



Objetivo:

- Poner en práctica los conceptos y procedimientos para el análisis formal del orden de magnitud del tiempo de ejecución de un código que resuelve un problema concreto.

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:

La problemática a resolver es la siguiente:

El almacén Salem necesita un sistema que le permita organizar todos los productos con los que cuenta en sus instalaciones, y para cumplir este propósito ha contratado un equipo de desarrolladores (ustedes estudiantes) que les puedan generar una solución.

El almacén maneja para cada producto dos datos: su nombre y su cantidad en stock; pero como hasta la fecha se han manejado los datos en cuadernos, se lleva un solo desorden, apuntando las cosas como van llegando, por lo que el gerente ha expresado que desea que el sistema a desarrollar le pueda proveer los datos de una forma donde le sea fácil ubicar cualquier producto de su interés.

El sistema debe identificar todos los productos que necesiten reabastecimiento, para que el gerente tenga en mente llamar a los respectivos proveedores. Se considera que un producto está en esta situación cuando quedan en bodega menos de 10 unidades.

Se busca que ustedes desarrollen un algoritmo que facilite la obtención de los n productos que tengan el menor stock, es decir, deben desarrollar una función (sin utilizar ningún tipo de librería estándar o externa) que devuelva un listado de los n productos con menor stock.

Por último, se requiere una manera de actualizar el stock de los productos.

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.

ETAPA III:

Una vez se tenga el software desarrollado se deberá hacer un análisis formal de cada línea de código producida, y se deberá generar el procedimiento completo de este análisis hasta llegar a la respuesta final que indicará el orden de magnitud del código escrito.

Ojo: No se requiere para este taller que el resultado final sea eficiente, en este punto solo interesa que sean capaces de analizar su programa y dar una respuesta de orden de magnitud.

Por lo tanto, se deberá escribir un documento reporte que contenga, de forma ordenada y legible, el procedimiento completo de análisis del código, junto con los resultados y su conclusión sobre estos resultados. El documento deberá tener una portada apropiada y bien elaborada.

Requerimientos de entrega:

- La entrega del proyecto será a través de Github Classroom. El enlace correspondiente será colocado en el Moodle de la materia.
- La entrega deberá hacerse a más tardar el día viernes 06 de septiembre a las 11:59pm.
- El esquema de carpetas deberá ser legible y adecuadamente estructurado de forma que sea fácil encontrar cada uno de los elementos de entrega solicitados:
 - Software que resuelve la problemática de la ETAPA II.
 - Documento reporte de la ETAPA III.
- 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 6 de septiembre, **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 III.