



Universidad de Buenos Aires
Facultad de Ingeniería
Algoritmos y Programación I

TRABAJO PRÁCTICO GRUPAL
“GESTIÓN DE BIBLIOTECA”
1º CUATRIMESTRE 2007

Grupo	Docente asignado		Calificación
		Nota	
		Firma	
		Fecha	

Padrón	Apellido y nombres

Fecha	Entrega
31/5	Entrega de diagramas
14/6	Compilación
21/6	Vencimiento

TP – Segunda parte

Implementar una extensión al sistema de gestión de la biblioteca, que consiste en una serie de funcionalidades para la gestión de préstamo de libros. Además de algunas pequeñas mejoras al sistema existente.

Extensiones

1. ABM de Socios, con la siguiente información:
 - Número de socio (código numérico)
 - Nombre del socio
 - DNI
 - Dirección
2. Préstamo de un libro a un socio, dados los siguientes datos ingresados por el usuario del sistema:
 - Número de socio
 - Número de ejemplarSe deberá considerar la fecha de préstamo (fecha del día).
3. Devolución de un libro dados los siguientes datos ingresados por el usuario del sistema:
 - Número de ejemplarEl sistema deberá indicar si el ejemplar fue devuelto fuera de término. El sistema establecerá un límite de una semana (7 días corridos) para su devolución a partir de la fecha de préstamo. No se considerarán excepciones respecto a los días feriados.
4. Importar autores: Cargar los datos de autores desde un archivo CSV, organizado del siguiente modo: *Número, Nombre*. Si el autor existe, debe sobrescribir los datos.
5. Importar títulos: Cargar los datos de títulos desde un archivo CSV, organizado del siguiente modo: *Número, Nombre, Número de autor, número de género*. Debe verificar la existencia de los números de autor y género, y la NO existencia previa de dicho título (sólo altas).
6. Visualización de datos de un socio: Dado un número de socio ingresado por teclado, listar sus datos personales y (número-nombre-fecha de devolución) de los libros que posee en préstamo.
7. Listado de libros adeudados: Emitir un listado de ejemplares (número ejemplar-nombre del título-número de socio-nombre del socio-fecha de devolución-días de atraso) cuya fecha de devolución haya vencido. El listado debe estar ordenado de mayor a menor, en cantidad de días de atraso.
8. Exportar información de deudas: Idem anterior, pero exportando los datos a un archivo CSV.
9. Ejecución en modo TEST: El programa debe contar con la opción de ejecutarse en modo test. Este modo se habilitará cuando se ejecute el programa recibiendo como primer y único parámetro un string con una fecha en formato AAAAMMDD. Esta será la fecha que utilizará el programa para todas sus operaciones. En caso de ejecutarse en modo "normal", se tomará la fecha de la computadora.

Mejoras

10. Implementar la persistencia de datos: Esto incluye todos los datos maestros y el contenido de ejemplares en la biblioteca. Es decir, que los datos se deberán mantener en archivos en disco.
11. Visualización de ejemplares de un título y su ubicación en la biblioteca: Se indicará pasillo y estante para los títulos que estén en la biblioteca, y para los ejemplares prestados, en lugar de pasillo y estante, se indicará su fecha de devolución. Los ejemplares prestados no contarán para el mapa de la biblioteca.

Requisitos

Se deberá entregar el trabajo práctico en las siguientes etapas:

1. Entrega de diagramas: Se debe entregar una carpeta (prolija y sin hojas sueltas), conteniendo:
 - a. Carátula. La misma que entregamos junto con este enunciado completando los datos requeridos.
 - b. Esquema del modelo de datos propuesto.
 - c. Diagramas básicos con los principales módulos a desarrollar.
 - d. Plan de asignación de tareas dentro del grupo.
2. Compilación:
 - a. Carátula. La misma que entregamos junto con este enunciado completando los datos requeridos.
 - b. Una primera versión pasada a máquina (en Pascal).
3. Vencimiento:
 - a. Carpeta (prolija y sin hojas o discos sueltos) conteniendo:
 - i. Carátula. La misma que entregamos junto con este enunciado completando los datos requeridos.
 - ii. Manual de usuario.
 - iii. Listado del código fuente.
 - iv. Explicación de los principales módulos.
 - b. Disquete ó CD con archivos fuente y archivo ejecutable (incluir un disco de backup).

Consideraciones:

El código fuente deberá ser modular y con los comentarios pertinentes. Deberá reutilizarse código (es decir, no implementar las mismas funcionalidades más de una vez en distintas partes del sistema). Los nombres de tipos, variables, procedimientos y funciones deberán ser claros y representativos.

Se considera importante que el diseño del soporte de datos y su manipulación (en memoria o archivos) sea eficiente según la utilización y las limitaciones de cada caso. Se aconseja revisar el diseño del mismo con el docente asignado antes de implementarlo.

Se considerará desaprobado aquel Trabajo Práctico que fuese entregado sin alguno de estos componentes básicos, o que fuese presentado fuera de término. Deberán respetarse estrictamente las fechas de presentación indicadas por la Cátedra. También serán desaprobados todos aquellos Trabajos Prácticos que la Cátedra considere que son copia, independientemente que alguno de los dos sea el verdadero original.

Importante:

Cualquier consideración o definición que no se encuentre detallada en este enunciado, deberá establecerse con el docente asignado.

Del mismo modo, el docente asignado es quien pedirá las correcciones pertinentes a cada una de las etapas (si fuera necesario) y establecerá los plazos para el cumplimiento de las mismas. Adicionalmente es el único que podrá, excepcionalmente, acordar otras fechas de presentación de los trabajos, distintas de las establecidas, siempre y cuando sean acordadas antes de la fecha de presentación; en caso contrario, el no cumplimiento significará la no aprobación de la materia. También será el único que podrá establecer o autorizar alguna modificación del alcance del mismo.