

## **Laboratorio Calificado 08**

**2025-2**

# ÍNDICE

1	Instrucciones .....	3
2	Contexto.....	3
3	Objetivos.....	3
4	Requisitos.....	4
5	Entregables.....	4
6	Rúbrica de Evaluación.....	4

## Laboratorio Calificado 08

# Programación de un mantenimiento para la entidad Máquina del Sistema de Gestión de Ensamblaje en una Línea de Producción Automatizada de Mecatrónica

## 1 Instrucciones

Se ha creado la tarea “Laboratorio Calificado 8” en la sección Laboratorio Calificado 8 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:

L8\_<Código del alumno de 8 dígitos>.<Extensión del archivo>

Como ejemplo, el nombre de archivo del alumno 20236969 sería “L8\_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

Durante los laboratorios anteriores se ha desarrollado un Sistema de Gestión de Ensamblaje en una Línea de Producción Automatizada de Mecatrónica, el cual incluye:

- Modelos como el Operador, con su respectivo mantenimiento CRUD.
- Controladores (Controllers) que gestionan entidades con repositorios en memoria y persistencia en archivo.
- Interfaces gráficas (GUI) implementadas en C++/CLI con Windows Forms, que permiten realizar las operaciones CRUD.
- Persistencia para almacenar la información de manera permanente entre sesiones.

En una nueva iteración del análisis del sistema, se ha identificado la necesidad de gestionar una nueva entidad denominada Máquina, la cual representa los equipos que forman parte de la línea de producción (por ejemplo, brazos robóticos, sensores, o actuadores especializados).

Cada máquina debe registrarse con los siguientes atributos:

- Identificador único (id)
- Nombre (nombre) — e.g., “Brazo Robótico 1”, “Sensor de Visión A2”
- Tipo (tipo) — e.g., “Robótico”, “Sensorial”, “Actuador”
- Estado (estado) — e.g., “Operativa”, “En Mantenimiento”, “Inactiva”
- Ubicación (ubicacion) — e.g., “Zona A-1”, “Zona B-3”

El objetivo es desarrollar un mantenimiento para la entidad Máquina, que permita **agregar nuevas máquinas** y **visualizar todas las registradas**, integrando todas las capas del sistema (modelo, control, persistencia y GUI).

## 3 Objetivos

Implementar las capas necesarias del sistema, garantizando que:

- Al iniciar el programa, se carguen automáticamente las máquinas almacenadas en el archivo.
- Las operaciones CRUD de **agregar máquina** y **consultar todas las máquinas** desde la GUI interactúen con la capa de persistencia.
- Al cerrar la aplicación, los datos de las máquinas se guarden correctamente.

## 4 Requisitos

### a) Implementación de la capa del modelo (2 puntos)

Definir una clase que modele a la Máquina, con los atributos mencionados (id, nombre, tipo, estado y ubicación) e integrarla a la capa del modelo.

### b) Implementación de la capa de persistencia (4 puntos)

Definir o adaptar la clase responsable de la lectura y escritura de los datos en un archivo (texto, CSV, XML o binario). Debe seguir el principio de separación de responsabilidades, evitando acoplar la lógica de persistencia con la lógica de negocio.

### c) Implementación de la capa de control (4 puntos)

Definir o adaptar el Controller o Service encargado de las operaciones:

- Agregar una máquina
- Consultar todas las máquinas

El controlador debe integrarse correctamente con la capa de persistencia.

### d) Implementación de la capa de presentación (GUI) (8 puntos)

Diseñar una interfaz gráfica en C++/CLI (Windows Forms) que permita:

- Agregar una máquina mediante un formulario con campos para cada atributo.
- Visualizar todas las máquinas registradas en un DataGridView.
- Validar datos básicos (campos vacíos, id único, formato de texto).
- Mostrar mensajes de confirmación o error según la operación realizada.

### e) Ejecución y pruebas (2 puntos)

Probar el sistema realizando las siguientes operaciones:

- Inserción de 2 máquinas diferentes (e.g., un brazo robótico y un sensor de visión).
- Consulta de todas las máquinas registradas.
- Verificar que los datos persisten después de reiniciar la aplicación.
- Incluir capturas de pantalla que muestren la GUI en cada operación.

## 5 Entregables

- Código fuente completo en C++/CLI, bien comentado y estructurado.
- Capturas de pantalla que muestren la GUI durante la inserción de 2 máquinas y la consulta de todas.

## 6 Rúbrica de Evaluación

**Puntaje Máximo: 20 puntos**

Criterio	Excelente (100%)	Bueno (80%)	Regular (60%)	Deficiente (40%)	No Cumple (0%)
<b>a) Capa del Modelo</b> (2 pts)	La clase Máquina está completamente definida e integrada al modelo.	Clase definida, pero con pequeños errores en atributos o métodos.	Clase incompleta o con atributos erróneos.	Clase mal estructurada o incoherente.	No se implementó la clase.
<b>b) Capa de Persistencia</b> (4 pts)	Persistencia totalmente funcional (lectura y escritura).	Funciona con errores menores o sin modularidad total.	Solo guarda o solo carga.	Guarda o carga incorrectamente.	No se implementó

					persistencia.
<b>c) Capa de Control</b> (4 pts)	CRUD 'Agregar' y 'Consultar todas' completamente funcionales e integrados con modelo y persistencia.	Controlador funcional con pequeños errores.	Solo una operación CRUD funciona correctamente.	Controlador con errores graves o incoherente.	No se implementó controlador.
<b>d) Capa de Presentación</b> (8 pts)	GUI funcional, intuitiva, con validaciones y visualización en grid.	GUI funcional, con pequeños errores de diseño o validación.	GUI incompleta (falta grid o formulario).	GUI con errores graves o poco funcional.	No se implementó GUI.
<b>e) Ejecución y Pruebas</b> (2 pts)	Operaciones ejecutadas correctamente, datos persistentes, capturas claras.	Funcionalidad completa con un error menor o falta de una captura.	Solo una operación funcional.	Ejecución parcial o sin persistencia.	No se realizaron pruebas.