

1. Presentación Visual de la Simulación

Para representar de forma clara y didáctica el comportamiento del algoritmo QuickSort, hemos utilizado la herramienta interactiva **Visualgo**. Esta simulación permite observar paso a paso el proceso de ordenamiento, desde la selección del pivote, el particionamiento del arreglo, hasta la ubicación final de cada elemento en orden.

Cada paso se acompaña de una visualización con colores diferenciados que permiten identificar:

- El pivote (en amarillo).
- Elementos menores y mayores al pivote.
- Las operaciones de intercambio (swap) entre elementos.

Adicionalmente, la simulación muestra estadísticas como el número de comparaciones y swaps, lo que ayuda a entender el costo computacional del algoritmo. Esta representación facilita la comprensión del algoritmo y sirve como apoyo visual fundamental durante la exposición oral del grupo.

Además de su valor pedagógico, la simulación en Visualgo permite repetir los pasos, avanzar manualmente y ajustar la velocidad de ejecución, lo que resulta útil para analizar con detalle los procesos internos del algoritmo.

2. Análisis Detallado del Algoritmo QuickSort

QuickSort es un algoritmo de ordenamiento basado en la estrategia "divide y vencerás". Su funcionamiento general consta de los siguientes pasos:

1. Seleccionar un elemento como **pivote**.
2. Reorganizar el arreglo para que todos los elementos menores al pivote queden a su izquierda y los mayores a su derecha (particionamiento).
3. Aplicar el mismo proceso recursivamente a los subarreglos resultantes hasta que estén ordenados.

Un fragmento representativo del algoritmo en pseudocódigo es el siguiente:

```
for each (unsorted) partition
```

```
    set first element as pivot
```

```
    storeIndex = pivotIndex + 1
```

```
    for i = pivotIndex + 1 to rightmostIndex
```

```
        if ((a[i] < a[pivot]) or (equal but 50% lucky))
```

```
swap(i, storeIndex)

++storeIndex

swap(pivot, storeIndex - 1)
```

Este enfoque permite ordenar grandes volúmenes de datos de manera eficiente en la mayoría de los casos.

Ventajas de QuickSort:

- Alta eficiencia en promedio (complejidad $O(n \log n)$).
- Utiliza memoria en el mismo arreglo, sin requerir estructuras adicionales.
- Tiene un excelente rendimiento en la práctica para la mayoría de los conjuntos de datos.

Desventajas:

- Su peor caso es $O(n^2)$, especialmente si el pivote se elige de forma poco efectiva.
- No es estable, es decir, puede alterar el orden de elementos iguales.
- Puede presentar dificultades al ordenarse listas muy pequeñas o muy grandes sin ajustes adecuados.

3. Justificación del Uso de Visualgo

La elección de **Visualgo** como herramienta de simulación se basa en varias razones técnicas y pedagógicas:

Ventajas:

- Interfaz amigable y fácil de usar.
- Visualización clara de cada operación: comparaciones, intercambios, elección del pivote.
- Permite a los estudiantes comprender visualmente lo que está ocurriendo en el algoritmo en tiempo real.
- Funciona directamente desde el navegador, sin necesidad de instalar programas adicionales.
- Útil tanto para principiantes como para usuarios avanzados que buscan una comprensión más profunda.

Desventajas:

- Requiere conexión a internet para funcionar.

- Tiene opciones limitadas de personalización (por ejemplo, no permite modificar la estrategia de selección del pivote).
- No permite visualizar código fuente real, lo que puede limitar su utilidad para usuarios más técnicos.

Aun con estas limitaciones, Visualgo sigue siendo una herramienta muy valiosa para presentaciones educativas por su capacidad de simplificar conceptos complejos.

4. Conclusión

El uso de QuickSort junto con Visualgo permitió no solo entender el funcionamiento del algoritmo, sino también presentarlo de forma clara, visual y accesible. La simulación fue clave para cumplir con el criterio de representar de forma precisa el comportamiento del algoritmo.

Además, el análisis permitió identificar las fortalezas y debilidades de QuickSort, mientras que Visualgo reforzó nuestra exposición con elementos visuales de gran valor didáctico. Esta herramienta no solo facilitó la explicación paso a paso del algoritmo, sino que también fomentó la colaboración del grupo, ya que cada integrante pudo observar, explicar y analizar diferentes momentos del proceso.

En conclusión, la combinación de una sólida comprensión teórica con una representación visual interactiva constituye una estrategia efectiva para enseñar, aprender y exponer algoritmos complejos como QuickSort.