

UTN – FRMDP Mar del Plata TSP - Laboratorio 1 Trabajo Práctico Final Mayo - Junio 2022	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 gestión de Clientes y sus Pedidos
- Administrar un archivo de **Clientes** (Alta, Baja, Modificación, Consulta y Listados)
- Administrar un archivo de **Pedidos**.

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

HaC **Pedidos**

Se nos pide desarrollar un sistema donde diferentes clientes puedan ingresar y realizar pedidos de diversos tipos (comidas, bebidas, compras de almacén, verdulería, etc... pensar y volar un poco en esta parte!!!)

Asimismo, podrán consultar sus pedidos generados previamente, darlos de baja (cancelarlos) y ver el costo total de sus pedidos.

El sistema proporcionará un acceso al perfil del cliente donde podrá verificar todos sus datos como así también acceder a los pedidos generados.

El sistema podrá visualizar listados de las estructuras administradas (vista admin), permitiendo también eliminar pedidos, clientes, etc.

Podrán optar por diferentes enfoques al momento de ingresar al sistema donde, por ejemplo, el propio sistema muestre promociones o “combos” para que los clientes realicen pedidos; o que cada cliente puede escribir su propio pedido; o que el cliente puede elegir entre diversos artículos para poder generar sus pedidos; etc.

Versión BETA 1.0

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

Estructura de Cliente

```
typedef struct
{
    int idCliente;                // auto incremental
    char nombre[30];
    char apellido[30];
    char userName[20];
    char password[20];
    char mail[30];
    char genero;
    int rol;                      ??? // 1: es admin - 0: es común
    int activo;                  // indica 1 o 0 si el cliente está activo
} stCliente;
```

Estructura de Pedido

```
typedef struct {
    int idPedido;                // auto incremental
    int idCliente;              // este valor corresponde al id del cliente que realizó el pedido
    char fecha[12];             // dd/MM/aaaa
    char descripcion[300];
    float costo;
    int pedidoAnulado;          // indica 1 o 0 si el pedido fue anulado o no
} stPedido;
```

Al cargar las dos estructuras se deben persistir en los archivos correspondientes. El campo `idCliente` en las estructuras `stCliente` y `stPedido` 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 2 (dos) archivos binarios: "clientes.dat" y "pedidos.dat".

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

Funciones para ABMCL de Clientes y Pedidos

- Alta, Baja, Modificación, Consulta y Listados de Clientes
- Alta, Baja, Modificación, Consulta y Listados de Pedidos.
- Validación en el ingreso de los Datos.
- Buscar Pedidos.
- Buscar Clientes.

Funciones para manejo de archivos binarios

- Persistencia de datos en Archivos.
- Listar Clientes (vista admin).
- Listar Pedidos (propios del cliente o todos en vista admin). Aplicar filtros
- Validar si el Cliente ya existe (para alta de Cliente).

Para organizar un poco el ingreso al sistema, podrá generar un menú con las siguientes características y funciones asociadas:

- Clientes
 - Alta de Clientes
 - Funciones:
 - Carga de Datos Clientes (verificar de no cargar un mismo cliente)
 - Guardar en Archivo
 - Baja de Clientes
 - Funciones:
 - Buscar cliente
 - Dar de baja el cliente
 - Guardar los cambios en el archivo
 - Modificación de Clientes
 - Funciones:
 - Buscar cliente
 - Modificar cliente
 - Guardar los datos en el archivo
 - Listar Clientes
 - Mostrar todos los clientes
 - Buscar un cliente en particular
- Pedidos
 - Alta del Pedido
 - Funciones:
 - Alta de Pedido
 - Guardar en Archivo
 - Baja de Pedido
 - Funciones:
 - Buscar Pedido en el Archivo
 - Anular el Pedido
 - Guardar los cambios en el archivo
 - Modificación de Pedido
 - Funciones
 - Buscar Pedido
 - Modificar el Pedido
 - Guardar las modificaciones en el archivo
 - Listado de Pedidos
 - Listar Pedidos por Cliente.
 - Estadística de Pedidos: Ejemplo ¿Cuál es el mejor cliente?