

TABLA COMPARATIVA: ALGORITMOS DE ORDENAMIENTO

Taller 5: Ordenación en Java

FECHA: 17/11/2025

Elaborado por:

- Cael Alejandro Soto Castillo.
- Ariel Ismael González Astudillo.

NOTA:

- Este documento resume, de manera cualitativa, el comportamiento observado en los tres algoritmos implementados: Inserción, Selección y Burbuja por medio de tablas comparativas.
- El análisis se basa en las trazas generadas para los datasets solicitados y nuestra opinión propia (antes de haber ejecutado los casos de prueba).

1. Ordenamiento por Inserción (Insertion Sort)

Cael Soto	Pienso que será rápido si el arreglo que nos dan ya está casi en orden, pues hará muy pocas comparaciones y casi ningún movimiento. Pero si nos dan el inverso a ello, significaría que habrá que desplazar muchos números, haciendo que se vuelva lento.
Ariel Ismael	Cuando los datos ya tienen algún grado de orden, el algoritmo de Inserción me parece el más eficaz y práctico, porque saca buen provecho de esa ventaja y evita trabajo que no es necesario.
En el programa	- Realiza recorridos cortos cuando los elementos están casi ordenados. - La mayoría de movimientos ocurren cuando un número debe desplazarse hacia la izquierda, generando varios "shifts". - Tiende a ser el más eficiente cuando la entrada ya está parcialmente ordenada. - En los conjuntos analizados, el número total de desplazamientos fue menor comparado con Selección y Burbuja.

2. Ordenamiento por Selección (Selection Sort)

Cael Soto	Lo que anticipo de Selección es que es muy predecible, no importa si nuestro arreglo imaginario está desordenado/ordenado, siempre va a tener que recorrer y comparar cada elemento para encontrar el mínimo en cada paso. Entonces, el número total de comparaciones siempre será el mismo, y será alto.
Ariel Ismael	A pesar de ser fácil de comprender y mostrar claramente la idea de seleccionar el mínimo en cada paso, resulta ser el más inflexible: realiza las mismas comparaciones constantemente, aun cuando el arreglo ya está ordenado, lo que disminuye su eficacia.

En el programa	<ul style="list-style-type: none"> - Recorre siempre toda la lista para encontrar el mínimo, sin importar si el arreglo ya está ordenado o no. - Genera pocos intercambios (generalmente uno por iteración), pero realiza muchas comparaciones. - En entradas desordenadas, mantiene un comportamiento estable, pero no es eficiente para vectores grandes.
----------------	--

3. Ordenamiento de Burbuja (Bubble Sort)

Cael Soto	<p>Anticipo que, si el arreglo está ordenado, será el más rápido de todos porque usaremos el "corte temprano" que lo detendrá de inmediato. Además, se sabe que va ordenando en base al mayor y siempre lo mandará al final constantemente, sin embargo, si nos dan un arreglo desordenado como [3, 2, 1], será un desastre. Pues tendrá que hacer el máximo de comparaciones y muchos intercambios en cada pasada para llevar los números a su lugar.</p>
Ariel Ismael	<p>Por otro lado, Bubble Sort, a pesar de ser considerado uno de los más lentos, gana puntos gracias al corte temprano, que lo vuelve competitivo en listas que casi no necesitan cambios.</p>
En el programa	<ul style="list-style-type: none"> - Su rendimiento es muy variable: empeora con listas invertidas y mejora cuando detecta orden temprano. - El uso de "corte temprano" permitió detener el algoritmo cuando ya no existían intercambios. - En entradas casi ordenadas, fue el más rápido en detenerse.

4. Conclusiones Generales

- Burbuja con corte temprano se adapta muy bien a distintos tipos de arreglos. Cuando no hay más cambios que hacer, detiene el proceso, lo que ahorra comparaciones y movimientos innecesarios, sobre todo en arreglos casi ordenados o con duplicados.
- Inserción funciona muy bien en arreglos pequeños o parcialmente ordenados, porque solo mueve los elementos que necesita. Sin embargo, en arreglos totalmente invertidos requiere más pasos y desplazamientos.
- Selección mantiene siempre el mismo número de comparaciones y solo hace intercambios cuando encuentra el elemento más pequeño. Esto lo hace predecible y constante, aunque menos flexible si los datos ya están ordenados o tienen valores repetidos.
- En arreglos con valores duplicados o todos iguales, los tres algoritmos logran ordenar correctamente. Burbuja e Inserción aprovechan para hacer menos movimientos, mientras que Selección sigue su patrón regular.
- En general, para datos pequeños o casi ordenados, Burbuja con corte temprano e Inserción muestran mejor desempeño, mientras que Selección es útil cuando se busca un comportamiento estable y predecible, sin depender del orden inicial de los datos.