

Laboratorio Calificado 01

2025-2

ÍNDICE

1	Instrucciones	3
2	Contexto.....	3
3	Requisitos del Problema.....	3
3.1	Datos del Sistema:.....	3
3.2	Funcionalidad:.....	3
3.3	Interacción con el Usuario:.....	4
4	Tareas para Resolver	4
5	Entregable.....	4
6	Código base para comenzar	4
7	Caso de Prueba	5
7.1	Condición inicial:.....	5
7.2	Entrada:	5
7.3	Salida esperada:.....	5
8	Rúbrica de Evaluación.....	6

Laboratorio Calificado 01

Control de Sensores de Temperatura en una Línea de Producción

1 Instrucciones

Se ha creado la tarea “Laboratorio Calificado 1” en la sección Laboratorio Calificado 1 de la página del curso-horario en PAIDEIA. Cada alumno deberá entregar un archivo comprimido (formato ZIP) con todos los archivos del desarrollo realizado en el laboratorio. El nombre del archivo comprimido deberá tener el formato:

L1_<Código del alumno de 8 dígitos>.<Extensión del archivo>

Como ejemplo, el nombre del archivo de alumno 20236969 del horario 06M1 sería “L1_20236969.zip”. Es COMPLETA RESPONSABILIDAD DE CADA ALUMNO el colocar nombres correctos a sus archivos y evitar así confusiones al momento de la calificación.

2 Contexto

En un laboratorio de ingeniería mecatrónica se desea llevar el control de sensores de temperatura instalados en diferentes estaciones de una línea de producción. Cada estación registra valores de temperatura (en °C) que deben ser almacenados en un arreglo.

El sistema permitirá:

- Registrar las temperaturas de cada estación.
- Calcular métricas básicas como temperatura promedio, máxima y mínima.
- Mostrar alertas si alguna estación supera un umbral de temperatura crítica.
- Permitir al usuario consultar y actualizar valores.

3 Requisitos del Problema

3.1 Datos del Sistema:

- Existen 10 estaciones de trabajo, cada una con un sensor de temperatura.
- Los datos se almacenan en un arreglo unidimensional de 10 enteros (int).
- El umbral de temperatura crítica se define como 70 °C.

3.2 Funcionalidad:

- El programa debe implementar al menos las siguientes funciones:

1. `void registrarTemperaturas(array<int>^ sensores)`
 - Permite al usuario ingresar las 10 temperaturas iniciales.
2. `void mostrarTemperaturas(array<int>^ sensores)`
 - Muestra todas las temperaturas registradas.
3. `double calcularPromedio(array<int>^ sensores)`
 - Retorna la temperatura promedio.
4. `int obtenerMaxima(array<int>^ sensores)`
 - Retorna la temperatura máxima registrada.
5. `int obtenerMinima(array<int>^ sensores)`
 - Retorna la temperatura mínima registrada.
6. `void verificarAlertas(array<int>^ sensores)`
 - Muestra un mensaje si alguna estación supera los 70 °C.

7. `void actualizarTemperatura(array<int>^ sensores, int estacion, int nuevaTemp)`
 - Permite actualizar la temperatura de una estación específica..

3.3 Interacción con el Usuario:

- El programa principal debe mostrar un menú con opciones y ejecutarse en un bucle hasta que el usuario ingrese "Q" para salir.
- Opciones mínimas del menú:
 - (1) Registrar temperaturas.
 - (2) Mostrar temperaturas.
 - (3) Calcular promedio.
 - (4) Mostrar máxima y mínima.
 - (5) Verificar alertas.
 - (6) Actualizar temperatura.
 - (Q) Salir.
- Se deben usar estructuras de control selectivas y estructuras iterativas para implementar la lógica.

4 Tareas para Resolver

1. Definir el arreglo de 10 estaciones.
2. Implementar todas las funciones solicitadas.
3. Desarrollar el menú principal para la interacción con el usuario en el caso que sea una aplicación por consola o un menú visual en el caso que sea una aplicación basada en GUIs.
4. Validaciones:
 - No permitir temperaturas negativas.
 - Validar que la estación a actualizar esté en el rango 1-10.

5 Entregable

Cree un nuevo proyecto en C++/CLI (consola o GUI) que contenga un programa completo que:

- Implemente correctamente las funciones descritas.
- Utilice tipos de datos adecuados, estructuras selectivas, iterativas y arreglos.
- Interactúe con el usuario a través de un menú sea por consola o por GUIs.
- Cumpla con las validaciones y requisitos del contexto.

6 Código base para comenzar

C++/CLI

```
#include "pch.h"
using namespace System;

void registrarTemperaturas(array<int>^ sensores) {
    for (int i = 0; i < sensores->Length; i++) {
        Console::Write("Ingrese temperatura para estación {0}: ", i + 1);
        int temp = Convert::ToInt32(Console::ReadLine());
        sensores[i] = temp;
    }
}

int main() {
    array<int>^ sensores = gcnew array<int>(10);
```

```
String^ opcion = "";

while (opcion != "Q") {
    Console::WriteLine("\n--- Menú de Control de Sensores ---");
    Console::WriteLine("1. Registrar temperaturas");
    Console::WriteLine("2. Mostrar temperaturas");
    Console::WriteLine("3. Calcular promedio");
    Console::WriteLine("4. Mostrar máxima y mínima");
    Console::WriteLine("5. Verificar alertas");
    Console::WriteLine("6. Actualizar temperatura");
    Console::WriteLine("Q. Salir");
    Console::Write("Seleccione una opción: ");
    opcion = Console::ReadLine();

    // Aquí implementarás el procesamiento con if-else
}

Console::WriteLine("Programa finalizado.");
return 0;
}
```

7 Caso de Prueba

7.1 Condición inicial:

El arreglo de temperaturas se encuentra vacío.

7.2 Entrada:

- Registro de temperaturas: 45, 52, 60, 71, 55, 68, 40, 75, 49, 62
- Mostrar temperaturas
- Consultar promedio
- Consultar máxima y mínima
- Verificar alertas
- Actualizar estación 4 a 65

7.3 Salida esperada:

- Seleccione una opción: 1
Ingrese temperatura para estación 1: 45
Ingrese temperatura para estación 2: 52
Ingrese temperatura para estación 3: 60
Ingrese temperatura para estación 4: 71
Ingrese temperatura para estación 5: 55
Ingrese temperatura para estación 6: 68
Ingrese temperatura para estación 7: 40
Ingrese temperatura para estación 8: 75
Ingrese temperatura para estación 9: 49
Ingrese temperatura para estación 10: 62
- Seleccione una opción: 2
Temperaturas registradas:
Estación 1: 45 °C
Estación 2: 52 °C
Estación 3: 60 °C
Estación 4: 71 °C
Estación 5: 55 °C

```

Estación 6: 68 °C
Estación 7: 40 °C
Estación 8: 75 °C
Estación 9: 49 °C
Estación 10: 62 °C

```

- Seleccione una opción: 3
Temperatura promedio: 57.7 °C
- Seleccione una opción: 4
Temperatura máxima: 75 °C
Temperatura mínima: 40 °C
- Seleccione una opción: 5
Alerta: Estación 4 supera el umbral con 71 °C
Alerta: Estación 8 supera el umbral con 75 °C
- Seleccione una opción: 6
Ingrese número de estación a actualizar (1-10): 4
Ingrese nueva temperatura: 65
Temperatura de la estación 4 actualizada a 65 °C

Resultado esperado después de la actualización:

- El promedio cambiará en futuras consultas.
- La estación 4 ya no genera alerta, pero la estación 8 sí.

En el caso de desarrollar un programa basado en GUIs deberá crear el formulario con los controles visuales necesarios para la interacción con el usuario y la invocación a cada una de las opciones.

8 Rúbrica de Evaluación

Puntaje Máximo: 20 puntos

Criterio	Excelente	Bueno	Regular	Deficiente	No Cumple
Uso de Arreglos (4 pts)	Declara y usa correctamente el arreglo para todas las operaciones	Declara y usa con leves errores	Uso parcial o limitado	Uso incorrecto	No usa arreglos
Definición y Uso de Funciones (5 pts)	Todas las funciones implementadas correctamente	Implementa la mayoría con leves errores	Implementa parcialmente	Implementa mínimamente	No usa funciones
Estructuras de Control (4 pts)	Aplica selectivas e iterativas de manera correcta y eficiente	Uso adecuado con pequeños errores	Uso parcial o incompleto	Uso incorrecto	No aplica
Validaciones (3 pts)	Valida entradas y límites correctamente	Valida parcialmente	Valida de forma incompleta	Validaciones mínimas	No valida
Funcionalidad Global (4 pts)	El programa cumple todos los requisitos y funciona sin errores	Funciona con leves errores	Funciona parcialmente	Funcionalidad mínima	No funciona