

Miniproyecto de Programación en Java: *Dragon Quest VIII*



El reino de Trodain está nuevamente en peligro. Cuatro valientes aventureros (el Héroe, Yangus, Jessica y Angelo) se preparan para una peligrosa misión: enfrentar a cuatro monstruos clásicos que amenazan con sembrar el caos en la región.

Tu misión es construir una simulación en Java que represente esta épica batalla. El sistema debe permitir crear una partida clásica con el grupo de cuatro héroes enfrentándose por turnos a cuatro enemigos controlados por la IA. Cada personaje tiene habilidades, estadísticas y comportamientos únicos, tanto en ataque como en defensa.

Tu simulación debe enfocarse en la lógica de combate por turnos, con decisiones estratégicas, variedad de personajes, habilidades y eventos durante el combate. No se debe hacer interfaz gráfica, debe correr en consola de manera clara, fluida y comprensible.

Requerimientos

1. Sistema de personajes jugables y enemigos:

- Cada personaje tiene atributos como HP, MP, ataque, defensa, velocidad, etc.
- Cada uno posee habilidades o ataques especiales que consumen recursos o tienen efectos únicos.

- Los enemigos también deben tener comportamientos diferenciados (por ejemplo, un enemigo más agresivo, otro más defensivo, etc.).

2. Sistema de turnos:

- El orden de acción debe respetar la velocidad de los personajes.
- Cada turno permite a los héroes elegir acciones como atacar, defender, usar habilidades o ítems.
- Los enemigos actuarán automáticamente según una lógica quemada.

3. Acciones especiales:

- Habilidades con efectos como curación, daño mágico, aumento de estadísticas, etc.
- Posibles estados alterados (envenenado, paralizado, dormido, etc.) que influyen en el combate.

4. Sistema de combate:

- Finaliza cuando todos los enemigos o todos los héroes son derrotados.
- Mostrar mensajes de combate y estado de los personajes tras cada acción.

5. Configuración inicial:

- Los personajes deben crearse con sus atributos y habilidades predefinidos al iniciar el programa, estos no deben ingresarse por consola (sería muy largo y tedioso)
- El combate se debe lanzar directamente o mediante un menú inicial simple.

Diseño

Aunque no se especifica cómo implementar cada parte, deberías organizar tu código de forma que te permita:

- Reutilizar comportamientos comunes entre tipos de personajes.

- Definir acciones o habilidades que puedan comportarse de manera diferente, pero comparten cierta estructura.
- Agrupar los distintos tipos de personajes, enemigos y habilidades de forma lógica.
- Controlar y almacenar listas de personajes, enemigos y habilidades de forma eficiente.
- Tomar decisiones en el flujo del combate de forma modular y mantenible.

Consideraciones Extra

- ¿Qué pasa si dos personajes tienen la misma velocidad?
- ¿Qué sistema se usará para definir las habilidades (y que puedan tener efectos distintos)?
- ¿Cómo se podrían representar los distintos tipos de personajes o enemigos?
- ¿Cómo se puede garantizar que el código sea extensible (por ejemplo, para el miniproyecto 2)?

Entregable

El proyecto debe entregarse en un repositorio (GitHub) por el campus en la fecha acordada:

- Código fuente bien estructurado.
- Instrucciones claras de compilación y ejecución.
- Un archivo **README.md** explicando la lógica general del diseño y con el nombre y código de los integrantes (los que no aparezcan aquí no les califico, si cometen este error al que falte le bajó la mitad de la nota, *[ojo aquí](#)*)
- Grupos de 3 (no se acepta individual)
- Nombres de los integrantes en ca