

```
<!--Proyecto programación-->
```

Pokemon Showdown {

```
<Por="Nicolae Poenaru"/>
```

```
}
```



Indice

01

`class Pokemon`

02

`class Movimiento`

03

`class EquipoPokemon`

04

`class Combate`

05

`class TableroJuego`

06

`class ShowdownJuego`

07

Prueba

```
class Pokemon {
```

```
// ----- ATRIBUTOS -----
```

```
public class Pokemon {  
    private String nombre;  
    private String tipo;  
    private int vidaMax;  
    private int vidaActual;  
    private int ataque;  
    private List<Movimiento> movimientos;  
    private boolean estaVivo;
```

```
// ----- CONSTRUCTOR -----
```

```
    public Pokemon(String nombre, String tipo, int vidaMax, int ataque, List<Movimiento> movimientos, boolean estaVivo) {  
        this.nombre = nombre;  
        this.tipo = tipo;  
        this.vidaMax = vidaMax;  
        this.vidaActual = vidaMax;  
        this.ataque = ataque;  
        if (movimientos == null) {  
            this.movimientos = new ArrayList<>();  
        } else {  
            this.movimientos = new ArrayList<>(movimientos);  
        }  
        this.estaVivo = estaVivo;  
    }  
}
```

```
}
```

```
class Pokemon {
```

```
// ----- GETTERS -----
```

```
public String getNombre() {  
    return this.nombre;  
}
```

```
public String getTipo() {  
    return this.tipo;  
}
```

```
public int getVidaMax() {  
    return this.vidaMax;  
}
```

```
public int getVidaActual() {  
    return this.vidaActual;  
}
```

```
public int getAtaque() {  
    return this.ataque;  
}
```

```
public boolean getEstaVivo() {  
    return this.estaVivo;  
}
```

```
public List<Movimiento> getMovimientos(){  
    return this.movimientos;  
}
```

```
public void aprenderMovimiento(Movimiento movimientoNuevo) {  
    this.movimientos.add(movimientoNuevo);  
}
```

```
public void recibirDaño(int daño) {  
    if (estaVivo) {  
        vidaActual = vidaActual - daño;  
        if (vidaActual <= 0) {  
            estaVivo = false;  
            vidaActual = 0;  
        }  
    }  
}
```

```
}
```

```
class Movimiento{
```

```
//----- ATRIBUTOS -----
```

```
public class Movimiento {  
    private String nombre;  
    private String tipo;  
    private int potencia;
```

```
// ----- GETTERS -----
```

```
    public String getNombre() {  
        return this.nombre;  
    }  
  
    public int getPotencia() {  
        return this.potencia;  
    }  
  
    public String getTipo() {  
        return this.tipo;  
    }
```

```
// ----- CONSTRUCTOR -----
```

```
    public Movimiento(String nombre, String tipo, int potencia) {  
        this.nombre = nombre;  
        this.tipo = tipo;  
        this.potencia = potencia;  
    }
```

```
}
```

class Movimiento{

```
public List<Movimiento> movs = new ArrayList<>();

public Movimiento() {

    // --- MOVIMIENTOS LEGENDARIOS / ESPECIALES ---
    movs.add(new Movimiento("Distorsión", "Dragon", 150));
    movs.add(new Movimiento("Corte Vacío", "Dragon", 100));
    movs.add(new Movimiento("Golpe Umbrío", "Fantasma", 120));
    movs.add(new Movimiento("Sentencia", "Normal", 100));
    movs.add(new Movimiento("Roar of Time", "Dragon", 150));
    movs.add(new Movimiento("Ataque Fulgor", "Eléctrico", 130));
    movs.add(new Movimiento("Llama Azul", "Fuego", 130));
    movs.add(new Movimiento("Pulso Primigenio", "Agua", 110));
    movs.add(new Movimiento("Ala Mortífera", "Siniestro", 80));
    movs.add(new Movimiento("Espada Suprema", "Acero", 100));
    movs.add(new Movimiento("Embate Supremo", "Lucha", 120));
    movs.add(new Movimiento("Cañón Dinamax", "Dragon", 100));
    movs.add(new Movimiento("Viento Aciago", "Fantasma", 60));
    movs.add(new Movimiento("Psicocorte", "Psíquico", 70));
    movs.add(new Movimiento("Esfera Aural", "Lucha", 80));

    // --- TIPO FUEGO ---
    movs.add(new Movimiento("Lanzallamas", "Fuego", 90));
    movs.add(new Movimiento("Llamarada", "Fuego", 110));
    movs.add(new Movimiento("Envite Ígneo", "Fuego", 120));
    movs.add(new Movimiento("Puño Fuego", "Fuego", 75));
    movs.add(new Movimiento("Sofoco", "Fuego", 130));
    movs.add(new Movimiento("Nitrocarga", "Fuego", 50));
```

```
// --- TIPO AGUA ---
movs.add(new Movimiento("Hidrobomba", "Agua", 110));
movs.add(new Movimiento("Escaldar", "Agua", 80));
movs.add(new Movimiento("Surf", "Agua", 90));
movs.add(new Movimiento("Cascada", "Agua", 80));
movs.add(new Movimiento("Acua Jet", "Agua", 40));
movs.add(new Movimiento("Martillazo", "Agua", 100));

// --- TIPO PLANTA ---
movs.add(new Movimiento("Rayo Solar", "Planta", 120));
movs.add(new Movimiento("Hoja Aguda", "Planta", 90));
movs.add(new Movimiento("Giga Drenado", "Planta", 75));
movs.add(new Movimiento("Tormenta Verde", "Planta", 130));
movs.add(new Movimiento("Latigazo", "Planta", 120));

// --- TIPO ELÉCTRICO ---
movs.add(new Movimiento("Rayo", "Electrico", 90));
movs.add(new Movimiento("Trueno", "Electrico", 110));
movs.add(new Movimiento("Voltio Cruel", "Electrico", 90));
movs.add(new Movimiento("Puño Trueno", "Electrico", 75));
movs.add(new Movimiento("Chispazo", "Electrico", 80));

// --- TIPO HIELO ---
movs.add(new Movimiento("Rayo Hielo", "Hielo", 90));
movs.add(new Movimiento("Ventisca", "Hielo", 110));
movs.add(new Movimiento("Chuzos", "Hielo", 85));
movs.add(new Movimiento("Canto Helado", "Hielo", 40));
movs.add(new Movimiento("Puño Hielo", "Hielo", 75));
```

}

```
class EquipoPokemon {
```

```
private List<Pokemon> listaPokemons = new ArrayList<>();

// ----- LISTA DE POKEMONS AÑADIDOS -----

public EquipoPokemon() {

    listaPokemons.add(new Pokemon("Mewtwo", "Psiquico", 212, 110, null, true));
    listaPokemons.add(new Pokemon("Lugia", "Psiquico", 212, 90, null, true));
    listaPokemons.add(new Pokemon("Ho-Oh", "Fuego", 212, 130, null, true));
    listaPokemons.add(new Pokemon("Rayquaza", "Dragon", 210, 150, null, true));
    listaPokemons.add(new Pokemon("Kyogre", "Agua", 200, 100, null, true));
    listaPokemons.add(new Pokemon("Groudon", "Tierra", 200, 150, null, true));
    listaPokemons.add(new Pokemon("Dialga", "Acero", 200, 120, null, true));
    listaPokemons.add(new Pokemon("Palkia", "Agua", 190, 120, null, true));
    listaPokemons.add(new Pokemon("Giratina", "Fantasma", 250, 100, null, true));
    listaPokemons.add(new Pokemon("Arceus", "Normal", 220, 120, null, true));
    listaPokemons.add(new Pokemon("Zekrom", "Electrico", 200, 150, null, true));
    listaPokemons.add(new Pokemon("Reshiram", "Fuego", 200, 120, null, true));
    listaPokemons.add(new Pokemon("Xerneas", "Hada", 226, 131, null, true));
    listaPokemons.add(new Pokemon("Yveltal", "Siniestro", 226, 131, null, true));
    listaPokemons.add(new Pokemon("Zacian", "Acero", 192, 170, null, true));
    listaPokemons.add(new Pokemon("Zamazenta", "Acero", 192, 130, null, true));
    listaPokemons.add(new Pokemon("Eternatus", "Dragon", 240, 85, null, true));
    listaPokemons.add(new Pokemon("Darkrai", "Siniestro", 170, 90, null, true));
    listaPokemons.add(new Pokemon("Deoxys", "Psiquico", 150, 150, null, true));
    listaPokemons.add(new Pokemon("Articuno", "Hielo", 190, 85, null, true));
    listaPokemons.add(new Pokemon("Zapdos", "Electrico", 190, 90, null, true));
```

```
}
```

```
class EquipoPokemon {
```

```
// ----- CODIGO GENERAR EQUIPO -----  
  
public List<Pokemon> generarEquipo() {  
    List<Pokemon> equipo = new ArrayList<>();  
    Movimiento listaMovimientos = new Movimiento();  
  
    for (int i = 0; i < 6; i++) {  
        int numero = (int) (Math.random() * listaPokemons.size());  
        Pokemon pBase = listaPokemons.get(numero);  
  
        Pokemon pNuevo = new Pokemon(pBase.getNombre(), pBase.getTipo(),  
                                       pBase.getVidaMax(), pBase.getAtaque(),  
                                       null, true);  
  
        while (pNuevo.getMovimientos().size() < 4) {  
            int numMov = (int) (Math.random() * listaMovimientos.movs.size());  
            Movimiento m = listaMovimientos.movs.get(numMov);  
  
            if (!pNuevo.getMovimientos().contains(m)) {  
                pNuevo.aprenderMovimiento(m);  
            }  
        }  
        equipo.add(pNuevo);  
    }  
    return equipo;  
}
```

```
}
```

```
class Combate{
```

```
    private Map<String, Map<String, Double>> tablaTipos;

    public Combate() {
        try {
            ObjectMapper mapper = new ObjectMapper();
            tablaTipos = mapper.readValue(new File("tipos.json"), Map.class);
        } catch (Exception e) {
            System.out.println("Error: No se pudo leer el archivo");
        }
    }
}
```

```
    public void realizarAtaque(Pokemon atacante, Pokemon defensor, Movimiento movimiento) {
        System.out.println(atacante.getNombre() + " usa " + movimiento.getNombre());

        double multiplicador = 1.0;
        if (tablaTipos != null && tablaTipos.containsKey(movimiento.getTipo())) {
            Map<String, Double> relaciones = tablaTipos.get(movimiento.getTipo());
            multiplicador = relaciones.getDefault(defensor.getTipo(), 1.0);
        }

        int daño = (int) (((atacante.getAtaque() + movimiento.getPotencia()) * 0.4) * multiplicador);

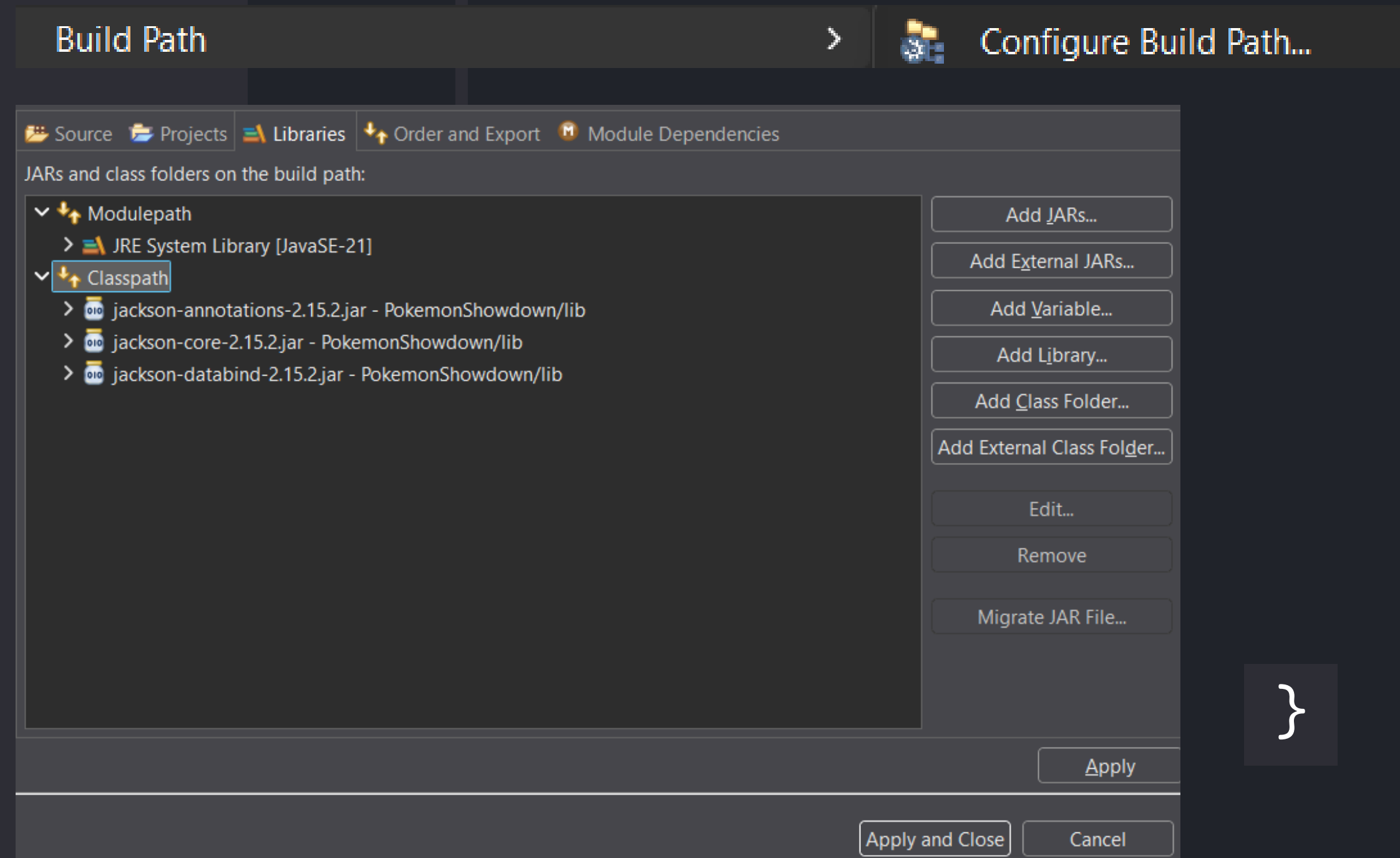
        if (multiplicador > 1.0) {
            System.out.println("¡Es muy eficaz!");
        } else if (multiplicador < 1.0 && multiplicador > 0) {
            System.out.println("No es muy eficaz, tontito.");
        } else if (multiplicador == 0) {
            System.out.println("No afecta a " + defensor.getNombre());
        } else {
            System.out.println("El daño es neutro.");
        }
        System.out.println("Ha causado " + daño + " puntos de daño!");
        defensor.recibirDaño(daño);
    }
}
```

```
}
```

```
class Combate{
```

```
import com.fasterxml.jackson.databind.ObjectMapper;  
import java.io.File;  
import java.util.Map;
```

- ▼ Referenced Libraries
 - > jackson-annotations-2.15.2.jar
 - > jackson-core-2.15.2.jar
 - > jackson-databind-2.15.2.jar
- ▼ lib
 - jackson-annotations-2.15.2.jar
 - jackson-core-2.15.2.jar
 - jackson-databind-2.15.2.jar



```
}
```

```
class TableroJuego {
```

```
    public class TableroJuego {
        Scanner sc = new Scanner(System.in);

// ----- CODIGO TABLERO JUEGO -----

        public void dibujarTablero(Pokemon Jugador1, Pokemon Jugador2) {
            System.out.println("\n" + "_".repeat(65));

            System.out.println();
            System.out.printf("%45s [%s]\n", "", Jugador2.getNombre().toUpperCase());
            System.out.println();
            System.out.printf("%45s HP: %d / %d\n", "", Jugador2.getVidaActual(), Jugador2.getVidaMax());

            System.out.println("\n                                VS\n");

            System.out.printf("[%s]\n", Jugador1.getNombre().toUpperCase());
            System.out.println();
            System.out.printf("HP: %d / %d\n", Jugador1.getVidaActual(), Jugador1.getVidaMax());
            System.out.println();

            List<Movimiento> misMovimientos = Jugador1.getMovimientos();

            System.out.printf(" 1. %-15s (Pot: %-3d) |  2. %-15s (Pot: %-3d)\n",
                misMovimientos.get(0).getNombre(), misMovimientos.get(0).getPotencia(),
                misMovimientos.get(1).getNombre(), misMovimientos.get(1).getPotencia());

            System.out.printf(" 3. %-15s (Pot: %-3d) |  4. %-15s (Pot: %-3d)\n",
                misMovimientos.get(2).getNombre(), misMovimientos.get(2).getPotencia(),
                misMovimientos.get(3).getNombre(), misMovimientos.get(3).getPotencia());
            System.out.println("_".repeat(65));
        }
    }
}
```

```
}
```

class ShowdownJuego{

```
// ----- GENERADOR DE TABLERO Y DE LOS EQUIPOS -----
```

```
TableroJuego tablero = new TableroJuego();
Scanner sc = new Scanner(System.in);
EquipoPokemon pokemons = new EquipoPokemon();
Combate combate = new Combate();

int pokemonActivo1 = 0;
int pokemonActivo2 = 0;

List<Pokemon> equipoJugador1 = pokemons.generarEquipo();
List<Pokemon> equipoJugador2 = pokemons.generarEquipo();

List<Pokemon> muertosJugador1 = new ArrayList<>();
List<Pokemon> muertosJugador2 = new ArrayList<>();
```

```
// ----- LISTA DE TUS POKEMONS -----
```

```
System.out.println("¡TUS POKEMONS!");
System.out.println();

for (Pokemon pokemon1 : equipoJugador1) {
    System.out.println(pokemon1.getNombre() + " -> " + pokemon1.getTipo());
}
System.out.println();
```

```
// ----- SHOWDOWN -----
```

```
// ----- IMPRIMIR TABLERO -----
```

```
do {
    boolean rivalAtaca = true;
    Pokemon activoJugador1 = equipoJugador1.get(pokemonActivo1);
    Pokemon activoJugador2 = equipoJugador2.get(pokemonActivo2);

    tablero.dibujarTablero(activoJugador1, activoJugador2);

    System.out.println("¿Que quieres hacer?");
    System.out.println("1. Atacar");
    System.out.println("2. Cambiar de pokemon");
    System.out.println("3. Rendirte");
    int opcion = sc.nextInt();

    switch (opcion) {
```

```
}
```

```
class ShowdownJuego{
```

```
// ----- ATACAR -----  
  
case 1 -> {  
  
    System.out.println("Elige el movimiento que quieres hacer (1-4): ");  
    int ataque = sc.nextInt() - 1;  
  
    Movimiento movEscogido = activoJugador1.getMovimientos().get(ataque);  
  
    combate.realizarAtaque(activoJugador1, activoJugador2, movEscogido);  
  
    if (!activoJugador2.getEstaVivo()) {  
        System.out.println("\n¡El " + activoJugador2.getNombre() + " rival se ha debilitado!");  
        muertosJugador2.add(activoJugador2);  
        pokemonActivo2++;  
        rivalAtaca = false;  
    }  
  
    if (muertosJugador2.size() < equipoJugador2.size()) {  
        activoJugador2 = equipoJugador2.get(pokemonActivo2);  
    } else {  
        System.out.println("¡HAS GANADO! Maquinon");  
        return;  
    }  
}  
}
```

```
}
```

```
class ShowdownJuego{
```

```
// ----- CAMBIAR -----
case 2 -> {
    boolean cambioCorrecto = false;

    while (!cambioCorrecto) {

        System.out.println("--- TU EQUIPO ---");
        String estado;
        for (int i = 0; i < equipoJugador1.size(); i++) {
            Pokemon p = equipoJugador1.get(i);
            if (p.getEstaVivo()) {
                estado = "VIDA: " + p.getVidaActual();
            } else {
                estado = "MUERTO";
            }
            System.out.println((i + 1) + ". " + p.getNombre() + " - " + estado);
        }
        System.out.println("Elige el numero del Pokémon al que cambiar:");
        int cambio = sc.nextInt() - 1;

        if (cambio >= 0 && cambio < equipoJugador1.size() && equipoJugador1.get(cambio).getEstaVivo()) {
            pokemonActivo1 = cambio;
            System.out.println("¡Adelante " + equipoJugador1.get(cambio).getNombre() + "!");
            cambioCorrecto = true;
        } else {
            System.out.println("Cambio no valido o Pokémon debilitado.");
        }
    }
}

// ----- FF -----

case 3 -> {
    System.out.println("Te has rendido, eres malisimo.");
    return;
}
```

```
}
```

```
class ShowdownJuego{
```

```
// ----- ATAQUE RIVAL RANDOM -----  
  
if (activoJugador2.getEstaVivo() && rivalAtaca) {  
  
    System.out.println("\n--- Turno del Rival ---");  
  
    int ataqueRandom = (int) (Math.random() * 4);  
    Movimiento ataqueRival = activoJugador2.getMovimientos().get(ataqueRandom);  
  
    combate.realizarAtaque(activoJugador2, activoJugador1, ataqueRival);  
  
    if (!activoJugador1.getEstaVivo()) {  
        System.out.println("\nTu " + activoJugador1.getNombre() + " se ha debilitado\n");  
        muertosJugador1.add(activoJugador1);  
    }  
}
```

```
}
```

```
class ShowdownJuego{
```

```
    if (muertosJugador1.size() < equipoJugador1.size()) {
        boolean cambioCorrecto = false;

        while (!cambioCorrecto) {

            System.out.println("--- TU EQUIPO ---");
            String estado;
            for (int i = 0; i < equipoJugador1.size(); i++) {
                Pokemon p = equipoJugador1.get(i);
                if (p.getEstaVivo()) {
                    estado = "VIDA: " + p.getVidaActual();
                } else {
                    estado = "MUERTO";
                }
                System.out.println((i + 1) + ". " + p.getNombre() + " - " + estado);
            }

            System.out.println("Elige el numero del Pokémon al que cambiar:");
            int cambio = sc.nextInt() - 1;

            if (cambio >= 0 && cambio < equipoJugador1.size()
                && equipoJugador1.get(cambio).getEstaVivo()) {
                pokemonActivo1 = cambio;
                System.out.println("¡Adelante " + equipoJugador1.get(cambio).getNombre() + "!");
                cambioCorrecto = true;
            } else {
                System.out.println("Cambio no valido o Pokémon debilitado.");
            }
        }
    } else {
        System.out.println("¡Has perdido! Pringao");
        return;
    }
}

} while (!equipoJugador1.isEmpty() && !equipoJugador2.isEmpty());
}
```

```
}
```

<!--Proyecto programación-->

Gracias {

<Por="Nicolae Poenaru"/>

}

