

# MEMORIA GRUPO

## Battle Terminal | Battle Royale en Consola (Java)

### Introducción

El siguiente documento explica el desarrollo del videojuego **Battle Terminal**, un proyecto basado en consola implementado y en interfaz gráfica (GUI) en **Java**. Este proyecto tiene como objetivo para desarrollar lo aprendido en la asignatura donde los jugadores deben enfrentarse en un tablero que se reduce progresivamente mientras toman decisiones estratégicas: moverse, atacar, recoger armas o activar habilidades especiales.

La implementación del juego se realizó siguiendo un enfoque **jerárquico y modular**.

Battle Terminal fue diseñado con una lógica flexible que permite **escalar** el proyecto hacia **futuras funcionalidades**, como la implementación de habilidades avanzadas, y una mayor complejidad en las mecánicas de combate.

### Cómo jugar

#### 1. Inicio del juego:

- Aparece un menú en consola que explica cómo comenzar a jugar.
- Al pasar al siguiente menú, se muestra una breve descripción del objetivo del juego, así como las funciones de cada personaje y arma.
- Se solicita al jugador ingresar el número de participantes.
- Cada jugador selecciona un personaje.
- El tablero inicial tiene un tamaño de **12x12**.
- Los jugadores y objetos iniciales se colocan aleatoriamente en el tablero.

#### 2. Acciones por turno:

En cada turno, cada jugador puede realizar una de las siguientes acciones:

- **Moverse:** Cambiar a una casilla adyacente (arriba, abajo, izquierda, derecha).

- **Recoger:** Si al moverse a una casilla hay un arma, el jugador puede elegir si desea cambiar si encuentra un arma, su arma actual por la nueva.
- **Atacar:** Si un jugador está en una casilla adyacente, puede atacar.
- **(POSIBLE IMPLEMENTACIÓN FUTURA) Usar habilidad:** Activar una habilidad especial (si está disponible). Las habilidades son:
  - **Curarse (+50 de vida):** Restaurar salud.
  - **Escudo:** Inmunidad al próximo ataque recibido.
  - **Colocar trampa (-60 de vida):** Dejar una trampa en tu casilla, la cual dañará a cualquier jugador que la pise.

### 3. Batallas:

- En cada ronda, se crea una (GUI) interfaz gráfica donde cada jugador verá únicamente a los enemigos en las casillas adyacentes.
- Solo se puede atacar si el enemigo está adyacente (arriba, abajo, izquierda o derecha).

### 4. Reducción del tablero:

- Al final de cada 3 **rondas**, el tablero se reduce, eliminando las filas y columnas exteriores.
- Si un jugador se encuentra en una casilla **eliminada**, muere automáticamente. (implementación futura)

### 5. Finalización del juego:

- El juego termina cuando solo queda un jugador y se muere todo el resto.

## Detalles del Juego

### Jugadores

Cada jugador selecciona uno de los siguientes personajes, cada uno con estadísticas y habilidades únicas. (BONUS) Además, cada personaje tiene un arma inicial asignada y recibe un **15% de bonificación al daño** si recoge su arma correspondiente durante el juego.

#### 1. Soldado:

- a. **Vida:** 150

- b. **Habilidad especial: Contraataque** (devuelve un 30% del daño recibido).

## 2. Explorador:

- a. **Vida:** 100
- b. **Habilidad especial: Movimiento rápido** (realiza dos movimientos en la siguiente ronda).

## 3. Médico:

- a. **Vida:** 120
- b. **Habilidad especial: Curación rápida** (restaura completamente su vida).

## 4. Ingeniero:

- a. **Vida:** 110
- b. **Habilidad especial: Colocar trampas** (coloca 2 trampas que dañan a los enemigos).

## Armas

A lo largo del juego, los jugadores pueden recoger diversas armas que están disponibles en el tablero:

1. **Cuchillo: 10** Daño bajo, siempre disponible.
2. **Fusil: 20** Daño Puede atacar en horizontal, vertical o diagonal.
3. **Escopeta: 15** Daño Puede atacar en horizontal o vertical, causando más daño que el fusil.
4. **Rifle de francotirador: 40** Daño Altísimo daño a larga distancia. Aumenta en 1 el número de casillas visibles por el jugador, pero solo puede atacar a enemigos en casillas adyacentes.
5. **Cañón de plasma: 50** Daño Ataque masivo en área (daña las casillas contiguas al objetivo).

## Habilidades especiales

### BONUS

Hemos puesto la mayor facilidad para poder implementar el código de habilidades en nuestro programa en una futura ampliación del proyecto

Las habilidades especiales aparecen aleatoriamente en el mapa y se activan por cada personaje cuando es posible. Las habilidades son las siguientes:

1. **Curarse (+50 de vida):** Recupera 50 puntos de vida, hasta el máximo inicial de vida.
2. **Escudo:** Inmunidad al próximo ataque recibido.
3. **Colocar trampa (-60 de vida):** Deja una trampa en la casilla del jugador. Cuando un enemigo pisa esa casilla, recibe 60 puntos de daño.

## Mapas

El tablero comienza con un tamaño de **12x12** y se reduce gradualmente cada 3 rondas conforme avanzan las rondas:

- **12x12:** Inicio del juego.
- **10x10:** Después de 2 rondas.
- **8x8:** Después de 4 rondas.
- **6x6:** Después de 6 rondas.
- **4x4:** Después de 8 rondas.

---

El tablero inicial es de 12x12 y se reduce gradualmente:

- **12x12:** Inicio del juego.
- **10x10:** Después de 3 rondas.
- **8x8:** Después de 6 rondas.
- **6x6:** Después de 9 rondas.
- **4x4:** Después de 12 rondas.

## Organización del Código: Battle Terminal

### Clases y Subclases

Esta organización sigue un enfoque jerárquico, de lo abstracto a lo concreto, con el fin de facilitar futuras expansiones o modificaciones del código.

---

## 1. Clases Abstractas y Base

### 1.1. Entidad

- Representa cualquier entidad del juego (jugadores, enemigos, etc.).
- **Atributos:**
  - nombre: Nombre de la entidad.
  - vida: Puntos de vida actuales.
  - posicion: Coordenadas actuales en el tablero.
- **Métodos:**
  - mover(posicionNueva): Cambia la posición de la entidad.
  - recibirDano(cantidad): Reduce la vida de la entidad.
  - estaVivo(): Retorna si la vida es mayor a 0.

### 1.2. Objeto

- Base para cualquier objeto interactivo en el tablero.
- **Atributos:**
  - tipo: Tipo de objeto (arma, trampa, etc.).
  - posicion: Ubicación en el tablero.
- **Métodos:**
  - usar(entidad): Aplica el efecto del objeto a una entidad.

---

## 2. Clases de Jugadores y Personajes

### \*\*2.1. Jugador (Hereda de Entidad)

- Representa a un jugador controlado por un usuario.
- **Atributos:**
  - personaje: Tipo de personaje seleccionado.
  - arma: Arma equipada actualmente.
  - habilidadEspecialDisponible: Indica si la habilidad especial está lista para usarse.
- **Métodos:**
  - usarHabilidadEspecial(): Activa la habilidad especial del personaje.
  - equiparArma(armaNueva): Cambia el arma actual.

### 2.2. Personaje (Clase Base para los Tipos de Personajes)

- Define características y habilidades comunes.
- **Subclases:**
  - Soldado:
    - Bonificación con fusil.
    - Habilidad: Contraataque.
  - Explorador:
    - Bonificación con escopeta.
    - Habilidad: Movimiento rápido.
  - Médico:
    - Bonificación con rifle de francotirador.
    - Habilidad: Curación rápida.
  - Ingeniero:
    - Bonificación con cañón de plasma.
    - Habilidad: Colocar trampas.

---

### 3. Clases de Objetos y Armas

#### **\*\*3.1. Arma (Hereda de Objeto)**

- Representa armas que los jugadores pueden usar.
- **Atributos:**
  - daño: Cantidad de daño que inflige el arma.
  - rango: Alcance del ataque.
  - bonificacion: Porcentaje de bonificación para personajes compatibles.
- **Subclases:**
  - Cuchillo:
    - Daño básico, sin restricciones de uso.
  - Fusil:
    - Ataque en línea recta y diagonal.
  - Escopeta:
    - Mayor daño, pero rango más limitado.
  - RifleFrancotirador:
    - Alto daño, mejora la visibilidad de enemigos.
  - CanonPlasma:
    - Ataque en área.

#### **\*\*3.2. Trampa (Hereda de Objeto)**

- Daña al jugador que pisa la casilla.
- **Atributos:**
  - daño: Cantidad de daño infligido.

---

### 4. Clases del Tablero y Gestión del Juego

#### **4.1. Tablero**

- Administra la representación y el estado del mapa.

- **Atributos:**
  - dimensiones: Tamaño actual del tablero.
  - casillas: Matriz que representa las casillas (vacías, ocupadas, etc.).
- **Métodos:**
  - inicializarTablero(jugadores, objetos): Coloca jugadores y objetos aleatoriamente.
  - actualizarTablero(): Reduce las dimensiones del tablero y ajusta posiciones.
  - imprimirMapa(jugador): Muestra el mapa visible al jugador.

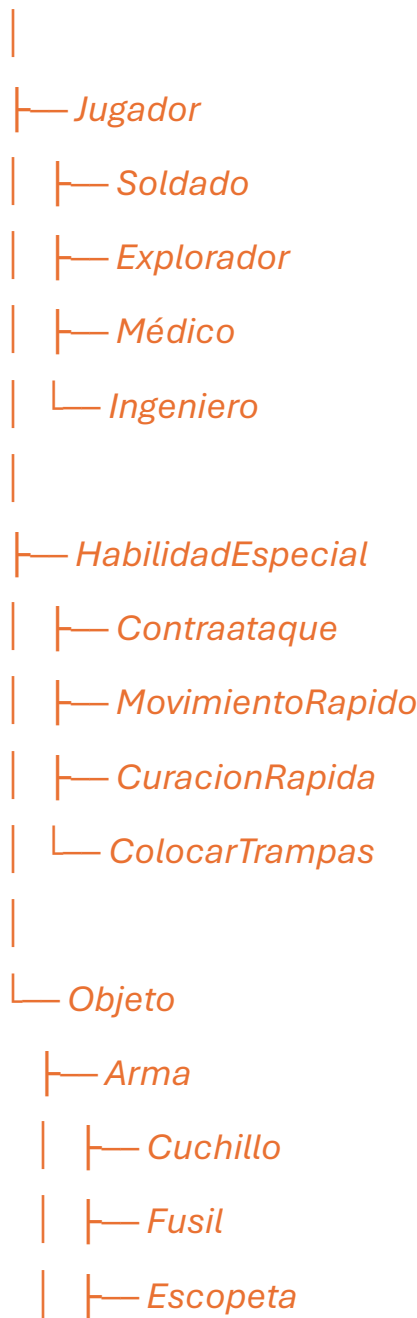
## 4.2. Partida

- Controla el flujo del juego.
- **Atributos:**
  - jugadores: Lista de jugadores activos.
  - tablero: Instancia de la clase Tablero.
  - rondaActual: Número de ronda en curso.
- **Métodos:**
  - iniciarJuego(): Configura el inicio de la partida.
  - realizarTurno(jugador): Maneja las acciones del jugador.
  - procesarBatallas(): Resuelve enfrentamientos entre jugadores adyacentes.
  - finalizarRonda(): Reduce el tablero y verifica condiciones de victoria.
  - mostrarGanador(): Anuncia al ganador.



JERARQUÍA DE CLASES:

Entidad



| |— *RifleFrancotirador*  
| |— *CanonPlasma*  
|— *Trampa*

El desarrollo de **Battle Terminal** ha permitido aplicar conceptos clave de programación orientada a objetos (POO) en Java, como la herencia, polimorfismo y abstracción, logrando un sistema modular y eficiente. El juego ofrece una experiencia estratégica única, en la que los jugadores deben optimizar sus movimientos, utilizar armas y habilidades de manera inteligente, y adaptarse a un tablero que se reduce progresivamente para sobrevivir.

**Trabajo realizado por: Santiago García, Javier González, Manuel Martínez, Fernando Contreras.**