

Parcial III. Algoritmos de Ordenamiento.

Dado el archivo json proporcionados con datos referenciales: datos_pruebas.json (El equipo tiene permitido modificar los datos para hacer pruebas) y la función de carga de datos. Programar los siguiente:

1. Identificar y programar cada una de las clases necesarias para el funcionamiento del programa.(**3 ptos**)
2. Programar una clase Reporte con un menú de opciones que realice lo siguiente(**1 pto**):
 - a. Consulta las tareas que están en un estado específico y ordénalas por el porcentaje de avance de forma descendente utilizando quicksort cuyo pivote sea el ultimo elemento (**2 ptos**)
 - b. Encuentra todos los proyectos cuyo nombre contenga una palabra clave específica y ordénalos por la fecha de inicio de forma ascendente utilizando mergesort (**2 ptos**)
 - c. Consulta todas las subtareas de un proyecto específico que están completadas y ordénalas por el nombre de forma ascendente utilizando shellsort(**2 ptos**)
 - d. Ordena los proyectos por la cantidad total de subtareas en todas las tareas de forma ascendente utilizando mergesort(**4 ptos**)
 - e. Encuentra todas las tareas de un proyecto que tengan un porcentaje de avance menor a un valor específico y ordénalas por la fecha de vencimiento en orden ascendente utilizando heapsort (**3 ptos**)
 - f. Consulta las tareas de un proyecto específico y almacenarlas en un diccionario hash con la fecha de inicio como clave.(**3 ptos**)