

Programación con Objetos (C1) - Trabajo Final

1. Introducción

Este documento presenta los detalles del Trabajo Final, que junto con la aprobación del examen parcial es uno de los requisitos para la aprobación de la materia.

El trabajo tiene las siguientes características:

- Consiste en el diseño e implementación de un modelo para un problema determinado.
- Deberá desarrollarse en grupos de 2 personas, y deben trabajar en conjunto.
- Deberá desarrollarse en clase pero, deben contemplar que deberán dedicar cierto tiempo extra fuera de clase.
- Deberá desarrollarse de manera iterativa e incremental utilizando la técnica de Test-Driven Development (TDD).
- El enunciado será designado por el equipo docente.
- Los docentes realizarán el seguimiento de cada trabajo, respondiendo las consultas que surjan, y podrán solicitar entregas parciales para validar el trabajo realizado hasta el momento.
- El resultado del trabajo final es la implementación, tests y el código resultante, y la elaboración de un informe que incluya el modelo y decisiones de diseño.
- **No subir el código a ningún repositorio público, ni debe de compartirse.**

2. Criterios de Evaluación

Cada grupo debe entregar su proyecto completo. La calificación final del trabajo será **individual** y dependerá de los siguientes criterios de evaluación:

- El diseño y la implementación. El modelo definido, la aplicación de los conceptos y metodologías aprendidas durante el curso, así como que el trabajo cumpla o no con las especificaciones mínimas requeridas.
- Comprensión y conocimiento sobre el dominio asignado.
- El tiempo de trabajo en clase se utilizará no sólo para realizar consultas sino para discutir acerca del diseño, y programar de a pares para ir resolviendo el trabajo.
- La calidad de la documentación. El trabajo entregado deberá cumplir con ciertos requisitos mínimos respecto a su documentación (detallados más abajo).

Tener en cuenta que todos los criterios de evaluación son importantes para definir la nota final del mismo.

2.1. Formato del Informe

El informe del trabajo desarrollado debe contener como mínimo las secciones descritas abajo. Además puede incluir cualquier información que considere relevante.

- **Carátula:** con enunciado asignado, integrantes del grupo, materia y fecha.
- **Alcance:** Escriba de 5 a 15 renglones
 - Qué partes del enunciado se implementaron, cuáles no y por qué.
 - En el caso de agregar alguna característica extra a las que figuraban en el enunciado, enumérelas y explique por qué se agregaron.
- **Modelo:** Describa cuáles son los objetos más importantes de su sistema, y sus principales responsabilidades.
- **Progreso:** Explique cómo se fue construyendo iterativa e incrementalmente el TP a lo largo de su realización. Que cosas asumieron que no estaban contempladas en el trabajo.
- **Dificultades:** Describa qué dificultades tuvo para implementar el TP. Especifique qué partes fueron más difíciles y cuáles fáciles. Puede incluir las dificultades del testing.
- **Conclusiones:** Escriba las conclusiones que haya obtenido a partir de este trabajo. Puede describir por ejemplo qué conceptos de los aprendidos utilizó en mayor medida y cuáles considera que no pudo aplicar.

2.2. Entrega

La entrega del Trabajo Final completo se subirá al **Campus**. Todo el contenido de la entrega debe ir en un archivo comprimido en formato **.zip**, y con nombre de acuerdo a la convención **TrabajoFinal-Grupo-X** donde **X** es el número de grupo asignado.

El contenido debe incluir como mínimo el archivo **.st** de la solución y un documento de texto (en cualquier formato estándar como **.pdf**, **latex**, **.docX**, etc)

3. Enunciados

Los enunciados sirven a modo de guía para comprender el dominio y saber lo que se espera de ellos, pero de ser necesario los docentes podrán ajustar el alcance, cambiar funcionalidades o dejar en claro que cosas se esperan, si eso ayuda a una mejor evaluación.

Cada grupo tendrá asignado de manera aleatoria uno de los siguientes cuatro enunciados:

Biblioteca

La biblioteca de la Universidad necesita renovar su aplicación para manejar los socios, el inventario y los préstamos que se otorgan.

Se nos pide poder registrar a los socios nuevos que ingresen, de estos necesitamos saber su nombre y apellido, su número de DNI y su fecha de registración. Cada socio debe renovar su membresía anualmente, de lo contrario no podrá pedir prestados libros.

Para poder registrar un nuevo libro en la biblioteca se necesita lo siguiente: su ISBN, autor, título, fecha de publicación, idioma y una lista de temas. No puede agregarse otro libro con el mismo ISBN. La biblioteca tiene una cantidad de ejemplares de cada libro, donde puede irse agregando más con el tiempo, de cada ejemplar agregado nos gustaría saber la fecha de ingreso y su número de inventario.

Para poder realizar un préstamo hay que cumplir una serie de reglas:

- No se pueden prestar ejemplares a un socio cuya registración/renovación expiró, o está sancionado/vetado.
- Un socio no puede pedir prestados más de 5 ejemplares al mismo tiempo.
- Un socio no puede pedir prestados dos ejemplares de un mismo libro.
- Tampoco se debe permitir prestar libros que no posean ejemplares disponibles.

Los préstamos pueden renovarse por una semana, siempre y cuando la renovación se realice dentro del plazo de entrega y el socio esté habilitado para hacerlo. Aunque no todos los libros tienen la misma prioridad, existen libros que no podremos realizar una renovación de un préstamo y otros que el tiempo es limitado a 1 día.

Además, existen los préstamos extendidos, que son determinados por una fecha de inicio y otra de fin, cuya duración es hasta la fecha de fin inclusive. Pero, no se pueden renovar.

En caso de que un socio no devuelva un libro en el plazo estipulado, se deberá sancionar automáticamente en el momento que realice la devolución.

La biblioteca puede verificar en cualquier momento si se debe sancionar y/o vetar a un socio. Se debe repetir este mismo proceso para todos los socios inscriptos.

Un socio que fue sancionado más de 3 veces quedará vetado de por vida, esto implica que si se quiere volver a registrar con el mismo DNI no podrá y tampoco deberá de poder pedir prestado ni renovar la membresía.

La biblioteca tiene una serie de listados recurrentes que utiliza todo el tiempo:

- Socios que están sancionados actualmente.
- **Socios solidarios:** un socio solidario es aquel que devuelve los libros rápidamente, para que otros puedan sacarlo. El listado debe incluir sólo a aquellos socios que hayan devuelto el préstamo en tiempo y forma.
- Poder buscar libros utilizando filtros de la siguiente manera:
 - Por autor
 - Por una lista de temas

- Fecha de inicio y/o fecha de fin
- Por idioma
- Por cantidad de ejemplares
- Que sean combinables

Ejemplo: *buscar aquellos libros que sean de Alan Kay, de Smalltalk, desde 1960, hasta 1990, en Inglés.*

Genere un archivo con el contenido de alguno de los listados mencionados anteriormente.

Del libro queremos saber toda su información, la cantidad de ejemplares que posee y la cantidad de préstamos actuales.

Algo que también desean poder hacer es la importación masiva de libros usando un archivo **.csv**

Mini-MiUNQ

La universidad quiere poder disponer de un sistema personalizado a-la-Unix para que el personal y estudiantes de la universidad puedan crear, subir, modificar y compartir diferentes tipos archivos.

Inicialmente, este sistema debe permitir tener archivos de texto y carpetas independientes para cada usuario del sistema.

Cuando un usuario inicia sesión, mediante usuario y contraseña, comienza en el directorio raíz. El usuario debería de poder ver en todo momento el contenido del directorio actual. Cada directorio puede contener cualquier tipo de archivo o carpetas dentro.

Para navegar basta con que se indique la ruta de la próxima carpeta a la que se quiere ir (de las que están listadas en el directorio actual) o volver a la anterior.

Obviamente, debería de ser capaz de crear carpetas y archivos, inicialmente alcanza con tener el nombre con la posibilidad de cambiarlo en cualquier momento. Además, debería tener la capacidad para poder editar un archivo de texto, al editarlo debería de cambiar el contenido que tiene por el nuevo. También se debería de permitir borrar tanto archivos como directorios, si se llegase a borrar un directorio que no está vacío, deberá de arrojar un error.

Para poder administrar el sistema existen los Administradores y ROOT, estos deben ser capaces de:

- Dar de alta, modificar Usuarios, solamente ROOT puede dar de baja Usuarios.
- Poder crear otros administradores pero no poder eliminarlos ni modificarlos, solo ROOT puede hacer esto.

ROOT representa el primer administrador, capaz de realizar toda acción, un administrador para iniciar sesión también utiliza credenciales de usuario y contraseña.

El principal foco de la aplicación es poder compartir contenido entre los diferentes usuarios de la plataforma. Para lograr esto necesitamos que:

Un usuario pueda compartir una carpeta o un archivo específico a otros usuarios con diferentes privilegios (por defecto todo contenido creado es privado):

- Lectura, para acciones de navegación o visualización de contenido.
- Escritura, para poder modificar o crear nuevos archivos o carpetas.
- Borrado, para poder eliminar archivos o carpetas.

También existe un usuario Público, que puede ingresar al sistema sin requerir usuario ni contraseña y solamente es capaz de visualizar todo el contenido que los demás usuarios declaren como "público".

Si un usuario quiere realizar alguna acción y no tiene permisos deberá arrojar un error.

Para poder agilizar la búsqueda entre diferentes carpetas y archivos, debemos permitir poder filtrar archivos según estos criterios:

- Que contengan un texto dado
- Una fecha de creación
- Que pertenezcan a un usuario específico
- Que tenga el permiso dado (lectura, escritura y/o borrado)
- Que sean combinables

Ejemplo: Obtener los archivos que contengan la palabra "Unix" con fecha de creación en el 1-9-2019 del usuario Pepe que sean de lectura

Como administrador, nos interesa poder escribir en un archivo del disco duro, la información de una carpeta. El resultado esperado debe ser algo parecido a:

Usuarios: <Nombre del Usuario>

/ruta/del/directorio: 3 elementos

archivo1, fecha de creación, usuarios compartidos

archivo2, fecha de creación, usuarios compartidos

directorio1, fecha de creación, cantidad de elementos dentro, usuarios compartidos

(El formato no tiene que ser igual, solo debe contener la misma información)

Sumado a esto, algo que también desean poder hacer una importación masiva de usuarios usando un archivo **.csv**