

## Estructuras de Datos y Algoritmos

### Práctico de máquina 1 - Año 2024

**Fecha de entrega: Jueves 29 de agosto de 2024 hasta las 8 hs.**

Se desea implementar una agenda de prestadores de servicios en la cual se guardaran los siguientes datos: **D.N.I.** del prestador que es único e identifica de forma unívoca toda la información asociada al mismo, nombre y apellido del prestador, servicios prestados, domicilio del prestador, correo electrónico y número de teléfono.

Para almacenar la información planteada se utilizará la siguiente estructura:

- Lista Secuencial Ordenada con examinación secuencial (LSO) con finalización dada por contenido (+infinito).

Para la solución completa al requerimiento, se deberá diseñar un programa en Lenguaje C en el cual se presente un menú que permita las siguientes operaciones: **ingreso de nuevos prestadores, eliminar prestadores existentes, modificar datos de un prestador, consultar la información completa asociada al prestador, Memorización Previa** y la opción **Mostrar Estructura**.

La opción **Mostrar Estructura** debe mostrar por pantalla todo el contenido de la estructura, listando la información completa de los elementos presentes en ella, en el orden en que están almacenados.

**Memorización Previa:** Esta opción deberá permitir cargar correctamente en la estructura el contenido del archivo de texto “*Prestadores.txt*” que contiene información de prestadores y será provisto por la cátedra (disponible en la *página web de la materia*).

El archivo de texto “*Prestadores.txt*” contiene un dato de la nupla en cada línea (renglón) del mismo. Un ejemplo de esa información se muestra a continuación:

21695670	/*DNI primer prestador*/
Jorge Icaza	/*nombre y apellido prestador*/
Albañil, Plomero	/*Servicios*/
4 DE JUNIO 100	/*domicilio prestador*/
jorge@mail.com	/*mail prestador*/
2664000000	/*teléfono prestador*/
.	
.	
.	
33565678	/*DNI n-ésimo prestador*/
Raul Perez	/*nombre y apellido prestador*/
Carpintero	/*Servicios*/
Rivadavia 1351	/*domicilio prestador*/
raul@mail.com	/*mail prestador*/
2664000001	/*teléfono prestador*/

#### Consideraciones a tener en cuenta:

- Se espera un máximo de 110 prestadores.
- La confirmación de la baja de un elemento deberá realizarse por pantalla.
- Los números de D.N.I. son enteros.
- El campo nombre y apellido puede contener un máximo de 80 caracteres en cada caso.
- El campo servicios puede contener un máximo de 120 caracteres.
- El domicilio puede contener un máximo de 80 caracteres.



- El correo electrónico puede contener un máximo de 50 caracteres.
- El número de teléfono puede contener un máximo de 30 caracteres.
- El pseudo-código genérico de los operadores puede verse en el apunte *Operaciones sobre Conjuntos*.
- El programa deberá desarrollarse en Lenguaje C, utilizando como entorno de desarrollo para tal fin **Code::Blocks** (disponible en [www.codeblocks.org](http://www.codeblocks.org)).

### Ejemplo de rutina para Lectura de Datos

El código que se presenta a continuación es una guía para programar una rutina que permita leer datos desde un archivo de texto. **Deberá adaptarlo a la situación planteada.**

```
int Lectura_datos ()
{
    .... //declaraciones
    FILE *fp;
    if (( fp = fopen ( "Prestadores.txt" , "r" ) )==NULL)
        return 0;
    else {
        while (!(feof(fp))){
            fscanf(fp, "%d",&aux.dni);
            fscanf(fp, "%[\n]",aux.nombre);
            fscanf(fp, "%[\n]",aux.servicios);
            fscanf(fp, "%[\n]",aux.domicilio);
            fscanf(fp, "%[\n",&aux.mail);
            fscanf(fp, "%[\n",&aux.telefono);
            //realizar el procesamiento del prestador
            ....
        }
        fclose(fp);
        return 1;
    }
}
```

### Importante:

- Los grupos deben ser de 2 integrantes.
- Los códigos fuente entregados que no compilen o estén incompletos respecto de la funcionalidad solicitada no serán revisados.
- La entrega del práctico se realiza por medio de la página de la materia y se debe enviar el archivo fuente del programa.
- El nombre del archivo deberá estar conformado de la siguiente manera: ***PnroP-GruponroG*** donde *nroP* es reemplazado por el número de práctico que se entrega y *nroG* por el número del grupo al que pertenece el programa. Por ejemplo, el nombre P1-Grupo22.c corresponde al práctico de máquina 1 enviado por el grupo 22. **Los programas cuyos nombres no respeten estas reglas de conformación no serán aceptados.**

