Práctico de máquina 2 - Año 2025

Fecha de entrega: lunes 22 de septiembre de 2025 hasta las 8 hs.

En la materia *Programación VI* se desea mantener un registro del resultado final de los alumnos que se inscribieron a la misma. Para ello se debe desarrollar una aplicación en la que se operará con los siguientes datos: **código de alumno** que es único e identifica de forma unívoca toda la información asociada al mismo, *apellido y nombre del alumno*, *correo electrónico*, *nota*, y *condición final*.

Se espera que la aplicación resultante sea lo más eficiente posible, por lo que hay que decidir qué estructura será la más adecuada para el requerimiento planteado. Por tal razón se compararán tres posibles alternativas para almacenar la información planteada, las cuales serán:

- Lista Secuencial Ordenada con examinación secuencial (**LSO**).
- Lista Invertida con búsqueda por Trisección (**LIBT**).
- Árbol Binario de Búsqueda (**ABB**).

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: **Comparación de Estrucuturas** y **Mostrar Estructura** (una opción de muestra por cada estructura).

Mostrar Estructura: debe mostrar por pantalla el contenido de la estructura seleccionada listando los alumnos presentes en cada una. Para el caso del **ABB** se deberá implementar un barrido pre-orden mostrando los datos de cada alumno y por cada nodo además mostrar el campo **código de alumno** de los nodos hijos (el barrido puede implementarse de forma recursiva).

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 alumno dado**. En el análisis debe considerar el peor escenario y el comportamiento esperado en cada caso.

Para el cálculo de los costos de ingreso y eliminación se considerarán los cambios estructurales que provoca cada operacion, para esto se asume que: cada corrimiento de nupla tiene costo **1 (uno)**. Las modificaciones o corrimientos de punteros tienen costo **0,5**.
Para las consultas el costo se determinará en cantidad de **celdas consultadas** para realizar dicha operación en cada una de las estructuras, **un punto (1) por cada celda**.

Para comparar las estructuras se utilizará una secuencia de operaciones detallada en el archivo de texto *"Operaciones-Alumnos.txt"* que contiene información de alumnos 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 mostrará por pantalla los costos obtenidos para cada estructura. Además una vez terminada la comparación en las estructuras **deben quedar** los datos resultantes de efectuar las operaciones detalladas en el archivo para ser alcanzados desde la opción mostrar de cada una de ellas.

Con los resultados obtenidos deberá realizar un análisis 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**).

El archivo de texto *"Operaciones-Alumnos.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)*/
0659A25                 /*código primer alumno*/
Wirth, Niklaus          /*apellido y nombre alumno*/
0659A25@mailbox.edu.ar  /*mail alumno*/
10                      /*nota*/
Promociona              /*condición final*/
3                       /*código de la segunda operación (Evocación)*/
0659A25                 /*código alumno*/
                .
                .
                .
2                       /*código de la n-ésima operación (Baja)*/
0308A25                 /*código n-ésimo alumno*/
Markoff, John           /*apellido y nombre alumno*/
0308A25@mailbox.edu.ar  /*mail alumno*/
7                       /*nota*/
Regular                 /*condición final*/
```

Consideraciones a tener en cuenta:

- Se espera un máximo de 130 alumnos. En **ABB** pueden insertarse más ya que es una estructura dinámica.
- No se utilizarán elementos ficticios de ningún tipo en las estructuras.
- Para la lista invertida (**LIBT**) la consigna a utilizar será límite inferior inclusivo, límite superior inclusivo y segmento mas grande a la izquierda.
- 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.
- La rutina de baja será automática, sin confirmación por parte del usuario. Es decir, la confirmación del elemento debe realizarse por código, comparando toda la nupla (en todas las estructuras).
- El campo código puede contener un máximo de 7 caracteres alfanumérico.
- El campo apellido y nombre puede contener un máximo de 80 caracteres en cada caso.
- El correo electrónico puede contener un máximo de 23 caracteres.
- La nota es un entero entre 0 y 10 inclusive.
- La condición final puede contener un máximo de 10 caracteres y será alguna de las siguientes: Promociona, Regular, Libre y Ausente (para el ausente la nota será 0).
- El ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**, esto significa que al buscar un código de un alumno deberá ser reconocido independientemente de cómo se ingresen las letras del mismo (1234A25 = 1234a25).
- 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 a **Code::Blocks**.

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-Alumnos.txt'' , ''r'') )==NULL)
                return 0;
        else {
                while(!(feof(fp))){
                        fscanf(fp,''%d'', &codigoOperador);
                        fscanf(fp,'' %[^\n]'',aux.codigo);

                        if(codigoOperador==1||codigoOperador==2){
                                fscanf(fp,'' %[^\n]'',aux.apellido);
                                fscanf(fp,'' %[^\n]'',aux.mail);
                                fscanf(fp,''%d'',&aux.nota);
                                fscanf(fp,'' %[^\n]'',aux.condicion);

                //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 (*link a la página de entregas*) y se debe enviar el archivo fuente del programa.
- El nombre del archivo deberá estar conformado de la siguiente manera: ***P****nroP*-***Grupo****nroG* 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**. Ante cualquier inconveniente para enviar los prácticos por la página comunicarse con la cátedra.