

## Estructuras de Datos y Algoritmos

### Práctico de máquina 2 - Año 2024

**Fecha de entrega: Lunes 29 de abril de 2024 hasta las 8 hs.**

La empresa de paquetería “*El Revoleo*” necesita un sistema capaz de manejar la información de los envíos realizados. La información que se mantiene sobre los mismos es: código de envío, que es único e identifica de forma unívoca toda la información asociada al mismo, D.N.I. del receptor, nombre y apellido del receptor, domicilio del receptor, D.N.I. del remitente, nombre y apellido del remitente, fecha en que se realizó el envío y la fecha en que se recibió.

Para almacenar la información planteada se considera utilizar una de las siguientes estructura:

- Lista Secuencial Ordenada con búsqueda po Bisección (**LSOBB**).
- Lista Invertida con búsqueda por Trisección (**LIBT**).
- 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 listando los envíos presentes en cada una. Para el caso del **ABB** se deberá implementar un barrido in-orden mostrando los datos de cada envío y por cada nodo además mostrar el campo código de envío de los nodos hijos (puede implementarse recursivo).

**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 envío 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: 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 **celdas consultadas** para todas 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-Envios.txt*” que contiene información de envíos 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 del archivo para ser alcanzados desde la opción mostrar de cada una de ellas.

El archivo de texto “*Operaciones-Envios.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)*/
AD09912	/*código del envío*/
21695670	/*DNI receptor*/
Jorge Icaza	/*nombre y apellido receptor*/
4 DE JUNIO 100	/*domicilio receptor*/
28499429	/*DNI remitente*/
Jose Rondeau	/*nombre y apellido remitente*/
2020-03-03	/*fecha envío*/
2020-03-18	/*fecha recepción*/
3	/*código de la segunda operación (Evocación)*/
AF09912	/*código del envío*/
.	.
.	.
2	/*código de la n-ésima operación (Baja)*/
AA07040	/*código del envío*/
32982224	/*DNI receptor*/
Jorge Luis Borges	/*nombre y apellido receptor*/
SARMIENTO 1335	/*domicilio receptor*/
24363665	/*DNI remitente*/
Jose de San Martin	/*nombre y apellido remitente*/
2020-03-04	/*fecha envío*/
2020-03-19	/*fecha recepción*/

### Consideraciones a tener en cuenta:

- Se espera un máximo de 250 envíos.
- No se utilizarán elementos ficticios en ninguna de las listas.
- Las listas deberán estar ordenadas lexicográficamente de menor a mayor respecto del código de envío.
- Para la lista invertida (**LIBT**) la consigna a utilizar será límite inferior inclusivo, límite superior exclusivo y segmento mas grande a la izquierda.
- Para la lista con búsqueda binaria por bisección (**LSOBB**) la consigna a utilizar será límite inferior exclusivo, límite superior exclusivo, testigo a derecha 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 confirmación del elemento en la rutina de baja debe realizarse por código, comparando toda la nupla (en todas las estructuras).
- El código del envío es una secuencia de 7 caracteres alfanuméricos.
- Los números de DNI son enteros.
- El campo nombre y apellido puede contener un máximo de 80 caracteres en cada caso.
- El domicilio puede contener un máximo de 80 caracteres.
- Las fechas de envío y recepción son una cadena de caracteres con el formato (AAAA-MM-DD).
- El ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**, esto significa que al buscar un código de un envío deberá ser reconocido independientemente de cómo se ingresen las letras del mismo (aa12345 = Aa12345 = AA12345 = aA12345).
- 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-Envios.txt" , "r" )) ==NULL)
        return 0;
    else {
        while (!(feof(fp))){
            fscanf(fp,"%d",&codigoOperador);
            fscanf(fp,"%d",&aux.codigo);

            if (codigoOperador==1||codigoOperador==2){
                fscanf(fp,"%d",&aux.dni_rec);
                fscanf(fp,"%[\n]",&aux.nombre_rec);
                fscanf(fp,"%[\n]",&aux.dom_rec);
                fscanf(fp,"%d",&aux.dni_env);
                fscanf(fp,"%[\n]",&aux.nombre_env);
                fscanf(fp,"%[\n]",&aux.fecha_env);
                fscanf(fp,"%[\n]",&aux.fecha_rec);

                //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.**

