

Laboratorio Calificado 07

2025

ÍNDICE

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

Laboratorio Calificado 07

Programación de la capa de persistencia para el Sistema de Gestión del Laboratorio de Mecatrónica

1 Instrucciones

Se ha creado la tarea “Laboratorio Calificado 7” en la sección Laboratorio Calificado 7 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:

L7_<Código del alumno de 8 dígitos>.<Extensión del archivo>

Como ejemplo, el nombre de archivo del alumno 20236969 sería “L7_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 los laboratorios anteriores se ha desarrollado un sistema de gestión de recursos del laboratorio de Mecatrónica, incluyendo:

- La capa del modelo (clase *Recurso*), con atributos como: identificador, nombre, cantidad, estado y ubicación.
- La capa de control (*Controller*), que gestiona los recursos mediante un repositorio en memoria.
- La capa de presentación, con una interfaz gráfica de usuario (*GUI*) en C++/CLI, que permite realizar las operaciones CRUD sobre los recursos.

Sin embargo, actualmente, los datos se pierden al cerrar la aplicación, ya que no se cuenta con una capa de persistencia que almacene la información de forma permanente.

En este laboratorio, se solicita desarrollar la capa de persistencia del sistema, de modo que los recursos del laboratorio puedan almacenarse en un archivo (por ejemplo, en formato texto plano, CSV, XML o binario) y se mantengan disponibles entre sesiones de ejecución del programa.

3 Objetivos

Implementar la capa de persistencia y lograr su integración con el sistema existente, de manera que:

- Al iniciar el programa, se carguen automáticamente los recursos almacenados en el archivo.
- Toda operación CRUD (crear, leer, actualizar, eliminar) realizada desde la GUI modifique también el archivo de persistencia.
- Al cerrar la aplicación, los datos actualizados se guarden correctamente.

4 Requisitos

El sistema debe cumplir con los siguientes puntos:

a) Implementación de la persistencia (10 puntos)

- Definir una clase o módulo responsable de la lectura y escritura de los datos del sistema en un archivo.

- El formato del archivo puede ser de texto, binario, XML o CSV.
- El diseño debe seguir el principio de separación de responsabilidades, evitando acoplar la lógica de persistencia con la lógica de negocio.

b) Integración de la persistencia con el Controller y la GUI (6 puntos)

- Al iniciar la aplicación, se deben cargar los datos del archivo en la memoria del sistema.
- Las operaciones CRUD realizadas desde la GUI deben reflejarse en el archivo de persistencia.
- El código debe mantener una estructura modular, clara y comentada.

c) Ejecución y pruebas (4 puntos)

- Probar el sistema ejecutando las siguientes operaciones:
 - Inserción de 3 recursos diferentes.
 - Modificación de un recurso existente.
 - Eliminación de un recurso.
 - Consulta de un recurso por su identificador.
 - Consulta de todos los recursos.
- Al reiniciar la aplicación, los cambios deben mantenerse.
- Incluir capturas de pantalla que muestren la GUI en cada operación y el contenido del archivo de persistencia antes y después de ejecutar el programa.

5 Entregables

- Código fuente completo en C++/CLI, bien comentado.
- Capturas de pantalla de una ejecución de prueba que muestre la GUI durante la inserción de 3 recursos, modificación, eliminación, consulta individual, y consulta de todos los recursos.

6 Rúbrica de Evaluación

Puntaje Máximo: 20 puntos

Criterio	Excelente (9-10 pts)	Bueno (7-8 pts)	Regular (5-6 pts)	Deficiente (3-4 pts)	No Cumple (0-2 pts)
Diseño e implementación del mecanismo de persistencia (10 puntos)	Se implementa correctamente una solución completa y robusta para la persistencia de los recursos (e.g., archivos planos, XML o binarios). La carga y el guardado son automáticos y transparentes para el usuario. La lógica de persistencia está desacoplada del controlador.	El mecanismo de persistencia funciona, pero presenta leves limitaciones (e.g., no guarda automáticamente o requiere reinicio manual). La estructura general es clara.	Se implementa persistencia básica, pero con errores recurrentes o falta de funciones clave (e.g., no carga correctamente los datos guardados).	La persistencia es parcial, inestable o depende de funciones que no están claramente integradas al flujo de uso. Se presentan fallas importantes.	No se implementa la persistencia o el código es erróneo/incompleto.
Integración de la persistencia con el controlador y GUI (6 puntos)	Toda la lógica de guardado y carga está correctamente integrada con el controlador, y la GUI refleja fielmente los cambios persistentes. No hay duplicación innecesaria de lógica. Código estructurado, modular y comentado.	Se logra una buena integración, aunque con pequeños errores o redundancias. La GUI funciona, pero puede requerir pasos adicionales para actualizarse.	La integración es parcial. Algunas operaciones CRUD no reflejan correctamente los cambios persistentes, o hay redundancia excesiva.	Integración mínima o con bugs frecuentes. La GUI no se actualiza adecuadamente con los cambios persistentes.	No se evidencia integración funcional. El sistema no utiliza los datos persistentes.
Ejecución y pruebas	El sistema compila y ejecuta correctamente. Se evidencia mediante	El sistema funciona y realiza las operaciones	Varias operaciones no funcionan bien.	El programa compila, pero fallan la	El programa no compila ni muestra

funcionales (4 puntos)	capturas que al iniciar la aplicación se cargan los datos persistidos, y que al cerrar se guardan cambios. Se presentan al menos 5 operaciones (insertar, modificar, eliminar, consultar individual y consultar todos). Capturas y PDF bien documentados.	principales. Una operación puede presentar un error menor. Capturas casi completas y documentación aceptable.	Capturas de pantalla incompletas o poco claras. Documento PDF deficiente.	mayoría de operaciones. Capturas insuficientes o falta de evidencia clara de persistencia.	evidencias de pruebas o persistencia.
-------------------------------	---	---	---	--	---------------------------------------