

Act 2.3 - Actividad Integral estructura de datos lineales (Evidencia Competencia)

El canal de Suez es un canal navegable que conecta el mar Mediterráneo con el mar Rojo a través de alrededor de 190 km, uniendo por un el lado del mar Mediterráneo desde el puerto Said hasta el golfo de Suez en el mar Rojo. Este canal es navegado entre 49 y 97 barcos diariamente. Se tiene un registro de los barcos que navegan por el canal con el siguiente formato:

<fecha> <hora> <punto-entrada> <UBI-Identificador único del buque>

En este segundo entregable, se realizó un código que pide el nombre de un archivo que contenga dichos registros, para después poder desplegar el número de tanto en mar rojo como en mediterráneo que han pasado por mes, esto con los primeros 3 dígitos de la UBI, por ejemplo, si en un mes 25 barcos del mar mediterráneo y 19 del mar rojo con cierta UBI pasan, se muestra lo siguiente:

Contador M-R 1AE:

Ene 20 25 19

Feb 20 1 0

Mar 20 0 0

Abr 20 0 0

May 20 0 0

Jun 20 0 0

Jul 20 0 0

Ago 20 0 0

Sep 20 0 0

Oct 20 0 0

Nov 20 0 0

Dic 20 0 0

Algoritmos de búsqueda y ordenamiento utilizados

Merge sort: este algoritmo es para ordenar los elementos que se le pasen, consiste en dividir en dos los datos para proceder a acomodarlos en cada mitad y al final, junta las dos mitades de manera ordenada.

Su complejidad temporal es de $O(n \log n)$ ya que consiste en una función auxiliar, en este caso *unir*, al multiplicar la complejidad de cada una de las partes, nos da de resultante $O(n \log n)$

Búsqueda secuencial: Este algoritmo se utiliza para buscar cierto elemento dentro de, en este caso una lista ligada, lo que realiza este método es, busca elemento por elemento de la lista el objetivo, pasando de uno por uno comparando.

Su complejidad temporal es de $O(n)$ en el peor de los casos ya que va elemento por elemento por lo largo de la lista, en el mejor de los casos presentará una complejidad de $O(1)$, si el elemento buscado es el primero de la lista.

En nuestro proyecto lo utilizamos para que agarre todos los elementos de la lista que su UBI coincida con la proporcionada con el usuario.

Casos de prueba:

Serie 1AE:

```
Ingrese 3 digitos de UBI: 1AE

Contador M-R 1AE:
Ene 20 26 19
Feb 20 0 0
Mar 20 0 0
Abr 20 0 0
May 20 0 0
Jun 20 0 0
Jul 20 0 0
Ago 20 0 0
Sep 20 0 0
Oct 20 0 0
Nov 20 0 0
Dic 20 0 0
```

En este caso se ve que los buques que tienen la clave UBI 1AE en sus primeros 3 dígitos, son en el mar mediterráneo 26, en el mar rojo 19.

Serie 1TL

```

Ingrese 3 digitos de UBI: 1TL
Contador M-R 1TL:
Ene 20  22 17
Feb 20   0 0
Mar 20   0 0
Abr 20   0 0
May 20   0 0
Jun 20   0 0
Jul 20   0 0
Ago 20   0 0
Sep 20   0 0
Oct 20   0 0
Nov 20   0 0
Dic 20   0 0
    
```

En este caso se ve que los buques que tienen la clave UBI 1TL en sus primeros 3 dígitos, son en el mar mediterráneo 22, en el mar rojo 17.

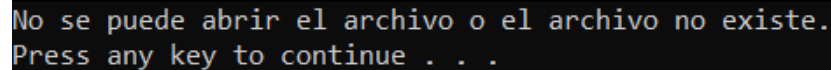
Serie inválida (000)

```

Ingrese 3 digitos de UBI: 000
Contador M-R 000:
Ene 20   0 0
Feb 20   0 0
Mar 20   0 0
Abr 20   0 0
May 20   0 0
Jun 20   0 0
Jul 20   0 0
Ago 20   0 0
Sep 20   0 0
Oct 20   0 0
Nov 20   0 0
Dic 20   0 0
    
```

En este caso se manda un numero de serie UBI que no existe, por lo que el indicador de barcos está vacío para cada mes.

Archivo mal escrito



```
No se puede abrir el archivo o el archivo no existe.  
Press any key to continue . . .
```

En este caso se puso el nombre del archivo incorrectamente, por lo que se manda un error indicando que el archivo no existe

Reflexión

Es importante el uso de distintos métodos de ordenamiento y búsqueda ya que hay algunos que se adaptan mejor al caso, por ejemplo, en esta actividad, se utilizó la búsqueda secuencial, ya que estamos utilizando listas ligadas, y en estas se tiene que acceder elemento por elemento

Por ejemplo, si tuviéramos un array de 100 números del 1 al 100 y pedimos que nos devuelva el index del número 100, en búsqueda secuencial tardará más que con búsqueda binaria, pero es la mas viable con listas ligadas.

Al igual que con los algoritmos de búsqueda, tenemos que encontrar un algoritmo de ordenamiento que satisfaga nuestras necesidades, en este caso se utilizó el merge sort, pero dependiendo de lo que tengamos que hacer y como tengamos que manipular los datos, podríamos utilizar distintos métodos.