

Parcial III. Proyecto.Algoritmos de Ordenamientos.

Una empresa de desarrollo de software está desarrollando un sistema de generación de reportes y te ha contratado a ti y tus compañeros para desarrollar los siguientes requerimientos:

Desarrollo de una aplicación de gestión de productos en Python, que permite crear, modificar, consultar y listar productos cuyas opciones se mostrarán en un menú:

1. La aplicación deberá permitir la creación de productos con atributos como **nombre, descripción, categoría, precio, imagen, SKU, cantidad, peso, dimensiones(alto y ancho en cm), fecha de creación y fecha de actualización**. Los productos deberán almacenarse en memoria en una lista. **(2 ptos)**
2. La aplicación deberá permitir la actualización de los atributos de los productos, actualizar el precio o la descripción, y modificar la imagen del producto. **(2 ptos)**
3. La aplicación deberá permitir cargar la lista de productos de prueba desde el inicio de la aplicación. Estos productos deberán estar guardados en un archivo csv **(2 ptos)**
4. La aplicación deberá permitir realizar los siguientes reportes junto a sus respectivos algoritmos de ordenamiento, que serán seleccionados a través de un submenú **(10 ptos 2 ptos c/u)**:
 - a. Listar productos según su cantidad de forma ascendente utilizando el algoritmo de quicksort
 - b. Listar productos de forma descendente según su peso en un rango de fecha de actualización introducido por el usuario, utilizando el algoritmo de mergesort
 - c. Listar productos de forma ascendente o descendente según su fecha de creación utilizando el algoritmo de shellsort. La opción de ordenamiento será seleccionada por el usuario
 - d. Listar los productos de forma ascendente según su cantidad utilizando el algoritmo de heapsort

- e. Listar productos de forma ascendente o descendente según sus dimensiones y antigüedad en un rango de fecha(Utilizar algoritmo de ordenamiento de su elección)

Pautas de Evaluación.

1. La evaluación es individual o en equipo de un máximo de tres personas.
2. Está permitido usar paradigma de programación estructurada o programación orientada a objetos
3. La defensa del proyecto tiene un valor de 4 puntos.
4. Los códigos iguales tendrán una penalización de puntos menos.
5. La entrega y defensa se realizará de forma presencial en hora de clases.
6. Realizar validaciones de datos introducidos por el usuario y el código debe estar comentado.
7. Tener datos por defectos para tomarlos como prueba.
8. En cada módulo se evaluará los siguiente:
 - a. Funcionamiento del módulo(errones, resultados correctos, independencia).
 - b. Legibilidad del código(nombres de variables, código comentado correctamente)
 - c. El alumno aplicó elementos conceptuales en la programación del módulo
 - d. Módulo entregado puntualmente.