

T.E.G (Táctica y Estrategia de Guerra) es un popular juego de mesa que busca simular enfrentamientos militares.

**1. Interfaz Medio:** Esta interfaz define la funcionalidad común de todos los "medios de combate".

#### Métodos

- `int getPoderCombate()`
- 

**2. Clase MedioCombate (Base para Herencia):** Esta clase implementará la interfaz **Medio** y servirá como base para los vehículos.

#### Atributos

- `double cantidadCombustible`

#### Constructor

- `public MedioCombate(double cantidadCombustible)`

#### Métodos

- `getCantidadCombustible(): double`
  - `getPoderCombate(): int` *(debe ser sobrescrito en las clases derivadas)*
- 

**3. Clases Derivadas de MedioCombate (Herencia):** Estas clases representarán los distintos **vehículos** con su cálculo de poder específico.

#### Clase Tanque

- `getPoderCombate(): int`  $\rightarrow 4 * cantidadCombustible$

#### Clase Helicoptero

- `getPoderCombate(): int → 8 * cantidadCombustible + 50`

#### Clase Avion

- `getPoderCombate(): int → 10 * cantidadCombustible + 100`
- 

**4. Clases de Soldados (Herencia y Especialización):** Cada clase tendrá un poder de combate fijo y habilidades especiales.

#### Clase Soldado

- `getPoderCombate(): int → 1`
- `dispararFusil(): void → Imprime "Dispara fusil"`

#### Clase Paracaidista (Extiende Soldado)

- `getPoderCombate(): int → 2`
- `saltar(): void → Imprime "Salta de un avión"`

#### Clase Comando (Extiende Paracaidista)

- `getPoderCombate(): int → 3`
  - `emboscar(): void → Imprime "Hace una emboscada"`
- 

**5. Clase Ejército:** Esta clase encapsula un conjunto de **MedioCombate** y define su comportamiento.

#### Atributos

- `String color`
- `MedioCombate[] medios (Arreglo de 3 medios)`

#### Constructor

- `public Ejercito(String color, MedioCombate a, MedioCombate b, MedioCombate c)`

## Métodos

- `getPoderTotal(): int` → Suma el poder de combate de todos los medios.
- 

**6. Clase Tablero:** La clase **Tablero** se encarga de enfrentar dos ejércitos y determinar el resultado de la batalla.

## Atributos

- `Ejercito ejercito1`
- `Ejercito ejercito2`

## Constructor

- `public Tablero(Ejercito ejercito1, Ejercito ejercito2)`

## Métodos

- `enfrentar(): void`
    - Compara el poder total de ambos ejércitos.
    - Imprime el resultado: "El ejército X ganó la batalla con un poder de combate de XXXX" o "Se produjo un empate".
- 

## 7. Clase Main

La clase principal crea instancias de los objetos y ejecuta la lógica del juego

```

public static void main(String args[]) {

Ejercito azul=new Ejercito("Azul", new Avion(10.50), new Comando(), new
Tanque(12.00));

Ejercito rojo=new Ejercito("Rojo", new Helicoptero(13.45), new Soldado(), new
Paracaidista())
;Tablero tablero=new Tablero(azul,rojo);

tablero.enfrentar();

Comando comando = new Comando();
comando.dispararFusil();
comando.saltar();
comando.emboscar();

Avion avion=new Avion(10.50);
System.out.println(avion.getCantidadCombustible());

Tanque tanque=new Tanque(12.00);
System.out.println(tanque.getCantidadCombustible());

}

```

## Resumen del Diseño

- **Interfaces:** Medio.
- **Herencia:** Clases de soldados y vehículos.
- **Composición:** La clase **Ejercito** contiene un arreglo de **MedioCombate**.
- **Encapsulamiento:** Atributos privados con métodos de acceso.
- **Polimorfismo:** Uso de métodos sobrescritos como `getPoderCombate()` y habilidades específicas.