**Práctica 4 – Introducción a las Ciencias de la Computación**

**Diseño de la Solución**

**María Fernanda Miranda Sánchez**

1. **Inicio y Bienvenida**

El programa comienza mostrando un mensaje de bienvenida y el menú principal con tres opciones: gestionar catálogo de juegos, jugar memorama o salir.

1. **Menú principal**

Se presenta al usuario el menú principal con las opciones numeradas (1, 2, 0).

* Si elige 1, se llama al módulo CatalogoJuegos.gestionar(sc).
* Si elige 2, se crea una instancia de Memorama y se ejecuta miJuego.jugar().
* Si elige 0, el programa imprime despedida y termina.

1. **Gestión del Catálogo de Juegos (submenú)**

Al entrar en la opción de catálogo, se muestra el submenú con las siguientes operaciones:

1. Agregar Juego
2. Ordenar Juegos por Nombre
3. Ordenar Juegos por Calificación
4. Ordenar Juegos por Género
5. Ranking de Juegos (Mejor y Peor)
6. Mostrar Todos los Juegos
7. Volver al menú principal
8. **Agregar Juego (flujo)**

* El sistema solicita Nombre, Calificación y Género.
* Se guarda en juegos[numJuegos] = new String[]{nombre, calificacion, genero}.
* Se incrementa numJuegos.
* Si numJuegos alcanza la longitud del arreglo, se informa que el catálogo está lleno.

1. **Estructura de datos del Catálogo**

* private String[][] juegos; (filas = capacidad, columnas = 3: nombre, calificación, género)
* private int numJuegos; contador de elementos activos

1. **Ordenamientos (algoritmo)**

Se emplea Bubble Sort (ordenamiento por burbuja) implementado en ordenar(int columna):

* Si columna == 1 (calificación): convertir cadenas a double con Double.parseDouble y comparar numéricamente.
* Si columna == 0 ó 2 (nombre/género): comparar alfabeticamente con compareToIgnoreCase.
* Tras ordenar, el programa imprime automáticamente la lista con mostrarJuegos.

1. **Ranking (mejor/peor)**

* Para obtener el mejor y el peor: recorre los numJuegos y lleva el índice del máximo y mínimo de la columna de calificación.
* Imprimir el nombre y la calificación del mejor y del peor.

1. **Memorama (flujo)**

* Tablero 4x4: char[][] simbolos y boolean[][] visible.
* Inicialización: crear 8 pares de símbolos en un arreglo de 16 posiciones, barajar aleatoriamente y rellenar el tablero.
* Ciclo del juego:
  1. Imprimir tablero.
  2. Pedir primera coordenada.
  3. Validar que la casilla no esté ya visible.
  4. Revelar, volver a imprimir.
  5. Pedir segunda coordenada (validar que no sea la misma ni ya visible).
  6. Revelar y comparar símbolos; si coinciden incrementar paresEncontrados; si no, volver a cubrirlas.
  7. Incrementar movimientos por cada par intentado.
  8. Repetir hasta paresEncontrados == 8.