

**LABORATORIO # 3 – ANÁLISIS DE ALGORITMOS**  
**ALGORITMOS DE ORDENAMIENTO**

Para esta actividad práctica, el propósito es “analizar y evaluar” la complejidad de algunos algoritmos de ordenamiento clásicos. El análisis se hará desde el punto de vista de la Complejidad Temporal y Espacial. Se utilizarán cuatro algoritmos de ordenamiento: Burbuja, Inserción, Selección y QuickSort, MergeSort, RadixSort (estos dos últimos solo aplicará para grupos de cinco integrantes).

Para llevar a cabo el desarrollo de la actividad, tenga en cuenta los conceptos explicados por el docente en la sesión de clases presencial anterior: comparaciones, intercambios, ordenamiento estable, ordenamiento inestable, inserción.

Se utilizarán cuatro lenguajes de programación (si el grupo es de cinco integrantes, serán cinco lenguajes de programación): java, Python, lenguaje C, Go, JavaScript (como quinto lenguaje).

Para cualquiera de los algoritmos de ordenamiento (recuerde que deben ser versiones iterativas) se tendrán que generar números aleatorios antes de iniciar el proceso de ordenamiento (para cualquier tamaño de entrada, los números aleatorios deberán estar en el rango de 300.000 y 302.500 inclusive). Para cada algoritmo de ordenamiento vs cada lenguaje de programación, deberá diligenciar las siguientes tablas:

Tabla 1 - Algoritmo de XXXXXXXXXXXX (Conceptos)<sup>1</sup>

Tamaño de la entrada	#comparaciones	#intercambios	Estable / Inestable	Inserción (SI/NO)
n = 1.000				
n = 1.500				
n = 2.000				
n = 2.500				
n = 3.000				

Tabla 2 - Complejidad Temporal - Algoritmo de XXXXXXXXXXXX<sup>1</sup>

Tamaño de la entrada	Java	Python	Lenguaje C	Go
n = 1.000				
n = 1.500				
n = 2.000				
n = 2.500				
n = 3.000				

Adicionalmente, para cada método de ordenamiento deberá indicar en qué momento ocurre el peor caso, mejor caso, cuál es su orden de complejidad temporal y complejidad espacial. Para este último caso, el grupo deberá decidir de qué manera obtendrá esta

información. Podría consultar fuentes de datos fiables y/o utilizar una herramienta de monitoreo de aplicaciones.<sup>1</sup>

### ¿Qué se debe entregar?

Un documento en formato WORD que contenga:

- Las tablas diligenciadas (y cualquier otra tabla que el grupo considere necesaria).
- Las gráficas solicitadas acorde a los datos registrados en la tabla, para el análisis de la Complejidad Temporal. Recuerde que las gráficas, deben ser realizadas utilizando una escala logarítmica, tal como fue explicada su realización en la clase anterior.
- Análisis de la Complejidad Espacial. Si el grupo lo considera necesario para verificar con las consultas realizadas por otras fuentes, podría generar gráficas o emitir sus propias conclusiones.
- Presentar la salida del algoritmo al final del proceso de ordenamiento, en donde se pueda evidenciar los datos de la tabla #1.
- Exponga en el documento sus conclusiones en términos de complejidad algorítmica, consumos de recursos, tiempos, etc para cada escenario analizado.
- Los códigos fuentes de cada lenguaje de programación utilizado.

**IMPORTANTE:** Lo ideal es ejecutar cada algoritmo en un entorno similar para cada uno de los casos de ordenamiento. El ordenamiento deberá hacerse de mayor a menor para cualquier caso.

### ¿Cómo debe realizarse la entrega?

Tanto el documento en formato Word como los códigos fuentes deberá colocarlos en una carpeta comprimida. Un integrante del grupo deberá realizar la entrega mediante el aula virtual. La carpeta puede estar comprimida en formato ZIP o RAR. Recuerde **evitar nombres extremadamente largos para el documento en formato Word**.

**Fecha/hora límite de entrega:** domingo 08 de septiembre de 2024 hasta las 23:59 horas.

---

<sup>1</sup> Para cada método de ordenamiento, se debe diligenciar la tabla #2. Para la tabla #1, el grupo deberá seleccionar el lenguaje de programación y luego diligenciar la tabla por cada método de ordenamiento.