**UNIVERSIDAD TECNOLOGICA NACIONAL**

COMISIÓN 3 - 2022

Magali Mouriño

LABORATORIO I

TRABAJO PRÁCTICO FINAL

|  |  |  |
| --- | --- | --- |
| ***UTN – FRMDP Mar del Plata***  ***TSP - Laboratorio 1***  ***Trabajo Práctico Final***    *Junio 2022* |  | ***Nota*** |

# Introducción

Con el propósito principal de integrar todo lo aprendido en la materia laboratorio 1, a los estudiantes que así lo deseen, propongan una temática particular elegida por los integrantes de su grupo.

El sistema deberá contemplar la administración de como mínimo, dos estructuras de datos, realizando la persistencia de estas en archivos binarios.

Si lo desean, podrán agregar una estructura de usuario (compartida en la propuesta de la cátedra), para efectuar el logueo al sistema.

La propuesta deberá ser formulada ante el equipo docente y aprobada por el mismo para la realización del trabajo.

# Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo 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**
* La aprobación del trabajo práctico estará sujeta a los puntajes considerados en la tabla debajo.
* Es obligatorio la presentación de este trabajo para aprobar la materia.

|  |  |  |
| --- | --- | --- |
| ***Apartado*** | ***Puntaje*** | ***Obtenido*** |
| Funciones para ABMCL de las estructuras de datos ● Alta, Baja, Modificación, Consulta y Listados.   * Validación en el ingreso de los Datos. * Busqueda de informacion en los archivos. | 35 |  |
| Funciones para manejo de archivos binarios ● Persistencia de datos en Archivos.   * Manejo de la información vista de administrador. * Manejo de la información vista de usuarios comunes. | 35 |  |
| Función main () y funciones de manejo de vistas del sistema Orden y prolijidad del código entregado. Explicación presencial del sistema   * Correcta modularización de las funciones. * Correcto uso de parámetros. * Prolijidad general del código. * Reutilización de las funciones. * Comentarios del código.     Se entregará una carpeta con la siguiente información:   * Impresión del presente enunciado * Diario de trabajo: Semana a semana qué actividades se desarrollaron y el responsable de cada una. * Matriz de soluciones: Que problema tuvieron y cómo lo resolvieron. * Diagrama de estructuras: Esquema de las estructuras utilizadas y sus relaciones. * Manual de usuario: Breve explicación de cómo funciona el sistema, pueden usar imágenes, videos, presentaciones, etc. | 30 |  |

Al realizar la entrega final, deberán tener en cuenta los siguientes puntos:

* Carpeta completa según lo requerido por la cátedra
* Código del sistema completo y compilado sin errores
* Explicación presencial del sistema

Además, tiene que cumplir con las siguientes funcionalidades:

* Organizar en Menús
* Registrar usuario y canciones.
* Persistir datos en el archivo

**Tabla de puntuación:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Obtenido** | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| **Nota** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

# PAUTAS GENERALES

**Entregar una narrativa de lo que se pretende solucionar o realizar con la propuesta del grupo**

--------------------------------------------------------------------------------------------------------------------------------

Actualmente para las empresas es necesario, por no decir imprescindible, contar con una tienda virtual que les permita hacer frente a la competencia, la cual está constantemente en busca de la implementación de nuevas estrategias que supongan una ventaja competitiva; Sino a su vez impactar en los mercados de distintas regiones y países, como así también brindarle al consumidor métodos de compra más simples y cómodos.

Lo que se pretende realizar con la propuesta elegida por el grupo es una tienda online de productos de indumentaria. La cual permita que el usuario pueda registrarse, iniciar sesión, realizar y ver sus pedidos, como también cancelarlos si eso quisiera. Asimismo, para el vendedor, agregar productos a la tienda y modificarlos. Como también poder ver todos los pedidos hechos por los usuarios, el precio total y su forma de pago.

--------------------------------------------------------------------------------------------------------------------------------

Detallar las estructuras, explicando sus campos:

|  |  |
| --- | --- |
| **Estructura 1**  typedef struct  {      char producto [20];      int precio;  int estado;    } Catalogo; | Guarda los nombres de los productos que la tienda tiene para ofrecer, así como también sus respectivos precios.  Estado es un campo booleano en el cual se guarda 1 si el producto está activo y 0 si el producto fue “eliminado” (concepto de “baja lógica”) por el administrador.  Se inicializa en 1, y se pasa a 0 solo si el Administrador decide eliminarlo. |
| **Estructura 2**  typedef struct  {      char producto [20];      int precio;      int color;      int talle;      int estado;      char nombreUsuario [20];    } Pedido; | Cada pedido guarda el nombre del producto que el Usuario desea comprar, siempre y cuando se encuentre en el Catálogo, como así también su precio.  Los colores están definidos por números, siendo estos los colores disponibles: //1- blanco // 2- negro // 3- beige // 4- azul // 5- marrón. Lo mismo sucede con los talles: | 1- S | 2- M | 3- L | 4- XL |.  Estará guardado al nombre de Usuario (el cual es único) de su respectivo comprador.  Estado es un campo booleano, se guarda 1 si el pedido esta activo y 0 si fue cancelado.  Todos los pedidos están guardados en un mismo archivo. |
| **Estructura 3**  typedef struct  {      char nombreYApellido[30];      char nombreUsuario[20];      int contrasenia;      Pago p;    } Usuario; | El nombre de Usuario es un string único, el cual es identificatorio de cada usuario. También guarda una contraseña compuesta de números (no más de 7).  Incluye otra estructura donde se almacena la manera en que el usuario pagará todos sus pedidos. |
| **Estructura 4**  typedef struct  {      int forma;      int precioTotal;      int desc;      int totalConDesc;    } Pago; | Acá es donde se almacenan los datos sobre la forma en que el Usuario pagara sus pedidos. Forma es un campo booleano, donde se guarda 0 si el usuario abonará en efectivo, y 1 si lo hará con tarjeta.  Dependiendo la forma de pagar y el precio total, se le realiza un descuento al cliente. Desc guarda el porcentaje de descuento que se realizó y en la última variable el total a pagar con este descuento aplicado. |

Matriz de soluciones:

**Semana 1:**

Problema:  El problema fue con la estructura catalogo y las opciones que queríamos brindar, como color y talle, pero nos dimos cuenta de que si poníamos esas variables en la misma estructura solo íbamos a poder cargar 1 talle y 1 color por producto.

Text

Description automatically generated(estructura disfuncional)

Solución:  En la opción catalogo solo cargamos producto y precio, y después dejamos que el usuario elija entre cinco colores y cuatro talles (generales, son los mismos para todos los productos cargados) que le mostramos por pantalla.

Text, letter

Description automatically generated

**Semana 2:**

Problema:  El problema fue decidir la manera de eliminar un producto del archivo catálogo. Se decidió entre sobrescribir el archivo o hacer una baja lógica. Finalmente se decidió optar por la ultima, agregando una variable de estado a la estructura Catalogo.

Problema:  El problema fue decidir cómo abordar las estructuras, si hacer que estén anidadas o relacionadas por algún tipo de variable, con el objetivo de manejar la información de una manera eficiente y prolija que supiéramos como manejar.

Solución: Las estructuras Usuario y Pedido están relacionadas por el nombre de usuario (único para cada usuario), de modo que el pedido se guarda con ese nombre. Por otro lado, dentro de la estructura Usuario se encuentra una variable de tipo estructura Pago.

Problema:  El problema fue con la función cancelar un pedido. Esto se solucionó, ya que se trataba de un error en el posicionamiento del cursor en la función.

**Semana 3:**

Timeline

Description automatically generatedProblema: la manera de actualizar la el total a pagar cuando se cancela un pedido, ya que los pedidos se almacenan en un archivo y la forma de pago en otro. Finalmente se realizaron las funciones de búsqueda, suma y modificación de archivos correspondientes para solucionar este problema.

