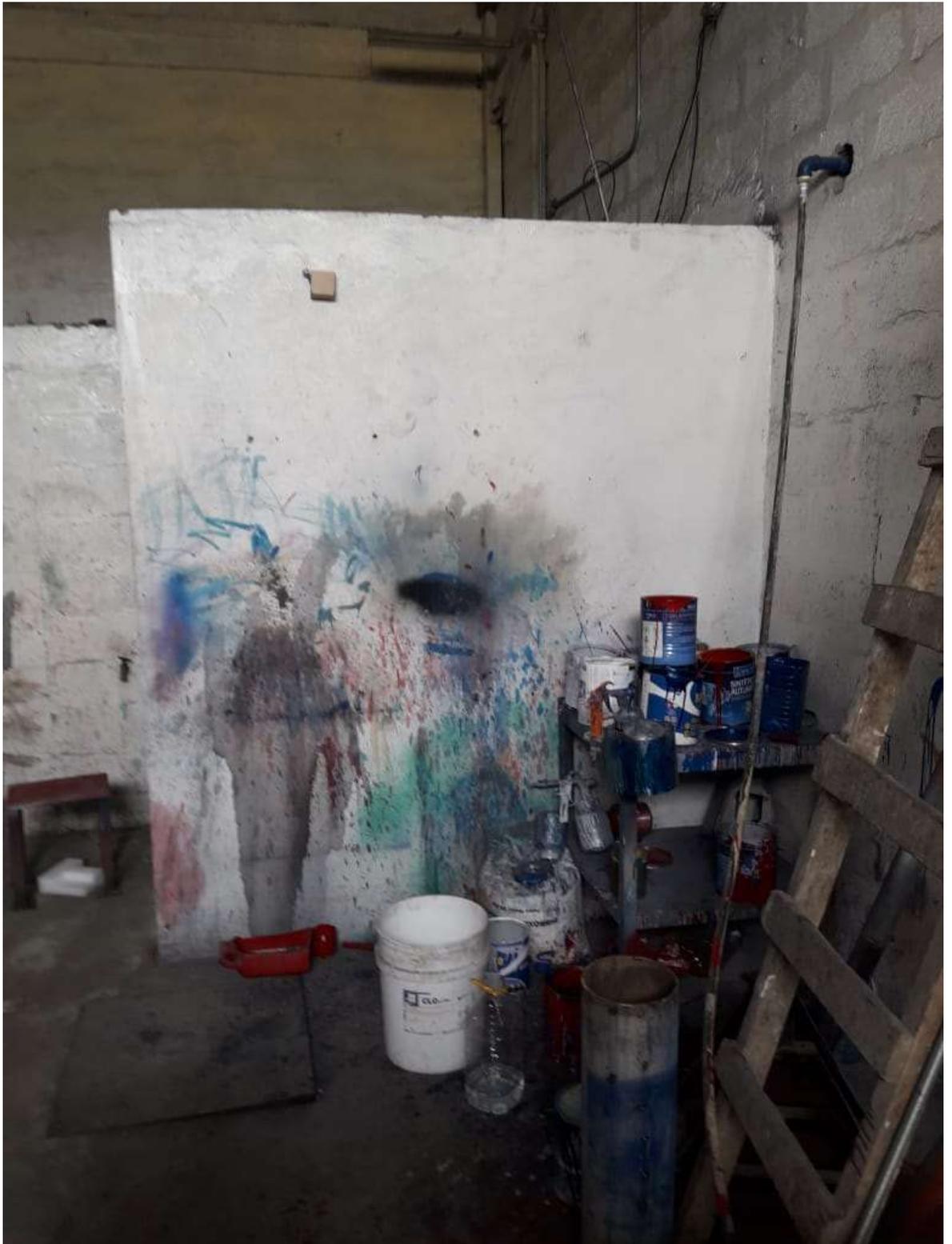
Referencias

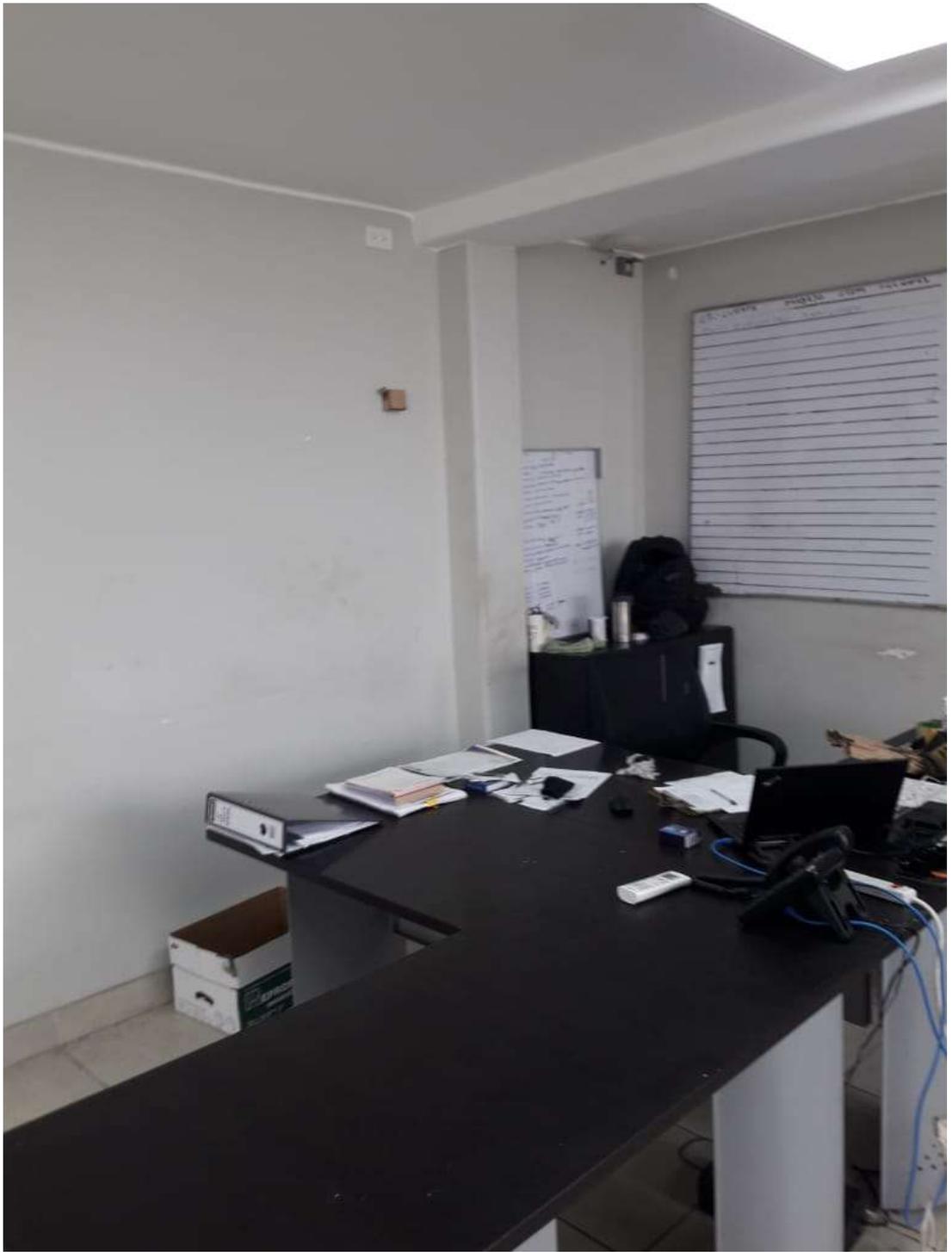
- Brik, V., Rayanchu, S., Saha, S., Sen, S., & Shrivastava, V. (2008). *A Measurement Study of a Commercial-grade*. Madison.
doi:<https://dl.acm.org/doi/pdf/10.1145/1452520.1452534>
- Camfil. (20 de Febrero de 2019). *camfil.com*. Obtenido de <https://www.camfil.com/es-es/insights/estandares-y-normativas/pm1-is-most-harmful>
- Fernández, Y. (24 de Septiembre de 2020). *Que es una red mesh y como funcionan y en que se diferencian*. Obtenido de <https://www.xataka.com/basics/que-red-mesh-como-funciona-que-se-diferencian-repetidor-plc>
- Ministerio para la transición ecológica y el reto demográfico. (2007). *PRTR-Espana*. Obtenido de <https://prtr-es.es/particulas-pm10,15673,11,2007.html>
- Mozilla Corporation. (25 de Noviembre de 2022). *MDN web docs*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- OEHHA. (2022). *oehha.ca.gov*. Obtenido de <https://oehha.ca.gov/calenviroscreen/indicator/pm25>
- Papayanopoulos, L. M. (2011). *Guía metodológica para la estimación de emisiones PM2.5*. Mexico. Obtenido de <http://www2.inecc.gob.mx/publicaciones2/libros/695/queson.pdf>
- Quesada, J. L. (2016). *IES San Juan Bosco*. Obtenido de http://dis.um.es/~lopezquesada/documentos/IES_1617/SRI/curso/UT2/UT2.pdf
- Somayya Madakam, R. R. (2015). *Internet of Things (IoT): A Literature*. India: Scientific Research Publishing. doi:10.4236/jcc.2015.35021

ANEXOS

Anexo 1: Montaje de los equipos







Anexo 2: Códigos de programación Arduino

Package.json

```
{
  "name": "web_server",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "nodemon src/index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2",
    "pg": "^8.8.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.20"
  }
}
```

Package-lock.json

```
{
  "name": "web_server",
  "version": "1.0.0",
  "lockfileVersion": 1,
  "requires": true,
  "dependencies": {
    "abbrev": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/abbrev/-/abbrev-1.1.1.tgz",
```

```

    "integrity": "sha512-
nne9/liQ/hzlhY6pdDnbBtz7DjPTKrY00P/zvPSm5pOFkl6xuGrGnXn/VtTNNf
NtAfZ9/1RtehkszU9qcTii0Q==",
    "dev": true
  },
  "accepts": {
    "version": "1.3.8",
    "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
    "integrity": "sha512-
PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSAeXEQ+
NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",
    "requires": {
      "mime-types": "~2.1.34",
      "negotiator": "0.6.3"
    }
  },
  "anymatch": {
    "version": "3.1.3",
    "resolved": "https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz",
    "integrity": "sha512-
KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyv
VV7D/zsdEbNnV6acZUutkiHQXvTr1Rw==",
    "dev": true,
    "requires": {
      "normalize-path": "^3.0.0",
      "picomatch": "^2.0.4"
    }
  },
  "array-flatten": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-
1.1.1.tgz",

```

```

    "integrity": "sha512-
PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx
8nNIM++NqRcK6CxxpUafjmhldKiHibqg=="
  },
  "balanced-match": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-
match-1.0.2.tgz",
    "integrity": "sha512-
3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbrRDZOpH0KW9+CeX4bRAaX0
Anxt0tx2MrpRpWwQaPwllISEJhYU5Pw==",
    "dev": true
  },
  "binary-extensions": {
    "version": "2.2.0",
    "resolved": "https://registry.npmjs.org/binary-extensions/-/binary-
extensions-2.2.0.tgz",
    "integrity": "sha512-
jDctJ/IVQbZoJykoeHbhXpOINBqGNcwXJKJog42E5HDPUwQTSdjCHdihj0
DlnheQ7blbT6dHOafNAiS8ooQKA==",
    "dev": true
  },
  "body-parser": {
    "version": "1.20.1",
    "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-
1.20.1.tgz",
    "integrity": "sha512-
jWi7abTbYwajOytWCQc37VulmWiRae5RyTpaCyDcS5/IMdtwSz5IOpDE67s
rw/HYe35f1z3fDQw+3txg7gNtWw==",
    "requires": {
      "bytes": "3.1.2",
      "content-type": "~1.0.4",
      "debug": "2.6.9",
      "depd": "2.0.0",

```

```

    "destroy": "1.2.0",
    "http-errors": "2.0.0",
    "iconv-lite": "0.4.24",
    "on-finished": "2.4.1",
    "qs": "6.11.0",
    "raw-body": "2.5.1",
    "type-is": "~1.6.18",
    "unpipe": "1.0.0"
  }
},
"brace-expansion": {
  "version": "1.1.11",
  "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-
expansion-1.1.11.tgz",
  "integrity": "sha512-iCuPHDFgrHX7H2vEI/5xpz07zSHB00TpugqhmYtVmmMO6518mCuRMoOY
FIdEBI0g187ufozdaHgWKcYFb61qGiA==",
  "dev": true,
  "requires": {
    "balanced-match": "^1.0.0",
    "concat-map": "0.0.1"
  }
},
"braces": {
  "version": "3.0.2",
  "resolved": "https://registry.npmjs.org/braces/-/braces-3.0.2.tgz",
  "integrity": "sha512-b8um+L1RzM3WDSzvhm6glz1yfTbBt6YTlcEKAvsmqCZZFw46z626IVj9j1y
EPW33H5H+IBQpZMP1k8l+78Ha0A==",
  "dev": true,
  "requires": {
    "fill-range": "^7.0.1"
  }
},

```

```

"buffer-writer": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/buffer-writer/-/buffer-writer-
2.0.0.tgz",
  "integrity": "sha512-
a7ZpuTZU1TRtnwyCNW3I5dc0wWNC3VR9S++Ewyk2HHZdrO3CQJqSpd
+95Us590V6AL7JqUAH2lwZ/398PmNFgw=="
},
"bytes": {
  "version": "3.1.2",
  "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",
  "integrity": "sha512-
/Nf7TyzTx6S3yRJOboAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8
P8zGkNXStjTSi6fzO6F0pBdcYbEg=="
},
"call-bind": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/call-bind/-/call-bind-1.0.2.tgz",
  "integrity": "sha512-
7O+FbCihRB5WGbFYesctwmTKae6rOilzmz1icreWJ+0aA7LJfuqhEso2T9nc
pcFtzMQtzXf2QGGueWJGTYSqrA==",
  "requires": {
    "function-bind": "^1.1.1",
    "get-intrinsic": "^1.0.2"
  }
},
"chokidar": {
  "version": "3.5.3",
  "resolved": "https://registry.npmjs.org/chokidar/-/chokidar-3.5.3.tgz",
  "integrity": "sha512-
Dr3sfKRP6oTcjf2JmUmFJfeVMvXBdegxB0iVQ5eb2V10uFJUCAS8OByZd
VAyVb8xXNz3GjjTgj9kLWsZTqE6kw=="
,
  "dev": true,
  "requires": {

```

```

    "anymatch": "~3.1.2",
    "braces": "~3.0.2",
    "fsevents": "~2.3.2",
    "glob-parent": "~5.1.2",
    "is-binary-path": "~2.1.0",
    "is-glob": "~4.0.1",
    "normalize-path": "~3.0.0",
    "readdirp": "~3.6.0"
  }
},
"concat-map": {
  "version": "0.0.1",
  "resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",
  "integrity": "sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFM RljaA7EXHaXZ8wsHdodFvbkhKmqg==",
  "dev": true
},
"content-disposition": {
  "version": "0.5.4",
  "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",
  "integrity": "sha512-FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1a AWA8cFIhHzMZUvLevkw5Rqk+tSQ==",
  "requires": {
    "safe-buffer": "5.2.1"
  }
},
"content-type": {
  "version": "1.0.4",
  "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.4.tgz",

```

```

    "integrity": "sha512-
hIP3EEPs8tB9AT1L+NUqtwOApis4mk2Zob89MwXMHjHWg9milF/j4osnnQ
LXBCBFBk/tvIG/tUc9mOUJiPBhPXA=="
  },
  "cookie": {
    "version": "0.5.0",
    "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.5.0.tgz",
    "integrity": "sha512-
YZ3GUyn/o8gfKJlnIX7g7xq4gyO6OSuhGPKaaGssGB2qgDUS0gPgtTvoyZ
LTt9Ab6dC4hfc9dV5arkvc/OCmrw=="
  },
  "cookie-signature": {
    "version": "1.0.6",
    "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-
signature-1.0.6.tgz",
    "integrity": "sha512-
QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQlrUgOWtHd
NDtPiywmFbiS12VjotlXlrKM3orQ=="
  },
  "debug": {
    "version": "2.6.9",
    "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
    "integrity": "sha512-
bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTK
Djm/E7AdgFBVeAPVMNcKGsHMA=="
    "requires": {
      "ms": "2.0.0"
    }
  },
  "depd": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",

```

```

    "integrity": "sha512-
g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3flh61/s/nPsfR6onyMwkC
AR/OIC3yBC0IESvUoQEAsslrw=="
  },
  "destroy": {
    "version": "1.2.0",
    "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",
    "integrity": "sha512-
2sJGJTaXllaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxB
pweaVcPevC2l3KpjYCx3NypQQgaJg=="
  },
  "ee-first": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
    "integrity": "sha512-
WMwm9LhRUo+WUaRN+vRuETqG89lgZphVSNkdFgeb6sS/E4OrDIN7t48
CAewSHXc6C8lefD8KKfr5vY61brQlow=="
  },
  "encodeurl": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
    "integrity": "sha512-
TPJXq8JqFaVYm2CWmPvnP2lYo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/r
TDf+rut/saiDiQEeVTNgAmJEdAOx0w=="
  },
  "escape-html": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-
1.0.3.tgz",
    "integrity": "sha512-
NiSupZ4OeuGwr68lGleym/kslZMJodUGOSZ/FSnTxcrekbvqrgdUxIJOMpij
aKZVjAJrWrGs/6Jy8OMuyj9ow=="
  },
  "etag": {

```

```

    "version": "1.8.1",
    "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
    "integrity": "sha512-
alL5Fx7mawVa300al2BnEE4iNvo1qETxLrPl/o05L7z6go7fCw1J6EQmbK4F
mJ2AS7kgVF/KEZWufBfdCIMcPg=="
  },
  "express": {
    "version": "4.18.2",
    "resolved": "https://registry.npmjs.org/express/-/express-4.18.2.tgz",
    "integrity": "sha512-
5/PsL6iGPdfQ/IKM1UuielYgv3BUoJfz1aUwU9vHZ+J7gyvwdQXFEBIEIaxe
Gf0GIcreATNyBExtalisDbuMqQ==",
    "requires": {
      "accepts": "~1.3.8",
      "array-flatten": "1.1.1",
      "body-parser": "1.20.1",
      "content-disposition": "0.5.4",
      "content-type": "~1.0.4",
      "cookie": "0.5.0",
      "cookie-signature": "1.0.6",
      "debug": "2.6.9",
      "depd": "2.0.0",
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "etag": "~1.8.1",
      "finalhandler": "1.2.0",
      "fresh": "0.5.2",
      "http-errors": "2.0.0",
      "merge-descriptors": "1.0.1",
      "methods": "~1.1.2",
      "on-finished": "2.4.1",
      "parseurl": "~1.3.3",
      "path-to-regexp": "0.1.7",
      "proxy-addr": "~2.0.7",

```

```

    "qs": "6.11.0",
    "range-parser": "~1.2.1",
    "safe-buffer": "5.2.1",
    "send": "0.18.0",
    "serve-static": "1.15.0",
    "setprototypeof": "1.2.0",
    "statuses": "2.0.1",
    "type-is": "~1.6.18",
    "utils-merge": "1.0.1",
    "vary": "~1.1.2"
  }
},
"fill-range": {
  "version": "7.0.1",
  "resolved": "https://registry.npmjs.org/fill-range/-/fill-range-7.0.1.tgz",
  "integrity": "sha512-qOo9F+dMUmC2Lcb4BbVvnKJxTPjCm+RRpe4gDuGrzkL7mEVI/djYSu2O
dQ2Pa302N4oqkSg9ir6jaLWJ2USVpQ==",
  "dev": true,
  "requires": {
    "to-regex-range": "^5.0.1"
  }
},
"finalhandler": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-
1.2.0.tgz",
  "integrity": "sha512-5uXcUVftIQMFnWC9qu/svkWv3GTd2PfUhK/3PLkYNAe7FbqjMt3515HaxE
6eRL74GdsriiwujiawdaB1BpEISg==",
  "requires": {
    "debug": "2.6.9",
    "encodeurl": "~1.0.2",
    "escape-html": "~1.0.3",

```

```

    "on-finished": "2.4.1",
    "parseurl": "~1.3.3",
    "statuses": "2.0.1",
    "unpipe": "~1.0.0"
  }
},
"forwarded": {
  "version": "0.2.0",
  "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",
  "integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9w
yablzMzOA5z9xA9L1KNjk/Nt6MT9aYow=="
},
"fresh": {
  "version": "0.5.2",
  "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
  "integrity": "sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9l/XBpm7kqgMywgmylMw
XHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q=="
},
"fsevents": {
  "version": "2.3.2",
  "resolved": "https://registry.npmjs.org/fsevents/-/fsevents-2.3.2.tgz",
  "integrity": "sha512-xiqMQR4xAeHTuB9uWm+fFRclOgKBMiOBP+eXiyT7jsgVCq1bkVygt00oA
SowB7EdtpOHaaPgKt812P9ab+DDKA==",
  "dev": true,
  "optional": true
},
"function-bind": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/function-bind/-/function-bind-
1.1.1.tgz",

```

```

    "integrity": "sha512-
ylovAzMX49sF8YI58fSCWJ5svSLuaibPxXQJFLmBOBtuCr0Mf1KiPopGM9
NiFjiYBCbfaa2Fh6breQ6ANVTI0A=="
  },
  "get-intrinsic": {
    "version": "1.1.3",
    "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-
1.1.3.tgz",
    "integrity": "sha512-
QJVz1Tj7MS099PevUG5jvnt9tSkXN8K14dxQlikJuPt4uD9hHAHjLyLBiLR5z
ELelBdD9QNRAXZzsJx0WaDL9A=="
  },
  "requires": {
    "function-bind": "^1.1.1",
    "has": "^1.0.3",
    "has-symbols": "^1.0.3"
  }
},
"glob-parent": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-
5.1.2.tgz",
  "integrity": "sha512-
AOlgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC
9hnbcOc9/SrMyM5RPQrkGz4aS9Zow=="
  },
  "dev": true,
  "requires": {
    "is-glob": "^4.0.1"
  }
},
"has": {
  "version": "1.0.3",
  "resolved": "https://registry.npmjs.org/has/-/has-1.0.3.tgz",

```

```

    "integrity": "sha512-
f2dvO0VU6Oej7RkWJGrehjzMAjFp5/VKPP5tTpWIV4JHHZK1/BxbFRtf/siA
2SWTe09caDmVtYYzWEIbBS4zw==",
    "requires": {
      "function-bind": "^1.1.1"
    }
  },
  "has-flag": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/has-flag/-/has-flag-3.0.0.tgz",
    "integrity": "sha512-
sKJf1+ceQBr4SMkvQnBDNDtf4TXpVhVGateu0t918bl30FnBE2m4vNLX+V
We/dpjlB+HugGYzW7uQXH98HPEYw==",
    "dev": true
  },
  "has-symbols": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-
1.0.3.tgz",
    "integrity": "sha512-
I3LCuF6MgDNwTDKkdYGEihYjt5pRPbEg46rtImnSPIUbgmB8LOIrKJbYYF
BSbnPaJexMKtiPO8hmeRjRz2Td+A=="
  },
  "http-errors": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",
    "integrity": "sha512-
FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnVeh2Z1pIBdXgbyjv34pHTSb9zoeH
MyDy33+DWy5Wt9Wo+TURtOYSQ=="
  },
  "requires": {
    "depd": "2.0.0",
    "inherits": "2.0.4",
    "setprototypeof": "1.2.0",
    "statuses": "2.0.1",

```

```

    "toidentifier": "1.0.1"
  }
},
"iconv-lite": {
  "version": "0.4.24",
  "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",
  "integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQ
Wne1TcRwhCL1HwLI21bEqdpj8/rA==",
  "requires": {
    "safer-buffer": ">= 2.1.2 < 3"
  }
},
"ignore-by-default": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/ignore-by-default/-/ignore-by-
default-1.0.1.tgz",
  "integrity": "sha512-lus2VYcGNk7T90CppJqclkS5ooHUZYlQK+CIZfMfMNF9VSE73Fq+906u
/CWu92x4gzZMWOWfFYckPObzdEbA==",
  "dev": true
},
"inherits": {
  "version": "2.0.4",
  "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
  "integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+En
BiSM9zGSoIFeMpXKxa4dYeZIQqewQ=="
},
"ipaddr.js": {
  "version": "1.9.1",
  "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",

```

```

    "integrity": "sha512-
0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW
3R+akZ6R/w18ZIXSHBYXiYUPO3g=="
  },
  "is-binary-path": {
    "version": "2.1.0",
    "resolved": "https://registry.npmjs.org/is-binary-path/-/is-binary-path-
2.1.0.tgz",
    "integrity": "sha512-
ZMERYes6pDydyuGidse7OsHxtbl7WVeUEozgR/g7rd0xUimYNlvZRE/K2M
gZTjWy725lfeLeVcEM97mmtRGXw==",
    "dev": true,
    "requires": {
      "binary-extensions": "^2.0.0"
    }
  },
  "is-extglob": {
    "version": "2.1.1",
    "resolved": "https://registry.npmjs.org/is-extglob/-/is-extglob-2.1.1.tgz",
    "integrity": "sha512-
SbKbANKn603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhG
J6QEDpOZIFkDtnq5JRxmvl3jsoQ==",
    "dev": true
  },
  "is-glob": {
    "version": "4.0.3",
    "resolved": "https://registry.npmjs.org/is-glob/-/is-glob-4.0.3.tgz",
    "integrity": "sha512-
xelSayHH36ZgE7ZWwhli7pW34hNbNI8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM
+mXWOZhFkgZfxhLSnrwRr4elSSg==",
    "dev": true,
    "requires": {
      "is-extglob": "^2.1.1"
    }
  }

```

```

    },
    "is-number": {
      "version": "7.0.0",
      "resolved": "https://registry.npmjs.org/is-number/-/is-number-7.0.0.tgz",
      "integrity": "sha512-41Cifkg6e8TyISpdtTpeLVMqvSBEVzTtHvERD741+pnZ8ANv0004MRL43Q
      KPDIK9cGvNp6NZWZUBlbGXYxxng==",
      "dev": true
    },
    "media-typer": {
      "version": "0.3.0",
      "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-
      0.3.0.tgz",
      "integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJY
      mUmknUF5HwMLOgb5O+a3KxfWapPQ=="
    },
    "merge-descriptors": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-
      descriptors-1.0.1.tgz",
      "integrity": "sha512-cCi6g3/Zr1iqQi6ySbseM1Xvooa98N0w31jzUYrXPX2xqObmFGHJ0tQ5u74
      H3mVh7wLouTseZyYlq39g8cNp1w=="
    },
    "methods": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
      "integrity": "sha512-cCi6g3/Zr1iqQi6ySbseM1Xvooa98N0w31jzUYrXPX2xqObmFGHJ0tQ5u74
      H3mVh7wLouTseZyYlq39g8cNp1w=="
    },
    "mime": {
      "version": "1.6.0",

```

```

    "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
    "integrity": "sha512-
x0Vn8spl+wuJ1O6S7gnbaQg8Pxb4NNHb7KSINmEWKiPE4RKOpIvijn+Nk
mYmmRgP68mc70j2EbeTFRsrswaQeg=="
  },
  "mime-db": {
    "version": "1.52.0",
    "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
    "integrity": "sha512-
sPU4uV7dYlvtWJxwwwHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwY
QQOKPJ8ClbE+1ETVIOoK1UC2nU3gYvg=="
  },
  "mime-types": {
    "version": "2.1.35",
    "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-
2.1.35.tgz",
    "integrity": "sha512-
ZDY+bPm5zTTF+YpCrAU9nK0UglCYPT0QtT1NZWFv4s++TNkcgVaT0g6
+4R2ul4MjQjzysHB1zXuWL50hzaeXiw==",
    "requires": {
      "mime-db": "1.52.0"
    }
  },
  "minimatch": {
    "version": "3.1.2",
    "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",
    "integrity": "sha512-
J7p63hRiAjw1NDEww1W7i37+BylR0WO5XQQAzz3VOcL0PNybwpmV/N0
5zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",
    "dev": true,
    "requires": {
      "brace-expansion": "^1.1.7"
    }
  },

```

```

"ms": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
  "integrity": "sha512-
Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGM
oFjC9ddhal5KVIYtAt97ix05A=="
},
"negotiator": {
  "version": "0.6.3",
  "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",
  "integrity": "sha512-
+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGIQk/k5giakcKsuxCObB
Ru6DSm9opw/O6sIWbJdghQM4bBg=="
},
"nodemon": {
  "version": "2.0.20",
  "resolved": "https://registry.npmjs.org/nodemon/-/nodemon-2.0.20.tgz",
  "integrity": "sha512-
Km2mWHKKY5GzRg6i1j5OxOHQtuvVsgskLfigG25yTtbyfRGn/GNvlbRyOf1
PSCKJ2aT/58TiuUsuOU5UToVViw==",
  "dev": true,
  "requires": {
    "chokidar": "^3.5.2",
    "debug": "^3.2.7",
    "ignore-by-default": "^1.0.1",
    "minimatch": "^3.1.2",
    "pstree.remy": "^1.1.8",
    "semver": "^5.7.1",
    "simple-update-notifier": "^1.0.7",
    "supports-color": "^5.5.0",
    "touch": "^3.1.0",
    "undefsafe": "^2.0.5"
  },
  "dependencies": {

```

```

    "debug": {
      "version": "3.2.7",
      "resolved": "https://registry.npmjs.org/debug/-/debug-3.2.7.tgz",
      "integrity": "sha512-
CFjzYYAi4ThfiQvizrFQevTTXHtnCqWfe7x1AhgEscTz6ZbLbfoLRLPugTQy
Bth6f8ZERVUSyWHFD/7Wu4t1XQ==",
      "dev": true,
      "requires": {
        "ms": "^2.1.1"
      }
    },
    "ms": {
      "version": "2.1.3",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
      "integrity": "sha512-
6FlzubTLZG3J2a/NVCAleEhjq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP
6tz7R9xAOtHnSO/tXtF3WRTIA==",
      "dev": true
    }
  },
  "nopt": {
    "version": "1.0.10",
    "resolved": "https://registry.npmjs.org/nopt/-/nopt-1.0.10.tgz",
    "integrity": "sha512-
NWmpvLSqUrgrAC9HCuxEvb+PSloHpqVu+FqcO4eeF2h5qYRrhA7ev6Kvel
yQAKtegUbC6RypJnlEOhd8vloNKYg==",
    "dev": true,
    "requires": {
      "abbrev": "1"
    }
  },
  "normalize-path": {
    "version": "3.0.0",

```

```

    "resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-
3.0.0.tgz",
    "integrity": "sha512-
6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwU
B6C6lKbQkY2Pnb/mD4WYojCRwcvLA==",
    "dev": true
  },
  "object-inspect": {
    "version": "1.12.2",
    "resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-
1.12.2.tgz",
    "integrity": "sha512-
z+cPxW0QGUUp0mcqcsGQyLVRDoXFQbXOwBaQyF7VIgl4TWNQsDHRBpU
QslRmlfAoYWdYzs6UIKJtB2XJpTaNspQ=="
  },
  "on-finished": {
    "version": "2.4.1",
    "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-
2.4.1.tgz",
    "integrity": "sha512-
oVlzkg3ENAhCk2zdV7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWA
LHM+brtuJjePWiYF/ClmuDr8Ch5+kg=="
  },
  "requires": {
    "ee-first": "1.1.1"
  }
},
  "packet-reader": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/packet-reader/-/packet-reader-
1.0.0.tgz",
    "integrity": "sha512-
HAKu/fG3HpHFO0AA8WE8q2g+gBJaZ9MG7fcKk+IJPLTGAD6Psw4443l+
9DGRbOIh3/aXr7Phy0TjilYivJo5XQ=="
  },

```

```

    "parseurl": {
      "version": "1.3.3",
      "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
      "integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ=="
    },
    "path-to-regexp": {
      "version": "0.1.7",
      "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.7.tgz",
      "integrity": "sha512-5DFkuoqlv1uYQKxy8omFBeJPQcdoE07Kv2sferDCrAq1ohOU+MSDswDIbnx3YAM60qlOnYa53wBhXW0EbMonrQ=="
    },
    "pg": {
      "version": "8.8.0",
      "resolved": "https://registry.npmjs.org/pg/-/pg-8.8.0.tgz",
      "integrity": "sha512-UXYN0ziKj+AeNNP7VDMwrehpACThH7LUI/p8TDFpEUuSejCUIwGSfxpHsPvtM6/WXFy6SU4E5RG4IJV/TZAGjw==",
      "requires": {
        "buffer-writer": "2.0.0",
        "packet-reader": "1.0.0",
        "pg-connection-string": "^2.5.0",
        "pg-pool": "^3.5.2",
        "pg-protocol": "^1.5.0",
        "pg-types": "^2.1.0",
        "pgpass": "1.x"
      }
    },
    "pg-connection-string": {
      "version": "2.5.0",

```

```

    "resolved": "https://registry.npmjs.org/pg-connection-string/-/pg-connection-string-2.5.0.tgz",
    "integrity": "sha512-r5oV//ORTA6TmUnyWZR9nCj1kIXCO2CEKNRIVuJptZe85QuhFayC7WeMicc7ndayT5IRIR0S0xFxFi2ousartIQ==",
  },
  "pg-int8": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/pg-int8/-/pg-int8-1.0.1.tgz",
    "integrity": "sha512-WCtabS6t3c8SkepDBUIb1kjOs7l66xsGdKpIPZsg4wR+B3+u9UAum2odSsF9tnvxg80h4ZxLWMy4pRjOsFlqQpw==",
  },
  "pg-pool": {
    "version": "3.5.2",
    "resolved": "https://registry.npmjs.org/pg-pool/-/pg-pool-3.5.2.tgz",
    "integrity": "sha512-His3Fh17Z4eg7oANLob6ZvH8xIVen3phEZh2Quyrll4dQSDVEabNducv6ysROKpDNPSD+12tONZVWfSgMvDD9w==",
  },
  "pg-protocol": {
    "version": "1.5.0",
    "resolved": "https://registry.npmjs.org/pg-protocol/-/pg-protocol-1.5.0.tgz",
    "integrity": "sha512-muRttij7H8TqRNU/DxrAJQITO4Ac7RmX3Klyr/9mJEObelpgnF8f9jAfRz5d3XwQZI5qBjF9gLSUtMPJE0vezQ==",
  },
  "pg-types": {
    "version": "2.2.0",
    "resolved": "https://registry.npmjs.org/pg-types/-/pg-types-2.2.0.tgz",
    "integrity": "sha512-qTAAIrEsl8s40iEQY69wDvcMldQN6wdz5ojQiOy6YRMuynxenON005oCpJl6lshc6scgAY8qvJ2On/p+CXY0GA==",
  }

```

```

    "requires": {
      "pg-int8": "1.0.1",
      "postgres-array": "~2.0.0",
      "postgres-bytea": "~1.0.0",
      "postgres-date": "~1.0.4",
      "postgres-interval": "^1.1.0"
    }
  },
  "pgpass": {
    "version": "1.0.5",
    "resolved": "https://registry.npmjs.org/pgpass/-/pgpass-1.0.5.tgz",
    "integrity": "sha512-
FdW9r/jQZhSeohs1Z3sl1yxFQNFvMcnmfuj4WBMUTxOrAyLMaTcE1aAMB
iTlbMNaXvBCQuVi0R7hd8udDSP7ug==",
    "requires": {
      "split2": "^4.1.0"
    }
  },
  "picomatch": {
    "version": "2.3.1",
    "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz",
    "integrity": "sha512-
JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fslvwaStHhAWJEEvd1o1QD8
OCU6+ZdEXXSLbSsuLwJjkCBWqRQUVA==",
    "dev": true
  },
  "postgres-array": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/postgres-array/-/postgres-array-
2.0.0.tgz",
    "integrity": "sha512-
VpZrUqU5A69eQyW2c5CA1jtLecCsN2U/bD6VilrFDWq5+5UIEVO7nazS3T
EcHf1zuPYO/sqGvUvW62g86RXZuA=="
  },

```

```

    "postgres-bytea": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/postgres-bytea/-/postgres-bytea-1.0.0.tgz",
      "integrity": "sha512-xy3pmLuQqRBZBXDULy7KbaitYqLcmxigw14Q5sj8QBVlqEwXfeybIKVWiqAXTlcvdvd0+xkOtdbfQMOF4IST1w=="
    },
    "postgres-date": {
      "version": "1.0.7",
      "resolved": "https://registry.npmjs.org/postgres-date/-/postgres-date-1.0.7.tgz",
      "integrity": "sha512-suDmjLVQg78nMK2UZ454hAG+OAW+HQPZ6n++TNDUX+L0+uUilywnoxJKDou51Zm+zTCjrCl0Nq6J9C5hP9vK/Q=="
    },
    "postgres-interval": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/postgres-interval/-/postgres-interval-1.2.0.tgz",
      "integrity": "sha512-9ZhXKM/rw350N1ovuWHbGxnGh/SNJ4cnxHiM0rxE4VN41wsg8P8zWn9hv/buK00RP4WvIOyr/RBDiptyxVbkZQ==",
      "requires": {
        "xtend": "^4.0.0"
      }
    },
    "proxy-addr": {
      "version": "2.0.7",
      "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
      "integrity": "sha512-llzwMqIw0wYD40YFwPcKdQYfX1YRhmWJ9zj7Hq2WqjwB93od7S9Rq6Y5PmKCY3kqkZRWdrfrzrWHdQ==",
      "requires": {

```

```

    "forwarded": "0.2.0",
    "ipaddr.js": "1.9.1"
  }
},
"pstree.remy": {
  "version": "1.1.8",
  "resolved": "https://registry.npmjs.org/pstree.remy/-/pstree.remy-1.1.8.tgz",
  "integrity": "sha512-77DZwxQmxKnu3aR542U+X8FypNzbfJ+C5XQDk3uWjWxn6151aIMGthWYRXTqT1E5oJvg+ljaa2OJi+VfvCOQ8w==",
  "dev": true
},
"qs": {
  "version": "6.11.0",
  "resolved": "https://registry.npmjs.org/qs/-/qs-6.11.0.tgz",
  "integrity": "sha512-MvjoMCJwEarSbUYk5O+nmoSzSutSsTwF85zcHPQ9OrIFoZOYljaqBAJlqlXjptyD5vThxGq52Xu/MaJzRklk4Q==",
  "requires": {
    "side-channel": "^1.0.4"
  }
},
"range-parser": {
  "version": "1.2.1",
  "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",
  "integrity": "sha512-HHcGhC835Q8W3u0b3B0YqWwV3qXz81kxZi613254T54z9Gq38N+jMVRxQ0G334b3B80y3664wVl3P7==",
  "integrity": "sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRozfG3ccAveqCBwr/2yxQ5BVd/GTI5agOwSg=="
},
"raw-body": {
  "version": "2.5.1",
  "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.1.tgz",

```

```

    "integrity": "sha512-
qqJBtEyVgS0ZmPGdCFPWJ3FreoqvG4MVQln/kCgF7OIq95IbOp0/BWyM
wbdtn4VTvkM8Y7khCQ2Xgk/tcrCXig==",
    "requires": {
      "bytes": "3.1.2",
      "http-errors": "2.0.0",
      "iconv-lite": "0.4.24",
      "unpipe": "1.0.0"
    }
  },
  "readdirp": {
    "version": "3.6.0",
    "resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz",
    "integrity": "sha512-
hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcffiMU3495FuTdQSm+7bhJ29
JvIOsBDEEan5DPu9t3To9VRIMzA==",
    "dev": true,
    "requires": {
      "picomatch": "^2.2.1"
    }
  },
  "safe-buffer": {
    "version": "5.2.1",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
    "integrity": "sha512-
rp3So07KcdmmKbGvgaNxQsJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5D
d8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ=="
  },
  "safer-buffer": {
    "version": "2.1.2",
    "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-
2.1.2.tgz",

```

```

    "integrity": "sha512-
YZo3K82SD7Riyi0E1EQPojLz7kpepnSQLI9IyPbHHg1XXXevb5dJI7tpyN2A
DxGcQbHG7vcyRHk0cbwqcQriUtg=="
  },
  "semver": {
    "version": "5.7.1",
    "resolved": "https://registry.npmjs.org/semver/-/semver-5.7.1.tgz",
    "integrity": "sha512-
sauaDf/PZdVgrLTNYHRtpXa1iRiKcaebiKQ1BJdpQIWH2ICvexQdX55snPFy
K7QzpuudqbCI0qXFfOasHdyNDGQ=="
    "dev": true
  },
  "send": {
    "version": "0.18.0",
    "resolved": "https://registry.npmjs.org/send/-/send-0.18.0.tgz",
    "integrity": "sha512-
qqWzuOjSFOuqPjFe4NOsMLafToQQwBSOEpS+FwEt3A2V3vKubTquT3v
mLTQpFgMXp8AIFWFuP1qKaJZOTppVXg=="
    "requires": {
      "debug": "2.6.9",
      "depd": "2.0.0",
      "destroy": "1.2.0",
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "etag": "~1.8.1",
      "fresh": "0.5.2",
      "http-errors": "2.0.0",
      "mime": "1.6.0",
      "ms": "2.1.3",
      "on-finished": "2.4.1",
      "range-parser": "~1.2.1",
      "statuses": "2.0.1"
    }
  },
  "dependencies": {

```

```

    "ms": {
      "version": "2.1.3",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
      "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP
6tz7R9xAOtHnSO/tXtF3WRTIA=="
    }
  },
  "serve-static": {
    "version": "1.15.0",
    "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-
1.15.0.tgz",
    "integrity": "sha512-XGuRDNjXUijsUL0vl6nSD7cwURuzEgglbOaFuZM9g3kwDXOWVTck0jLzjP
zGD+TazWbboZYu52/9/XPdUgne9g==",
    "requires": {
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "parseurl": "~1.3.3",
      "send": "0.18.0"
    }
  },
  "setprototypeof": {
    "version": "1.2.0",
    "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-
1.2.0.tgz",
    "integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrl
ODCrkoxMEeNi/XZlWuRvY4XNwYMJpw=="
  },
  "side-channel": {
    "version": "1.0.4",

```

```

    "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-
1.0.4.tgz",
    "integrity": "sha512-
q5XPytqFEIKHkGdiMlrY10mvLRvnQh42/+GoBIFW3b2LXLE2xxJpZFdm94
we0BaoV3RwJyGqg5wS7epxTv0Zvw==",
    "requires": {
      "call-bind": "^1.0.0",
      "get-intrinsic": "^1.0.2",
      "object-inspect": "^1.9.0"
    }
  },
  "simple-update-notifier": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/simple-update-notifier/-/simple-
update-notifier-1.1.0.tgz",
    "integrity": "sha512-
VpsrsJSUcJEseSbMHkrsrAVSdvVS5I96Qo1QAAQ4FxQ9wXFcB+pjj7FB7/us
9+GcgfW4ziHtYMc1J0PLczb55mg==",
    "dev": true,
    "requires": {
      "semver": "~7.0.0"
    },
    "dependencies": {
      "semver": {
        "version": "7.0.0",
        "resolved": "https://registry.npmjs.org/semver/-/semver-7.0.0.tgz",
        "integrity": "sha512-
+GB6zVA9LWh6zovYQLALHwv5rb2PHGIJi3IfiqIHxR0uuwCgefcoJc59v9fv
1w8GbStwxuuqqAjl9NMAOOgq1A==",
        "dev": true
      }
    }
  },
  "split2": {

```

```

    "version": "4.1.0",
    "resolved": "https://registry.npmjs.org/split2/-/split2-4.1.0.tgz",
    "integrity": "sha512-VBiJxFkxiXRIUleyMQi8s4hgvKCSjtknJv/LVYbrgALPwf5zSKmEwV9Lst25A
kvMDnvxODugjdl6KZgwKM1WYQ=="
  },
  "statuses": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
    "integrity": "sha512-RwNA9Z/7PrK06rYLizFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPX
QvB2fOC4AjuYSEndS7N/mTCbkdQ=="
  },
  "supports-color": {
    "version": "5.5.0",
    "resolved": "https://registry.npmjs.org/supports-color/-/supports-color-
5.5.0.tgz",
    "integrity": "sha512-QjVjwdXIt408MliAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFN
Tr4uMNZma0ey4f5IgLrkB0aX0QMow==",
    "dev": true,
    "requires": {
      "has-flag": "^3.0.0"
    }
  },
  "to-regexp-range": {
    "version": "5.0.1",
    "resolved": "https://registry.npmjs.org/to-regexp-range/-/to-regexp-range-
5.0.1.tgz",
    "integrity": "sha512-65P7iz6X5yEr1cwcgvQxbblw7Uk3gOy5dldtZ4rDveLqhrdJP+Li/Hx6tyK0NE
b+2GCyneCMJiGqrADCSNk8sQ==",
    "dev": true,
    "requires": {

```

```

    "is-number": "^7.0.0"
  }
},
"toidentifier": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
  "integrity": "sha512-
o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenISox/2O/BT
IZUtEe+JG7s5YhEz608PIAHRA=="
},
"touch": {
  "version": "3.1.0",
  "resolved": "https://registry.npmjs.org/touch/-/touch-3.1.0.tgz",
  "integrity": "sha512-
WBx8Uy5TLtOSRtlq+M03/sKDrXCLHxwDcquSP2c43Le03/9serjQBIztjRz6
FkJez9D/hleyAXTBGLwwZUw9IA==",
  "dev": true,
  "requires": {
    "nopt": "~1.0.10"
  }
},
"type-is": {
  "version": "1.6.18",
  "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
  "integrity": "sha512-
TkRKR9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCn
IFPW4ky9Q+iA+ma9BGm06XQBy8g==",
  "requires": {
    "media-typer": "0.3.0",
    "mime-types": "~2.1.24"
  }
},
"undefsafe": {
  "version": "2.0.5",

```

```

    "resolved": "https://registry.npmjs.org/undefsafe/-/undefsafe-2.0.5.tgz",
    "integrity": "sha512-
WxONCrssBM8TSPRqN5EmsjVrsv4A8X12J4ArBiiayv3DyyG3ZIIg6yysuuS
YdZsVz3TKcTg2fd//Ujd4CHV1iA==",
    "dev": true
  },
  "unpipe": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
    "integrity": "sha512-
pjy2bYhSsufwWIKwPc+l3cN7+wuJIK6uz0YdJEOIQDbI6jo/YIPi4mb8agUkV
C8BF7V8NuzeyPNqRksA3hztKQ=="
  },
  "utils-merge": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-
1.0.1.tgz",
    "integrity": "sha512-
pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybb
sHHoED/ztMtkv/VoYTYyShUn81hA=="
  },
  "vary": {
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
    "integrity": "sha512-
BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQ
OzmAKzYqYRAISoRhdQr3eIZ/PXqg=="
  },
  "xtend": {
    "version": "4.0.2",
    "resolved": "https://registry.npmjs.org/xtend/-/xtend-4.0.2.tgz",
    "integrity": "sha512-
LKYU1iAXJXUgAXn9URjju+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAxD
82/mCWbtLsGjFlad0wlsod4zrTAEQ=="
  }
}

```

```
    }  
  }  
}
```

Controller.js

```
const {Pool} = require('pg');
```

```
const pool= new Pool({  
  host:'localhost',  
  user:'gruiz',  
  password:'gruiz1234',  
  database:'esp32database',  
  port:'5432'  
});
```

```
const getdevices = async (req,res)=>{  
  const response = await pool.query("SELECT * FROM devices");  
  res.status(200).json(response.rows);  
};
```

```
const createDevices = async(req,res)=>{  
  const name = req.params.name;  
  const location = req.params.location;  
  const response = await pool.query('INSERT INTO devices (name,  
location) VALUES ($1, $2)',[name,location]);  
  res.json({  
    message: 'Measure Added Succesfully',  
    body: {  
      user:{name, location}  
    }  
  })  
};
```

```

const uploadMeasures = async (req,res)=>{
  const name = req.params.name;
  const measure = req.params.measure;
  const time = req.params.time;
  const response = await pool.query('INSERT INTO measures (name,
measure, time_measure) VALUES ($1, $2, $3)',[name,measure,time])
  res.json({
    message: 'Measure Added Succesfully',
    body: {
      user:{name, measure, time}
    }
  })
};

```

```

module.exports = {
  getdevices,
  uploadMeasures,
  createDevices
}

```

Routes index.js

```

const { Router } = require('express');
const router = Router();
//funciones a usar
const {getdevices, uploadMeasures, createDevices} =
require('../controllers/index.controller')

```

```

module.exports = router;
//rutas implementadas
router.get('/devices',getdevices)
router.post('/devices/:name.:location',createDevices)
router.post('/measure/:name.:measure.:time',uploadMeasures)

```

```
Index.js
const express = require('express');
const app = express();
const port = 5000
// middlewares
app.use(express.json());
app.use(express.urlencoded({extended: false}));

//definiendo las rutas
app.use(require('./routes/index'));

app.listen(port)
console.log('server on port 5000')
```