



---

# CÁTEDRA DE INGENIERÍA DE SOFTWARE I

## Profesor M. en C. Ing. Luis Alberto Muñoz Gómez Codificación y Pruebas

***Objetivo: Codificar el sistema diseñado mediante Ingeniería de Software Orientada a Objetos. Mostrar evidencia del desarrollo del software, en código fuente y documentación.***

### Descripción

Efectuar la implementación del sistema acorde a lo especificado en todas las actividades anteriores. El proyecto puede implementarse como una aplicación de consola, una aplicación con interfaz gráfica de usuario (GUI), o una aplicación Web, a elección del profesor (o del cliente).

### Requerimientos

1. Cumplir con el objetivo y la descripción de esta actividad.
2. Es indispensable para derecho a calificación mayor a 5/100 que para la aplicación desarrollada, su código fuente:
  - a) Se elabore en un lenguaje orientado a objetos (solo C++, Java o C# a elección del profesor o cliente); y
  - b) Presente cero errores de compilación; y
  - c) Muestre para los nombres de las clases y sus métodos, así como los nombres de variables atributo y parámetros de métodos, que sean exactamente los mismos que los ilustrados tanto en el diagrama de clases, como en los diagramas de secuencia del sistema y los diagramas de secuencia de instancias (aplicar en código fuente el camel case, a diferencia de que para las tablas del ER se use snake case para cada entidad del diagrama ER); y
  - d) El código ejecutable correspondiente funcione correctamente ante el usuario de la aplicación, esto es, sin bugs (haber sido plenamente probado el ejecutable); y
  - e) Sea compilable mediante un archivo de configuración de proyecto (diseñado para CodeBlocks, o Eclipse o Netbeans), o bien, sea compilable mediante comandos de consola (citando los comandos necesarios en un archivo de texto cuyo nombre sea "ComoCompilarEnConsola.txt"); y
  - f) En su totalidad aplicar el paradigma orientado a objetos, esto es, evitar subrutinas que no estén albergadas dentro de una clase; y

g) Sea entregado junto con todos los archivos necesarios para compilarlo (acorde al inciso "e"); y

h) No entregar el código ejecutable correspondiente, sino exclusivamente los códigos fuente;

i) Sea original y nada en lo absoluto parecido en diseño de código fuente, al de alguno de sus compañeros de equipo, ni de grupo, ni de otro grupo de semestres anteriores (excepto por lo dictado en el diagrama de clases actualizado del proyecto en cuanto a las firmas de los métodos a implementar); y

j) Muestre en ejecución un menú con opciones, propias a los casos de uso que le correspondan al alumno, que efectúen según los casos de uso a sea alta, baja, consulta, modificación, listado, búsqueda de registros dado un patrón de búsqueda, que guardar información de forma persistente (en base de datos) o bien la consulten, donde todas estas funcionalidades operen a plenitud ante el usuario; y

k) Almacenar toda la información correspondiente a los casos de uso asignados al alumno en una base de datos, y cargar de ella la información necesaria al ejecutarse el caso de uso, esto es, almacenar en memoria RAM solo la información necesaria previo a ejecutar el caso de uso, y que cada que se ejecute un caso de uso, hasta entonces acceder a la base de datos, ya sea para insertar, eliminar, consultar o actualizar la información; y

l) Cada miembro de cada equipo se encargará de gestionar, tanto en RAM como en disco, al menos dos casos de uso del proyecto, cuyos casos no sean los mismos que los de otro de sus compañeros de equipo y también diferente a los casos de uso de los demás equipos; y

m) Provea conexión a un SGBD MySQL, donde la base de datos se nombre como indique el profesor (o el cliente), y cuya contraseña para acceso total sea "12345". El nombre de la base de datos se cita en el excel Actividades; y

n) se realice la conexión a la base de datos solo hasta el momento de ejecutar una funcionalidad dada del sistema (las que utilizaría el usuario final de la aplicación), y que se cierre dicha conexión lo antes posible, una vez que termine de ejecutarse dicha funcionalidad, esto es, evitar abrir una conexión a base de datos que sirviera para todo el programa y que durara toda la ejecución del programa, o que una misma conexión se usara para ejecutar dos o más funcionalidades del sistema.

3. Es indispensable para derecho a calificación mayor a 60/100: Implementar una clase (modelada como una clase administradora) que se encargue de almacenar el listado de objetos sobre los cuales se aplicarán las operaciones ABC, esto es, una clase que cuente con un método particular, para cada una de las siguientes operaciones:

a) agregar un objeto al listado; y

b) eliminar un objeto del listado; y

c) obtener un objeto del listado dada una posición (correspondiente a un índice de arreglo); y

d) modificar un objeto del listado; y

e) obtener la lista de todos los objetos; y

f) buscar y obtener en su caso, un objeto del listado dado un patrón de búsqueda; y

g) que ninguno de los métodos implementados dentro de dicha clase contenga instrucciones que interactúen con el usuario de la aplicación (por ejemplo, en C++, evitar usar "cin2 y "cout" al interior de la clase).

4. Es indispensable para derecho a calificación mayor a 80/100: Implementar una clase que se encargue exclusivamente de escribir y leer el medio secundario usando una conexión a base de datos, modelada como un objeto de acceso a datos (DAO). Implementar dicha clase de tal manera que, si la reemplazáramos en un futuro por otro DAO, este último en lugar de usar una base de datos, escribiera y leyera un archivo de texto, o un archivo binario, y que este otro DAO tuviera los mismos métodos con las mismas firmas que el primer DAO.
5. Entregar los documentos solicitados en actividades anteriores, y que cuenten con todas las correcciones que hayan sido detectadas en la retroalimentación durante las sesiones clase, con especial atención a todos los artefactos de ingeniería que correspondan a los casos de uso que le tocaron de manera individual a cada alumno en la repartición que acordaron previamente; ningún caso de uso debe estar siendo manejado por otro compañero del grupo.
6. Si se trata de un trabajo individual, los casos de uso que corresponden al alumno son los que se le evaluarán para todos los artefactos, desde la actividad III hasta esta actividad. En el caso de un trabajo de equipo, lo entregado contenga los desarrollos de cada integrante en todos y cada uno de los artefactos y el código fuente muestre la integración del trabajo de todos los miembros del equipo.
7. Nótese que, no se está solicitando que cada miembro del grupo implemente la aplicación del proyecto final de forma completa, solo lo que corresponda a su(s) caso(s) de uso que le haya(n) sido asignado(s) en la repartición de casos de uso.
8. **INDISPENSABLE PARA DERECHO A CALIFICACIÓN MAYOR A 5/100** de este trabajo, cuando se elabore como trabajo de equipo, y por consecuencia para el proyecto final: el trabajo de equipo habrá de incluir la fusión de las propuestas de los compañeros del equipo en un solo trabajo, cuidando que no se copien su forma de programar, es decir, por ejemplo pensando en entregar la aplicación en C++, cada integrante elaborará sus propios archivos ".h" individuales, que funcionen con su aplicación que entregará de forma individual, y al elaborar el trabajo de equipo, se reúnen de forma presencial (no en línea) para entre todos formar un solo main.cpp que llame a las subrutinas de cada librería, pero sin que los compañer@s del equipo vean los ".h" de l@s otr@s compañer@s; si el profesor nota que los códigos fuente de los miembros del equipo se parecen, se anula la evaluación de su proyecto final de todo el equipo; si algún(a) compañer@ del equipo no logra hacer funcionar su entregable individual, será excluido de la integración de trabajo del equipo y por consiguiente, de la presentación del proyecto final, esto es, es imprescindible saber programar una aplicación con manejo de base de datos con su estilo individual, esto es, el alumno nunca vea el código fuente de las clases administradoras, ni los DAOs, ni los ".h" de sus compañeros de equipo, sino hasta después de que hubiesen presentado su proyecto final, lo único que pueden ver todos en conjunto es el "main.cpp" de sus compañeros y ese "main.cpp" no muestre más que solo la

subrutina “main”. El archivo principal de la aplicación, por decir en C++ el “main.cpp” contenga solo una subrutina (la “main”), y las demás subrutinas necesarias para su aplicación se encuentren en uno o más archivos “.h” que sean llamados desde “main”; en este caso “main” solo se encargue de mostrar el menú principal de la aplicación; en el caso de una aplicación para interfaz gráfica de usuario (GUI), la clase que modele la ventana principal contenga solo el menú del programa y que la acción de un botón (o comando de un menú desplegable) del menú principal implique solo mostrar el contenido de la ventana necesaria, esta última ventana estará bajo el control de un solo alumno responsable de ella y l@s demás compañer@s del equipo eviten saber cómo funciona esa ventana en su interior.

9. Actualizar el archivo Excel que muestre las tareas y recursos utilizados según se describió en la actividad previa sobre “Ingeniería de Requerimientos”.
10. Entregar todos los artefactos de actividades anteriores (desde la actividad III hasta esta actividad), en una carpeta diferente cada artefacto.

## **Criterios de Evaluación**

- Los establecidos en las “Reglas de Operación y Evaluación” del curso.
- Cumplir con la fecha límite de entrega citada en el Excel de Actividades.
- Cumplir con lo establecido en el Formato Estándar para Entrega de Actividades en Documento.
- Cumplir con Requerimientos de Valor Agregado en Código Fuente (hasta el req. “ZZ”)
- Entrega en al menos un lenguaje: C++ (ANSI) o Java o C#.
- Es indispensable la entrega de un programa con variables, comentarios e impresiones a consola completamente en idioma Español.
- Calificación en base a cobertura de requerimientos y fecha de entrega.