Proyecto Final – Técnicas de Programación

Desarrollarás un sistema de consola con temática de casino que permita simular la gestión de jugadores, la participación en diferentes juegos de azar, y los procesos internos del establecimiento. Este proyecto tiene como propósito integrar y aplicar las distintas técnicas de programación y estructuras de datos vistas a lo largo del curso, tales como listas, pilas, colas, recursividad, backtracking, y algoritmos de búsqueda y ordenamiento.

El sistema debe permitir registrar jugadores ingresando datos por consola como nombre completo, ID único (alfanumérico) y saldo inicial. Esta información debe almacenarse y leerse desde archivos de texto o formato JSON para garantizar la persistencia. Ejemplo: un jugador se registra como 'Juan Pérez', ID 'JP001', saldo: $500. El sistema debe permitir consultar, modificar y eliminar información de jugadores según se requiera.

Debe implementarse un historial de actividades por jugador, incluyendo apuestas realizadas, juegos ganados, juegos perdidos y saldo actualizado. Estas actividades pueden manipularse como cadenas de texto y deben almacenarse en una pila para recuperar, por ejemplo, las últimas 10 acciones. Ejemplo: ['Apostó $50 en ruleta', 'Ganó $100', 'Perdió $30', ...].

Los jugadores que esperan ingresar a mesas populares deben estar en una cola por orden de llegada. Simula este comportamiento usando una cola, por ejemplo: ['Jugador A', 'Jugador B', 'Jugador C'] en espera para la ruleta.

Implementa al menos dos juegos. Ejemplos (están en la libertad de generar unos diferentes):
- Tragamonedas: usa fuerza bruta para generar combinaciones aleatorias y evaluar si son ganadoras.
- Blackjack simplificado: usa recursión de pila para sumar cartas hasta alcanzar o acercarse a 21.
- Juego de adivinanza: con recursión de cola para calcular el número óptimo de intentos.
- Simulación de combate entre jugador y casino (recursión cruzada): el jugador y el sistema lanzan jugadas alternas hasta que uno gana.

Utiliza backtracking para resolver un problema estratégico como: 'Dado un saldo inicial y una tabla de apuestas posibles, encontrar la mejor ruta de apuestas para maximizar las ganancias sin irse a cero'. El juego y la lógica deben estar bien documentados y explicados.

Para la gestión de datos se deben aplicar algoritmos de búsqueda y ordenamiento:
- Búsqueda lineal por nombre del jugador.
- Búsqueda binaria por ID (una vez que los datos estén ordenados alfabéticamente o por ID).
- Ordenamiento burbuja, selección e inserción para organizar los jugadores por saldo, número de juegos ganados, o total apostado.
- Ordenamiento por mezcla para generar rankings globales eficientes.

Además, el sistema debe generar al menos 5 reportes distintos, por ejemplo:

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

1. Jugadores con mayor saldo.
2. Historial de un jugador específico.
3. Ranking de los mejores jugadores.
4. Jugadores que más veces han perdido.
5. Juegos con mayor cantidad de participaciones.

Todo el código debe estar correctamente documentado y dividido en funciones o módulos claros. Implementa validaciones en cada entrada (por ejemplo, evitar que un jugador tenga saldo negativo o que se registren IDs duplicados).

Recomendaciones

- Utiliza archivos de texto o JSON para guardar información persistente.
- Divide el proyecto en módulos como: Gestión de Jugadores, Simulación de Juegos, Gestión de Archivos, Reportes, etc.
- Asegúrate de que cada juego tenga reglas claras, retroalimentación para el usuario y lógica documentada.
- Comenta cada función del sistema explicando qué hace, qué parámetros recibe y qué devuelve.
- Realiza pruebas con al menos 3 jugadores y 2 juegos distintos.
- Documenta los juegos que usan recursividad y backtracking con diagramas o pseudocódigo.

Entregables

- Análisis y estrategias de solución para cada componente del proyecto.
- Código fuente en Python completamente funcional y documentado.
- Archivos de datos de ejemplo (mínimo 3 jugadores y 2 juegos).
- Informe que incluya: descripción general del sistema, estructuras de datos utilizadas, algoritmos implementados, resultados de ejecución y justificación de decisiones.
- Video o presentación mostrando el funcionamiento del sistema y explicación del diseño.

Nota importante: la documentación deberá estar en inglés. Los videos narrados en inglés tendrán valoración adicional.

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia