**Ejercicio Grupo Análisis de tiempo y espacio en algoritmos de ordenamiento**

Integrantes del grupo:

Maria Stefanie Bermudez 2220231016

Juan David Fandiño 2220221087

Miguel Angel Avila 2220222016

Juan Felipe Medina 2220221080

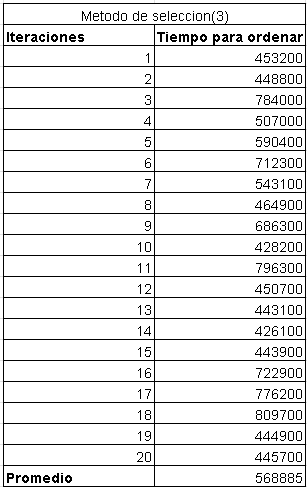
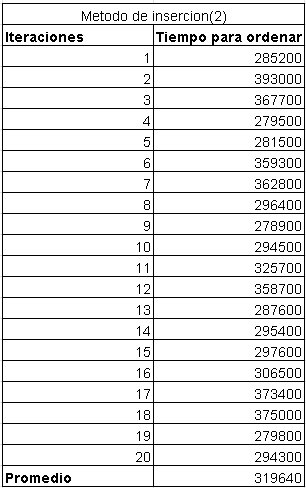
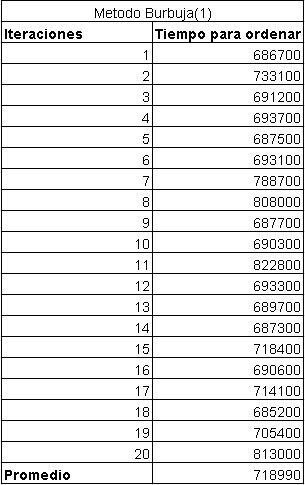
Mildred Sofia Lozano 2220232038

Calculate the execution time of three (3) sorting algorithms in any programming language. To do this, use static arrays on a topic of your choice. Additionally, apply time measurement methods between lines of code. Perform at least 20 time measurements and provide the following:

**a) The measurement method and programming language used**

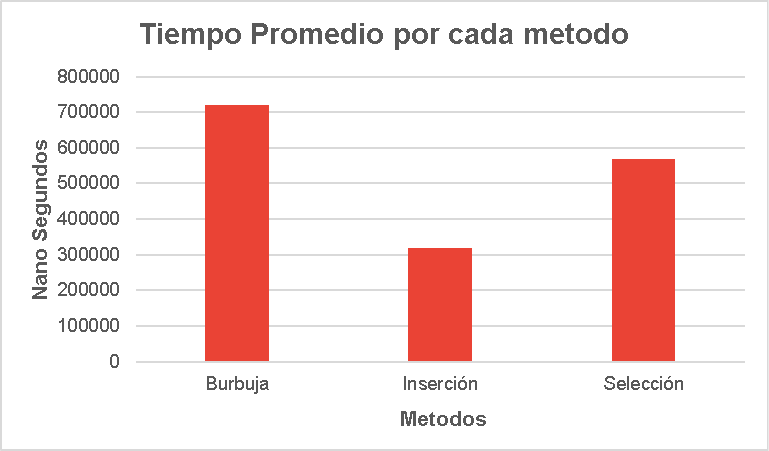
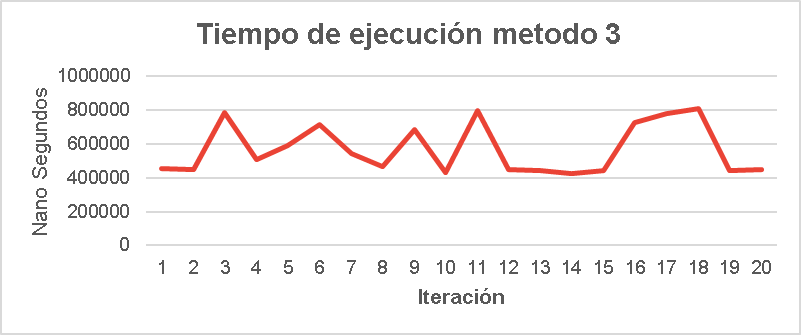
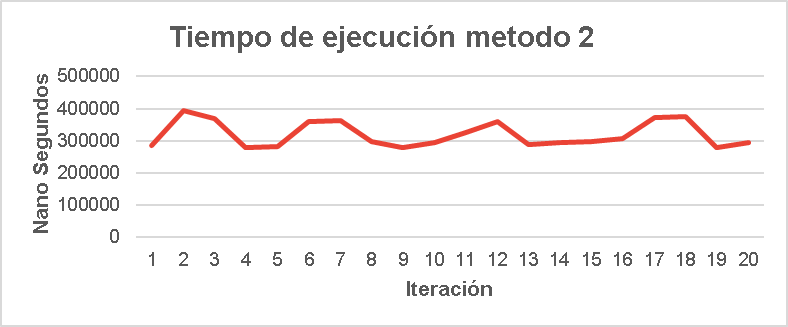
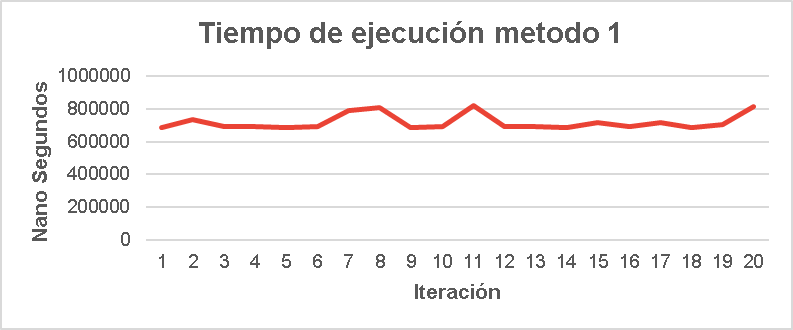
* Método de Medición: System.nanoTime() (medición de tiempo en nanosegundos).
* Métodos de programación: Ordenamiento de burbuja, ordenamiento de inserción y ordenamiento de selección.
* Lenguaje de Programación: Java.

**b) Results table**





**c) Results graph**



**d) Results analysis**

El Método de Inserción es la mejor opción para el cliente, ya que demostró ser el más rápido en los tests realizados, con un tiempo de ejecución de 319,000 ns, superando a Selección (516,000 ns) y Burbuja (718,000 ns). Este algoritmo es especialmente eficiente cuando el arreglo está parcialmente ordenado o es pequeño, lo que es común en aplicaciones prácticas. Su eficiencia no solo reduce el tiempo de procesamiento, sino que también optimiza el uso de recursos, permitiendo ahorrar en personal y costos operativos.

**e) Time and space complexity**

Metodo Burbuja:

* Complejidad del tiempo: o(n^2)
* Complejidad del espacio:o(n)

Metodo Insercion:

* Complejidad del tiempo: o(n^2)
* Complejidad del espacio:o(n)

Método Selección:

* Complejidad del tiempo: o(n^2)
* Complejidad del espacio:o(n)

**f) Compare items c and e**

Aunque los tres algoritmos de ordenamiento (Burbuja, Inserción y Selección) tienen una complejidad temporal de O(n^2) En el peor caso, el Método de Inserción es el más eficiente en la práctica. Esto se debe a que su implementación es más compacta y aprovecha mejor casos como arreglos parcialmente ordenados, donde su complejidad puede reducirse a O(n).

En los tests, el Método de Inserción fue el más rápido con 319,000 ns, superando a Selección (516,000 ns) y Burbuja (718,000 ns). Esto demuestra que, aunque todos comparten la misma complejidad teórica, el Método de Inserción es superior en la práctica debido a su optimización y adaptabilidad.