

Universidad de San Carlos de Guatemala.  
Facultad de ingeniería.  
Escuela de ingeniería en ciencias y sistemas.  
Área: Ciencias de la computación  
Nombre del curso: Arquitectura de computadores y ensambladores 1  
Catedrático: m.sc. Luis Fernando Espino Barrios  
Auxiliar: Carlos Antonio Velásquez Castellanos



# Manual de Usuario

## Proyecto #2

No	Nombre:	Carné
1	Henderson Migdo Baten Hernandez	201019694
2	Selim Idair Ergon Castillo	201801300
3	Jemima Solmaira Chavajay Quiejú	201801521
4	Giovanni Saul Concohá Cax	202100229
5	Johan Moises Cardona Rosales	202201405
6	Estiben Yair Lopez Leveron	202204578
7	Santiago Julián Barrera Reyes	201905884

**28 de junio, 2024**

## Contenido

Introducción.....	3
Requisitos del Sistema.....	4
Requisitos de Hardware .....	4
Requisitos de Software .....	4
Guía de Instalación .....	5
Preparación del Hardware .....	5
Configurar la Raspberry Pi.....	5
Conectar los Sensores.....	5
Uso del Sistema .....	6
Acceso a la Página Web de Control .....	6
Descripción de la Interfaz de Usuario .....	7
Funciones Avanzadas .....	8
Resultado del código fuente.....	9
Solución de Problemas .....	10
Cómo Contactar Soporte Técnico .....	12

# Introducción

El cambio climático es uno de los mayores desafíos globales contemporáneos, manifestándose a través del aumento de las temperaturas, alteraciones en los patrones de precipitación y la creciente frecuencia e intensidad de eventos climáticos extremos. En este contexto, el monitoreo continuo y preciso de las condiciones meteorológicas locales es esencial para comprender y mitigar estos efectos adversos.

La implementación de una estación meteorológica, como la descrita en este proyecto, proporciona datos cruciales para la investigación climática, la gestión de recursos naturales y la protección de las comunidades frente a eventos climáticos adversos. Este proyecto tiene como objetivo desarrollar una estación meteorológica que pueda recolectar, procesar y visualizar datos meteorológicos en tiempo real, utilizando una combinación de sensores y tecnologías de procesamiento de datos.

Para lograrlo, se integrarán sensores para medir diversos parámetros atmosféricos (temperatura, humedad, velocidad del viento, luminosidad, calidad del aire y presión barométrica) con un sistema de procesamiento basado en un lenguaje ensamblador ARM, respaldado por un backend en Python y un frontend adaptable para la visualización de los datos. Este manual técnico documenta el diseño, implementación y funcionamiento de la estación meteorológica, proporcionando una guía detallada para su replicación y uso.

# Requisitos del Sistema

## Requisitos de Hardware

- Raspberry Pi 3 Model B o superior
- Sensor de Temperatura (DHT11/DHT22)
- Sensor de Humedad (DHT11/DHT22)
- Sensor de Velocidad del Viento (Anemómetro)
- Fotorresistencia (LDR)
- Sensor de Calidad del Aire (MQ-135)
- Sensor de Presión Barométrica (BMP180/BMP280)

## Requisitos de Software

- Sistema Operativo: Raspbian Buster o posterior
- Lenguaje de Programación: Python 3.7 o superior
- Bibliotecas Python: RPi.GPIO, Adafruit\_DHT, smbus, flask
- Herramientas de Desarrollo ARM: gcc-arm-none-eabi, gdb-multiarch, QEMU
- Navegador Web: Chrome, Firefox, u otro navegador compatible con HTML5

# Guía de Instalación

## Preparación del Hardware

### Configurar la Raspberry Pi

#### 1. Preparar la Tarjeta microSD:

- Descargue la imagen de Raspbian Buster (o posterior) desde el sitio oficial de Raspberry Pi.
- Use una herramienta como Balena Etcher para escribir la imagen de Raspbian en la tarjeta microSD.
- Inserte la tarjeta microSD en la ranura de la Raspberry Pi.

#### 2. Conectar Periféricos:

- Conecte un teclado, ratón y monitor a la Raspberry Pi, conecte la Raspberry Pi a una fuente de alimentación.

## Conectar los Sensores

#### 1. Temperatura y Humedad (DHT11/DHT22):

- Conecte el pin de datos al GPIO correspondiente en la Raspberry Pi.
- Conecte los pines de VCC y GND a los pines de 3.3V y GND de la Raspberry Pi.

#### 2. Velocidad del Viento (Anemómetro):

- Conecte el anemómetro a los pines GPIO según las especificaciones del modelo.

#### 3. Luminosidad (LDR):

- Conecte un extremo de la LDR a 3.3V y el otro a un pin GPIO con una resistencia de 10k ohmios a tierra.

#### 4. Calidad del Aire (MQ-135):

- Conecte el pin de datos al GPIO correspondiente en la Raspberry Pi.
- Conecte los pines de VCC y GND a los pines de 5V y GND de la Raspberry Pi.

#### 5. Presión Barométrica (BMP180/BMP280):

- Conecte los pines de SDA y SCL a los pines I2C de la Raspberry Pi.

# Uso del Sistema

## Acceso a la Página Web de Control

### 1. Inicia el servidor web:

- Asegúrate de que la Raspberry Pi esté encendida y conectada a la red.
- Inicia sesión en la Raspberry Pi y abre una terminal.
- Navega al directorio del proyecto donde está el servidor Flask.
- Ejecuta el comando `python3 main.py` (o el nombre del script que inicia el servidor Flask).
- Nota: Si el servidor está configurado para ejecutarse en un puerto específico, asegúrate de conocer este puerto (por ejemplo, 5000) `script.py`

### 2. Accede a la interfaz web:

- Desde cualquier dispositivo en la misma red, abre un navegador web.
- Ingresa la dirección IP de la Raspberry Pi seguida del puerto del servidor Flask (por ejemplo, [http://<IP\\_Raspberry\\_Pi>:5000](http://<IP_Raspberry_Pi>:5000)).

## Descripción de la Interfaz de Usuario

La interfaz de usuario para la aplicación de control de sensores probablemente incluiría varias páginas o secciones principales para cada tipo de sensor. Aquí hay un desglose de cómo podría estructurarse la UI:

### 1. Página de Inicio (Dashboard)

- **Descripción:** Un panel de control que proporciona una visión general de todos los sensores conectados.
- **Estado de los Sensores:** Indicadores visuales (luces verdes/rojas) que muestran si cada sensor está encendido o apagado.
- **Datos Resumidos:** Muestra datos clave como la última lectura de cada sensor.

### 2. Control de Sensores

- **Descripción:** Una sección donde los usuarios pueden encender y apagar los sensores individualmente.
- **Botones de Control:** Botones para encender y apagar cada sensor (On\_Sensor y Off\_Sensor).
- **Indicadores de Estado:** Indicadores que muestran el estado actual (encendido/apagado) de cada sensor.

### 3. Estadísticas de Sensores

- **Descripción:** Una sección dedicada a mostrar estadísticas detalladas de los datos recopilados por los sensores.
- **Gráficas y Tablas:** Visualizaciones que muestran estadísticas como promedio, mediana, desviación estándar, máximos, mínimos y moda de los datos de los sensores.
- **Selector de Sensores:** Dropdown o lista para seleccionar el sensor del cual se quieren ver las estadísticas.

### 4. Datos en Tiempo Real

- **Descripción:** Una página donde los usuarios pueden ver las lecturas actuales de los sensores en tiempo real.
- **Lecturas en Vivo:** Visualización en tiempo real de los datos de los sensores (por ejemplo, temperatura, humedad, calidad del aire).
- **Gráficos en Tiempo Real:** Gráficos que se actualizan en tiempo real para mostrar los cambios en los datos del sensor.

# Funciones Avanzadas

## Configuración de Parámetros del Sistema

### 1. Notificaciones y Alertas

Implementar notificaciones y alertas en tiempo real para advertir a los usuarios sobre condiciones críticas o anormales detectadas por los sensores.

Implementación:

- Backend: Añadir lógica en las funciones de lectura de los sensores para verificar si los valores superan ciertos umbrales y, en ese caso, enviar notificaciones.
- Frontend: Usar una biblioteca de notificaciones (como Toastr) para mostrar alertas visuales al usuario.

### 2. Control Remoto y Automatización

Permitir a los usuarios configurar reglas para la automatización del encendido y apagado de los sensores según ciertos parámetros y condiciones.

Implementación:

- Backend: Crear endpoints adicionales para configurar reglas de automatización.
- Frontend: Interfaz para que los usuarios puedan definir y gestionar estas reglas.

### 3. Visualización Avanzada de Datos

Implementar gráficos y visualizaciones más avanzadas para los datos de los sensores, incluyendo tendencias, comparaciones y análisis históricos.

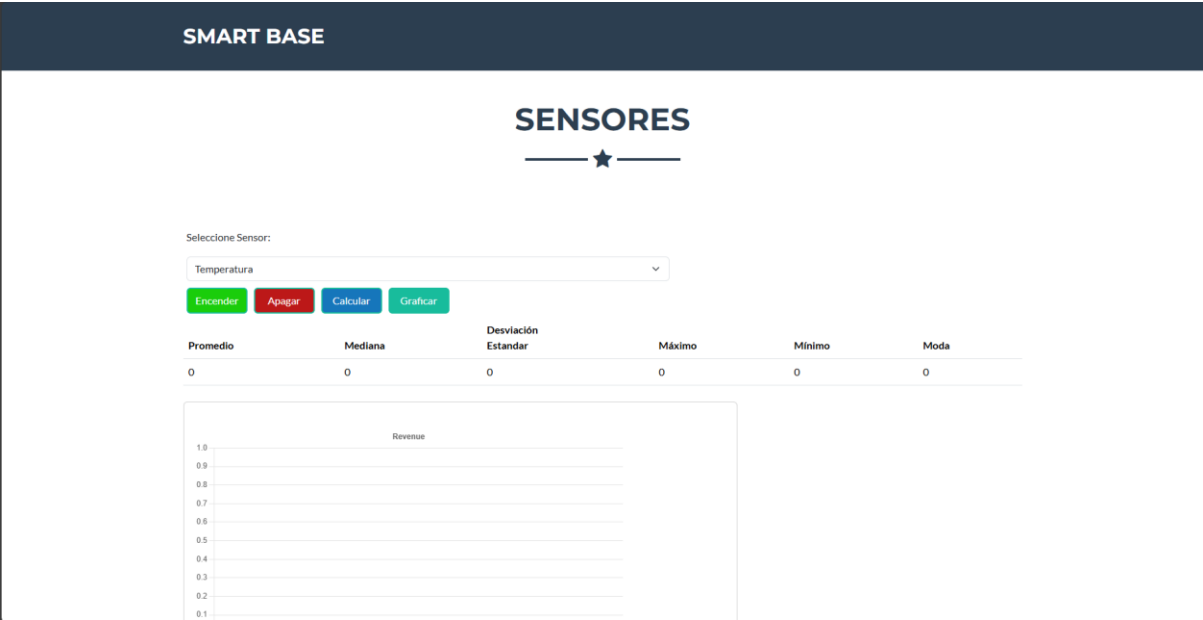
Implementación:

- Backend: Endpoint para obtener datos históricos y procesados.
- Frontend: Uso de bibliotecas de gráficos avanzadas como Chart.js, D3.js o Plotly.

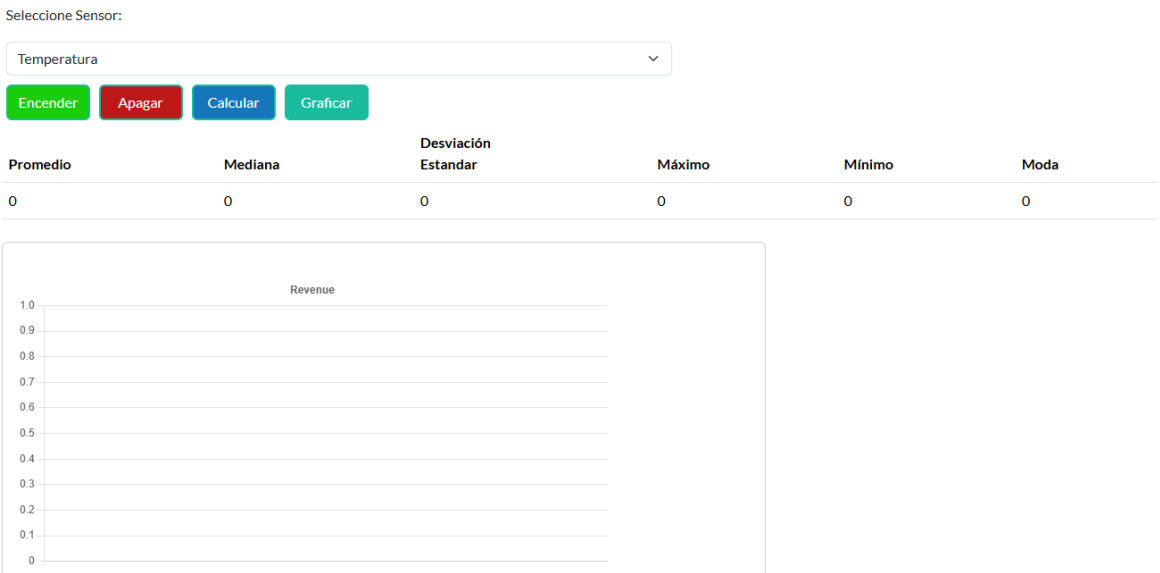


# Resultado del código fuente

La interfaz que se muestra en la imagen es de una aplicación llamada “SMART BASE” que parece estar diseñada para interactuar con varios sensores. En la sección “SENSORES”, puedes seleccionar un sensor específico de una lista desplegable que incluye opciones como Temperatura, Presión, Humedad, Aire, Calidad del aire y Gases.



Una vez seleccionado un sensor, la interfaz muestra estadísticas para ese sensor en particular. Las estadísticas incluyen el Promedio, la Mediana, la Desviación Estándar, el Máximo, el Mínimo y la Moda. En la imagen proporcionada, todos estos valores se muestran como cero, lo que podría indicar que aún no se han recopilado datos del sensor seleccionado o que el sensor está apagado.



# Solución de Problemas

1. **Errores de importación:** Si alguna de las bibliotecas importadas no está instalada en el sistema donde se ejecuta el script, se producirá un error de importación. Esto se puede solucionar instalando las bibliotecas necesarias.

**Solución:** Asegúrate de que todas las bibliotecas necesarias estén instaladas en tu sistema. Puedes hacer esto utilizando el administrador de paquetes de Python, pip, con comandos como `pip install flask`.

2. **Problemas con los sensores:** Si los sensores no están conectados correctamente o si hay un problema con los sensores en sí, las lecturas podrían ser incorrectas o podrían no obtenerse en absoluto.

**Solución:** Verifica las conexiones de tus sensores y asegúrate de que estén funcionando correctamente. Si es posible, prueba los sensores individualmente para descartar cualquier problema.

3. **Problemas con la biblioteca compartida:** Si la biblioteca compartida “`calculos.so`” no está en el mismo directorio que el script o si no se puede cargar por alguna razón, se producirá un error.

**Solución:** Asegúrate de que la biblioteca compartida “`calculos.so`” esté en el mismo directorio que tu script. Si la biblioteca no se puede cargar, puede que necesites revisar su compilación y enlace.

4. **Problemas con las rutas API:** Si las solicitudes a las rutas API no incluyen los datos esperados (por ejemplo, el ‘`sensor`’ en las solicitudes a ‘`/api/on`’ y ‘`/api/off`’), se devolverá un error.

**Solución:** Asegúrate de que todas las solicitudes a tus rutas API incluyan los datos necesarios. Podrías manejar los errores en tu código para devolver mensajes útiles cuando falten datos.

5. **Problemas de concurrencia:** Como este código parece estar diseñado para leer de varios sensores en paralelo, podría haber problemas de concurrencia o sincronización si varias funciones intentan leer del mismo sensor al mismo tiempo.

**Solución:** Considera el uso de bloqueos o semáforos para manejar el acceso concurrente a los sensores. Esto puede ayudar a prevenir condiciones de carrera y otros problemas de concurrencia.

6. **Problemas de rendimiento:** Si se recopilan grandes cantidades de datos de los sensores, podría haber problemas de rendimiento o de memoria.

**Solución:** Si estás recopilando grandes cantidades de datos, considera formas de limitar el uso de la memoria. Esto podría implicar el almacenamiento de datos en disco en lugar de en la memoria, o la reducción de la cantidad de datos que se recopilan.

7. **Problemas de red:** Como este es un servidor Flask, podría haber problemas si la red se cae o si hay problemas con el host o el puerto especificados para la aplicación.

**Solución:** Asegúrate de que tu red sea estable y de que el host y el puerto que estás utilizando para tu aplicación Flask estén disponibles. Podrías considerar la posibilidad de manejar los errores de red en tu código para que tu aplicación pueda recuperarse de los problemas de red.

# Cómo Contactar Soporte Técnico

Si los problemas persisten y no se pueden resolver con las soluciones anteriores, puede ser necesario contactar al soporte técnico para obtener ayuda adicional.

## 1. Recopila Información:

- Anota los síntomas del problema y cualquier mensaje de error que aparezca.
- Toma nota de cualquier cambio reciente en la configuración del sistema o en el hardware.

## 2. Contacta Soporte Técnico:

- Correo Electrónico: Envíe un correo electrónico detallando el problema a [yaircarrito@gmail.com](mailto:yaircarrito@gmail.com).
- Teléfono: Llame al número de soporte técnico: +502 3330-5195

## 3. Proporciona Detalles:

- Proporcione una descripción clara del problema, los pasos que ha seguido para intentar resolverlo y cualquier información adicional relevante (como mensajes de error y configuraciones de hardware).

## 4. Seguimiento:

- Manténgase disponible para responder a cualquier pregunta adicional del equipo de soporte.
- Siga las instrucciones proporcionadas por el soporte técnico para resolver el problema.