



Universidad Politécnica de Victoria
Ingeniería en Tecnologías de la Información e Innovación Digital

Actividad 1
“Unidad 2”

Asignatura:
Estructura de datos

Alumna:
Barbie Lumbreras Olvera

Doctor:
Said Polanco Martagón

Ciudad Victoria, Tamaulipas
Octubre de 2025



INTRODUCCIÓN

Este proyecto desarrolla un sistema de monitoreo inteligente para infraestructura crítica que puede manejar diferentes tipos de sensores de manera unificada. El sistema está diseñado para ser flexible, eficiente y escalable, permitiendo trabajar con múltiples tipos de sensores sin modificar el código base.

El objetivo principal es demostrar conceptos avanzados de programación en C++ como polimorfismo, templates, gestión manual de memoria y arquitecturas modulares, todo aplicado a un problema real del mundo.

MANUAL TÉCNICO

** DISEÑO

1. Arduino:

El arduino simula sensores reales enviando datos cada cierto tiempo. Se obtuvo un formato sencillo para las salidas: empezando por los tipos de sensores, continuando con la identificación y terminando con el valor medido.

2. Aplicación en C++

Aquí se reciben los datos del arduino y los organiza. Se desarrollaron diferentes tipos de sensores en el código y cada uno de estos sabe como se maneja su propia información.

3. Sistema de almacenamiento

Para el guardado de las lecturas de cada sensor, se implementó listas que pueden crecer según sea necesario. Cada sensor tiene su propio historial de mediciones.

** DESARROLLO DEL PROYECTO

- Planificación: Comenzamos recaudando la información necesaria para abordar dicho proyecto, una vez teniendo la información comenzamos.
- Codificación básica: Se comenzaron creando las clases principales. La clase padre define qué pueden hacer todos los sensores y las clases hijas implementan las funciones específicas para temperatura y presión.
- Comunicación serial: Configuramos la comunicación entre la computadora y el Arduino.

** COMPONENTES PRINCIPALES

- Aplicación principal (main.cpp): Este contiene un menú interactivo y toda la lógica para comunicarse con el arduino. Permite crear sensores manualmente o automáticamente cuando llegan datos nuevos.
- Clases de sensores:
 - SensorBase: La plantilla que define como debe comportarse cualquier sensor
 - SensorTemperatura: Especializado en manejar datos de temperatura
 - SensorPresion: Especializado en manejar datos de presión
- Sistema de listas:
 - ListaSensor: Guarda el historial de lecturas de cada sensor
 - ListaGestion: Administra todos los sensores del sistema

CAPTURAS DE IMPLEMENTACIÓN

1. Compilación del proyecto

```
admin1@admin1-barbie:~/Escritorio/act1u2/src$ g++ main.cpp -o SistemaIoT
admin1@admin1-barbie:~/Escritorio/act1u2/src$ ./SistemaIoT
```

2. Menú principal

```
--- Sistema IoT de Monitoreo Polimórfico ---

===== MENU =====
1. Crear sensor
2. Registrar lectura
3. Ejecutar procesamiento polimorfico
4. Listar sensores
5. Abrir/usar puerto serial y leer 1 linea del Arduino
6. Leer continuamente del Arduino (demo)
7. Salir
Elige opcion: 
```

3. Creación de sensores

```
Elige opcion: 1
Tipo de sensor (T=Temperatura, P=Presion): T
ID del sensor (ej. T-001): T-001
Sensor de Temperatura 'T-001' creado e insertado.
```

4. Código en arduino

```
sketch_oct30a
}

void loop() {
  unsigned long now = millis();
  if (now - lastSend >= 2000) { // cada 2 segundos manda algo
    lastSend = now;

    // Alterna entre temperatura y presión
    static bool sendTemp = true;
    if (sendTemp) {
      tempFake += 0.3; // variar un poco
      Serial.print("T;T-001;");
      Serial.println(tempFake, 1); // 1 decimal
    } else {
      presionFake += 1;
      Serial.print("P;P-105;");
      Serial.println(presionFake);
    }
    sendTemp = !sendTemp;
  }
}
```



CONCLUSIÓN

Este proyecto me resultó muy útil para manejar diferentes tipos de sensores, así como poner en práctica la conexión con el Arduino.

La documentación con Doxygen no solo cumplió con los requisitos del proyecto, sino que me ayudó a escribir código más claro y mantenible.

Logramos crear un sistema funcional que combina el hardware y software de manera efectiva, aplicando lo aprendido de programación