



## ANÁLISIS DE ALGORITMOS – CICLO 02/2024

### DESCRIPCIÓN DE TALLER 2

#### Objetivo:

- Poner en práctica los conceptos y procedimientos para el análisis formal del orden de magnitud del tiempo de ejecución de un código recursivo, así como también las técnicas de resolución de las Recurrencias que los representan.

#### ETAPA I:

Los estudiantes serán organizados por los docentes en equipos 3 personas de forma aleatoria. Esta aleatoriedad puede permitir que hagan equipo personas de distintas secciones de la materia.

Los equipos estarán colocados en el Moodle en el mismo momento que se publique esta descripción.

#### ETAPA II:

Se deberá investigar sobre la estructura de Datos conocida como Montículo (Heap), sus características, y sus variantes.

Con esto, deberá elaborar un infográfico que explique todo el material investigado. Este infográfico deberá ser entregado como una imagen JPG o PNG.

#### ETAPA III:

La problemática a resolver es la siguiente:

*El almacén Salem, gracias a la ayuda que le brindaron en la ocasión anterior, aumentó mucho su eficiencia y eso le permitió crecer rápidamente, y ahora se ha vuelto una empresa de alto prestigio y buena fama.*

*Debido a este rápido crecimiento se está preparando para distribuir los bonos de fin de año a sus empleados. Para hacerlo de manera efectiva, el departamento de Recursos Humanos necesita ordenar los salarios de los empleados en **orden descendente**, de modo que los porcentajes de los bonos puedan calcularse desde los que ganan más hacia los que ganan menos; además, es importante tomar en cuenta que, con la expansión de la empresa, la nómina ha crecido considerablemente y ahora contiene miles de empleados con salarios variados.*

*Recursos Humanos ha encargado a tu equipo la tarea de ordenar esta enorme lista de salarios. Dado el tamaño del conjunto de datos, el proceso de ordenamiento debe ser eficiente y utilizar la menor cantidad de memoria posible, ya que los recursos de la empresa están siendo utilizados al máximo en otros departamentos. Se necesita un*

*método de ordenamiento rápido y confiable para garantizar que el plan de distribución se prepare a tiempo.*

La problemática debe ser resuelta con un algoritmo que utilice Montículos (Heaps) como su componente principal.

La problemática debe ser resuelta mediante un software que deberá ser desarrollado utilizando exclusivamente uno de los siguientes lenguajes de programación (a elección de cada equipo de trabajo):

- C++
- Java
- Python
- Kotlin

El software debe funcionar exclusivamente a nivel de consola, no se requiere ningún tipo de interfaz gráfica.

El software desarrollado debe ser capaz de ejecutarse y de producir resultados, de lo contrario, el procedimiento de las siguientes etapas recibe automáticamente una nota de cero. Para esto, se recomienda contar con un archivo que contenga los datos, y que su programa lo lea en un inicio; adicionalmente, ya que se trata de visualizar el trabajo con cantidades de datos grande, el mínimo de datos a procesar en su programa debe ser 1000.

**OJO:** A partir de este taller, sí se requerirá que toda parte del código sea eficiente, no solamente el algoritmo que resuelve la problemática.

#### **ETAPA IV:**

Una vez se tenga el software desarrollado se deberá hacer un análisis formal de cada línea de código producido, identificando las partes del código que sean recursivas y las que no.

Toda la porción del código que no sea recursivo deberá ser analizado formalmente paso a paso con su el procedimiento completo hasta determinar su orden de magnitud.

Luego, en base a la parte del código que sea recursiva, se deberá definir la Recurrencia que representa el código completo. En esta parte, se deberá explicar a detalle el razonamiento empleado para definir cada pieza de la Recurrencia definida.

Como siguiente paso, se deberá resolver la Recurrencia **utilizando el Teorema Maestro** (a menos que no sea posible, en cuyo caso utilicen el método del árbol), y se deberá hacer todo el procedimiento hasta obtener el orden de magnitud del código en su totalidad.

Por último, se deberá escribir un pequeño ensayo, de no más de 3 párrafos, con una reflexión del equipo acerca de los resultados obtenidos.

Todo lo anterior deberá ser colocado y escrito en un documento reporte construir de forma ordenada y legible. El documento deberá tener una portada apropiada y bien elaborada.

**Requerimientos de entrega:**

- El equipo deberá colocar todos los productos en un repositorio privado de Github que deberá ser compartido con ambos docentes de la materia.
- En Moodle se colocará una actividad donde deberán colocar además el enlace a este repositorio.
- La entrega deberá hacerse a más tardar el día sábado 12 de octubre a las 11:59pm.
- El esquema de carpetas en el repositorio deberá ser legible y adecuadamente estructurado de forma que sea fácil encontrar cada uno de los elementos de entrega solicitados:
  - Infografía de la parte II.
  - Software que resuelve la problemática de la ETAPA III.
  - Documento reporte de la ETAPA IV.
- Si se detectan códigos iguales entre dos equipos, por la razón que sea, implicará automáticamente una nota igual a cero.
- Si se detectan documentos iguales entre dos equipos, por la razón que sea, implicará automáticamente una nota igual a cero.
- Si se detecta uso de ChatGPT (o una herramienta similar), por la razón que sea, implicará automáticamente una nota igual a cero.
- Si se entrega la tarea más allá de las 11:59pm del 12 de octubre, por la razón que sea, implicará automáticamente una nota igual a cero.
- Si los códigos entregados no pueden ejecutarse, implicará automáticamente una nota igual a cero en la ETAPA IV.