

Sistema de Monitoreo Climático

Manual Técnico

Universidad San Carlos de Guatemala

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1 -Grupo 3

1. Introducción

Este manual técnico describe la arquitectura, configuración, y funcionamiento del Sistema de Monitoreo Climático. El sistema combina sensores, procesamiento de datos y visualización en tiempo real para proporcionar información precisa sobre condiciones ambientales. La aplicación utiliza Flask para la API backend, React + Vite para el frontend y Python para los scripts de adquisición de datos desde los sensores.

2. Arquitectura del Sistema

El sistema se compone de tres capas principales:

2.1 Capa de Sensores

- Sensores utilizados:
 - DHT11: Mide temperatura y humedad relativa.
 - BMP280/BME280: Mide presión barométrica.
 - Anemómetro: Mide velocidad del viento.
- Conexión: Los sensores están conectados a los pines GPIO del Raspberry Pi.

2.2 Backend (API)

Framework utilizado: Flask.

Descripción:

- Procesa y almacena los datos recibidos de los sensores.
- Ofrece endpoints para el frontend:
- GET /data: Devuelve datos actuales y/o históricos.
- POST /config: Permite configurar los parámetros del sistema (ej. frecuencia de lectura).

Base de datos:

- **MySQL en la nube:**
 - La base de datos almacena los datos recopilados por los sensores y permite consultas rápidas tanto para el dashboard como para las gráficas históricas.
 - **Configuración:**
 - Se debe tener acceso a un servidor MySQL en la nube.
 - Credenciales y configuración de conexión se almacenan en un archivo .env para seguridad.
 - La estructura de las tablas incluye:
 - **sensores:** Información de configuración de los sensores.
 - **lecturas:** Datos recolectados, con campos como id, timestamp, temperatura, humedad_relativa, humedad_absoluta, velocidad_viento, y presion.

2.3 Frontend (Dashboard)

Framework utilizado: React con Vite.

Características:

- Visualización en tiempo real de variables climáticas.
- Filtros para consultas de datos históricos.
- Gráficas generadas con librerías como Chart.js .

3. Instalación y Configuración

3.1 Requisitos Previos

Hardware:

- Raspberry Pi (modelo 3B+ o superior).
- Sensores: DHT11, BMP280/BME280, anemómetro.

Software:

- Python 3.8 o superior.
- Node.js v16 o superior.
- Pipenv o virtualenv para gestión de dependencias en Python.
- Git para control de versiones.

3.2 Configuración del Backend

1. Crear un entorno virtual y activar:

```
python -m venv venv
```

```
source venv/bin/activate # En Windows: venv\Scripts\activate
```

3. Instalar dependencias:

```
pip install -r requirements.txt
```

4. Ejecutar el servidor Flask:

```
flask run
```

Por defecto, el servidor estará disponible en `http://127.0.0.1:5000`.

3.3 Configuración del Frontend

1. Entrar al directorio del frontend:

```
cd frontend
```

2. Instalar dependencias: `npm install`

3. Iniciar el servidor de desarrollo:

```
npm run dev
```

El dashboard estará disponible en el puerto configurado por Vite (por defecto `http://127.0.0.1:5173`).

3.4 Scripts de Sensores

1. Entrar al directorio de los scripts:

```
cd sensors
```

2. Ejecutar el script de adquisición de datos:

```
python sensors.py
```

Este script recopila información de los sensores y la envía al servidor Flask.

4. Uso del Sistema

4.1 Flujo de Trabajo

1. Iniciar los scripts de sensores para recopilar datos.
2. Verificar que el servidor Flask esté corriendo y recibiendo los datos.
3. Acceder al dashboard en el navegador para visualizar los datos.

4.2 Endpoints Principales del API

- GET /data: Devuelve datos actuales o filtrados por rango de tiempo.

```
curl http://127.0.0.1:5000/data?start_date=2024-12-01&end_date=2024-12-13
```
- POST /config: Configura el sistema (frecuencia de lectura, etc.).

```
curl -X POST http://127.0.0.1:5000/config -H "Content-Type: application/json" -d '{"frequency": 10}'
```

5. Mantenimiento y Solución de Problemas

5.1 Mantenimiento

- Limpieza de sensores para evitar lecturas erróneas.
- Actualizaciones periódicas del software mediante `git pull`.

5.2 Problemas Comunes

1. El backend no responde:

- Verifique que Flask esté corriendo.
- Revise los logs para errores específicos.

2. El frontend no carga:

- Confirme que el servidor de desarrollo esté activo.
- Asegúrese de que las dependencias estén instaladas correctamente.

3. Los sensores no envían datos:

- Revise las conexiones físicas de los sensores.
- Ejecute los scripts de prueba incluidos para verificar funcionalidad.

6. Documentación del Código

6.1 Estructura del Proyecto

- /backend: Contiene el código de la API Flask.
- /frontend: Contiene la aplicación React + Vite.
- /sensors: Scripts en Python para comunicación con los sensores.

6.2 Buenas Prácticas

- Realice commits frecuentes y bien documentados.
- Mantenga el código limpio y modular.