

**Universidad Nacional de San Agustín**  
**Escuela Profesional de Ingeniería de Sistemas**  
**Fundamentos de Programación II**  
**Práctica de Laboratorio N° 12:**  
**Definición de Clases de Usuario**  
**Clase Soldado - Menú**

**Nombre:** Jhonatan Benjamin Mamani Céspedes

**CUI:** 20232188

**Link de GitHub:** <https://github.com/JBenjamin01/fp2-24b/tree/main/Laboratorio>

Usar como base el diagrama de clases UML siguiente:

Soldado
-nombre -nivelAtaque -nivelDefensa -nivelVida -vidaActual -velocidad -actitud -vive
+Soldado() +atacar() +defender() +avanzar() +retroceder() +serAtacado() +huir() +morir() +setVidaActual() +getVidaActual()

1. Puede reutilizar todo el código del laboratorio 11, pero ahora el objetivo es gestionar los ejércitos autogenerados.
2. Al ejecutar el videojuego, el programa deberá dar las opciones:
  1. Juego rápido (tal cual como en el laboratorio 11) Al acabar el juego mostrar las opciones de volver a jugar y de volver al menú principal. También se deberá tener la posibilidad de cancelar el juego actual en cualquier momento, permitiendo escoger entre empezar un juego totalmente nuevo o salir al menú principal.

2. Juego personalizado: permite gestionar ejércitos. Primero se generan los 2 ejércitos con sus respectivos soldados y se muestran sus datos

Luego se tendrá que escoger cuál de los 2 ejércitos se va a gestionar, después se mostrarán las siguientes opciones:

- 1) Crear Soldado: permitirá crear un nuevo soldado personalizado y añadir al final del ejército (recordar que límite es de 10 soldados por ejército)
- 2) Eliminar Soldado (no debe permitir un ejército vacío)
- 3) Clonar Soldado (crea una copia exacta del soldado) y se añade al final del ejército (recordar que límite es de 10 soldados por ejército)
- 4) Modificar Soldado (con submenú para cambiar alguno de los atributos nivelAtaque, nivelDefensa, vidaActual)
- 5) Comparar Soldados (verifica si atributos: nombre, nivelAtaque, nivelDefensa, vidaActual y vive son iguales)
- 6) Intercambiar Soldados (intercambia 2 soldados en sus posiciones en la estructura de datos del ejército)
- 7) Ver soldado (Búsqueda por nombre)
- 8) Ver ejército
- 9) Sumar niveles (usando Method-Call Chaining), calcular las sumatorias de nivelVida, nivelAtaque, nivelDefensa, velocidad de todos los soldados de un ejército 1. Por ejemplo, si ejército tendría 3 soldados: 2.  $s = s1.sumar(s2).sumar(s3)$ ; 3. s es un objeto Soldado nuevo que contendría las sumatorias de los 4 atributos indicados de los 3 soldados. Ningún soldado cambia sus valores
- 10) Jugar (se empezará el juego con los cambios realizados) y con las mismas opciones de la opción 1 del menú principal.
- 11) Volver (muestra el menú principal) Después de escoger alguna de las opciones 1) a 9) se podrá volver a elegir uno de los ejércitos y se mostrarán las opciones 1) a 11)

3. Salir

## Clase Soldado.java:

```
1  public class Soldado {
2      private String nombre;
3      private int puntosVida;
4      private int fila;
5      private char columna;
6      private int nivelAtaque;
7      private int nivelDefensa;
8      private int velocidad;
9      private String actitud;
10     private boolean vive;
11
12     // Constructores sobrecargados del Soldado
13     public Soldado(String nombre, int puntosVida, int fila, char columna, int nivelAtaque, int nivelDefensa) {
14         this.nombre = nombre;
15         this.puntosVida = puntosVida;
16         this.fila = fila;
17         this.columna = columna;
18         this.nivelAtaque = nivelAtaque;
19         this.nivelDefensa = nivelDefensa;
20         this.velocidad = 0;
21         this.actitud = "Defensiva"; // Estado inicial
22         this.vive = true; // Aquí dejo que por defecto, el soldado está vivo
23     }
24     public Soldado(String nombre, int puntosVida, int fila, char columna) {
25         this.nombre = nombre;
26         this.puntosVida = puntosVida;
27         this.fila = fila;
28         this.columna = columna;
29         this.nivelAtaque = (int)(Math.random() * 5) + 1;
30         this.nivelDefensa = (int)(Math.random() * 5) + 1;
31         this.velocidad = 0;
32         this.actitud = "Defensiva";
33         this.vive = true;
34     }
35     public Soldado(String nombre) {
36         this(nombre, (int)(Math.random() * 5) + 1, (int)(Math.random() * 10), (char)('A' + (int)(Math.random() * 10)),
37             (int)(Math.random() * 5) + 1, (int)(Math.random() * 5) + 1);
38     }
39     public Soldado() {
40         this("Soldado" + (int)(Math.random() * 100));
41     }
42
43     // Getters
44     public String getNombre() {
45         return nombre;
46     }
47
48     public int getFila() {
49         return fila;
50     }
51     public char getColumna() {
52         return columna;
53     }
54     public int getPuntosVida() {
55         return puntosVida;
56     }
57     public int getNivelAtaque() {
58         return nivelAtaque;
59     }
60     public int getNivelDefensa() {
61         return nivelDefensa;
62     }
63     public int getVelocidad() {
64         return velocidad;
65     }
66     public String getActitud() {
67         return actitud;
68     }
69     public boolean isVivo() {
70         return vive;
71     }
72 }
```

```

72 // Setters
73 public void setFila(int fila) {
74     this.fila = fila;
75 }
76 public void setColumna(char columna) {
77     this.columna = columna;
78 }
79 public void setPuntosVida(int puntosVida) {
80     this.puntosVida = puntosVida;
81 }
82
83 // Métodos del UML de referencia
84 public void atacar() {
85     velocidad += 1;
86     actitud = "Ofensiva";
87     System.out.println(nombre + " ha atacado, su velocidad es ahora " + velocidad);
88 }
89 public void defender() {
90     actitud = "Defensiva";
91     System.out.println(nombre + " está en modo defensivo.");
92 }
93 public void huir() {
94     velocidad += 2;
95     actitud = "Fuga";
96     System.out.println(nombre + " está huyendo, su velocidad es ahora " + velocidad);
97 }
98 public void avanzar() {
99     velocidad += 1;
100    System.out.println(nombre + " avanza, su velocidad es ahora " + velocidad);
101 }
102 public void retroceder() {
103     if (velocidad > 0) {
104         velocidad = 0;
105         actitud = "Defensiva";
106         System.out.println(nombre + " se ha detenido, velocidad actual: " + velocidad);
107     } else {
108         velocidad -= 1;
109         System.out.println(nombre + " ha retrocedido, velocidad negativa: " + velocidad);
110     }
111 }
112 public void serAtacado(int daño) {
113     recibirAtaque(daño);
114 }
115 public void recibirAtaque(int daño) {
116     puntosVida -= daño;
117     if (puntosVida <= 0) {
118         puntosVida = 0;
119         System.out.println(nombre + " ha muerto.");
120     } else {
121         System.out.println(nombre + " ha recibido " + daño + " de daño. Vida restante: " + puntosVida);
122     }
123 }
124 public void morir() {
125     vive = false;
126     System.out.println(nombre + " ha muerto.");
127 }
128
129 @Override
130 public String toString() {
131     return "Nombre: " + nombre
132         + " | Vida: " + puntosVida
133         + " | Ataque: " + nivelAtaque
134         + " | Defensa: " + nivelDefensa
135         + " | Velocidad: " + velocidad
136         + " | Actitud: " + actitud
137         + " | Posición: " + columna + fila
138         + " | Vive: " + (vive ? "Sí" : "No");
139 }
140 }

```

## Clase VideoJuego.java

```
1 // LABORATORIO N° 12
2 // AUTOR: JHONATAN BENJAMIN MAMANI CÉSPEDES
3 // TIEMPO: 41 MINUTOS
4 import java.text.DecimalFormat;
5 import java.util.*;
6 public class VideoJuego {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         boolean exit = false;
10
11         while (!exit) {
12             System.out.println("\n===== MENU PRINCIPAL =====");
13             System.out.println("1. Juego rápido");
14             System.out.println("2. Juego personalizado");
15             System.out.println("3. Salir");
16             System.out.print("Seleccione una opción: ");
17
18             int option = sc.nextInt();
19
20             switch (option) {
21                 case 1:
22                     ArrayList<ArrayList<Soldado>> tablero = new ArrayList<>();
23                     ArrayList<Soldado> e1 = new ArrayList<>();
24                     ArrayList<Soldado> e2 = new ArrayList<>();
25
26                     inicializarTablero(tablero);
27                     inicializarEjercitos(e1, e2, tablero);
28
29                     System.out.println("\n¡Inicia la batalla!");
30                     mostrarTablero(tablero, e1, e2);
31                     juego(tablero, e1, e2);
32                     break;
33                 case 2:
34                     System.out.println("Juego personalizado (Aun estoy definiendo que métodos puedo usar)...");
35                     break;
36                 case 3:
37                     System.out.println("Saliendo del juego. ¡Hasta luego!");
38                     exit = true;
39                     break;
40                 default:
41                     System.out.println("Opción inválida. Intente nuevamente.");
42                     break;
43             }
44         }
45
46
47
48     }
49
50     public static void inicializarTablero(ArrayList<ArrayList<Soldado>> tablero) {
51         for (int i = 0; i < 10; i++) {
52             ArrayList<Soldado> fila = new ArrayList<>();
53             for (int j = 0; j < 10; j++)
54                 fila.add(null);
55             tablero.add(fila);
56         }
57     }
58
59     public static void inicializarEjercitos(ArrayList<Soldado> e1, ArrayList<Soldado> e2,
60         ArrayList<ArrayList<Soldado>> tablero) {
61         for (int i = 0; i < 2; i++) {
62             int n = (int) (Math.random() * 10) + 1;
63             for (int j = 0; j < n; j++) {
64                 int fila, columna;
65                 do {
66                     fila = (int) (Math.random() * 10);
67                     columna = (int) (Math.random() * 10);
68                 } while (tablero.get(fila).get(columna) != null);
69
70                 String nombre = "Soldado" + (i + 1) + "X" + (j + 1);
71                 int puntosVida = (int) (Math.random() * 5) + 1;
72                 Soldado soldado = new Soldado(nombre, puntosVida, fila + 1, (char) ('A' + columna));
73
74                 if (i == 0) {
75                     e1.add(soldado);
76                     tablero.get(fila).set(columna, soldado);
77                 } else {
78                     e2.add(soldado);
79                     tablero.get(fila).set(columna, soldado);
80                 }
81             }
82         }
83     }
84 }
```

```

85     public static void mostrarTablero(ArrayList<ArrayList<Soldado>> tablero, ArrayList<Soldado> e1,
86         ArrayList<Soldado> e2) {
87         System.out.println("Tablero de la batalla:");
88         + "\nLas unidades del ejército 1 están con sus puntos de vida entre corchetes ([x])."
89         + "\nLas del ejército 2 con sus puntos de vida entre signos menor y mayor que (<x>):");
90         System.out.println("\n      A   B   C   D   E   F   G   H   I   J");
91         System.out.println();
92         System.out.println("-----");
93
94         for (int i = 0; i < tablero.size(); i++) {
95             System.out.print(i + 1 + "\t| ");
96             for (int j = 0; j < tablero.get(i).size(); j++) {
97                 Soldado soldado = tablero.get(i).get(j);
98                 if (soldado == null)
99                     System.out.print("   | ");
100             }
101             else if (e1.contains(soldado))
102                 System.out.print("[ " + soldado.getPuntosVida() + " ] | ");
103             else if (e2.contains(soldado))
104                 System.out.print("< " + soldado.getPuntosVida() + " > | ");
105             else
106                 System.out.print(soldado.getPuntosVida() + "   | ");
107         }
108         System.out.println();
109         System.out.println("-----");
110     }
111 }
112
113
114 public static void juego(ArrayList<ArrayList<Soldado>> tablero, ArrayList<Soldado> e1, ArrayList<Soldado> e2) {
115     Scanner sc = new Scanner(System.in);
116     boolean turnoEjercito1 = true;
117     while (!e1.isEmpty() && !e2.isEmpty()) {
118         System.out.println(turnoEjercito1 ? "\nTurno del Ejército 1 ---> [x]" : "\nTurno del Ejército 2 ---> <x>");
119
120         // Paso 1: Ingresar coordenada y verificarla antes de solicitar la dirección
121         System.out.print("Ingrese coordenada del soldado a mover (Ej. C5): ");
122         String coordenada = sc.nextLine().toUpperCase();
123
124         // Buscar el soldado en la coordenada ingresada
125         Soldado soldadoSeleccionado = buscarSoldado(tablero, coordenada);
126         if (soldadoSeleccionado == null || (turnoEjercito1 ? !e1.contains(soldadoSeleccionado) : !e2.contains(soldadoSeleccionado))) {
127             System.out.println("Movimiento inválido. No hay soldado en la posición o es del ejército contrario.");
128             continue; // El continue deja volver al inicio del ciclo sin solicitar la dirección de movimiento
129         }
130
131         // Paso 2: Solicitar la dirección si la coordenada es válida
132         System.out.print("Ingrese dirección de movimiento (W para arriba, S para abajo, A para izquierda, D para derecha): ");
133         char direccion = sc.next().toUpperCase().charAt(0);
134         sc.nextLine();
135
136         boolean movimientoExitoso = moverSoldado(soldadoSeleccionado, direccion,
137             tablero, turnoEjercito1 ? e1 : e2, turnoEjercito1 ? e2 : e1);
138
139         if (!movimientoExitoso) {
140             System.out.println("Movimiento no válido, intente nuevamente.");
141             continue;
142         }
143
144         mostrarTablero(tablero, e1, e2);
145         turnoEjercito1 = !turnoEjercito1; // Cambios de turno
146     }
147
148     System.out.println(e1.isEmpty() ? "¡El Ejército 2 ha ganado!" : "¡El Ejército 1 ha ganado!");
149     sc.close();
150 }
151
152 public static Soldado buscarSoldado(ArrayList<ArrayList<Soldado>> tablero, String coordenada) {
153     int columna = coordenada.charAt(0) - 'A';
154
155     // Este paso se hizo para poder verificar el caso de que hayan dos cifras
156     int fila;
157     if (coordenada.length() == 3) {
158         // Aquí son dos dígitos, por ejemplo "H10"
159         fila = Integer.parseInt(coordenada.substring(1, 3)) - 1;
160     } else {
161         // Y este de solo un dígito, por ejemplo "A5"
162         fila = Character.getNumericValue(coordenada.charAt(1)) - 1;
163     }
164
165     if (fila >= 0 && fila < 10 && columna >= 0 && columna < 10)
166         return tablero.get(fila).get(columna);
167
168     return null;
169 }

```

```

170     public static boolean moverSoldado(Soldado soldado, char direccion, ArrayList<ArrayList<Soldado>> tablero,
171                                       ArrayList<Soldado> ejercitoAliado, ArrayList<Soldado> ejercitoEnemigo) {
172         int fila = soldado.getFila() - 1;
173         int columna = soldado.getColumna() - 'A';
174
175         switch (direccion) {
176             case 'W': fila--; break; // Arriba
177             case 'S': fila++; break; // Abajo
178             case 'A': columna--; break; // Izquierda
179             case 'D': columna++; break; // Derecha
180             default: return false;
181         }
182
183         // Primero verifico si el movimiento está dentro de los límites del tablero
184         if (!verificarMovimientoValido(fila, columna, tablero))
185             return false;
186
187         Soldado soldadoEnemigo = tablero.get(fila).get(columna);
188
189         // En caso de que la posición contiene un enemigo, se inicia la batalla
190         if (soldadoEnemigo != null && ejercitoEnemigo.contains(soldadoEnemigo)) {
191             batalla(soldado, soldadoEnemigo, ejercitoAliado, ejercitoEnemigo, tablero);
192         } else if (soldadoEnemigo == null) { // Si no hay enemigo, se va a mover el soldado a la posición
193             tablero.get(soldado.getFila() - 1).set(soldado.getColumna() - 'A', null);
194             tablero.get(fila).set(columna, soldado);
195             soldado.setFila(fila + 1);
196             soldado.setColumna((char) ('A' + columna));
197         } else {
198             return false; // Aquí simplemente sería por un movimiento inválido
199         }
200
201         return true;
202     }
203
204     public static boolean verificarMovimientoValido(int fila, int columna, ArrayList<ArrayList<Soldado>> tablero) {
205         return fila >= 0 && fila < 10 && columna >= 0 && columna < 10;
206     }
207
208     public static void batalla(Soldado s1, Soldado s2, ArrayList<Soldado> ejercitoAliado,
209                              ArrayList<Soldado> ejercitoEnemigo, ArrayList<ArrayList<Soldado>> tablero) {
210         int vidaTotal = s1.getPuntosVida() + s2.getPuntosVida();
211         double probS1 = (double) s1.getPuntosVida() * 100 / vidaTotal;
212         double probS2 = (double) s2.getPuntosVida() * 100 / vidaTotal;
213
214         DecimalFormat df = new DecimalFormat("#.##"); // Aquí estoy usando un objeto para controlar a 3 decimales
215
216         Random random = new Random();
217         double resultado = random.nextDouble() * 100; // Genera un número entre 0 y 1 para decidir el ganador

```

```

226
227         System.out.println("\nBatalla entre " + s1.getNombre() + " y " + s2.getNombre());
228         System.out.println("Las probabilidades de victoria son | " + s1.getNombre() + " : " + df.format(probS1) + "% | y | "
229                             + s2.getNombre() + " : " + df.format(probS2) + "% |");
230
231         if (resultado < probS1) {
232             System.out.println(s1.getNombre() + " ha ganado la batalla contra " + s2.getNombre() + " y ha sumado 1 punto de vida.");
233             ejercitoEnemigo.remove(s2);
234             s1.setPuntosVida(s1.getPuntosVida() + 1);
235
236             tablero.get(s2.getFila() - 1).set(s2.getColumna() - 'A', s1);
237
238             tablero.get(s1.getFila() - 1).set(s1.getColumna() - 'A', null);
239             s1.setFila(s2.getFila());
240             s1.setColumna(s2.getColumna());
241         } else {
242             System.out.println(s2.getNombre() + " ha ganado la batalla contra " + s1.getNombre() + " y ha sumado 1 punto de vida.");
243             ejercitoAliado.remove(s1);
244             s2.setPuntosVida(s2.getPuntosVida() + 1);
245
246             tablero.get(s1.getFila() - 1).set(s1.getColumna() - 'A', null);
247         }
248     }
249
250     public static void juegoRapido() {
251         Scanner sc = new Scanner(System.in);
252         boolean jugarDeNuevo;
253
254         do {
255             ArrayList<ArrayList<Soldado>> tablero = new ArrayList<>();
256             ArrayList<Soldado> e1 = new ArrayList<>();
257             ArrayList<Soldado> e2 = new ArrayList<>();
258
259             inicializarTablero(tablero);
260             inicializarEjercitos(e1, e2, tablero);
261
262             System.out.println("\n¡Inicia la batalla!");
263             mostrarTablero(tablero, e1, e2);

```

```

265 // Muestra opciones de finalización solo si el juego no fue cancelado
266 if (!cancelado) {
267     System.out.println("\nEl juego ha finalizado. ¿Qué desea hacer?");
268     System.out.println("1 --> Volver a jugar un juego nuevo");
269     System.out.println("2 --> Volver al menú principal");
270     System.out.print("Seleccione una opción: ");
271     int opcion = sc.nextInt();
272     jugarDeNuevo = (opcion == 1);
273 } else {
274     jugarDeNuevo = false;
275 }
276 } while (jugarDeNuevo);
277 }
278
279
280
281 public static void juegoPersonalizado() {
282     ArrayList<ArrayList<Soldado>> t = new ArrayList<>();
283     ArrayList<Soldado> e1 = new ArrayList<>();
284     ArrayList<Soldado> e2 = new ArrayList<>();
285
286     inicializarTablero(t);
287     inicializarEjercitos(e1, e2, t);
288
289     System.out.println("\nDatos del ejército 1:");
290     mostrarDatosEjercito(e1);
291     System.out.println("\nDatos del ejército 2:");
292     mostrarDatosEjercito(e2);
293
294     Scanner sc = new Scanner(System.in);
295     boolean exit = false;
296
297     while (!exit) {
298         System.out.println("\n===== GESTION DE EJÉRCITOS =====");
299         System.out.println("Seleccione un ejército para gestionar:");
300         System.out.println("1. Ejército 1");
301         System.out.println("2. Ejército 2");
302         System.out.println("3. Volver al menú principal");
303
304         int seleccionEjercito = sc.nextInt();
305         ArrayList<Soldado> ejercito = seleccionEjercito == 1 ? e1 : e2;
306
307         if (seleccionEjercito == 3) {
308             exit = true;
309             continue;
310         }
311
312         boolean volver = false;
313         while (!volver) {
314             System.out.println("\n===== OPCIONES DE GESTION DE SOLDADOS =====");
315             System.out.println("1. Crear Soldado");
316             System.out.println("2. Eliminar Soldado");
317             System.out.println("3. Clonar Soldado");
318             System.out.println("4. Modificar Soldado");
319             System.out.println("5. Comparar Soldados");
320             System.out.println("6. Intercambiar Soldados");
321             System.out.println("7. Ver Soldado");
322             System.out.println("8. Ver Ejército");
323             System.out.println("9. Sumar Niveles");
324             System.out.println("10. Jugar");
325             System.out.println("11. Volver a selección de ejército");
326             System.out.print("Seleccione una opción:");
327
328             int opcion = sc.nextInt();
329
330             switch (opcion) {
331                 case 1: crearSoldado(ejercito); break;
332                 case 2: eliminarSoldado(ejercito); break;
333                 case 3: clonarSoldado(ejercito); break;
334                 case 4: modificarSoldado(ejercito); break;
335                 case 5: compararSoldados(ejercito); break;
336                 case 6: intercambiarSoldados(ejercito); break;
337                 case 7: verSoldado(ejercito); break;
338                 case 8: verEjercito(ejercito); break;
339                 case 9: sumarNiveles(ejercito); break;
340                 case 10:
341                     System.out.println("\n¡Inicia la batalla!");
342                     mostrarTablero(t, e1, e2);
343                     juego(t, e1, e2); volver = true; break;
344                 case 11: volver = true; break;
345                 default: System.out.println("Opción inválida. Intente nuevamente.");
346             }
347         }
348     }
349 }
350
351 public static void crearSoldado(ArrayList<Soldado> ejercito) {
352     if (ejercito.size() >= 10) {

```



```

353         System.out.println("El ejército ya tiene 10 soldados. No se pueden añadir más.");
354         return;
355     }
356     Scanner sc = new Scanner(System.in);
357     System.out.print("Ingrese nombre del soldado: ");
358     String nombre = sc.nextLine();
359     System.out.print("Ingrese puntos de vida: ");
360     int puntosVida = sc.nextInt();
361     System.out.print("Ingrese nivel de ataque: ");
362     int nivelAtaque = sc.nextInt();
363     System.out.print("Ingrese nivel de defensa: ");
364     int nivelDefensa = sc.nextInt();
365
366     Soldado nuevoSoldado = new Soldado(nombre, puntosVida, 0, 'A', nivelAtaque, nivelDefensa);
367     ejercito.add(nuevoSoldado);
368     System.out.println("Soldado creado y añadido al ejército.");
369 }
370
371 public static void eliminarSoldado(ArrayList<Soldado> ejercito) {
372     if (ejercito.size() <= 1) {
373         System.out.println("No se puede eliminar, el ejército no puede quedarse sin soldados.");
374         return;
375     }
376     Scanner sc = new Scanner(System.in);
377     System.out.print("Ingrese el índice del soldado a eliminar (1-" + ejercito.size() + "): ");
378     int index = sc.nextInt() - 1;
379
380     if (index >= 0 && index < ejercito.size()) {
381         ejercito.remove(index);
382         System.out.println("Soldado eliminado del ejército.");
383     } else {
384         System.out.println("Índice inválido.");
385     }
386 }
387
388 public static void clonarSoldado(ArrayList<Soldado> ejercito) {
389     if (ejercito.size() >= 10) {
390         System.out.println("El ejército ya tiene 10 soldados. No se pueden añadir más.");
391         return;
392     }
393     Scanner sc = new Scanner(System.in);
394     System.out.print("Ingrese el índice del soldado a clonar (1-" + ejercito.size() + "): ");
395     int index = sc.nextInt() - 1;
396
397     if (index >= 0 && index < ejercito.size()) {
398         Soldado clon = new Soldado(ejercito.get(index).getNombre(), ejercito.get(index).getPuntosVida(),
399             ejercito.get(index).getFila(), ejercito.get(index).getColumna(),
400             ejercito.get(index).getNivelAtaque(), ejercito.get(index).getNivelDefensa());
401         ejercito.add(clon);
402         System.out.println("Soldado clonado y añadido al ejército.");
403     } else {
404         System.out.println("Índice inválido.");
405     }
406 }
407
408
409 public static void modificarSoldado(ArrayList<Soldado> ejercito) {
410     Scanner sc = new Scanner(System.in);
411     System.out.print("Ingrese el índice del soldado a modificar (1-" + ejercito.size() + "): ");
412     int index = sc.nextInt() - 1;
413
414     if (index >= 0 && index < ejercito.size()) {
415         Soldado soldado = ejercito.get(index);
416         System.out.println("Modificando al soldado: " + soldado.getNombre());
417
418         System.out.print("Ingrese nueva vida actual: ");
419         soldado.setPuntosVida(sc.nextInt());
420
421         System.out.println("Soldado modificado exitosamente.");
422     } else {
423         System.out.println("Índice inválido.");
424     }
425 }
426
427
428 public boolean compararSoldados(Soldado soldado1, Soldado soldado2) {
429     return soldado1.getNombre().equals(soldado2.getNombre()) &&
430         soldado1.getNivelAtaque() == soldado2.getNivelAtaque() &&
431         soldado1.getNivelDefensa() == soldado2.getNivelDefensa() &&
432         soldado1.getPuntosVida() == soldado2.getPuntosVida();
433 }
434
435 public static void intercambiarSoldados(ArrayList<Soldado> ejercito) {
436     Scanner sc = new Scanner(System.in);
437     System.out.print("Ingrese el índice del primer soldado a intercambiar (1-" + ejercito.size() + "): ");
438     int index1 = sc.nextInt() - 1;

```

```

439     System.out.print("Ingrese el índice del segundo soldado a intercambiar (1-" + ejercito.size() + "): ");
440     int index2 = sc.nextInt() - 1;
441
442     // Validar Los índices
443     if (index1 >= 0 && index1 < ejercito.size() && index2 >= 0 && index2 < ejercito.size()) {
444         // Intercambiar soldados
445         Soldado temp = ejercito.get(index1);
446         ejercito.set(index1, ejercito.get(index2));
447         ejercito.set(index2, temp);
448         System.out.println("Soldados intercambiados exitosamente.");
449     } else {
450         System.out.println("Índices inválidos.");
451     }
452 }
453
454 public static void verSoldado(ArrayList<Soldado> ejercito) {
455     Scanner sc = new Scanner(System.in);
456     System.out.print("Ingrese el índice del soldado a ver (1-" + ejercito.size() + "): ");
457     int index = sc.nextInt() - 1;
458
459     if (index >= 0 && index < ejercito.size()) {
460         Soldado soldado = ejercito.get(index);
461         System.out.println("Datos del Soldado:");
462         System.out.println(soldado);
463     } else {
464         System.out.println("Índice inválido.");
465     }
466 }
467
468 public static void verEjercito(ArrayList<Soldado> ejercito) {
469     System.out.println("Datos del Ejército:");
470     for (int i = 0; i < ejercito.size(); i++) {
471         System.out.println((i + 1) + ". " + ejercito.get(i));
472     }
473 }
474
475 public static void sumarNiveles(ArrayList<Soldado> ejercito) {
476     int totalVida = 0;
477     int totalAtaque = 0;
478     int totalDefensa = 0;
479
480     for (Soldado soldado : ejercito) {
481         totalVida += soldado.getPuntosVida();
482         totalAtaque += soldado.getNivelAtaque();
483         totalDefensa += soldado.getNivelDefensa();
484     }
485
486     System.out.println("Puntaje combinado del ejército:");
487     System.out.println("Puntos totales de vida: " + totalVida);
488     System.out.println("Nivel total de ataque: " + totalAtaque);
489     System.out.println("Nivel total de defensa: " + totalDefensa);
490 }
491
492 public static void compararSoldados(ArrayList<Soldado> ejercito) {
493     Scanner sc = new Scanner(System.in);
494     System.out.print("Ingrese el índice del primer soldado a comparar (1-" + ejercito.size() + "): ");
495     int index1 = sc.nextInt() - 1;
496     System.out.print("Ingrese el índice del segundo soldado a comparar (1-" + ejercito.size() + "): ");
497     int index2 = sc.nextInt() - 1;
498
499     if (index1 >= 0 && index1 < ejercito.size() && index2 >= 0 && index2 < ejercito.size()) {
500         Soldado soldado1 = ejercito.get(index1);
501         Soldado soldado2 = ejercito.get(index2);
502         if (soldado1.equals(soldado2)) {
503             System.out.println("Los soldados son iguales.");
504         } else {
505             System.out.println("Los soldados son diferentes.");
506         }
507     } else {
508         System.out.println("Índices inválidos.");
509     }
510 }
511 }

```