



# **Actividad Evaluativa EPE 2**

## **PROGRAMACIÓN MODULAR Y VECTORES**

**Nombre: Felipe Rodríguez Candia**  
**Profesor: Luis Alvarado Barraza**  
**Mayo, 2023**

## Detalles de la actividad

Un nuevo cliente, una empresa de aduana de la Región de Valparaíso, necesita un sistema amigable y fácil de usar que les permita ordenar por cantidad de carga los distintos camiones que ingresan cada día. Se ingresa el vehículo con patente y luego la carga en toneladas.

Al final del día se necesita un informe que entregue en orden los vehículos que ingresaron y la carga que llevaban. El orden es por carga **DESCENDENTE**, primero el camión o camiones con más carga hacia camiones con la menor carga.

Cliente además asegura que los computadores que poseen no son de última generación y no están en posición de realizar una actualización a estos y les han funcionado perfectamente el tiempo que llevan con ellos.



## **¿Por qué elegirías un lenguaje estructurado como C para implementar una solución al requerimiento del Cliente?**

Si tomamos en cuenta los requerimientos y especificaciones técnicas de los equipos de la empresa de aduana de la Región de Valparaíso, podemos sugerir el uso del lenguaje de programación en C, ya que es una muy buena opción para computadores con pocos recursos como los que posee el cliente.



El lenguaje estructurado C permite un gran control sobre los recursos del sistema, logrando una alta eficiencia controlando grandes cantidades de datos, sin requerir grandes cantidades de memoria. Además, el lenguaje estructurado C es un lenguaje de programación muy eficiente y rápido, lo que lo hace ideal para aplicaciones en las que el rendimiento es crítico, como este caso, donde se espera procesar grandes cantidades de datos proveniente de los camiones que pasen por la aduana.

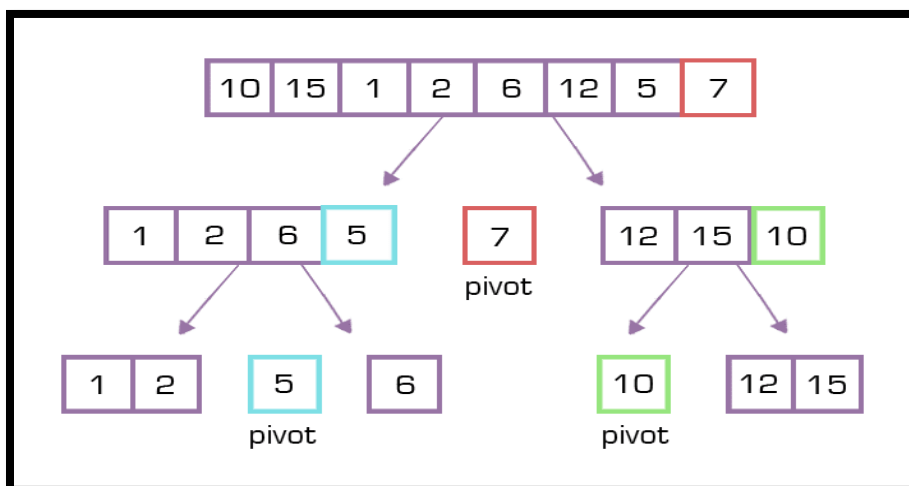
De la misma forma, tenemos que tomar en cuenta que los requerimientos de la empresa de aduana de la Región de Valparaíso son bajos en nivel de complejidad, de este modo podemos implementar un programa de forma estructurada o procedimentaria lo suficientemente fácil, funcional y eficiente. Implementando el uso de funciones que permitan el ordenamiento de los camiones y sus cargas, de este modo el lenguaje estructurado C resulta ser un lenguaje de programación muy eficiente y rápido, lo que lo hace ideal para aplicaciones en las que el rendimiento es crítico, como en este caso donde se espera procesar grandes cantidades de datos.

## ¿Qué tipo de ordenamiento ocuparías y por qué?

Para solucionar el requerimiento de la empresa de aduana de la Región de Valparaíso yo usaría el método de ordenamiento rápido, ya que este es muy eficiente a la hora de procesar grandes cantidades de datos. Lo que lo hace más rápido que muchos otros algoritmos de ordenamiento, especialmente en comparación con el algoritmo de burbuja y el algoritmo de selección. Además, el algoritmo de ordenamiento rápido es in situ, lo que significa que el algoritmo ordena los elementos en el mismo conjunto de datos sin necesidad de almacenar el conjunto de datos ordenado en una lista separada.

El algoritmo de ordenamiento rápido utiliza la técnica de "divide y vencerás". El conjunto de datos se divide en dos subconjuntos más pequeños, a menudo denominados subconjunto izquierdo y subconjunto derecho. A continuación, se ordena cada subconjunto recursivamente utilizando el mismo algoritmo. Finalmente, los subconjuntos ordenados se combinan en uno solo, lo que resulta en la lista ordenada completa, en este caso se ordenarían de forma descendiente para así ordenar los camiones de mayor a menor peso. El proceso de ordenamiento rápido se puede resumir en los siguientes pasos:

- Seleccionar un elemento del arreglo como "pivote". El pivote puede ser cualquier elemento del arreglo, aunque lo más común es seleccionar el primer o el último elemento.
- Dividir el arreglo en dos subarreglos: uno que contenga los elementos menores o iguales que el pivote y otro que contenga los elementos mayores que el pivote. Esto se hace mediante la comparación de cada elemento del arreglo con el pivote.
- Ordenar cada subarreglo de forma recursiva aplicando el mismo proceso de selección de pivote y división del arreglo hasta que todos los elementos estén ordenados.
- Concatenar los subarreglos ordenados en el orden correcto.



Un ejemplo de código de implementación del código de ordenamiento rápido para los requerimientos del cliente sería el siguiente:

```
void ordenamiento_Rapido_Descendente(struct Camion camiones[], int primero, int ultimo) {
    int i, j, medio;
    struct Camion pivote, temp;
    medio = (primero + ultimo) / 2;
    pivote = camiones[medio];
    i = primero;
    j = ultimo;
    do {
        while (camiones[i].carga > pivote.carga && i < ultimo) {
            i++;
        }
        while (camiones[j].carga < pivote.carga && j > primero) {
            j--;
        }
        if (i <= j) {
            temp = camiones[i];
            camiones[i] = camiones[j];
            camiones[j] = temp;
            i++;
            j--;
        }
    } while (i <= j);
    if (primero < j) {
        ordenamiento_Rapido_Descendente(camiones, primero, j);
    }
    if (ultimo > i) {
        ordenamiento_Rapido_Descendente(camiones, i, ultimo);
    }
}
```

De esta forma, al usar el programa para ingresar la patente y el peso de los camiones, el método de ordenamiento rápido los ordenará imprimiendo el resultado en forma decreciente de la siguiente manera:

```
Ingrese la patente del camion 1 (o ingrese fin para terminar): dft6578
Ingrese la carga en toneladas del camion 1: 1
Ingrese la patente del camion 2 (o ingrese fin para terminar): cre4584
Ingrese la carga en toneladas del camion 2: 5
Ingrese la patente del camion 3 (o ingrese fin para terminar): ted5478
Ingrese la carga en toneladas del camion 3: 3
Ingrese la patente del camion 4 (o ingrese fin para terminar): cde4587
Ingrese la carga en toneladas del camion 4: 7
Ingrese la patente del camion 5 (o ingrese fin para terminar): fin
Informe de camiones:
Patente      Carga (ton)
cde4587      7.00
cre4584      5.00
ted5478      3.00
dft6578      1.00

Process returned 0 (0x0)   execution time : 51.681 s
Press any key to continue.
```

El algoritmo de ordenamiento rápido es más eficiente en términos de tiempo y uso de memoria que los algoritmos de ordenamiento de burbuja y selección, y es especialmente útil para manejar grandes conjuntos de datos. Por último, el algoritmo de ordenamiento rápido también es fácil de implementar y es muy versátil, ya que se puede adaptar para trabajar con diferentes tipos de datos y en diferentes situaciones. Todo esto lo hace una excelente opción para ordenar grandes conjuntos de datos.

## ¿Recomendarías actualizar los equipos informáticos?, ¿por qué?

Si bien no considero que haya una respuesta correcta para esta pregunta, hay varios factores a considerar, por eso lo más importante es tomar en cuenta la opinión del cliente, ya que este indica que no están en posición de realizar una actualización a los equipos y además comenta que los equipos actuales les han funcionado perfectamente el tiempo que llevan con ellos.

Si bien, no se puede determinar con certeza las razones por la cual no están en posición de actualizar los equipos de la empresa, existen algunas posibles razones por las cuales una empresa podría optar por no actualizar sus equipos informáticos, las cuales pueden ser las siguientes:

- **Costo de actualización de los equipos:** La actualización de los equipos informáticos puede ser una inversión costosa. Si una empresa no tiene los recursos financieros para invertir en nuevos equipos, entonces la actualización podría no ser una opción.
- **Compatibilidad:** Algunos programas y sistemas operativos pueden no ser compatibles con equipos más nuevos. Si una empresa utiliza software antiguo que no es compatible con sistemas más modernos, entonces una actualización podría no ser una opción viable.
- **Tiempo:** La actualización de los equipos informáticos puede ser un proceso que requiere tiempo y puede interrumpir la productividad de la empresa. En una empresa como la de aduana de la Región de Valparaíso no se puede permitir un tiempo de inactividad prolongado, entonces la actualización no puede ser una opción viable.

Como se mencionó anteriormente, los equipos actuales de la empresa satisfacen adecuadamente las necesidades de la empresa. Si los empleados de la empresa no necesitan nuevas funciones o características que solo están disponibles en equipos más modernos, entonces la actualización podría no ser necesaria. No actualizar los equipos de la empresa puede ser una decisión racional si los costos y la compatibilidad lo justifican, y si los equipos existentes funcionan bien y cumplen con las necesidades de la empresa.