

***Instrucciones:**

Proceda a resolver el ejercicio propuesto, usando como lenguaje Java, comprima su proyecto en un .zip / .rar y proceda a subirlo a la plataforma de mediación antes de la fecha propuesta. Este proyecto es individual.

Fecha límite para la entrega: **sábado 06 de julio 11:00 pm.**

Detalle del problema: Usando como base el juego realizado el proyecto programado #2, debe implementar las siguientes mejoras/cambios:

1. **La matriz** de juego ahora debe ser una matriz de objetos en lugar de una matriz de enteros (ahora de 10 filas x 10 columnas), y los valores aleatorios deben ser generados del 0 (inclusive) al 99 (inclusive), la **clase a instanciar** debe tener atributos para el valor entero aleatorio y para mantener el control si dicho espacio está ya seleccionado (usado), puede ser un valor comodín como boolean, int ó char. También debe programar un método o función para guardar los elementos de la matriz en un archivo de texto con formato CSV.
2. **Función adicional 1:** Una vez generados los valores aleatorios enteros para cada objeto que compone la matriz de juego, debe recorrer la matriz y depositar en una lista enlazada doble los objetos correspondientes a los valores IMPARES, para esto debe implementar una clase "Lista" con las funciones pertinentes vistas en clase. Las inserciones deben realizarse al principio de la lista.
3. **Función adicional 2:** Una vez generados los valores aleatorios enteros para cada objeto que compone la matriz de juego, debe recorrer la matriz y depositar en árbol binario los objetos correspondientes a los valores PARES, para esto debe implementar una clase "Arbol" con las funciones pertinentes vistas en clase, la raíz del árbol, o sea el primer elemento a insertar en el árbol, debe ser el valor constante 50 (cincuenta), para que así los valores MENORES O IGUALES a 50 se inserten a la izquierda de la raíz y los valores MAYORES a 50 se inserten a la derecha.
4. Usando la lista enlazada doble del punto 2, debe implementar métodos en la clase "Lista" para las siguientes funciones:
 - Obtener el elemento más grande de la lista (entero más grande).
 - Obtener el elemento más pequeño de la lista (entero más pequeño).
 - Obtener el promedio de los valores enteros.
 - Obtener cuántos números primos tiene la lista enlazada.
 - Guardar cada elemento de la lista en un archivo BINARIO.
 - Cargar los elementos de un archivo binario en una lista nueva, en este caso insertando siempre al final y mostrar en pantalla el recorrido de dicha lista.
5. Usando el árbol binario del punto 3, debe implementar métodos en la clase "Arbol" para las siguientes funciones:
 - Contar y obtener la cantidad de nodos que hay a la izquierda de la raíz del árbol.
 - Contar y obtener la cantidad de nodos que hay a la derecha de la raíz del árbol.
 - Guardar los elementos / nodos (valor numérico) del árbol en un archivo de texto, usando los recorridos PreOrder, Order y PostOrder para recorrer cada elemento y guardarlo en cada archivo (uno por recorrido), cada elemento entero de cada archivo debe ir una línea.
 - Cargar los datos del archivo correspondiente al recorrido PreOrder y con cada uno crear un árbol binario, luego debe desplegar en pantalla los recorridos del árbol recién creado en PreOrder, Order y PostOrder.

En esta ocasión **NO** debe implementar/copiar la funcionalidad de la tarea 2, únicamente la creación de la matriz de 10x10 y luego implementar las mejoras/nuevas funciones. Puede implementar un menú de opciones para que el usuario seleccione que proceso ejecutar, ejemplo:

- 1- **Crear/cargar – Guardar matriz**
 - a. **Crear nueva matriz.**
 - b. **Guardar matriz en archivo CSV.**
 - c. **Cargar matriz desde archivo (esto en caso de tener una previamente guardada).**
- 2- **Crear Lista Enlazada Doble**
- 3- **Crear Árbol Binario**
- 4- **Operaciones con la Lista Enlazada Doble:**
 - a. **Obtener el elemento más grande.**
 - b. **Obtener el elemento más pequeño.**
 - c. **Obtener el promedio.**
 - d. **Obtener cuántos números primos tiene la lista.**
 - e. **Guardar cada elemento de la lista en un archivo BINARIO.**
 - f. **Cargar los elementos de un archivo binario en una nueva lista.**
- 5- **Operaciones con el Árbol Binario:**
 - a. **Obtener cantidad de nodos a la izquierda de la raíz del árbol.**
 - b. **Obtener cantidad de nodos a la derecha de la raíz del árbol.**
 - c. **Guardar cada recorrido del árbol en archivos de texto.**
 - d. **Cargar los datos del archivo recorrido PreOrder.**

Debe validar que no se seleccione una función u opción que dependa de una anterior, sin que ésta se haya ejecutado.

Criterios de evaluación:

Para el desarrollo de esta tarea, debe usar cualquiera de los conceptos estudiados hasta este momento durante el semestre, como objetos, manejo de errores, estructuras de datos, lectura y escritura de archivos. Debe seguir las métricas descritas a continuación:

1. 15% Uso de clases y objetos
2. 20% Lectura/ escritura de archivos binarios y de texto
3. 10% Manejo de errores
4. 25% Implementación de estructuras de datos adecuadas, listas y árboles
5. 20% Lógica y desarrollo del problema
6. 10% Orden, documentación interna