

Parcial III. Algoritmos de Ordenamiento.

Eres el encargado de desarrollar un sistema para gestionar torneos de videojuegos donde los jugadores compiten en partidas y obtienen puntajes. Se requiere ordenar a los jugadores en función de diferentes criterios para determinar clasificaciones y enfrentar a los mejores entre sí.

La clase jugador debe contener los siguientes atributos(3 ptos):

victorias (entero) → Cantidad de partidas ganadas.

derrotas (entero) → Cantidad de partidas perdidas.

efectividad (flotante) → Proporción de victorias sobre partidas jugadas. (Se calcula como $victorias / (victorias + derrotas)$, con un mínimo de 0 si no ha jugado).

país (cadena) → Nacionalidad del jugador.

fecha_registro (datetime) → Cuándo se unió al torneo (puede usarse para ordenar por antigüedad).

tipo_jugador (cadena) → Puede ser "novato", "intermedio" o "experto" según el nivel.

La clase Torneo debe contener una lista de jugadores, un menú de opciones y los métodos necesarios para mostrar los siguientes reportes: (Menú de opciones: 1 pto)

1. Ordenar por número de victorias y, en caso de empate, por menor número de derrotas (**usando Quicksort**). 2ptos
2. Encuentra todos los jugadores con nivel entre X e Y y ordénalos por efectividad. (**Utilizar Heapsort**) 2 ptos
3. Muestra todos los jugadores agrupados por país y ordenados por nivel dentro de cada grupo.(**Utilizar Mergesort**) 2 ptos
4. ¿Cuáles son los 5 jugadores con la mejor efectividad (>80%) que tengan al menos 10 partidas jugadas? (**utilizar Shellesort**) 2 ptos
5. ¿Quién fue el primer jugador con nivel N que se unió al torneo? (**Utilizar quicksort**) (2 ptos)
6. Determinar los Jugadores Más Regulares: (4 ptos)
 - a. Un jugador es considerado **regular** si su número de **victorias y derrotas es similar** (es decir, la diferencia entre ambos es menor a un 10%).
 - b. Encuentra todos los jugadores regulares y ordénalos por **puntaje usando quicksort, heapsort o shellsort**.
 - c. Si hay empate, desempata por **fecha de registro con Mergesort** (los más antiguos primero).

Pautas de Evaluacion.

1. Usar Programación Orientada a objetos
2. Es presencial
3. Individual o en equipo máximo de dos integrantes.
4. Los códigos iguales tienen penalización de puntos menos.
5. No usar material de apoyo.
6. Utilizar el código torneo.py para cargar los datos del archivo jugadores.json.