

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática



Ingeniería Informática

**PROGRAMACIÓN ORIENTADA
A OBJETOS**

UNIDAD 10
Trabajo Práctico Final
2023

UNIDAD 10

Trabajo Práctico Final

Todos los alumnos de Programación Orientada a Objetos de la carrera Ingeniería en Informática de la FICH-UNL deben realizar un Trabajo Práctico Final consistente en una aplicación escrita en C++, utilizando diseño orientado a objetos y con interfaz gráfica, que permita modelar un pequeño sistema real.

La exposición y defensa de dicho proyecto -cuyas características se indican a continuación- formará parte del examen final de la materia.

Premisas para la realización del Trabajo Práctico Final

1. Consideraciones Previas:

- 1.1. Será posible efectuar el diseño y desarrollo en grupos de hasta 3 personas. La conformación de los grupos se deja a cargo de los alumnos.
- 1.2. Proponer un tema o solicitar alguno a los profesores integrantes de la cátedra. El proyecto a desarrollar puede ser de cualquier índole: software de gestión, software didáctico o educativo, videojuegos, etc.
- 1.3. Cada grupo debe presentar los objetivos y el sistema que se pretende modelar **previamente** al examen final¹ para que un docente de la cátedra ajuste el nivel de desarrollo/dificultad de ser necesario. Según el tipo y tamaño de problema, el docente podrá limitar o expandir el alcance del proyecto para ajustarlo a la dificultad y duración esperada.
- 1.4. El sistema debe incluir técnicas de diseño orientado a objetos. Esto quiere decir que debe existir un modelo de objetos que represente la lógica del problema que el software resuelve.
- 1.5. La implementación de las clases específicas del problema que se resuelve

¹ Para esto el grupo deberá enviar una descripción detallada del proyecto o preferentemente solicitar una clase de consulta para mostrar las primeras versiones del mismo.

debe ser INDEPENDIENTE de la lógica de presentación (interfaz).² Existen, sin embargo, algunos casos particulares en los que se justifica omitir este requisito. Dichos casos deberán ser consultados y acordados con un docente de la cátedra previo a la presentación del trabajo final.

1.6. La codificación se hará en C++, utilizando cualquier combinación de compilador, sistema operativo, IDE, biblioteca, etc., a elección del usuario y aplicando los conceptos desarrollados en el cursado de la materia.

1.7. El manejo de archivos es obligatorio y no se admitirán componentes visuales para el manejo de bases de datos. La elección del tipo de archivo (texto o binario) deberá estar justificado según su función dentro del programa.

2. Presentación del TP final en el Examen:

2.1. Antes del examen final donde será evaluado el trabajo, es conveniente (aunque no obligatorio) que los integrantes del grupo realicen una o varias consultas para presentar el desarrollo con el objetivo de que el docente verifique los detalles de funcionamiento y documentación.

2.2. Durante el examen final el trabajo debe estar COMPLETO, y será defendido en forma individual por TODOS los alumnos que aprueben el examen escrito y por los promocionados por parciales que se presenten a la mesa examinadora. Los integrantes de un determinado grupo pueden presentarse juntos en una misma mesa, o individualmente en diferentes turnos de examen y se los podrá evaluar sobre el mismo proyecto.

2.3. El TP Final es parte del examen final de la materia, por lo tanto, si un alumno aprueba el escrito del examen final y no presenta el TP Final o este estuviera incompleto, será calificado con INSUFICIENTE, debiendo rendir nuevamente el examen en otra fecha.

2.4 Un alumno que haya finalizado el cursado en condición de regular o libre no podrá defender su proyecto en caso de no aprobar el examen escrito.

3. Material a presentar:

3.1. Código fuente. Codificación en C++ con adecuada documentación interna (programa fuente) en formato digital.

3.2. Programa de instalación: debe entregarse un programa de instalación (o en su defecto un archivo comprimido) que contenga todo lo necesario (ejecutable

² Esto quiere decir, por ejemplo, que las clases que representan un componente visual NO deben incluir código relacionado a guardar o leer datos en los archivos, ni realizar cálculos, sino simplemente debe llamar a métodos de los objetos que modelan el problema, y estos métodos NO deben hacer referencia a los componentes visuales ni a tipos de datos específicos de la biblioteca gráfica utilizada.

compilado, archivos dlls requeridos, recursos del programa como imágenes, archivos de datos, etc) para verificar el funcionamiento del software.³

El código fuente del proyecto debe ser entregado (digitalmente, enviando al docente un link de descarga, o copiándolo con un pendrive el día de la defensa) por el último integrante del grupo que presente dicho sistema.

³ El instalador (o archivo comprimido) es simplemente para garantizar la posibilidad de ejecutar el programa aunque la PC donde el alumno lo defienda no tenga las herramientas de desarrollo.