

## Estructuras de Datos y Algoritmos

### Práctico de máquina 3 - Año 2023

<b>Fecha de entrega: Jueves 19 de Octubre de 2023 hasta las 8 hs.</b>
---

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 estructuras:

- Rebalse Abierto Lineal (RAL).
- Rebalse Abierto Cuadrático (RAC).
- Rebalse Separado (RS).

Para decidir cuál sería la mejor para esto, se deberá desarrollar una aplicación que las compare. La aplicación deberá presentar un menú de opciones principal que permita seleccionar las siguientes opciones: **Comparación de estructuras** y **Mostrar Estructura**.

La opción **Mostrar Estructura** deberá tener una opción de muestra por cada estructura, y debe mostrar por pantalla el contenido de la estructura seleccionada. Para los **Rebalses Abiertos** se deben mostrar las **M** posiciones de la 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.

La opción **Comparación de Estructuras** debe realizar y mostrar una comparación adecuada de lo que cuesta, en cada una de las estructuras, **consultar un envío dado**. En el análisis debe considerar el peor escenario y el comportamiento esperado en cada caso (tanto para éxito como para fracaso). 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 la función de costo a utilizar es celdas consultadas para todas las estructuras **1 punto 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 300 envíos.
- La confirmación del elemento en la rutina de baja debe realizarse por código, comparando toda la nupla (en todas las estructuras).
- Para el **RAL** se tendrá  $\rho = 0.84$ .
- Para el **RAC** se tendrá  $\rho = 0.75$ .
- Para el **RS** se tendrá  $\rho = 1.45$ .
- Se utilizará una ranura por balde en cada Rebalse.
- No se utilizarán baldes en la cabecera del **RS**.
- Se debe utilizar la siguiente función de hashing:

```

int hashing (char* x, int M) {
    int longitud, i;
    int contador=0;
    longitud=strlen(x);
    for (i=0; i< longitud; i++)
        contador+=((int)x[i]) * (i+1);
    return (contador % M);
}

```

- 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 , "%s" , aux.nombre_rec );
                fscanf ( fp , "%s" , aux.dom_rec );
                fscanf ( fp , "%d" , &aux.dni_env );
                fscanf ( fp , "%s" , aux.nombre_env );
                fscanf ( fp , "%s" , aux.fecha_env );
                fscanf ( fp , "%s" , 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.**

