

### **Entrega Final Estructura de Datos: Implementación del Juego “Buscaminas”**

Los estudiantes deberán desarrollar el juego “Buscaminas” como una aplicación funcional en Java, aplicando los conceptos teóricos y prácticos de la asignatura. El juego debe permitir generar tableros de distintos tamaños, distribuir minas de manera aleatoria, calcular y mostrar el número de minas adyacentes, y gestionar el flujo lógico del juego (victoria o derrota).

Se evaluará la implementación de estructuras de datos adecuadas y el diseño eficiente del algoritmo que controla la interacción entre los componentes del juego.

#### **Requerimientos Técnicos**

1. **Lenguaje:** Java o Python.
2. **Estructuras de Datos a aplicar (mínimo tres):**
  - **Listas enlazadas o circulares:** manejo del tablero o almacenamiento de celdas.
  - **Pilas:** para controlar movimientos anteriores o retrocesos (deshacer jugadas).
  - **Colas:** para el procesamiento de celdas adyacentes o expansión automática al revelar celdas vacías.
  - **Árboles binarios (opcional o avanzado):** para representar niveles de dificultad o clasificación de puntuaciones.
3. **Generación aleatoria de minas:** uso de generador Random o algoritmo propio.
4. **Interfaz:** puede ser gráfica (Swing/JavaFX) o consola interactiva, siempre y cuando el juego sea completamente funcional.
5. **Código comentado y modularizado** (uso de clases y métodos coherentes).
6. **El programa no debe depender de librerías externas** salvo las estándar de Java.

#### **Entregables**

1. **Código fuente completo** del juego (en carpeta comprimida .zip con estructura organizada o GitHub).
2. **Manual de usuario** (en formato PDF) que incluya:
  - Descripción general del juego.
  - Reglas y condiciones de victoria/derrota.
  - Instrucciones de uso (instalación y ejecución).
  - Capturas de pantalla o ejemplos de uso.
  - Consideraciones técnicas y posibles errores.
3. **Presentación (diapositivas)** que explique:
  - Estructura del programa y clases principales.
  - Estructuras de datos implementadas y su justificación.

- Flujo lógico del algoritmo principal.
  - Retos técnicos y mejoras propuestas.
4. **Sustentación oral presencial** (10-15 minutos por equipo) donde se muestre el funcionamiento y se respondan preguntas técnicas.

#### Indicaciones para la Sustentación

Durante la sustentación, cada grupo deberá:

1. **Ejecutar el programa** en vivo y demostrar su funcionalidad completa (inicio, juego, detección de minas, fin de partida).
2. **Explicar la estructura de datos usada**, justificando por qué fue la más adecuada para cada parte del sistema.
3. **Argumentar decisiones de diseño**, mostrando el uso de POO (clases, herencia, composición, encapsulamiento).
4. Responder preguntas técnicas sobre el funcionamiento interno del código y posibles mejoras futuras.

#### Criterios de Evaluación

Criterio	Descripción	Ponderación (%)
<b>Funcionalidad del juego</b>	El juego cumple con las reglas y responde correctamente a las interacciones del usuario.	25%
<b>Aplicación de estructuras de datos</b>	Uso coherente y justificado de al menos tres estructuras (listas, pilas, colas, árboles).	25%
<b>Calidad del código y modularidad</b>	Código limpio, comentado, organizado y con aplicación de POO.	20%
<b>Documentación y manual de usuario</b>	Claridad, presentación profesional, explicaciones comprensibles.	10%
<b>Presentación y sustentación oral</b>	Explicación técnica, dominio del tema, capacidad de argumentación.	20%
<b>Total</b>		<b>100%</b>