

## Ejercicio 4: Modelación con herencia Juego de Batalla

### Análisis del Sistema

#### 1.1 Requisitos funcionales del sistema

- El programa debe tener al menos 2 enemigos de distintos tipos con una versión normal y jefe de cada uno
- Debe poder dejar que jueguen 1 o más jugadores.
- Debe tener el simulador de la batalla
- La batalla debe ser por turnos
- Cada enemigo o jugador debe tener un nombre, pts de vida y ataque.
- Debe dejar que el combatiente o enemigo tenga un mensaje en su turno y debe dar la opción de atacar, pasar turno o utilizar un ítem.
- A los combatientes se les debe dar la posibilidad de poder elegir un objetivo y al atacar se deben reducir los pv de el combatiente al que seleccionen como objetivo.
- Se debe asignar una habilidad especial a cada enemigo
- A los enemigos jefes el programa debe darles una habilidad extra, además deben tener mejores stats que sus versiones normales.
- Los ítems de los jugadores solo tienen que ser ítems que tienen efecto a 1 o más jugadores y pueden tener efectos positivos en sus stats.
- El programa debe permitir que el jugador elija entre 2 roles guerrero y explorador
- Las batallas deben tener de 1 a 3 enemigos, se eligen al azar la cantidad
- Debe finalizar el simulador de la batalla cuando se reduzcan por completo los pv de los jugadores o enemigos.
- Debe permitir al usuario controlar tanto a enemigos como jugadores con menús.
- Debe llevar un registro de los últimos 3 turnos y mostrar los stats actuales de enemigos y jugadores.

#### 1.2 Clases necesarias y su propósito

Clase	Propósito
<b>Combatientes</b>	Clase base que encapsula atributos y comportamientos generales de todos los personajes (jugadores y enemigos).
<b>Jugador</b>	Representa al usuario, con un rol y un inventario inicial de ítems. Permite atacar y usar objetos.
<b>Enemigos</b>	Representa a los enemigos que enfrentan al jugador. Poseen habilidades especiales y, si son jefes, una segunda habilidad.
<b>Items</b>	Representa objetos que los jugadores pueden usar para curarse, obtener escudos, aumentar ataque o ganar turnos extra.
<b>HabilidadEspecial</b>	Define los poderes de los enemigos (ej. envenenar, inhabilitar, dañar múltiples).

<b>Batalla</b>	Gestiona el flujo de la pelea, alternando turnos, mostrando estados y resolviendo acciones.
<b>MainEjer4</b>	Driver program que inicializa los objetos y arranca la simulación de la batalla.

### 1.3 Atributos de cada clase

Clase: Combatientes

Atributo	Tipo de dato	Visibilidad	Propósito
nombre	String	private	Identificar al combatiente
PV	int	private	Puntos de vida
PA	int	private	Poder de ataque
inhabilitado	boolean	private	Indica si no puede actuar
aumentoAtaque	int	private	Bono temporal de ataque
escudoActivo	int	private	Escudo de defensa activo
turnoExtra	boolean	private	Permite jugar un turno adicional
turnosInhabilitado	int	private	Contador de turnos inhabilitado
turnosVeneno	int	private	Contador de turnos envenenado
danoVenenoPorTurno	int	private	Daño por veneno en cada turno

Clase: Jugador

Atributo	Tipo de dato	Visibilidad	Propósito
inventario	List	private	Ítems disponibles para usar
rol	String	private	Tipo de jugador (Guerrero, Explorador, etc.)

Clase: Enemigos

Atributo	Tipo de dato	Visibilidad	Propósito
tipo	String	private	Tipo elemental del enemigo

habilidadEspecial	HabilidadEspecial	private	Habilidad básica del enemigo
habilidadJefe	HabilidadEspecial	private	Habilidad secundaria (solo si es jefe)

Clase: Items

Atributo	Tipo de dato	Visibilidad	Propósito
nombre	String	private	Nombre del ítem
efecto	String	private	Tipo de efecto aplicado
valor	int	private	Intensidad del efecto

Clase: HabilidadEspecial

Atributo	Tipo de dato	Visibilidad	Propósito
nombre	String	private	Nombre de la habilidad
efecto	String	private	Tipo de efecto
valor	int	private	Intensidad del efecto
duracion	int	private	Duración en turnos

Clase: Batalla

Atributo	Tipo de dato	Visibilidad	Propósito
jugador	Jugador	private	Jugador en combate
enemigos	ArrayList	private	Lista de enemigos enfrentados

## 1.4 Métodos de cada clase

Clase: Combatientes

Método	Parámetros	Visibilidad	Propósito
recibirDano	int dano	public	Recibir daño o curación
aplicarAumentoAtaque	int valor	public	Incrementa ataque
resetearAumentoAtaque	-	public	Restablece bono de ataque
activarEscudo	int valor	public	Activa defensa temporal
aplicarVeneno	int dano, int turnos	public	Aplica estado de veneno

avanzarEstadosDeTurno	-	public	Actualiza efectos activos
estaVivo	-	public	Retorna si está vivo
otorgarTurnoExtra	-	public	Concede un turno adicional
usarTurnoExtra	-	public	Consume turno adicional

Clase: Jugador

Método	Parámetros	Visibilidad	Propósito
usarItem	String nombreItem, Combatientes objetivo	public	Usar un ítem sobre objetivo
atacar	Enemigos enemigo	public	Atacar a un enemigo
mostrarInventario	-	public	Mostrar ítems disponibles
mensajeInicioBatalla	-	public	Mensaje inicial del jugador

Clase: Enemigos

Método	Parámetros	Visibilidad	Propósito
usarHabilidadEspecial	Combatientes objetivo	public	Aplica habilidad básica
usarHabilidadJefe	Combatientes[] objetivos	public	Aplica habilidad de jefe
actuar	Combatientes objetivoPrincipal	public	Ejecuta acción en turno
mensajeInicioBatalla	-	public	Mensaje inicial del enemigo

Clase: Items

Método	Parámetros	Visibilidad	Propósito
usar	Combatientes objetivo	public	Aplica efecto del ítem

Clase: HabilidadEspecial

Método	Parámetros	Visibilidad	Propósito
aplicarHabilidad	Combatientes objetivo	public	Aplica efecto sobre un combatiente

aplicarHabilidadMultiple	Combatientes[] objetivos	public	Aplica efecto a múltiples combatientes
--------------------------	-----------------------------	--------	--

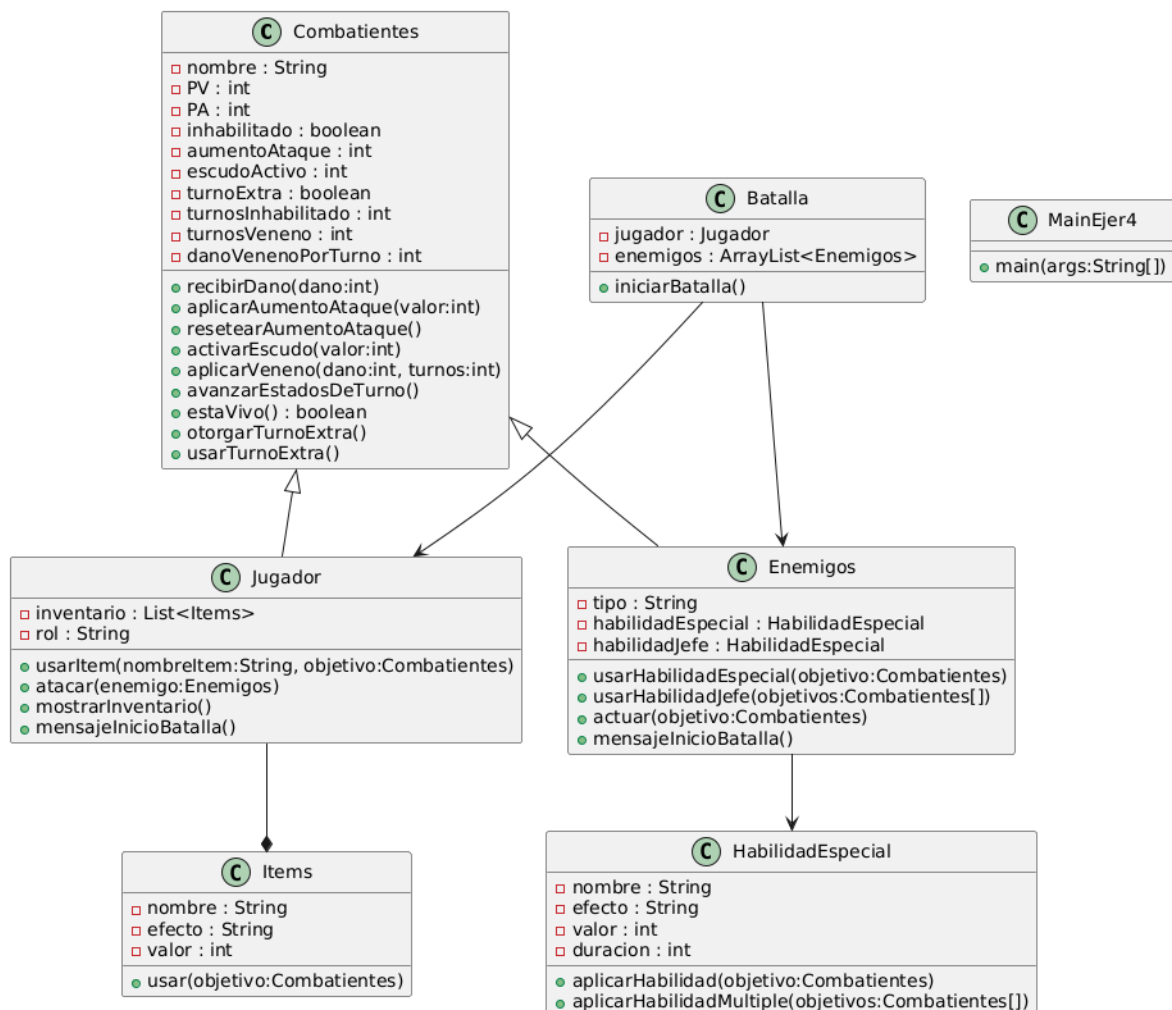
Clase: Batalla

Método	Parámetros	Visibilidad	Propósito
iniciarBatalla	-	public	Controla el flujo de la batalla

Clase: MainEjer4

Método	Parámetros	Visibilidad	Propósito
main	String[] args	Public static	Inicia programa y arranca la batalla

## 2.0 Diseño: Diagrama de Clases



### **3.0 Programa**

Github: <https://github.com/Maquito365/Ejer4-Herencia>

<https://github.com/Maquito365/Ejer4-Herencia>