Estructuras de Datos y Algoritmos

Práctico de máquina 2 - Año 2022

Fecha de entrega: Martes 11 de octubre de 2022 hasta las 8 hs.

Una empresa de ventas con múltiples franquicias desea registrar el desempeño de sus vendedores manteniendo la siguiente información: el *número de documento* que identifica a cada vendedor, *Nombre y Apellido, teléfono, monto vendido, cantidad vendida* y *canal de venta*.

Se necesita diseñar una aplicación que permita resolver los requerimientos y para ello se cuenta con las siguientes estructuras de datos para almacenar la información mencionada:

a) Árbol Binario de Búsqueda (**ABB**).

b) Rebalse Abierto Cuadrático (**RAC**).

c) Rebalse Separado (**RS**).

La aplicación deberá presentar un menú de opciones principal que permita seleccionar la estructura con la que se desea trabajar y para cada una de ellas un nuevo menú que muestre las opciones para administrarla. Este menú debe presentar por pantalla las siguientes opciones: **ingreso de nuevos vendedores, eliminación de vendedores existentes** y **consulta de vendedores**. Además, debe contener las opciones **Mostrar Estructura** y **Memorización Previa**.

La opción **Memorización Previa** es una rutina que permite guardar en una estructura la información incluida en el archivo de texto *"Vendedores.txt"* provisto por la cátedra. Esta rutina debe leer desde el archivo la información correspondiente a un vendedor e insertarla en la estructura correspondiente.

La opción **Mostrar Estructura** debe mostrar por pantalla el contenido de la estructura seleccionada, listando la **estructura completa**. Para el **Rebalse Abierto Cuadrático** se deben mostrar las **M** posiciones de cada estructura, presentando la información completa de los elementos presentes en ella, y para las celdas que no están ocupadas, diferenciar claramente las celdas *libres* de las celdas *nunca usadas*. En el caso del **Rebalse Separado** debe mostrar para las M listas los elementos presentes en ellas o indicar que la lista está vacía, y para el **Árbol Binario de Búsqueda** implementar un barrido **preorden** mostrando los datos de cada vendedor y por cada nodo además mostrar el campo dni de los nodos hijos (puede implementarse recursivo).

Consideraciones a tener en cuenta:

- El número esperado de vendedores a almacenar es de 110.
- El número de documento es un entero no mayor a 99.999.999.
- El campo nombre y apellido puede contener un máximo de 50 caracteres.
- El teléfono puede contener un máximo de 15 caracteres.
- El monto es un valor real positivo.
- La cantidad es un número entero positivo.
- El canal de venta puede contener un máximo de 20 caracteres.
- El ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**.

- La confirmación del elemento en la rutina de baja debe realizarse por pantalla.
- La politica de reemplazo en la baja del **ABB** cuando el nodo tiene dos hijos es el **menor de los mayores**.
- Para el **RAC** se tendrá ρ= 0.77.
- Para el **RS** se tendrá ρ= 1.84.
- Se utilizará una ranura por balde en cada Rebalse.
- Se debe utilizar la siguiente función de hashing:

```
int hashing (int dni, int M) {
        char x[8];
        int longitud, i;
        int contador=0;
        sprintf(x,''%d'', dni);
        longitud=strlen(x);
        for (i=0; i< longitud; i++)
                contador+=((int)x[i]) * (i+1);

        return (contador %M );
}
```

- El programa deberá desarrollarse en Lenguaje C, utilizando como herramienta para tal fin **Code::Blocks** (disponible en www.codeblocks.org).

Nota Importante: 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: ***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**.

Ejemplo de rutina para Memorización Previa

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 Memorización_Previa()
{
        FILE *fp;
        if (( fp = fopen ( ''Vendedores.txt'' , ''r'') )==NULL)
                return 0;
        else {
                while(!(feof(fp))){
                        fscanf(fp,''%d'',&aux.dni);
                        fscanf(fp,'' %[^\n]'',aux.Nombre);
                        fscanf(fp,'' %[^\n]'',aux.telefono);
                        fscanf(fp,''%f'',&aux.Valor);
                        fscanf(fp,''%d'',&aux.Cant);
                        fscanf(fp,'' %[^\n]'',aux.tipoventa);
                        /* Donde aux tiene los campos adecuados que se
                        corresponden con la información guardada en el
                        archivo Vendedores.txt en la posición corriente */
                        .
                        .
                        .
                        /* Invocar los procedimientos que correspondan */
                        .
                        .
                        .
                        }
                fclose(fp);

                return 1;
        }

}
```