

# Reporte Técnico

## Sistema IoT de Gestión Polimórfica de Sensores para Infraestructura Crítica

---

**Autor:** Juan Francisco Ortega Pulido  
**Matrícula:** 2430450  
**Fecha:** 30 Octubre 2025  
**Institución:** Universidad Politécnica de Victoria

---

### 1. INTRODUCCIÓN

#### 1. INTRODUCCIÓN (El Problema y la Solución)

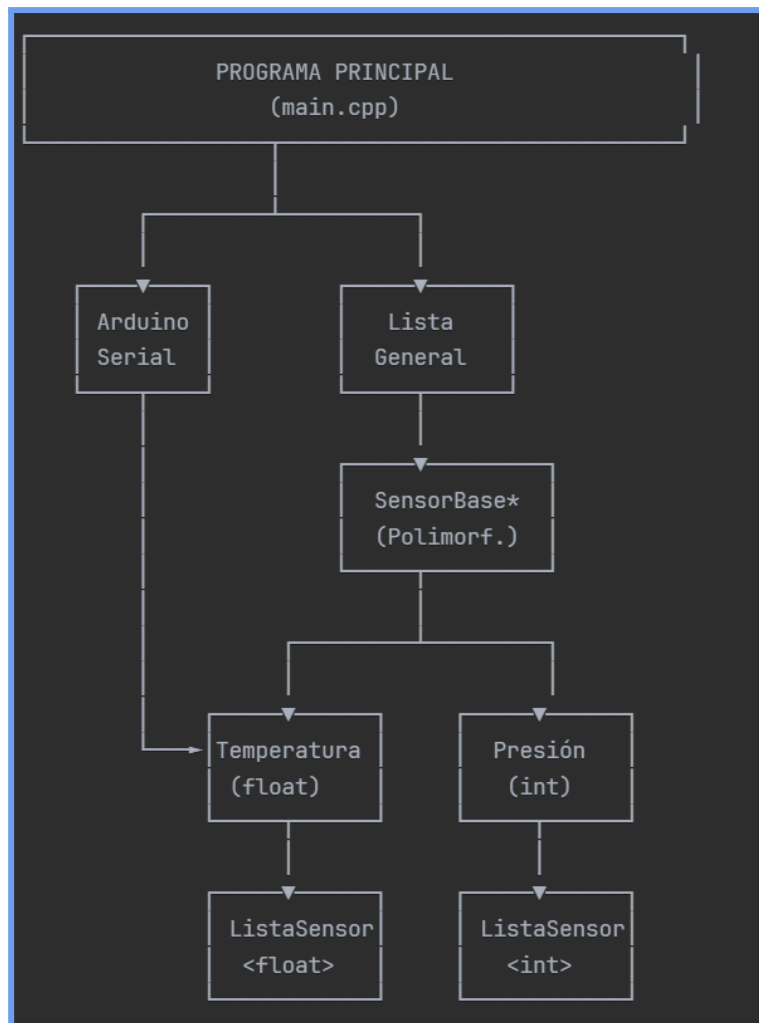
El proyecto aborda la rigidez de los sistemas de monitoreo de Infraestructura Crítica (IC) ante la gestión de múltiples tipos de sensores (como float y int).

El objetivo principal fue desarrollar un sistema de bajo nivel en C++ que logre:

- **Gestión Polimórfica:** Unificar el manejo de sensores heterogéneos (Temperatura y Presión) usando una jerarquía de clases abstractas (SensorBase).
- **Almacenamiento Dinámico y Genérico:** Implementar listas enlazadas simples genéricas (ListaSensor<T>) completamente manuales.
- **Gestión de Memoria Rigurosa:** Asegurar cero fugas de memoria mediante la aplicación estricta de la Regla de los Tres/Cinco y la prohibición total del uso de la STL (Standard Template Library).
- **Captura de Datos:** Leer señales en tiempo real desde dispositivos Arduino vía puerto serial.

## 2. REPORTE TÉCNICO (Arquitectura y Componentes Clave)

### 2.1 Arquitectura de POO Avanzado



El sistema se basa en una jerarquía de clases que permite el Polimorfismo:

- **SensorBase (Clase Base Abstracta):** Define la interfaz común (procesarLectura() virtual puro) y utiliza un destructor virtual para la correcta liberación de memoria en cascada.
- **Clases Derivadas:** SensorTemperatura (usa ListaSensor<float>) y SensorPresion (usa ListaSensor<int>).
- **ListaGeneral:** Contiene punteros SensorBase\* para gestionar todos los sensores de forma unificada y llamar a procesarLectura() de manera polimórfica.

## 2.2 Gestión de Datos y Memoria

- **Listas Genéricas (ListaSensor<T>):** Implementación manual de listas para almacenar lecturas (T) de forma dinámica. El uso de Templates permite su reutilización para float y int.
- **Regla de los Tres:** Implementación manual de destructor, constructor de copia y operador de asignación (operator=) para garantizar la copia profunda y evitar fugas o dobles liberaciones de memoria.
- **Procesamiento:** Ocurre polimórficamente cuando se acumulan lecturas. Por ejemplo, SensorTemperatura elimina el valor más bajo antes de calcular el promedio.

## 2.3 Flujo de Datos

El sistema lee líneas en formato **TIPO:VALOR\n** (ej: TEMP:23.5) del puerto serial (gestionado por la clase multiplataforma SerialReader). El programa principal parsea manualmente la cadena, registra el valor en el sensor correspondiente de ListaGeneral, y ejecuta el procesamiento polimórfico cuando es necesario.

---

## 3. CONCLUSIONES

El proyecto demostró el dominio de POO avanzado (Polimorfismo, Destrucción Virtual, Programación Genérica (Templates) y Gestión de Memoria de Bajo Nivel (Regla de los Tres/Cinco, no-STL). Se logró un sistema modular, extensible y verificado con Valgrind para tener cero fugas de memoria.