

Objetivo

U1.2. Aplicar conceptos de análisis de complejidad de algoritmos para describir y comparar formalmente (utilizando elementos matemáticos como funciones y notación asintótica) diferentes estrategias de solución al mismo problema.

U1.3. Evaluar los resultados de un proceso de análisis computacional e identificar los elementos relevantes que soportan la elección e implementación de una solución en particular.

U1.4. Implementar pruebas unitarias automáticas utilizando el nuevo lenguaje de programación.

U1.6. Llevar a cabo experimentos relacionados con el desarrollo de programas.

Importante: antes de iniciar la lectura del enunciado y el desarrollo del taller, cada uno de los integrantes del grupo debe registrarse e instalar esta herramienta para la medición de tiempo: <https://wakatime.com>. El registro con la herramienta debe ser llevado a cabo la cuenta de GitHub y una vez se autentique una persona debe crear un leaderboard y agregar a sus compañeros de grupo. Luego cada uno debe instalar el plugin correspondiente en los siguientes tres programas: Google Chrome, Visual Studio .NET y eclipse (si es que también va a programar en Java). Los documentos de texto no los desarrolle utilizando MS Word sino Google Docs, y para los diagramas utilice Visual Paradigm en línea (<https://online.visual-paradigm.com>) o draw.io.

Enunciado

Teniendo en cuenta que “Un experimento es un cambio en las condiciones de operación de un sistema o PROCESO, que se hace con el objetivo de medir el efecto del cambio sobre una o varias propiedades del producto o resultado”¹, usted llevará a cabo un experimento sobre el ORDENAMIENTO en un arreglo de valores. Dicho lo anterior, queda claro cuál es el proceso sobre el cual se hará la experimentación. Ahora usted debe decidir cuales son los cambios en las condiciones que se hará sobre la operación de dicho proceso.

Esos cambios sobre la operación del proceso se definen a través de los FACTORES. Factores cuyo cambio en sus valores pueden llegar a afectar el proceso anterior podrían ser:

- Algoritmo de Ordenamiento (Burbuja, Selection, QuickSort, HeapSort, etc)
- Tamaño del arreglo (10^1 , 10^2 , 10^3 , 10^4 , etc)
- Estado de los valores en el arreglo (en orden aleatorio, ordenado ascendente, ordenado descendente)
- Lenguaje de programación (Java, C#, Python, etc)
- RAM del computador donde se ejecuta el algoritmo (2GB, 4GB, 8GB, 16GB, etc)
- Procesador del computador donde se ejecuta el algoritmo
- Sistema Operativo
- Cantidad de procesos que se están ejecutando en el computador mientras se ejecuta el algoritmo
- Nivel de fragmentación del disco duro del computador donde se ejecuta el algoritmo
- Tamaño del registro del procesador (8bits, 16bits, 32bits, 64bits)

¹ Gutierrez, Humberto. De la Vara, Roman. Análisis y Diseño de Experimentos. Segunda Edición. Editorial McGraw-Hill. 2008.

Únicamente es obligatorio el primero de los factores, todos los demás son opcionales teniendo en cuenta que debe usted elegir un total de mínimo 3 factores controlables y un máximo de 5.

La cantidad de niveles del primer factor controlable (Algoritmo de Ordenamiento) deben ser dos (2), por lo cual debe elegir dos (2) algoritmos de ordenamiento diferentes que cumplan con que la complejidad temporal asintótica sea la misma (es decir, la función que describe el comportamiento del algoritmo tiempo vs entrada es asintóticamente igual).

Usted debe definir para el experimento: cuál es la unidad experimental, cuales son los factores controlables, no controlables y los estudiados. Igualmente debe definir los niveles de dichos factores y todos los tratamientos y el número de repeticiones. También debe definir la variable o variables de respuesta.

Debe llevar a cabo las siguientes etapas del Diseño de Experimentos:

- Planeación y Realización
- Análisis
- Interpretación
- Control y conclusiones finales

Adicionalmente, usted debe registrar en el informe el proceso de análisis de complejidad temporal y espacial de sus algoritmos y compararlos con los resultados obtenidos en el experimento.

El desarrollo de sus algoritmos debe incluir el diseño y construcción de pruebas unitarias automáticas. Las pruebas deben ser sobre entradas pequeñas fijas y entradas grandes generadas aleatoriamente.

Entregables

El enlace al repositorio en GitHub donde se encuentre su proyecto completo el cual debe incluir:

1. Diseño del diagrama de clases, diagrama de objetos y diseño de pruebas.
2. Código fuente en un proyecto de Visual Studio en C#.
3. Informe de: el diseño e implementación del experimento llevado a cabo y el análisis de complejidad temporal y espacial de los algoritmos seleccionados.
4. Datos obtenidos como resultado de llevar a cabo los tratamientos en todas sus repeticiones, en formato de hoja electrónica.
5. Informe de tiempo

Condiciones

- El taller deberá ser desarrollado por grupos de máximo 4 personas.
- Todos deben estar involucrados en el desarrollo del programa en C# (sólo así aprenderán a programar en este lenguaje). Esto implica que habrá preguntas individuales de parte de los profesores sobre este aspecto en la presentación.
- La evaluación de la actividad se hará utilizando los criterios de calificación indicados en la pestaña de notas del taller 4.

- Ya que su grupo debe experimentar sobre un par de algoritmos diferentes a todos los demás grupos de Proyecto Integrador I, debe registrar sus algoritmos en el [Formulario Experimento Ordenamiento](#), habiendo confirmado previamente que sus dos algoritmos no han sido elegidos ya por otro grupo en el [Listado Público de Algoritmos Elegidos](#). Tenga en cuenta que otro grupo puede elegir uno de sus dos algoritmos, lo que no es posible es que elija los mismos dos que su grupo.