

<u>UTN – FRMDP Mar del Plata</u> <i>TSP - Programación 1</i> <i>Trabajo Práctico Final</i> <i>Mayo - Junio 2024</i>	<i>Integrantes del grupo</i>	<i>Nota</i>
---	-------------------------------------	--------------------

Introducción

Con el propósito principal de integrar todo lo aprendido en la materia Programación 1 hemos planteado la siguiente problemática:

- Codificar un sistema de Logueo que gestione la estructura **Usuario**.
- Administrar un archivo de **Libros** (Alta, Baja, Modificación, Consulta y Listados)
- Administrar un listado de libros favoritos/colección para cada usuario.
- Administrar un archivo de **comentarios/valoraciones** (Alta, Baja, Modificación, Consulta y listados)

Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo (se formarán grupos de 3 o 4 estudiantes) a partir del desarrollo de un proyecto de software. Para que este tipo de proyectos sea más exitoso, deben llevarse a cabo desde un enfoque que facilite alcanzar los Objetivos de Aprendizaje propuestos.

Una de las ideas centrales es desarrollar competencias profesionales y preparar al futuro programador para el mundo laboral y el trabajo en equipo.

En un ambiente de aprendizaje colaborativo, los estudiantes:

- Construyen conocimiento y en lugar de recibirlos en forma pasiva;
- Se involucran y se comprometen directamente con el descubrimiento de nuevo conocimiento;
- Se exponen a puntos de vista alternativos e ideas contrapuestas, de forma tal que pueden sacar sus propias conclusiones y así transformar conocimientos y experiencias previas y de esta manera comprender con mayor profundidad;
- Transfieren conocimientos y habilidades a nuevas situaciones o circunstancias;
- Se responsabilizan y apropian tanto de su aprendizaje continuo de contenidos curriculares, como del desarrollo propio de competencias;
- Los estudiantes colaboran para el aprendizaje del grupo y el grupo colabora en el aprendizaje individual de estos.

Objetivos

De aprendizaje:

- Incorporar Arreglos, Matrices, Modularización, Estructuras de Datos y Archivos Binarios.
- Trabajar en forma colaborativa.

Metodológicos:

- Ser capaces de trabajar en un proyecto complejo, aplicando técnicas de desarrollo de software.
- Lograr integrar contenidos de otras asignaturas.
- El grupo deberá ir mostrando el avance sobre el trabajo en clase.

Modo de Evaluación del Trabajo Práctico

- Se establece el desarrollo de un trabajo práctico final, brindando una fecha límite de entrega del mismo: **Según planificación de cada comisión**
- Si el sistema está codificado en su totalidad y funciona correctamente, se considerará aprobado con una nota mínima de 6.
- Si el sistema no está codificado en su totalidad (como mínimo un 60 % en cada inciso), se considerará desaprobado y el grupo presentará la versión final en la fecha de recuperatorio.
- En la fecha de recuperatorio deberá cumplir las pautas mínimas establecidas precedentemente para la aprobación de la instancia recuperatoria.
- **Es obligatorio la presentación de este trabajo para aprobar la materia.**

Pautas Generales

Como metodología de trabajo, se requiere crear un documento de texto (o carpeta) en Google Drive que será compartido a todos los miembros del grupo (y también al equipo docente, publicando el link vía campus virtual en el foro correspondiente), con el fin de plasmar los avances del proyecto de forma de construir la siguiente documentación a entregar:

Informe final - Documentación a entregar: [15 puntos]

- Diario de trabajo: Día a día qué actividades se desarrollaron y el responsable de cada una.
- Matriz de soluciones: Que problema tuvieron y cómo lo resolvieron.
- Manual de usuario: Breve explicación de cómo funciona el sistema, pueden usar imágenes, videos, presentaciones, etc.
- Diagrama de estructuras: Esquema de las estructuras utilizadas y sus relaciones.
- Documento con usuarios y passwords de prueba. Archivos cargados con 100 usuarios, 150 libros y por lo menos 100 comentarios para probar el sistema.

Asimismo, deberán crear un tablero en www.trello.com para distribuir las tareas entre los integrantes del grupo y trabajar de forma organizada. **A medida que avancen con el desarrollo del trabajo, realizarán capturas de pantalla y las adjuntarán al Diario de trabajo.** Deberán compartir el tablero con el equipo docente, publicando el link vía campus virtual en el foro correspondiente.

Al momento de efectuar la entrega del proyecto, deberán agregar todo el material digital solicitado precedentemente.

La función principal - Main() y menús: [15 puntos]

IMPORTANTE: LA NAVEGABILIDAD DEL PROGRAMA.

El sistema deberá contar con una presentación amigable con el usuario, construir menús de acceso a las diferentes estructuras y funcionalidades del sistema, **y de manera directa o indirecta, permitir probar todas las funciones desarrolladas.**

El desarrollo del sistema deberá ser ordenado, identificando con comentarios cada una de las funciones realizadas, explicando brevemente lo que realizan. Se tendrá en cuenta, al momento de evaluar, la prolijidad del código y la organización de los módulos. Se recomienda agrupar los mismos por funcionalidad.

Importante, el sistema debe trabajar en memoria principal consumiendo las estructuras generadas para tal fin. Al inicio del sistema se consultarán los archivos binarios y se actualizará esa información al finalizar la ejecución del programa.

Detalle de estructuras y funcionalidad básicas: [70 puntos]

Al momento de iniciar el sistema se le debe mostrar al usuario una pantalla que le permita ingresar con su email y contraseña al sistema. Una vez ingresadas las credenciales, si el usuario existe, dependiendo de su rol (si es un usuario normal o un administrador) se le debe mostrar un menú con el listado de las siguientes funcionalidades.

Libros:

- Agregar un libro al sistema. El usuario **no** podrá establecer una valoración por defecto.
- Ver libros. Ver listado de libros según categoría. Ver libros por autor.
- Búsqueda de libros por título.
- Agregar/Quitar un libro como favorito.
- Cada libro tiene un puntaje de valoración que es calculado bajo el promedio de todas las calificaciones de los comentarios de ese libro.

Asimismo, además de las funcionalidades anteriores, un usuario administrador podrá:

- Dar de baja un libro, eliminar definitivamente un libro. Eso implica eliminar los comentarios/reseñas de ese libro y eliminar como favorito de todos los usuarios el id de ese libro.

Comentarios/Reseña:

- Agregar un comentario a un libro.
- Ver los comentarios de otros usuarios sobre un libro.
- Modificar y eliminar una reseña/comentario que el mismo realizó con anterioridad.

Asimismo, además de las funcionalidades anteriores, un usuario administrador podrá:

- Eliminar a demanda comentarios/valoraciones de cualquier usuario.

Usuario:

- Un usuario puede crearse una cuenta (Registrarse). Para este proceso de registro/onboarding se deben validar los siguientes campos: email y contraseña.
 - o El email debe contener un @ y un .com. Además se debe validar que el email no este registrado previamente por otro usuario.
 - o La contraseña debe contener por lo menos 1 mayúscula y 1 minúscula.Al momento en que finaliza el proceso de registro, se lo debe redirigir a la pantalla de login.
- Un usuario debe poder ingresar al sistema con sus credenciales: email y contraseña. En caso de que el email no exista se le debe indicar que no existe

usuario con esos datos. En caso de que la contraseña no sea correcta, se le debe indicar que algunos de los campos son incorrectos.

- Un usuario una vez logueado en el sistema podrá modificar su información personal.
- Una opción de deslogueo para salir del sistema y que pueda ingresar con otra cuenta.

Asimismo, además de las funcionalidades anteriores, un usuario administrador podrá:

- Ver todos los usuarios registrados en el sistema.
- Dar de baja de usuarios.

Para la persistencia de datos en el archivo utilizaremos las siguientes estructuras de datos:

```
typedef struct{
    int idLibro; /// único, autoincremental
    char titulo[100];
    char editorial[50];
    char autor[50];
    char categoria[50];
    float valoracion;
    int eliminado; /// 0 si está activo - 1 si está eliminado
}stLibro;
```

```
typedef struct {
    int idComentario; /// único, autoincremental
    int idLibro;
    int idUsuario;
    char tituloComentario[100];
    char descripcion[500];
    int puntaje; /// de 0 a 5 "estrellas"
    char fechaComentario[20]; /// o se puede hacer con dia, mes, anio.
    int eliminado; /// 0 si está activo - 1 si está eliminado
}stComentario;
```

```
typedef struct{
    char calle[50];
    int altura;
    int cp;
    char ciudad[100];
    char localidad[50];
    char pais[100];
} stDomicilio;
```

```
typedef struct{
    int idUsuario;
    char email[100];
    char password[20];
    char username[20];
    int esAdmin; /// 1 = si 0 = no
    char genero;
    char fechaNacimiento[20]; /// o se puede hacer con dia, mes, anio.
    int librosFavoritos[50]; /// se van a guardar los id de los libros favs.
    char dni[10];
    stDomicilio domicilio;
```

```
    int eliminado; /// 0 si está activo - 1 si está eliminado
}stUsuario;
```

Como ya se ha dicho, toda la información administrada por el sistema se persistirá en 3 (tres) archivos binarios: “usuarios.dat”, “libros.dat” y “comentarios.dat”.

(si lo desean, podrán agregar funcionalidades)