

| | | |
|--|------------------------------|-------------|
| UTN – FRMDP Mar del Plata TSP - Laboratorio 1 Trabajo Práctico Final <i>Junio 2021</i> | Integrantes del grupo | Nota |
|--|------------------------------|-------------|

Introducción

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

- Codificar un sistema de Logueo que gestione la estructura **Usuario**.
- Administrar un archivo de **Contenidos** (Alta, Baja, Modificación, Consulta y Listados)
- Administrar un listado de contenidos favoritos para cada usuario.
- Administrar un archivo de **Mensajería** entre usuarios.

Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo (se formarán grupos de 2, 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. Vale aclarar que no puede aprobar de manera directa.
- **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.

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]

Un poco de narrativa

La red -social

Se nos pide desarrollar un sistema donde diferentes usuarios puedan ingresar y consumir contenidos de diversos tipos, de interés general o los que a ellos les interesan según sus gustos configurados. Asimismo, podrán generar contenido de su gusto y publicarlo, guardándolo en la base de datos general de la red para que otros puedan consumirlo.

Al ingresar por primera vez, cada usuario podrá elegir sobre diversas categorías (Gaming, Electrónica, Deportes, Turismo, Música, Panes de masa madre, etc) y el sistema podrá ofrecerle diversas publicaciones sobre cada temática.

Asimismo, los usuarios podrán guardar en su colección las publicaciones o contenidos que más les interesen para no tener que volver a buscarlos.

Al iniciar el sistema, una vez logueado un usuario, se presentará una pantalla con publicaciones acordes a la elección del usuario o bien, contenido random si este no ha seleccionado algún tema en particular.

El sistema proporcionará un acceso al perfil del usuario donde podrá verificar todos sus datos como así también acceder al contenido guardado.

Como la frutilla del postre (**aquellos grupos que quieran sumar puntos extras - los grupos de 4 integrantes lo tienen que hacer obligatoriamente**), cada usuario podrá recomendar a otro una publicación en particular. Esto generará un mensaje que el usuario destinatario visualizará cuando se loguee al sistema, y para el usuario remitente le permitirá sumar puntos a su cuenta y subir de nivel (no sabemos aún para que... pero pinta que va a estar bueno). El puntaje se determinará en base a los puntos que tenga la publicación y un algoritmo maaagique (leerlo con voz de Jose Maria Listorti).

El sistema podrá visualizar listados de las estructuras administradas (vista admin), permitiendo también dar de alta contenidos, verificar puntajes de los usuarios, etc.

Beta version 1.0

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

Estructura de Usuario

```
typedef struct
{
    int idUsuario;                // auto incremental
    char nombre[30];
    char apellido[30];
    char userName[20];
    char password[20];
    char mail[30];
    char genero;
    int listadoIdsContenidosGuardados[50];    8  19  33
    int validosContenidos;                3
    int puntaje;
    int nivel;
    ??? contenidosFavoritos ??? ; // pensar como administrarlos
    int rol;                ??? // 1: es admin - 0: es comun
    int activo;                // indica 1 o 0 si el usuario está activo
} stUsuario;
```

Estructura de Contenido

```
typedef struct {
    int idContenido; // auto incremental
    char titulo[30];
    char descripcion[300];
    char categoria[30];
    int likes;
    int puntosPorCompartir;
    int activo;                // indica 1 o 0 si el contenido está activo
} stContenido;
```

Estructura de Mensajería

```
typedef struct {
    int idUsuarioEmisor;
    int idUsuarioReceptor;
    int idContenidoEnviado;
    char mensaje[300];
    int leído;                // 1:si 0:no
} stMensajeria;
```

Al cargar las tres estructuras se deben persistir en los archivos correspondientes. El campo `listadoIdsContenidosGuardados[i]` (`stUsuario`) e `idContenido` (`stContenido`) están relacionados, para poder recuperar la información a partir de dicho campo.

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

El sistema tendrá que proporcionar el acceso a las diferentes funcionalidades mínimas :
(si lo desean, podrán agregar funcionalidades)

Funciones para ABMCL de Usuarios, Contenido y Mensajería

- Alta, Baja, Modificación, Consulta y Listados de Usuarios.
- Alta, Baja, Modificación, Consulta y Listados de Contenido.
- Alta, Baja, Modificación, Consulta y Listados de Mensajería.
- Validación en el ingreso de los Datos.
- Buscar Usuario.
- Buscar Contenido.

Funciones para manejo de archivos binarios

- Persistencia de datos en Archivos.
- Listar Usuarios (vista admin).
- Listar Contenido.
- Mostrar mensajes al iniciar sesion (si los hubiera)
- Validar si el Usuario ya existe (para alta de usr).