Sistema IoT de Monitoreo Polimórfico

José Emiliano Puga Saucedo

30 de octubre de 2025

Resumen

Este documento resume el propósito, la arquitectura y el uso básico del proyecto "Sistema IoT de Monitoreo Polimórfico". Incluye una descripción de las clases principales, ejemplos de uso y las instrucciones mínimas para compilar y generar la documentación con Doxygen.

1. Introducción

El proyecto consiste en un conjunto de clases en C++ para gestionar sensores de distintos tipos (temperatura y presión). Se aprovecha el polimorfismo para mantener una interfaz común (`SensorBase`) y especializar el comportamiento en subclases (`SensorTemperatura`, `SensorPresion`). Además, se utiliza una lista enlazada genérica para almacenar el historial de lecturas.

2. Estructura general

Los ficheros relevantes son:

- `main.cpp` – Interfaz de usuario y bucle principal.
- `SensorBase.h` – Clase base abstracta con la interfaz común.
- `SensorTemperatura.h`, `SensorPresion.h` – Implementaciones concretas.
- `ListaSensor.h` – Implementación de una lista enlazada genérica.
- `ListaGestion.h` – Colección que administra los sensores (polimórfica).

3. Clases principales

3.1. SensorBase

Contrato mínimo que obliga a las subclases a implementar:

- `agregarLecturaDesdeTexto(const char*)`
- `procesarLectura()`
- `imprimirInfo() const`

3.2. SensorTemperatura

Guarda lecturas en una `ListaSensor<float>`. Antes de calcular el promedio elimina la lectura más baja como filtrado de valores anómalos.

3.3. SensorPresion

Guarda lecturas en una `ListaSensor<int>` y calcula un promedio simple.

4. Uso rápido

Ejemplo mínimo de interacción (modo manual desde la consola):

1. Compilar el binario: `g++ main.cpp -o SistemaIoT`
2. Ejecutar: `./SistemaIoT`
3. Seguir el menú para agregar sensores e ingresar lecturas.

5. Compilación y recomendaciones

Se recomienda usar C++11 (o superior) y, para proyectos más grandes, CMake. En este repositorio existe un fichero `CMakeLists.txt` que puede usarse para generar los ficheros de proyecto:

```
mkdir -p build && cd build
cmake ..
make
```

Alternativamente, para una compilación rápida:

```
g++ main.cpp -o SistemaIoT
```

6. Generar documentación con Doxygen

Si desea generar la documentación HTML con Doxygen:

1. Asegúrese de tener instalado `doxygen` (y opcionalmente `graphviz` si quiere diagramas).
2. En la raíz del proyecto ejecute: `doxygen Doxyfile` (o use `Doxyfile.clean` si prefiere la versión limpia).
3. La salida por defecto aparecerá en la carpeta `docs/html` o la que esté indicada en el `Doxyfile`.

7. Buenas prácticas y notas

- Documente nuevas clases y métodos usando el formato Doxygen en los headers.
- Mantenga el fichero `Doxyfile` actualizado con las rutas correctas en `INPUT` y `STRIP_FROM_PATH`.
- Para diagramas de clases instale Graphviz y active `HAVE_DOT = YES` en el `Doxyfile`.

8. Licencia y autoría

Este proyecto es de carácter académico. Autor: José Emiliano Puga Saucedo. Si quiere incluir una licencia explícita, añada un fichero `LICENSE` con la licencia deseada.

Documento preparado para acompañar el repositorio; si desea, puede generar un PDF a partir de este fichero e integrarlo en el flujo de *build*.