

## Estructuras de Datos y Algoritmos

### Práctico de máquina 2 - Año 2024

**Fecha de entrega: Jueves 26 de Septiembre 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 utilizarán las siguiente estructuras:

- Lista Secuencial Ordenada con examinación secuencial (LSO) con finalización dada por contenido (+infinito).
- Lista Secuencial Ordenada con búsqueda binaria (LSOBB).
- Árbol Binario de Búsqueda (ABB).

La aplicación deberá presentar un menú de opciones principal que permita seleccionar las siguientes opciones: **Comparación de estructuras** y **Mostrar Estructura** (una opción de muestra por cada estructura).

La opción **Mostrar Estructura** debe mostrar por pantalla el contenido de la estructura seleccionada. Para las listas mostrar los prestadores presentes y para el Árbol Binario de Búsqueda implementar un barrido preorden mostrando los datos de cada prestador y por cada nodo además mostrar el campo DNI del prestador de los nodos hijos (**implementar la solución iterativa**).

**Comparación de Estructuras:** Esta opción debe realizar y mostrar una comparación adecuada de lo que cuesta, en cada una de las estructuras, **realizar ingresos, eliminaciones y consultas de un prestador dado**. En el análisis debe considerar el peor escenario y el comportamiento esperado en cada caso. Una vez finalizada esta operación, deberá realizar un análisis de los resultados obtenidos y sacar una conclusión de los mismos; dicha conclusión deberá quedar plasmada al principio de su programa principal (donde se encuentra el main) como comentario (**incluir los resultados de la comparación**).

Para el cálculo de los costos de ingreso y eliminación: en las listas secuenciales la función de costo será la cantidad de corrimientos y cada corrimiento de elemento tiene costo **1 (uno)**. En el ABB se considerarán modificaciones de punteros (**0,5 por cada modificación**) y en el caso de utilizar la política de reemplazo se deberá sumar **un costo más (1)** por la copia de datos.

Para las consultas el costo se determinará en **cantidad de comparaciones por el campo DNI** para todas las estructuras, **un punto (1) por cada comparación**.

Para comparar las estructuras se utilizará una secuencia de operaciones detallada en el archivo de texto "*Operaciones-Prestadores.txt*" que contiene información de prestadores y será provisto por la cátedra (disponible en la *página web de la materia*). Esta secuencia de operaciones se deberá realizar sobre cada una de las estructuras, asegurando que las mismas **no contengan ningún dato inicialmente**. Una vez finalizada la secuencia de operaciones se mostrarán por pantalla los costos obtenidos para cada estructura. Además en las estructuras **deben quedar** los datos resultantes de efectuar las operaciones del archivo para ser alcanzados desde la opción mostrar de cada una de ellas.

El archivo de texto "*Operaciones-Prestadores.txt*" contiene una línea con el código de operación (1-Alta, 2-Baja y 3-Evocación) y a continuación los datos de la nupla necesarios para la operación en cada línea (renglón) del mismo. Un ejemplo de esa información se muestra a continuación:



1	/*código de la primera operación (Alta)*/
21695670	/*DNI prestador*/
Jorge Icaza	/*nombre y apellido prestador*/
Albañil, Plomero	/*Servicios*/
4 DE JUNIO 100	/*domicilio prestador*/
jorge@mail.com	/*mail prestador*/
2664000000	/*teléfono prestador*/
3	/*código de la segunda operación (Evocación)*/
26545670	/*DNI prestador*/
.	.
2	/*código de la n-ésima operación (Baja)*/
33565678	/*DNI 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.
- Las listas deberán estar ordenadas de menor a mayor respecto del DNI.
- Para la lista con búsqueda binaria (**LSOBB**) la consigna a utilizar será **trisección**, límite inferior inclusivo, límite superior exclusivo y segmento mas grande a la izquierda. Para ella no se utilizará ningún elemento ficticio para indicar fin de lista.
- La **confirmación de la baja** en la rutina de baja debe realizarse por código, es decir comparando toda la nupla (en todas las estructuras).
- La política de reemplazo en la baja del **ABB** cuando el nodo tiene dos hijos es el **mayor de los menores** y el reemplazo deberá realizarse con copia de datos.
- 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 ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**, esto significa por ejemplo que Juan Fernandez = JUAN FERNANDEZ = JUan fernandez.
- 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 Operaciones

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_Operaciones ()
{
    .... //declaraciones
    FILE *fp;
    if (( fp = fopen ( "Operaciones-Prestadores.txt" , 'r' )) ==NULL)
        return 0;
    else {
        while (!(feof(fp))){
            fscanf(fp, "%d", &codigoOperador);
            fscanf(fp, "%d", &aux.dni);

            if (codigoOperador==1||codigoOperador==2){
                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);

                //llamar al operador correspondiente (Alta o Baja)
                //en todas las estructuras

            } else if (codigoOperador==3){
                //llamar a Evocar en todas las estructuras
            } else{
                //error, código operación no reconocido
            }
            codigoOperador=0;
        }
        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.**

