|  |  |  |
| --- | --- | --- |
| ***UTN – FRMDP Mar del Plata TUP - Laboratorio 2 Trabajo Práctico Final***  *Octubre 2020* | ***Integrantes del grupo*** | ***Nota*** |

## Introducción

Con el propósito principal de integrar contenidos aprendidos a lo largo de todo el año hemos planteado la siguiente problemática:

* El proyecto será una extensión del proyecto final del 1er cuatrimestre.
* Administrar un archivo de Clientes y consumes (Alta, Baja, Modificación, Consulta y Listados)
* Generar un árbol de clientes que contenga una lista de sus consumos telefónicos.

## Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo **(se formarán grupos de 3 alumnos)** y la integración de contenidos de otras asignaturas 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:

* Gestión de archivos binarios.
* Recursividad.
* Listas enlazadas simples o dobles
* Arboles binarios.
* Estructura de datos compuestos. (arreglo de listas, listas de listas, listas de árboles, etc.)
* 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

1. Se establece el desarrollo de un trabajo práctico final, brindando una fecha límite de entrega del mismo.
2. Si el sistema está codificado en su totalidad y funciona correctamente, se considerará aprobado con una nota mínima de 6.
3. 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.
4. En la fecha de recuperatorio deberá cumplir las pautas mínimas establecidas en el inciso 2.
5. Es obligatoria la presentación de este trabajo para aprobar la materia.

# Pautas Generales

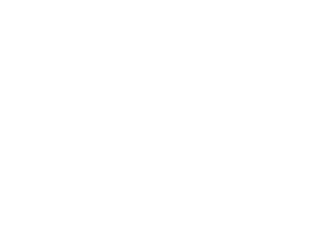
Se solicita que el equipo desarrolle un sistema que pueda lograr un manejo dinámico de la información y lograr persistencia de la información a partir de la gestión de archivos binarios.

El sistema deberá permitir la gestión de Clientes con sus respectivos consumos telefónicos. Constará de un archivo con por lo menos 50 clientes y otro con no menos de 30 consumos por cliente.

Sugerimos la utilización de la aplicación Trello [www.trello.com](http://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.

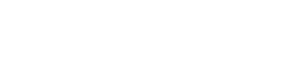
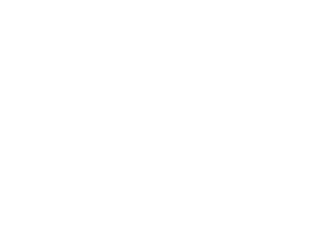
Para la persistencia de datos en los archivos (“clientes.dat” y “consumos.dat”) utilizaremos las siguientes estructuras de datos:





Cliente

1



Consumo

N

|  |  |
| --- | --- |
| typedef struct {  int id; /// campo único y autoincremental  int nroCliente;  char nombre[30];  char apellido[30];  int dni;  char email[30];  char domicilio[45];  int movil;  int baja; /// 0 si está activo - 1 si está eliminado  }stCliente; | typedef struct{  int id; /// campo único y autoincremental  int idCliente  int anio;  int mes; /// 1 a 12  int dia; /// 1 a … dependiendo del mes  int datosConsumidos; /// expresados en mb.  int baja; /// 0 si está activo - 1 si está eliminado  }stConsumo; |
| typedef struct \_nodoArbol{  stCliente dato;  nodoLista\* consumos;  struct \_nodoArbol\* izq;  struct \_nodoArbol\* der;  }nodoArbol; | typedef struct \_nodoLista{  stConsumo dato;  struct \_nodoLista\* sig;  }nodoLista; |

# El sistema debe tener la siguiente lógica de funcionamiento

El sistema deberá contar con un CRUD de clientes y otro de consumos por cliente, pudiendo listar los consumos ordenados por fecha. (Esto fue realizado el cuatrimestre anterior).

Se deberá recorrer el archivo de consumos y por cada consumo leído habrá que ir a buscar al archivo correspondiente el cliente al cual pertenece el consumo leído. Ahora se tendrán 2 variables, un consumo y un cliente. Con estos datos deberá utilizar una función que busque el cliente en el árbol, si el cliente no existe deberá crear un nodo árbol, agregar el dato cliente, inicializar la lista y agregar en ella el consumo.

Deberán codificar todas las funciones necesarias para administrar el Árbol y la gestión del Archivo binario, ordenadas en librerías y respetando la responsabilidad en cada una de ellas.

En el programa principal (main) se deberán crear todas las variables necesarias para el correcto funcionamiento de la aplicación y la función main deberá contar con un menú de opciones que permitan al usuario elegir la opción deseada. Deberá existir como opción la carga del árbol a partir de los datos de los archivos. Una vez cargado el árbol el usuario podrá optar por liquidar un período (mes) de consumos de un usuario en particular, toda ésta lógica deberá realizarse sobre el árbol de listas para luego

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

### Tabla de puntuación:

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